Benjamin Walker

Masters in Micro and Nanotechnologies for Integrated Systems
2025

Université Paris Saclay
3 rue Joliot Curie, 91190 Gif-sur-Yvette

Spintronic on-chip learning with Bayes' Rule

from 03/03/25 to 12/09/25, with 2 week interruption

Under the supervision of :

- **Company supervisor :**
  Damien, Querlioz, damien.querlioz@universite-paris-saclay.fr
- **Phelma Tutor :**
  Miquel, Jonathan, jonathan.miquel@grenoble-inp.fr

Confidentiality : ☐ yes ☒ no

Ecole nationale
supérieure de physique,
électronique, matériaux

Phelma
Bât. Grenoble INP - Minatec
3 Parvis Louis Néel - CS 50257
F-38016 Grenoble Cedex 01

Tél +33 (0)4 56 52 91 00
Fax +33 (0)4 56 52 91 03

http://phelma.grenoble-inp.fr

Contents

List of Figures

1 Introduction

It would be an understatement to state that artificial intelligence (AI) models have altered the course of our society in the last five years, for better or for worse. While smaller models such like those used in optical character recognition or automated fraud detection have been successfully implemented and used as early as the 2000s, large language models (LLMs) have taken the world by storm. Partially pushed on consumers by companies and investors looking to cash in on a craze [1], these massive models predict the most-likely next word [1] given a preexisting set of words. By training on any and all text/data available on the internet and other sources (regardless of the permission of the copyright holders [2]), these models end up encoding a relatively vivid depiction of natural language, logic, and understanding of the world, just to predict the next most likely word. Then, by using a system prompt (e.g. "You are a friendly and helpful AI chatbot who wishes to help the user with their questions.") and fine-tuning with humans, they end up with a very useful model for helping with information-based tasks [3].

These LLMs have had and will continue to have profound impacts across multiple domains of society. These technologies have the potential to multiply worker productivity several-fold, improving quality of life for the average human worldwide if harnessed and implemented equitably. However, the benefits have served to reinforce existing capital, and businesses and investors are ready to replace entire workforces with AI. The replacement of call center workers is already having massive effects in Southeast Asia [4], and tech companies have severely slowed the hiring of junior developers, with hopes that AI systems can replace most of them [5]. The already addictive TikTok and similar video recommendation algorithms have removed agency from the user to curate their own content [6], [7]. Now, with the proliferation of "AI-slop" and generative AI content [8], people are being not only passivated but further and further isolated from real human interactions. Beyond these and other societal problems such as AI scams [9], deepfakes [10], disinformation [11], and potentially psychosis [12], just simply operating these AI systems presents a societal risk due to their energy consumption, both for training and inference. While OpenAI's GPT-3 was trained with 1.29 GWh [13], GPT-4 used ~ 55 GWh [14], and GPT-5 likely used on the order of ~ 550 GWh[2]. For GPT-5, this corresponds to 200,000 metric tons of CO_2 emissions from training alone (using the U.S. grid emissions factor [16]). Perhaps the more persistent problem is with inference. Researchers at the University of Rhode Island [13] estimate GPT-5 (high) to use 17.7 Wh per prompt, which equates to 6.06 g of CO_2. With millions of users, this easily creates hundreds of thousands of metric tons of CO_2 emissions per year, especially after considering scope 2 factors. Even the manner in which these data centers are powered has presented their own controversies outside of emissions in my home state, Louisiana [17]. The energy used by data centers is expected to surge from 2% to 9% of U.S. electricity consumption from 2020 to 2030 in large part due to machine learning training/inference demand [18]–[20].

While some of the broader ethical dilemmas posed by AI systems are beyond the scope of this work,

[1] Large-language models actually generate a set of most likely tokens, not words. Tokens can be punctuation, parts of words, or multiple words- all determined by the tokenizer

[2] OpenAI refuses to disclose both the number of parameters of GPT-4 and GPT-5 and the energy cost of training [15]. However, the cost of inference and training scales with parameter count (for non-reasoning models)[13], so one can estimate 10×, from the inference costs

the emissions and second order effects posed by these systems can be mitigated by designing systems with energy efficiency from the bottom up. One such proposal, is a tiered approach known as edge computing which reduces load on large data centers, while obtaining a higher fidelity of data processing and inference. This paradigm will be explained in more detail in 2.1.2, but it requires ultra-low power devices at the edge, with increased performance and power consumption as one moves closer to the datacenter level. While improvements in complementary metal-oxide semiconductor (CMOS) digital technologies have resulted in impressive improvements in energy efficiency per operation, one can gain even more efficiency and processing power by shifting to the analog domain, using emerging technologies such as magnetoresistive random access memory (MRAM) [21]. These analog inference arrays are a form of compute-in-memory (CiM), which reduces how much data needs to be shuffled around the chip, improving efficiency even further. The hope is, that by using emerging technology we can first improve the efficiency and feasibility of simple edge computing devices, and that the structures or algorithms for training and inference can be generalized and scaled to improve the efficiency of the entire computing stack.

Therefore, in this 3A thesis report, I outline the continuation of my inference array design from my 2A internship. After providing a detailed background to understand the context of my work, I then recap the array design created during my last internship. Then, similarly to my last report, I detail the design decisions chronologically as they lead to my finalized design. Beyond just the array itself, the chip will also need robust control circuitry, an analog-to-digital converter, and a digital interface to communicate with a micro-controller. The parameter spaces of each of these components had to be thoroughly explored before achieving a reasonably optimal solution. Particularly, the analog-to-digital converter (ADC) proved to be the most difficult component to design—which is typical of analog compute-in-memory arrays as I will explain in the next section. The design of the array ended up taking up the majority of the time of my thesis, therefore the title of the thesis "Spintronic on-chip learning with Bayes' Rule" does not correspond well to the content described herein— a better title could be "Design finalization of a SOT-MRAM Inference chip" However, my future work will combine my inference array with Bayesian learning rules, to be able to create an efficient on-chip learning methodology, which can hopefully reduce the energy consumption of not just inference for artificial intelligence (AI), but also for the training process as well.

2 Background

To better understand the context and technical details of my work, in this section I motivate the technical reasons for designing my array, then I discuss the technologies used to create my array, and then finally the algorithms which my array implements and that are used to train my array.

2.1 Technical Motivations

The overall goal of computing engineering is to increase the speed and performance of the entire system while minimizing power consumption. Since the start of Dennard scaling [3], power consumption per

[3]The bulk of transistor scaling up until ~2006 maintained a constant electric field across the gate, as both voltage and physical size were reduced in tandem.

operation has been halving every 1.5 years [22], a scaling rule known as Koomey's law. However, since Dennard scaling stopped in 2006, the gate charging energy has not scaled as quickly [23], resulting in massive power densities. Thus, alternative architectures have been proposed to increase performance per Watt not only for each computational unit but for the entire computing stack. It turns out that the less data has to be moved around, the more efficient the entire architecture becomes both from minimizing the cost of transmission and by allowing for the use of lower-bandwidth higher-efficiency components. Thus, both in-memory computing and edge computing seek to process data where it is collected.

2.1.1 In-memory computing

The von Neumann Bottleneck is a limitation of traditional computer architectures where the memory is physically separated from the processor. As the speed of the processor has increased over the years, the limiting factor to the processing speed has not been the processor or the memory, but rather the time delay caused by moving data between the processor and the memory elements [24]. As static power dominates modern computing nodes [25], if one were to reduce (or eliminate) the distance between the CPU (central processing unit) and the memory, one massively improves the performance per watt. This is why modern CPUs have integrated L1 memory caches, integrated as close to the CPU as possible. However, a new computing architecture aims to completely eliminate the separation of the computing element and the memory: compute-in-memory (CiM) [26]. This paradigm allows for ultra-high processing speeds, as no data needs to be moved within the memory unit. However, in-memory computation needs to integrate the processor within the memory unit (or vice-versa) which results in lower information density and/or a limited set of instructions/operations that can be performed. For this reason, simple operations between an input and the memory, such as sum and dot product, are preferred to be implemented in such systems. It turns out, as we shall soon see, that these are the fundamental operations which form the basis of a neural network.

2.1.2 Edge Computing

Beyond changing the architecture of individual computing units or machines, edge computing tries to optimize the entire computing stack to improve performance per Watt. The foundational idea is simple: certain computing architectures are optimized for ultra-low electrical power at lower processing speeds and some are optimized for maximum processing power which requires large quantities of power. The purpose of the computing stack is to be able to draw inferences from data collected at the edge. For example, the Oura Ring is a low-power smart device which processes biometric data within the ring, sends it to the user's phone for further processing and basic interpretation, and then sends those inferences to Oura's own servers to generate detailed trends and inferences about the user's health patterns [28], [29]. If all of the processing were to be done on-ring, it would have to use a higher-power (per bit processed) architecture with more memory, creating a bulky ring with horrific battery life. Likewise, if all of the raw data were simply streamed to Oura's servers, the extra data would have to be processed at a higher cost per bit than if it were done with a less performant processor, and the cost of streaming the data would be extraordinarily wasteful, not to mention the privacy concerns which are created by streaming

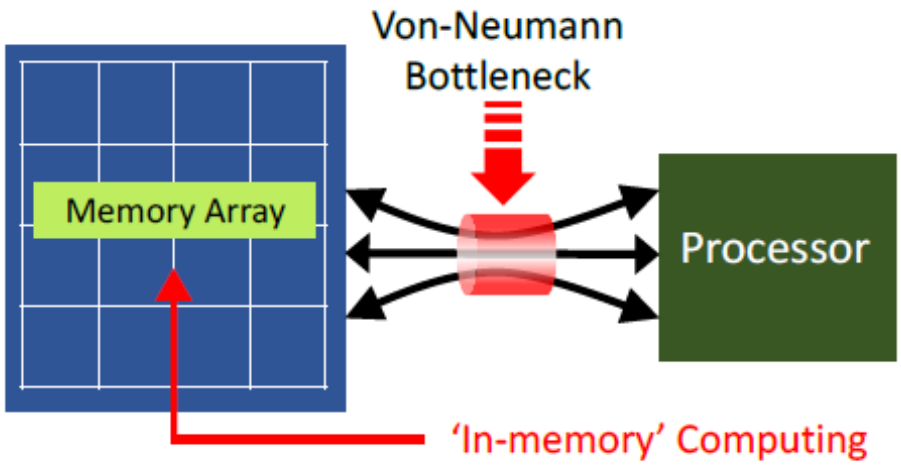


Figure 1: A diagram depicting the von Neumann bottleneck, in red, which represents the lack of bandwidth between memory and the processor. In-memory computing is represented by an arrow to the memory array, proposing that the computation takes places solely within the memory element to remove the bottleneck[27]

health information in real-time. Therefore, to minimize total computational energy cost and to maximize the efficiency/utility of the system, the modern computing stack can be broken down into three broad categories: the edge, the fog, and the cloud [30], [31] (Figure 2). The niche of this project is to create edge computing devices that optimize solely computational efficiency at the cost of lower computational throughput.

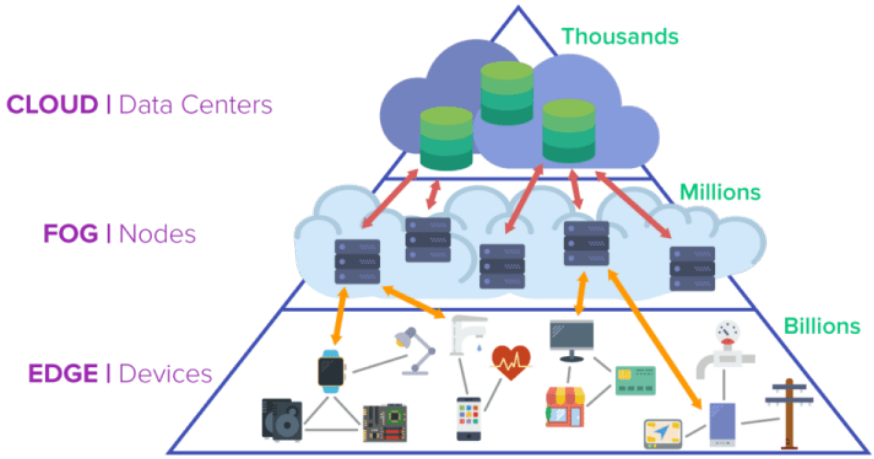


Figure 2: The edge computing stack. Edge devices (bottom) connect and send inferences/data to nodes on the fog (middle), which send data to data centers on the cloud (top) [32]

2.2 Technology

My design proposed in this work relies on two device technologies: fully-depleted silicon-on-insulator (FD-SOI) and spin–orbit torque (SOT) MRAM. The low power consumption of the two technologies

complement each other in order to create a highly-efficient final design.

2.2.1 FD-SOI

In CMOS process nodes, the two dominant technologies have been fin field-effect transistor (FinFET) and FD-SOI (Figure 3). FinFET is dominant in industry and performance-critical designs due to its reliability and improved on-off ratio, and FD-SOI is useful for modern low-power designs, as one is able to dynamically adjust the threshold voltage of the gates post-fabrication. Currently, foundries are beginning to switch towards gate all-around (GAA) and nanowire technologies beyond FinFET [33], and it is unknown if FD-SOI will be able to continue to be competitive for logic technologies, though it maintains a strong advantage for specific application domains [34], such as ultra-low power designs.

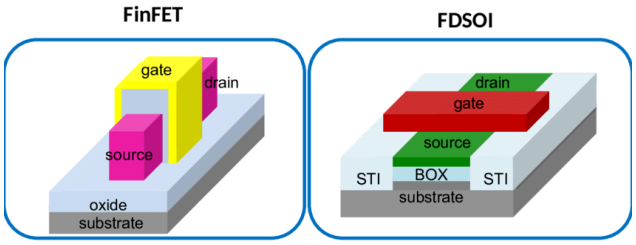


Figure 3: Device geometry comparison between FinFET (left) and FD-SOI (right) [35]

My lab and its industrial partners have access to a commercial 22 nm FD-SOI process, which will be used for our design. As the name suggests, SOI utilizes a buried oxide layer on top silicon to isolate the channel from the bulk. SOI comes in two flavors: fully-depleted (FD) and partially-depleted (PD) (Figure 4). In PD-SOI, the channel region is relatively thick, and there are regions where carriers are not fully-removed from the channel; the depletion regions are not merged. In FD-SOI, the channel is sufficiently thin that it remains fully depleted, allowing for enhanced carrier-conduction and control, and the near elimination of drain-induced barrier lowering, allowing shorter channel / smaller area devices.

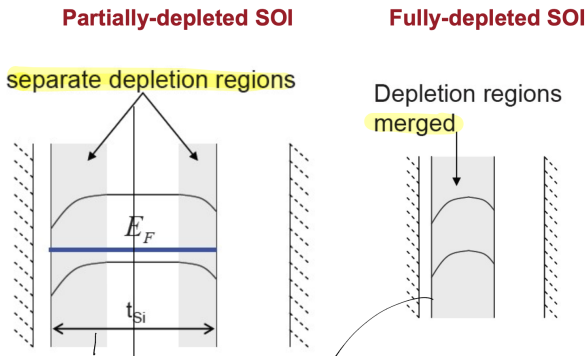


Figure 4: Comparison of depletion regions for partially versus fully depleted SOI. The lines represent the Fermi level (E_F) across the channel.

The process design kit (PDK) provided by our 22 nm FD-SOI foundry has two flavors of CMOS configuration: conventional well and flip well (Figure 5). In the conventional well configuration, the CMOS

cell maintains the same structure as without the buried oxide: n-channel MOS (NMOS) in a p-type well (p-well) and p-channel MOS (PMOS) in a n-type well (n-well). In order to increase the threshold voltage of the device, a positive voltage should be applied to the n-well of the PMOS and a negative voltage should be applied to the p-well of the NMOS device. This is known as reverse body biasing (RBB) Because the n-well is at a higher voltage than the p-well, the diode formed by the doped regions remains in reverse bias, and no conduction takes place. However, if one were to try to reduce the threshold voltage of the device by applying a negative voltage to the n-well and a positive voltage to the p-well, the diode created by the wells is forward biased and conduction takes place between the diffusion regions [36]. This forward-biasing of the well diode should be avoided at all costs. Thus, because the well region below the buried oxide does not have to correspond with the diffusion regions which contact the source/drain of the transistor, one can create the flip well configuration where the NMOS is placed in a deep n-well and the PMOS in a deep p-well. Thus, when forward body biasing (FBB) (negative to n-well, positive to p-well) is applied, the diode formed between the two wells remains reverse-biased, preventing forward conduction through the deep wells. Therefore, one can either increase or decrease the threshold voltage beyond typical, but once the deep well has been determined, it cannot switch between FBB and RBB modes.

With FBB, one can decrease the threshold bias to reduce the on-resistance of the transistor (R_{on}), with the cost of also decreasing the off resistance (R_{off}). Therefore, with FBB one can increase the performance with a cost of energy efficiency of a given group of transistors, and with RBB one can increase the efficiency at the cost of performance. In my design, we use both conventional-well and flip-well devices to provide sufficient current for the write operation while reducing the leakage associated with the read operation. Further, the ability to dynamically control the threshold voltage ends up being crucial for the ability to write the SOT-MRAM devices, while maintaining high energy efficiency for the read operation.

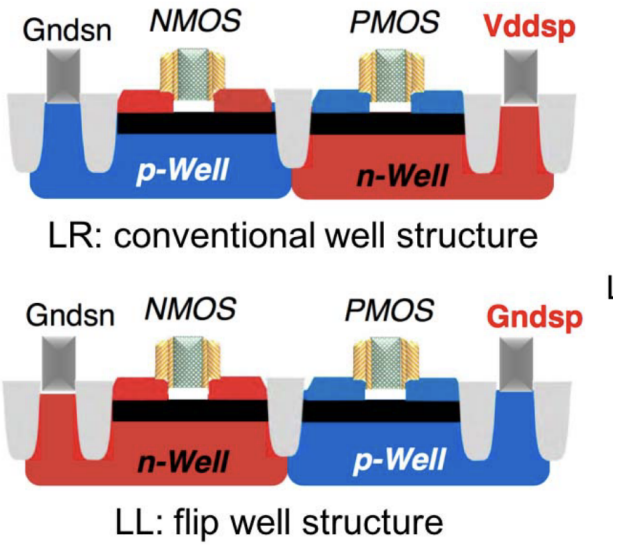


Figure 5: Comparison between flip-well and conventional-well structures for FD-SOI [37]

2.2.2 SOT-MRAM

Spin-orbit torque MRAM differs from other magnetic random access memory (RAM) technologies due to its efficient switching and complete isolation of its read and write paths. Most magnetic MRAM technologies encode the state '0' or '1' by the magnetization direction of a nanomagnet, and consequently the resistance of its associated read path.

The first integrated magnetic RAM utilized the magnetic fields induced by electric current flowing through row and column wire to write the state of a specific cell [38]. The induced fields were insufficient on either an individual row or column, but at their intersection, they added together to switch the state of the free nanomagnet, encoding the written data (Figure 6a). The resistance of the devices magnetic tunnel junction (MTJ) stack changes due to the new magnetization as per the tunnel magnetoresistance (TMR) effect, and it is read via an access transistor. The use of external fields to switch the nanomagnet was very energetically costly, and quickly spin-transfer torque (STT) MRAM was developed.

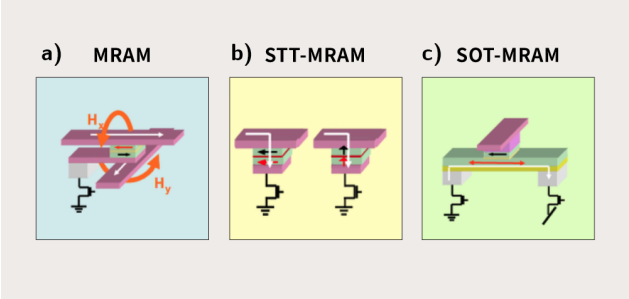


Figure 6: Comparison of the three leading types of MRAM a)Field-Based MRAM, b)STT-MRAM, and c) SOT-MRAM [39]

Now utilized in consumer-facing products [40], STT-MRAM (Figure 6b) uses the spin-transfer torque effect to induce torque on, and eventually switch, the free nanomagnet. Electric current flows across the STT-stack, and becomes spin-polarized in the fixed layer. Then, as it enters and relaxes in the free-layer, it imparts magnetic torque sufficient enough to switch the magnetization of the free magnet. This method is much more efficient on write than traditional MRAM, however, it removes the isolation between read and write paths, as one reads the differential resistance with the TMR effect. Therefore, one cannot isolate the read/write circuitry, requiring increased overhead and a careful design to prevent read-disturbance. Despite this, STT-MRAM is leading the magnetic memory market.

However, there exists an even more efficient type of magnetic memory which has complete isolation of the read and write paths: SOT-MRAM (Figure 6c). By using the spin orbit torque effect, SOT-MRAM (Figure 6c) flows electric current in $+x$ through a material which exhibits a non-zero spin-hall angle, causing excess spin accumulation at the interface. This forces the magnetization to be in $\pm y$. Finally, an external magnetic field in $+x$ biases the pinned magnetization to go in $\mp z$ depending on the cross product of the magnetization and current direction, allowing for deterministic switching.

The electronic current required for this switching depends on many properties, including the spin-hall angle of the SOT layer, saturation magnetization of the free layer, the volume of the free layer squared, and the time for which the signal is applied: $I_{\text{switch}} \propto M_{\text{sat}} V_{\text{free}}^2 / \tau_{\text{pulse}}$. The relationship between switching

frequency (or inverse to pulse length) and critical switching current is characterized empirically in Figure 7. SOT-based switching is thermally and stochastically mediated, as a region of the magnetization at the interface must spontaneously switch before the spin accumulation can propagate the domain wall to switch the entire nanomagnet. This means there must be lots of heat to switch at high frequencies, and there should be a minimum required current to switch at low frequencies. The current required to deterministically switch at a given frequency is known as the critical current and the minimum switching current in low-frequency is called the activated critical current, and is a critical parameter for the array design. The relatively high activated critical current has caused our and our collaborators' transistors to be relatively large compared to the footprint of the MRAM itself.

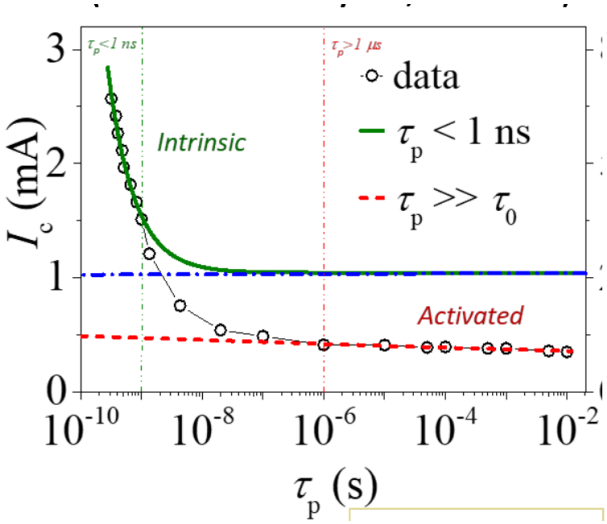


Figure 7: Relationship between critical current and pulse duration for an SOT-MRAM device [41]

2.2.3 Co-Integration

Both FD-SOI and the SOT-MRAM will be integrated onto the same piece of silicon. Our industrial partner for the FD-SOI process will create the active elements, and will deposit only the first four metal layers. Then, the chip will be sent to our second industrial partner, who will then proceed to deposit the layers of magnetic materials needed for the SOT-MRAM, along with three more metal layers and the pads which allow for the chip to be connected to the outside world. This co-integration allows us to combine transistors for digital and analog signal processing/control below the SOT-MRAM on the same piece of silicon. The majority of users of this process are looking to develop traditional memory cells, where a low resistance area (RA) product is desired for faster writing operations necessary for highly-performant memory. However, as we will see, write operations are rare for neuromorphic arrays, and one would typically prefer a high-RA product design to reduce read current and increase energy efficiency for inference. My design actually uses a low-RA product MRAM element, in order to be in the first batch of manufactured chips. Despite this limitation, I was able to create a competitive design.

2.3 Neural Networks

As alluded earlier, our crossbar array will perform the simple dot-product operation in-memory. By performing several of these in parallel, this allows us to compute a vector-matrix product in the analog domain. We then choose to apply a non-linear activation function to the output of this result. This set of operations forms the heart of simple neural network (NN) inference; that is, given a trained set of weights and new input data, the NN can infer conclusions based on the new data and its previous training. This differs from NN training, when training data (and sometimes labels) are provided to a NN in order to optimize its parameters, minimizing the error between the learned function, the ideal function belonging to the space of the model, and hopefully the true function. This master's thesis report only discusses inference for binarized neural networks (BNNs); however, it could be paired with learning algorithms in the future to create an efficient on-chip learning system. Thus, in this section I will discuss the basics of inference, training, and the common approaches to accelerating them with dedicated hardware.

Neural networks are a particularly powerful class of models which can parameterize any non-linear function given a few constraints on their depth or width (see universal approximation theorems) [42], [43]. Neural networks contain several important parts which allow them to accomplish this seemingly incredible task. The standard NN architecture [4] contains several layers of nodes called neurons, which accept data from input sources or other neurons. Each neuron in a layer is fully connected to each neuron in the next layer via edges called synapses. Data flows through this architecture from the left to the right, and each neuron sums the product of the weights on each synapse and the output of the previous neuron (or input to the model in the case of the first layer), and applies a non-linear activation function to the data before sending it to the next layer. After the activation has been applied, the output of a given neuron is called its activation. At the output layer, one often uses the softmax function to normalize the output activations to a probability distribution, when the outputs correspond to a group of classifications.

Binary Neural Networks (BNNs) are a subset of NNs that restrict the range of possible weights and activations from the space of real numbers (or floating point with a given precision) to binary values. This restriction inherently requires an increased number of neurons to compute functions at the same accuracy of a traditional neural network, however, with sufficient size, they can be as accurate as traditional neural networks [44]. Rather than requiring expensive floating point operations, BNNs can directly use the rules of binary logic for computation. Therefore, to propagate data through a BNN, one must multiply the set of input activations by the weights for each neuron, sum these products, and apply a threshold function. In binary, the multiplication and sum operations map directly to XNOR and popcount operations respectively. For the activation, the most common and simple function is a simple threshold (+1 if above a threshold, and -1 if below). However, some non-monotonic functions such as a window (where the output is +1 if within a range, and -1 if outside the range), can outperform the simple threshold as it extracts more information from the preactivation.

[4] Here we discuss only the architecture of multilayer perceptrons (MLPs) which have only fully-connected layers which feed forward. Different tasks such as image processing or translation have more complicated layers such as convolution or attention mechanisms, but still matrix operations remain the core part of all neural network architectures

2.4 Neuromorphic Acceleration

Therefore, one may look to find more efficient ways of performing inference and training that either increases the performance or reduces the energy per operation. The process of designing hardware fit for a specific algorithmic use case to improve performance metrics is known as hardware acceleration. When it is applied to improve the performance of neural networks, it is known as neuromorphic acceleration. In this section, I outline the state of the art for neuromorphic acceleration for inference and for learning.

The concept of analog CiM naturally arises for neuromorphic acceleration. Not only does the system benefit from the efficiency of the analog domain, but it maintains the flexibility of weight modification post-fabrication, and it avoids the memory bottleneck problem altogether. The simplest and most-well studied analog CiM system is that of the memristor crossbar (Figure 8). Memrisistors are a theoretically predicted fourth fundamental element which have memresistance (M) in Ohms that relates magnetic flux (Φ) to electrical charge (q) ($M = d\Phi/dq$), such that the memresistance gains hysteresis with respect to a varying input signal. resistive RAM (ReRAM) devices maintain some characteristics of an ideal memristor, and commonly use a conducting filament which can be grown or shrunk via electrical current to program their resistance. When arranged in a crossbar (Figure 8), inputs can be supplied as voltages, and the current into a given node will be given as the product of the input voltage and the conductance of the memristor via Ohm's law. Then, by Kirchhoff's current law, all of the currents to a given neuron will sum, and the dot product between the input voltage and the weights can be computed completely in-analog.

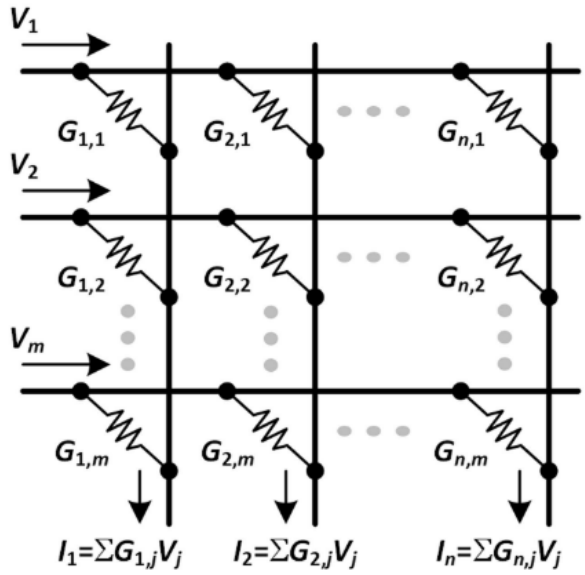


Figure 8: Depiction of the working principle of a memristor crossbar array. Input voltages are provided, and the output currents via Ohm's law are equivalent to a multiplication by the conductance matrix of the memristor states [45]

This powerful concept is used to massively reduce the power consumption of inference circuits, however, it is not without its drawbacks. Notably, signal degradation and noise significantly alter the operation of analog systems, and the accuracy of the analog computation is often lower compared to the

predicted accuracy given by the offline training. Further, the variation between fabricated devices can significantly alter the effective versus idealized conductance for each ReRAM element, which can affect neural network performance. Finally, the activation function still needs to be computed given the output current, and the current needs to be converted to a voltage for it to be effectively propagated to another layer. This typically requires expensive and careful analog design.

Fortunately, if one transitions from the continuous-valued voltages and weights associated with filament-based memristor arrays and instead implements a binary neural network in-analog, many of these issues disappear. This is because the signal energy and separation between neighboring states tends to be much greater when working with discrete signals. Therefore, previous works [46], [47] have implemented STT-MRAM into crossbar arrays. These MRAM crossbar arrays implement BNN functionality natively, as the MRAM cell can only store a high or low resistance state based on the magnetization of the free layer. Most MRAM crossbar arrays have utilized the same sum-of-currents approach to sum the dot product in parallel. However in 2022, Samsung research laboratories showed how STT-MRAM devices can be placed in series, and the sum can be calculated by the sum-of-resistances via time-to-digital conversion (TDC) [45]. Therefore, there exist two ways to connect crossbar arrays to calculate the dot product: in-parallel using sum-of-currents, which is best suited for high-resistance devices, and in-series using sum-of-resistances, which is best suited for low-resistance devices. This is because applying a nominal voltage to a low-resistance element creates too much current and therefore power. Because the Elmore delay of the column is proportional to the sum of resistances (with a sufficiently sized output capacitor), the TDC method measures the Elmore delay of a column using a digital timer to measure the resistance of the column, and therefore the sum of the dot product of activations. These ADC and digital-to-analog converter (DAC) methods are the Achilles heel of analog computation—the beauty and simplicity of summing currents disappears when one performs the necessary step of connecting it to digital hardware.

3 Inference Chip Design

The bulk of my thesis period, and thus this report, will be devoted to the description of the design process which went into my SOT-MRAM inference array chip. The components of the chip can be visualized at a high-level in Figure 9. The core of the array (represented in grey) was primarily completed during my 2A internship, though it was slightly modified during the course of my 3A thesis period. In blue, the control circuitry was entirely specified and tested during my 3A period. The ADC circuitry in orange was the most complicated part of the project, and most of my time of the 3A period was spent optimizing and ensuring the functionality of the readout process. Finally, digital design had to be done to create the serial programming interface (SPI) block to connect the chip to a microcontroller for testing, and finally the pads themselves in pink, representing the physical pads that will specify how many connections the chip will be able to have to the outside world. The design of the chip has been following our Gantt chart (Figure 37), in order to meet the tapeout deadline of December 16th.

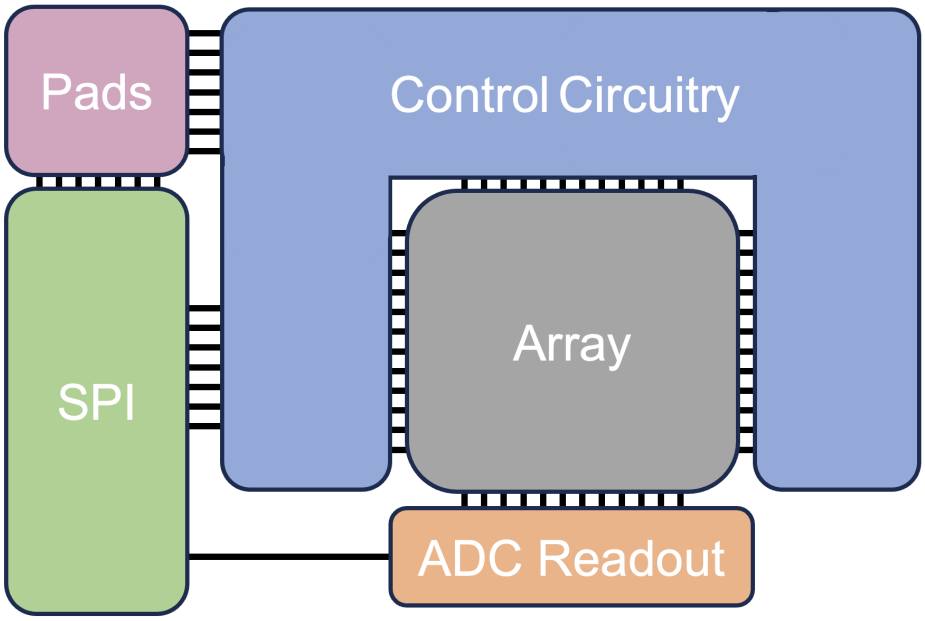


Figure 9: Chip design overview. The major components necessary to complete the tapeout are shown above. The array (grey) is connected to the control circuitry (blue) and the ADC readout (orange). The registers for both the control circuitry and the ADC are connected to the SPI interface (green), which is interfaced off-chip by the pads (pink).

3.1 Cell

This section of the report discusses the design of the core inference array, which utilizes SOT-MRAM devices to compute the vector matrix product, taking in a voltage and outputting a resistance presented to the ADC unit. While the majority of the array design occurred during the 2A internship, it needs to be recapped here to properly understand the context of the periphery which was developed during this 3A period.

The proposed array design is composed of 64 neurons, each with 64 synapses, resulting in a size of 64×64 cells in the entire array (Figure 10). Each of these cells is composed of two SOT-MRAM devices, one on the left denoted as the positive, and one on the right denoted as the negative side. Further, the unit cell contains five total transistors: two transistors to control the inference operation (inference line (IL) and inference line bar (ILB)), two to control the write operation (word line (WL) and word line bar (WLB)), plus one transistor to connect the programming lines for the write operation (word programming line (WPL)) (Figure 11). At the end of the 2A internship, all of the transistors belonged to the thin oxide variety, and they used the super-low voltage threshold (SLVT) variant, in order to have enough current to meet the activated critical current of the MRAM, in order to ensure the controllability of the entire array. Further, the size of the IL/ILB and WL/WLB transistors was set to 3μm and the WPL to 2μm to ensure that the on-resistance was sufficiently small to have good inference performance for the expected TMR, and to once again ensure that enough current could be passed to write the cells deterministically.

To perform the inference operation, the resistance presented to the neuron changes as a function of

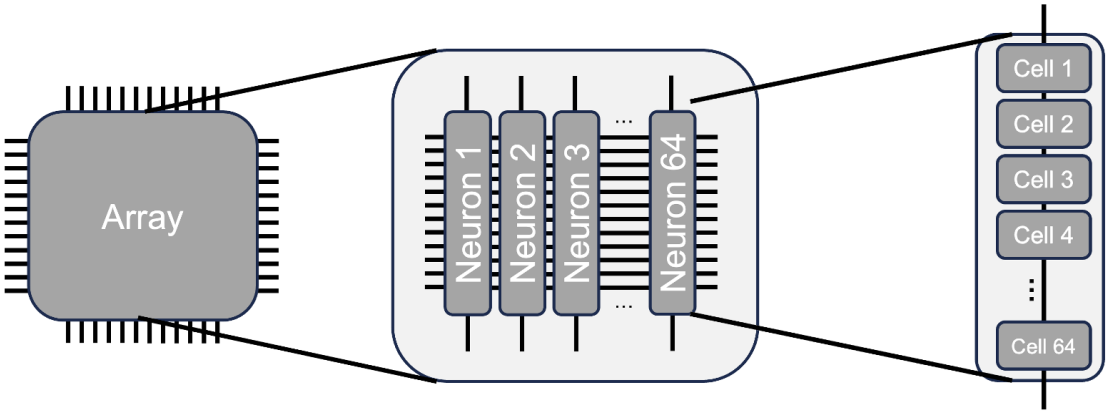


Figure 10: Array structure. The array (left) is composed of 64 neurons (middle), each of which are composed of 64 cells (right)

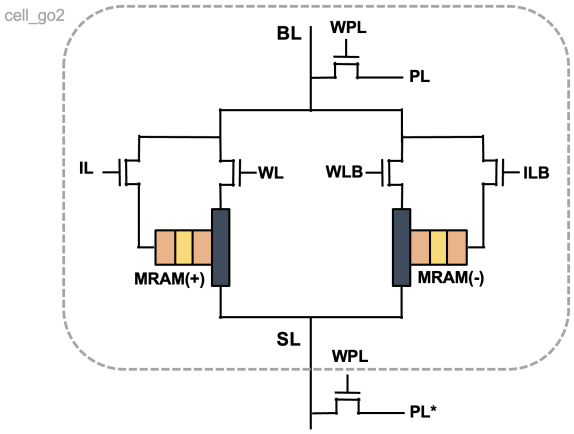


Figure 11: Cell structure. The unit cell of the array is surrounded by a line in grey. Each cell has 1 PL transistor, 2 IL/ILB transistors, 2 WL/WLB transistors, and 2 MRAM units.

the encoding of the cell and the applied activation. Figure 12 helps visualize the read process at the cell level. When the inference line for the given row is +1, the cell in the positive side of the array is selected to present its resistance to the stack. Similarly, when -1 arrives, the negative side's MTJ resistance gets added to the stack. Because the cell is always encoded differentially (the positive side will always be oppositely encoded to the negative side), this results in a signed multiplication of the activation and the cell encoding to determine the presented resistance, which is well represented in Table 1.

A similar process occurs to write any individual cell. The WL/WLB transistors select which of the two MTJs will receive current, and the WPL lines are set to either supply voltage (V_{DD}) or ground (GND) in order to supply voltage to either side of the SOT material depending on the desired cell encoding (Figure 13). The write operation must occur in two passes, as the multiplication operation breaks down if both MTJs are encoded into the same state.

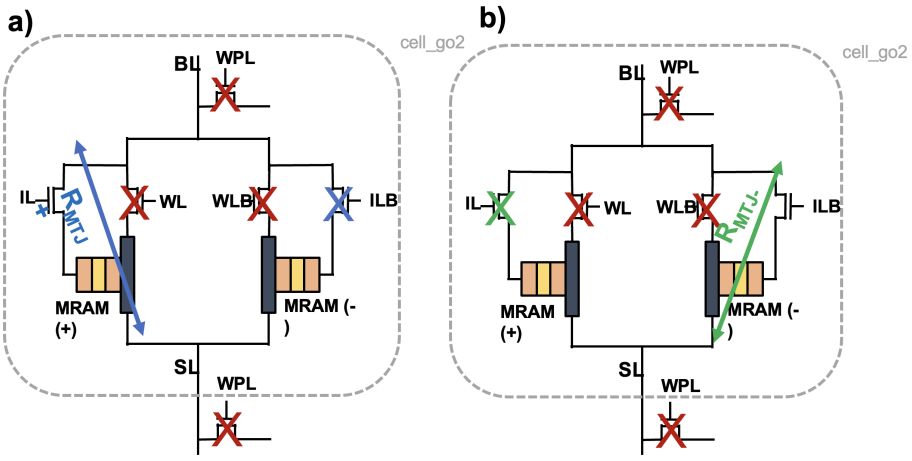


Figure 12: Cell read operation diagram. (a) Read process when IL=+1, the + side MTJ resistance is connected in-series to the other cells (b) Read process when IL=0, the - side MTJ resistance is connected in-series to the other cells

Read Line	Cell State	Resistance
$IL = V_{dd}$ (+1)	$R^{+}_{\mathrm{MTJ}} = R_{\mathrm{H}}$ (+1)	R_{H} (+1)
$IL = V_{dd}$ (+1)	$R^{+}_{\mathrm{MTJ}} = R_{\mathrm{L}}$ (−1)	R_{L} (−1)
$IL = 0$ (−1)	$R^{-}_{\mathrm{MTJ}} = R_{\mathrm{L}}$ (+1)	R_{L} (−1)
$IL = 0$ (−1)	$R^{-}_{\mathrm{MTJ}} = R_{\mathrm{H}}$ (−1)	R_{H} (+1)

Table 1: Readout table for the MTJ cell.

3.2 Array

By now looking at the entire array, one can begin to see and plan how the control circuitry and the ADC unit may interface to it.

3.2.1 Inference

Figure 14a demonstrates how the resistance will be presented to a theoretical ADC at the bottom of the array. To perform the multiply and accumulate operation, x_i is presented to each row of the array, and is encoded as $+1 := (IL = V_{dd})$, $-1 := (IL = 0)$. Thus, the resistance presented by each cell is either R_L or R_H, depending on the multiplication of the weight by the activation: $w_i \cdot x_i$ (see Table 1). As the resistances sum in series, this gives a total resistance equal to the sum of the individual products. If one defines y to be total number of +1s resulting from the multiplication in the neuron, then the equation can be simplified to remove the summation term (Eq. 1). Furthermore, because the high resistance state can be rewritten as a factor of the low resistance state and the device TMR ($R_H = R_L \cdot (1 + TMR)$, the equation can be simplified, adding together the common mode elements (Eq. 2). One can can clearly see that the TMR is crucial to having a measurable signal. In the real devices, the common mode (fixed) resistance of the column is greater than $64 \cdot R_L$ because of the on-resistance of the transistors.

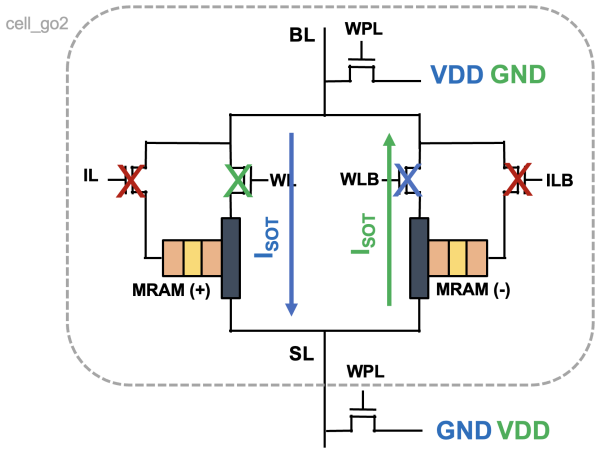


Figure 13: Cell write operation diagram. The first write pass (blue) sends $+I_{SOT}$ through the + side MRAM, and the second pass (green) sends $-I_{SOT}$ through the - side MRAM.

$$R_1 \approx \sum_{i=1}^{64} \left(R_H \frac{1+w_i x_i}{2} + R_L \frac{1-w_i x_i}{2} \right) = R_H y + R_L (64-y). \tag{1}$$

$$R_1 = R_L \left[64 + \text{TMR} \cdot y \right]. \tag{2}$$

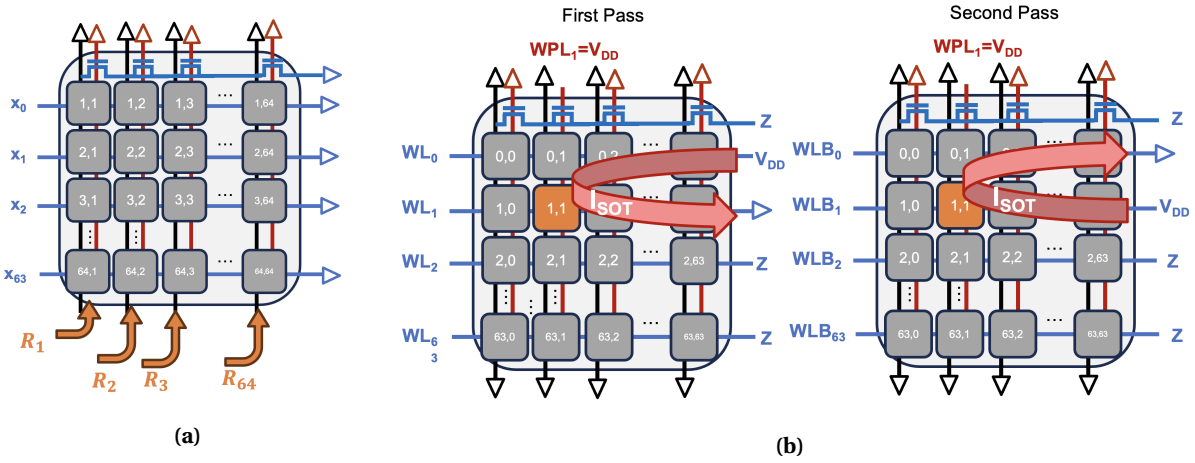


Figure 14: (a) Array level read operation; (b) Array write operation.

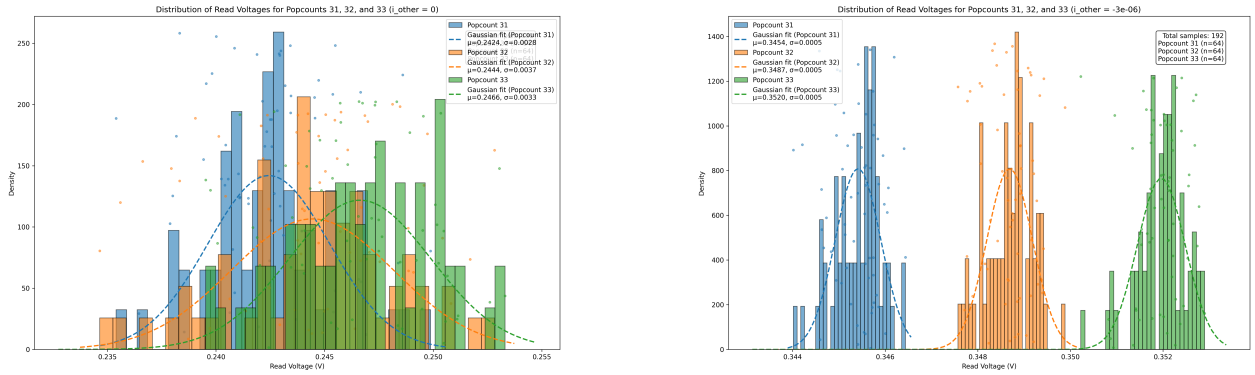
3.2.2 Writing

Figure 14b demonstrates how the cell write operation needs to take place within the context of the entire array. As mentioned earlier, the write operation takes place in two passes, one to write the positive side of the cell and one to write the negative side of the cell. The design of the array exploits the fact that the WPL columns run perpendicular to the WL/PL rows in order to select individual cells; if either WPL is off or PL is in high-Z, then the cell will not be able to receive any current. Thus, only cells where WPL, WL/PL are

active can be written, and when only one of each is active, only one cell is selected. Thus, one can iterate the chosen cells across the array in several passes to write to all of the memory elements.

3.2.3 Leakage Problem

During the 2A internship report, I presented Figure 15, which explained how due to leakage through the SLVT transistors, one had to perform all inference operations in parallel in order to properly differentiate neighboring states. However, at the start of my 3A period I presented these results to my group, and we quickly realized that this property of my array, while not detrimental to final operation, would make the testing of my chip a nightmare. One would be unable to read an individual column without applying current to all other columns. Therefore, the first task of my 3A internship was to resolve this problem. Fortunately, by switching from the SLVT to normal low voltage threshold (LVT) transistors, the subthreshold leakage was reduced by an order of magnitude, allowing one to properly differentiate neighboring popcounts by looking at the voltage of the column with a current source for the typical typical corner. After this change, the distribution of voltages were separate, even for the worst case corner (slow-slow) (Figure 16), indicating that indeed the solution of using the LVT transistor solved the problem.



(a) Current only to read column

(b) Current to all columns

Figure 15: Distribution of read voltages for popcounts of 31 (blue), 32 (orange), and 33 (green); dotted lines show best-fit Gaussians. (a) Current provided only to the read column. (b) Current provided to all columns.

3.3 Control Circuitry

Looking back at the high-level block diagram (Figure 9), we can see that circuitry is needed to properly control the operation of the array in order to perform the inference operation. In this section, we motivate and describe the design of the control block, and evaluate its performance using transient simulations.

3.3.1 Enable Block

The core of the enable block is formed by two transmission gates, as shown in Figure 17a. The transmission gate is simple analog building block which consists of a PMOS and an NMOS connected in parallel. Because the PMOS can send signals close to V_{DD} without degradation and NMOS can send signals

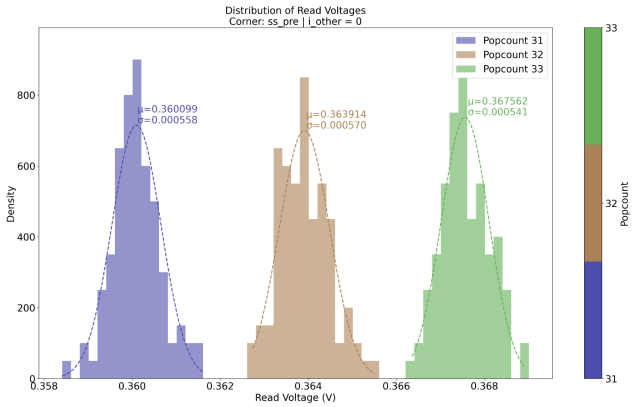


Figure 16: Inference voltage distribution for slow-slow corner

close to GND without degradation, the transmission gate can be used to connect nets of the design to analog voltages/currents with minimal signal loss. Here, by using two of these transmission gates, with independent indexing, we can choose to connect the input of the enable cell to either pad zero, pad one, or we can leave it floating, in a high-Z state. Because the programming lines (PLs) of the array each need to be connected in one of these 3 ways, by placing one enable cell per row of the array, one can individually control the programming lines, allowing for the write operation to take place (Figure 17b).

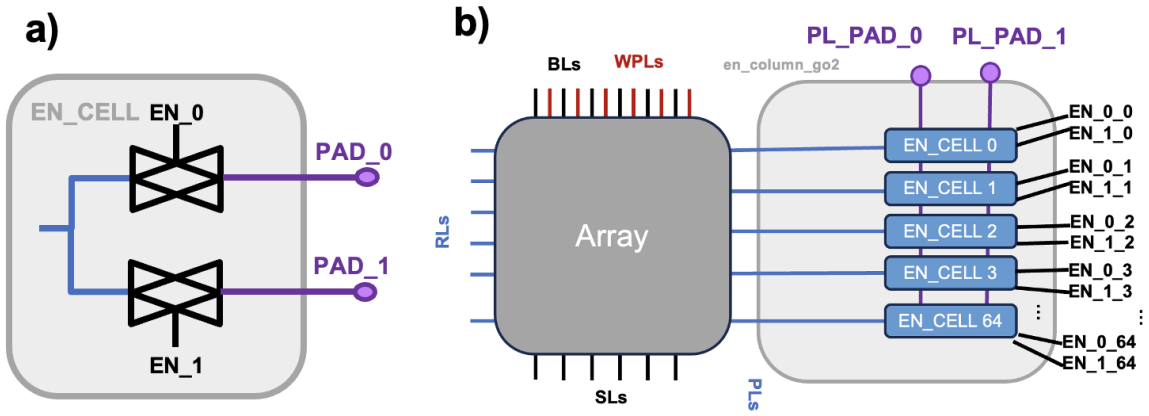


Figure 17: Enable block design. a) The schematic of the enable cell, which is composed of two transmission gates that connects the input to pad_0 or pad_1 b) the schematic of the full enable column, composed of 65 enable cells, connecting any programming line to pad_0 or pad_1

My connection of the programming lines to external pads instead of to on-chip digital voltages such as V_{DD} or GND allows for lots of flexibility during the testing process, which is very often needed when creating demonstrator boards with novel technologies, because the devices always have a higher chance of not meeting specifications. For example, let us imagine for a moment that all of my ADC circuitry is broken for whatever reason. Then, to perform inference, one can simply connect PL_0 to pad zero, and PL_{64} to pad one, and measure the resistance directly with a multimeter, or whichever circuitry we decide to build off-chip. Furthermore, if for example, the TMR were greatly diminished, the flexibility allows

one to measure the resistance of any number of devices to perform inference. One can even measure the resistance of an individual device, in order to validate the functionality of individual MRAM elements. Thus, this control circuitry essentially allows one to probe any two points in the array— a feature essential for testing.

3.3.2 Validation

The ability for the control block to write individual cells without disturbing others was verified using Spectre transient simulations. Furthermore, I performed Monte Carlo simulations for the write process to ensure that even in the worst case, the cell received enough current using the LVT thin-oxide transistors. The distributions are summarized in Figure 18. One can clearly see, that even in the worst case, the current remains over the activated critical current of 450μA even at the 5σ point from the fitted distribution.

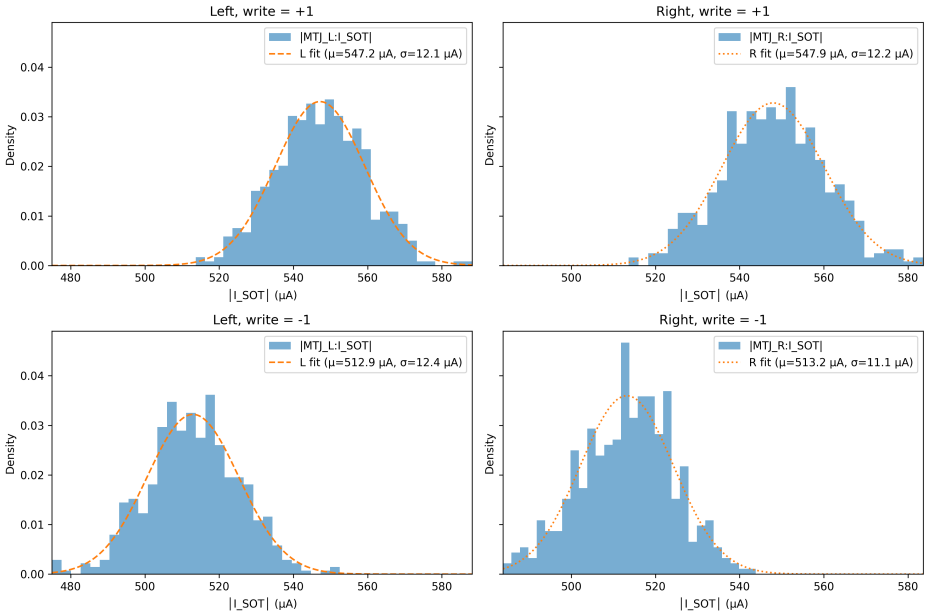


Figure 18: Monte Carlo simulation results for writing a single cell (top left) writing a +1 to the left MRAM (top right) writing a +1 to the right MRAM (bottom left) writing a -1 to the left MRAM (bottom right) writing a -1 to the right MRAM

Secondly, the speed by which the write operation can take place is also of great interest. However, the MRAM model used for the device contains no time-dependent behavior, and switches instantaneously when the activated critical current is reached. Nonetheless, these simulations were important to see if the delays introduced by the capacitances of the array dominated the write time compared to the estimated write duration needed by Figure 7, ~ 100ns. The two-stage writing process was simulated using Spectre in Figure 19. The time necessary to write two cells back to back was ~ 10 ns, which is an order of magnitude lower than the activated critical current, so therefore the write time is not limited by our on-chip circuitry, and one should expect write frequencies in the low MHz range, which is fine as updates to the weights of the array should be uncommon.

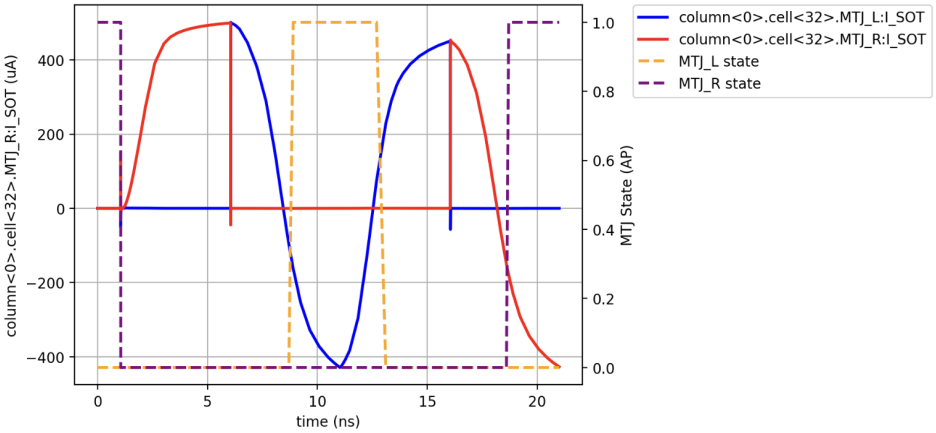


Figure 19: Transient simulation results for writing a single cell. The left (+) side current is shown in red, and the right (-) side current is shown in blue. The MTJ states (yellow/purple) switch from (1,0) to (0,0) to (0,1) to (0,0) and back to (1,0)

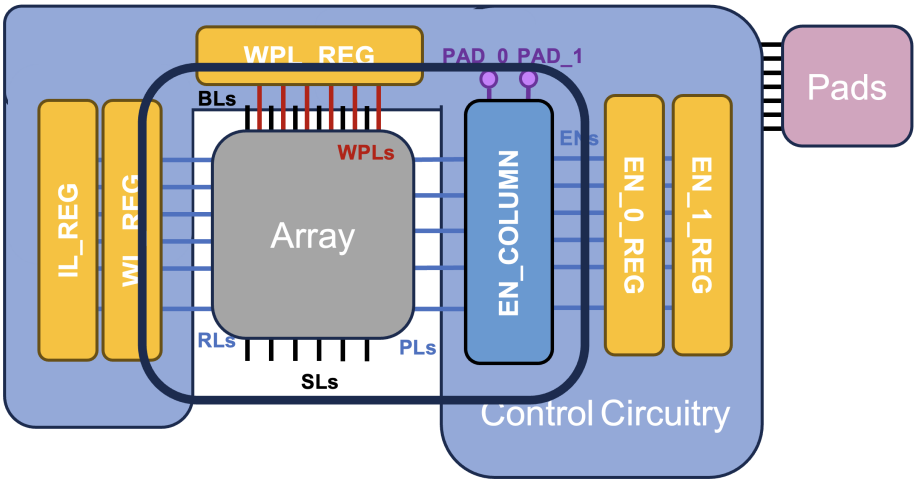


Figure 20: Block diagram of control circuitry with registers, with the control block represented in blue, and the registers represented in yellow. By setting the values in the control registers, the array is configured for read or write.

3.3.3 Integration

Finally, here we discuss how the control circuitry itself will be controlled. Rather than have one pad per IL/WL/WPL/Enable block, it is far more efficient to have these digital voltages be controlled by on-chip registers. Figure 20 best visualizes this register-based control system. The inference line, word line, and word programming line states will be stored in digital registers on-chip. Then the barred versions will be computed from the output of the registers for the inference line bar and word line bar signals. Likewise, the enable signals for the enable cells will be stored in registers for each block. The number of registers for the control system is expected to be $64 \times 5 + 65 \times 2 = 450$, as there are 65 enable cells in the column to properly control the 64 array cells. These registers need to be programmable from an external control unit, therefore section 3.5.2 discusses the digital design that makes this possible using only four pads.

3.4 Analog-to-Digital Conversion (ADC)

The design of the analog-to-digital conversion block was the most difficult part of the chip design. It is easily the most complicated block, and it took the majority of my thesis period to design and validate its performance. In this section, I outline the various ADC methodologies that I attempted, and the reasons that I selected my final design to be manufactured on our upcoming tapeout. From the end of my 2A internship, I suggested several potential options to perform the ADC, but several of them turned out to be inherently worse than others. For example, using a voltage fixed source and measuring the current in order to derive the popcount requires at least as much circuit overhead as using a current source, but instead has the nasty behavior of non-linearity between the measured current and the true popcount. While this could be used to help create a non-linear activation function, for our simple network it simply would have worsened the read performance at higher resistances, as the slope of the $1/x$ transfer function would be lower. Therefore, in this section I outline the methodology to select and optimize the between using a pre-charge sense amplifier and a current source to perform the ADC. All of the decisions and exploration contained in this section occurred in parallel, though I will do my best to present it in a linear fashion, rederiving what brought me towards my final design.

3.4.1 Pre-charge sense amplifier

The first circuit that I attempted to build was that which utilized pre-charge sense amplifier (PCSA). PCSAs are a powerful building block used in circuit design to remove the dependence on an external voltage or current reference. They work by pre-charging a unstable set of two inverters to be at V_{DD}. Then, by connecting the PCSA to two different resistances to ground, two different current strengths will be seen by the PCSA unit. In the column with the low resistance, the output will be driven to ground, and V_{DD} will be seen on the high-resistance column. This circuit not only removes dependence on a reference voltage, but also minimizes power-consumption, as there is no static current consumption, only the transient currents needed to charge and discharge the capacitors.

During my internship, I then explored two ways of implementing the PCSA block—one was fully differential (Figure 21a), and one was one-sided (Figure 21b). To implement the fully differential array, one would need to compare the resistance of two columns of the array by comparing the relative speed of discharge. One column would receive the inputs of the inference operation, and the other column would remain a fixed value, encoding the threshold for the activation function applied to the popcount. When the resistance of the neuron column is higher than the threshold column (or when the popcount is higher than the threshold), the voltage falls more slowly on that side, and the race condition eventually forces the voltage to rise to V_{DD} on that column. A similar condition occurs in the one-sided PCSA implementation (Figure 21b), except that the size of the threshold column has been condensed from 64 MRAM devices to only 6. This reduction in size is only made possible by the binary weighting of each MRAM element to the total delay by exponentially scaling the capacitor size towards the end of the delay element. However, this comes at the cost of losing the match between the two sides, therefore if the fully-matched PCSA design could not be made to work; then the one-sided design would have no chance.

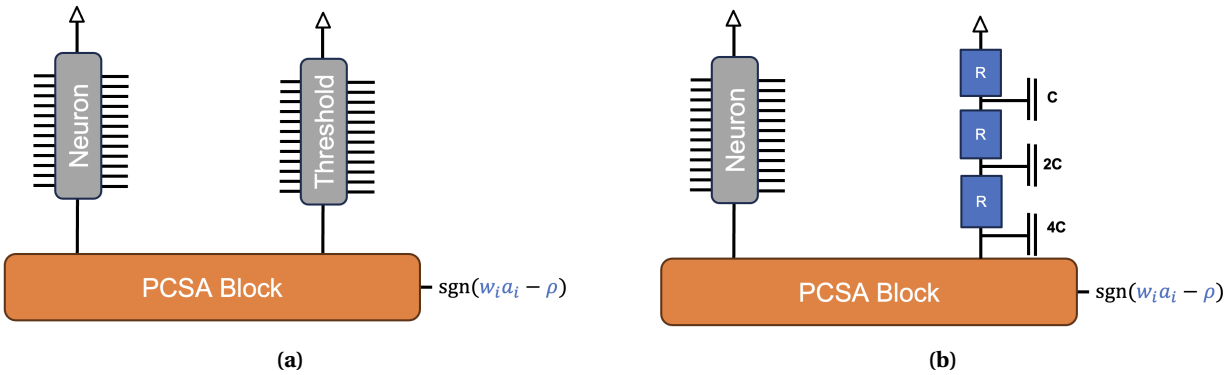


Figure 21: Fully-differential (a) versus one-sided (b) PCSA designs

Therefore, I got to work optimizing and testing the two sided idea, to see if something could be made of it. The general testing methodology is to start simple and see if it can work in an idealized case, then start throwing in more nonidealities, and improve the design until it is robust enough for the real-world. An ideal PCSA unit would only have capacitance at the gate of the transistors in the race condition, such that only the total resistance of the two sides affected the timing. Unfortunately, because of the Elmore delay effect, the charge discharging from the bottom of the column (closer to the PCSA unit) has to flow through all of the other resistors before reaching ground, giving it an outsized contribution to the delay term. For the one-sided design this delay is exploited, but here it is the main driver of error.

Therefore, I started with a typical-typical (TT) corner, and ran several transient simulations with a random spatial distribution of the activations to determine the error rate as a function of the threshold due to Elmore delay, for a constant popcount of 32. Figure 22 summarizes my results for a 100 ns inference time, attempting to differentiate between the popcount and the threshold. Each threshold was tested 100 times, with random distributions of the activations with a popcount of 32. When the threshold is near 32, the error rate reaches fifty percent, indicating near random behavior. However, as one increases the distance between the popcount and the threshold, the measurement becomes more accurate, as the larger difference in resistance is easier to measure. As expected, using a larger capacitor resulted in an overall lower error rate than the smaller capacitor, as most of the delay came from discharging the final capacitor, rather than the parasitics within the columns. The 10 pF capacitor had an error rate of 100%, because it was not able to converge within the 100 ns read time allocated to the transient simulation.

Nonetheless, the error rate for the PCSA system was simply unacceptable, regardless of how well-tuned the output capacitance was. Therefore, I knew that I had to discard both PCSA ideas in order to focus on designing a current-source-based implementation for the ADC.

3.4.2 Current Source

The current source ADC contains two core components: the current source, and the comparator. In order to function well, the current source should have a high input impedance, that way the current changes minimally as the resistance of the neuron changes. Further, a good comparator design will have a high sensitivity and a fast measurement speed. And of course, the blocks should be designed to optimize for

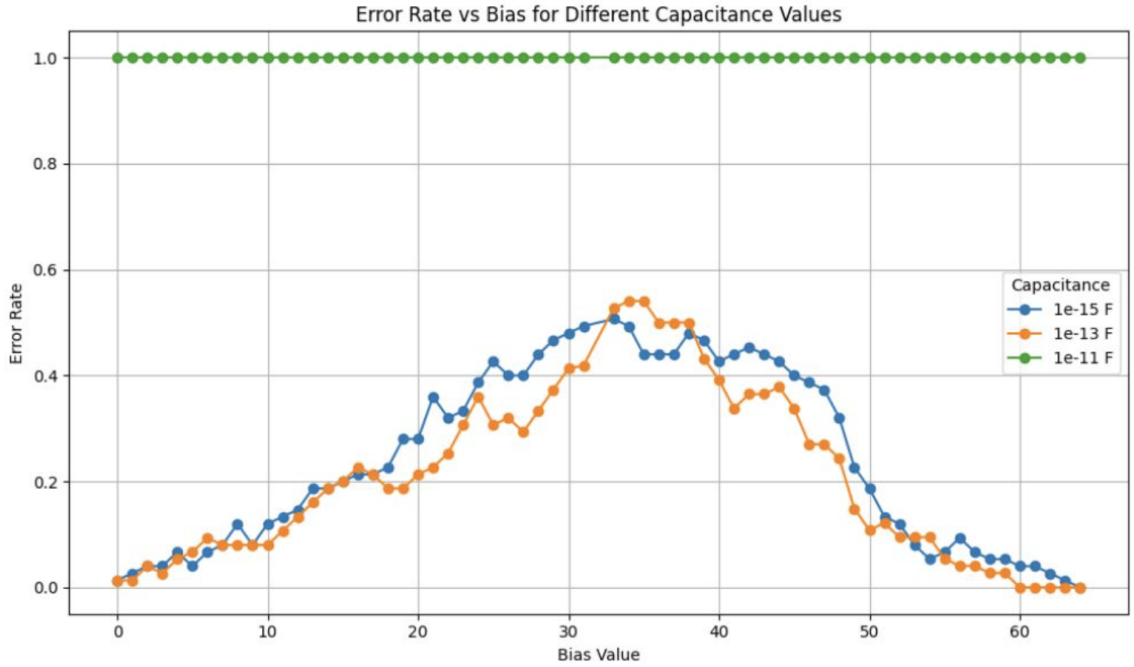


Figure 22: Error rate as a function of bias value for the PCSA ADC methodology tested for output capacitance of 1 fF (blue), 100 fF (orange), and 10 pF (green)

performance per power.

3.4.2.1 Optimization Methodology

Several types of current sources (Figure 23) were tested and evaluated during the course of the design. For each topology discussed, an optimization occurred to select the width and length of all of the involved transistors to optimize both between linearity, signal amplitude, and mismatch. I used three principles to guide my manual exploration of the parameter space, as well as automated optimization algorithms such as differential evolution and Nelder Mead to search within reasonable bounds given by these principles.

The first guiding principle to the optimization of the current source was Pelgrom's Law (Eq. 3). It simply states that the standard deviation of threshold voltage and current of a transistor is inversely proportional to the square root of the transistor gate area. This makes sense from a foundational level, as if the variations are proportional to the width or the height, increasing the physical size of the gate would proportionally reduce the variation as a function of the total area. The clear takeaway is that bigger transistors are better to reduce mismatch, however, this comes at the cost of chip area.

The second principle is to optimize for the output impedance of the current source (Eqs. 6 and 7). The higher this impedance, the less sensitive the output current of the source is to variations in the output resistance, and therefore the transistor drain to source voltage (V_{DS}). The channel length modulation parameter (λ) is constant across the relevant PMOS cells available in the PDK, therefore the best way to increase the impedance is to increase the length of the transistor.

However, this reduces the amount of current that can be carried across the transistor, leading to the

third principle, transistor on-resistance. The drain current of a transistor in saturation is given by Eq. 4, therefore to keep a fixed current amount if one reduces the W/L, the transistor must increase V_{DS} to compensate, thereby increasing the "on-resistance" of the device (Eq. 5).

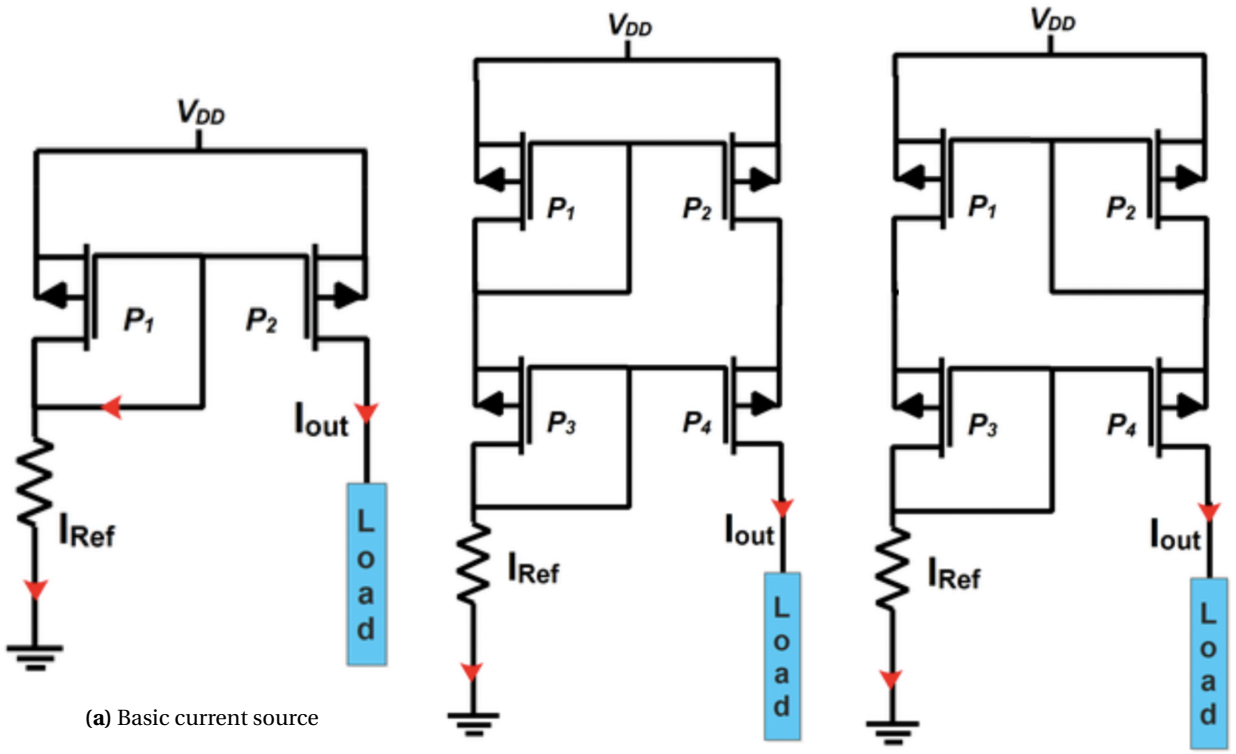
For each topology tested, the search space was well explored using soft constraints such as a maximum transistor size of $20\mu m^2$, a minimum current through the column of $\sim 3\mu A$, and a linearity less than 5%.

$$\sigma(\Delta V_{TH}, \beta) \propto \frac{1}{\sqrt{WL}} \tag{3}$$

$$I_D = \tfrac{1}{2}\mu C_{ox}\frac{W}{L}(V_{GS} - V_{TH})^2(1 + \lambda V_{DS}) \tag{4}$$

$$R_{on} \approx \frac{1}{\mu C_{ox}\frac{W}{L}(V_{GS} - V_{TH})} \tag{5}$$

3.4.2.2 Current Mirror Topologies



(a) Basic current source

(b) Cascode current mirror

(c) Wilson current mirror

Figure 23: Current mirror implementations: (a) basic current source, (b) cascode current mirror, and (c) Wilson current mirror. Figure modified from [48]

Initially, I naively attempted to use the traditional current mirror topology (Figure 23a). However, this failed almost immediately. The large range of resistances possible for a given column makes a large

swing in voltage, which changes V_{DS} of the active transistor, modifying its current due to channel length modulation. I found that the current reduced by over 50% from the lowest resistance to the highest resistance state, which was a non-linearity which was unacceptable for the goals of this project. Increasing the current increases the voltage signal for a given resistance, however, I noticed that when the drain voltage begins to reach rise beyond $V_{DD} - V_{th}$, the PMOS gets put out of saturation, and the current source leaves saturation, destroying the read operation. Therefore, the maximum output voltage of a column is equal to $V_{DD} - V_{th}$.

$$R_{\text{out, simple}} \approx r_o = \frac{1}{\lambda I_D} \tag{6}$$

From my analog design coursework, I knew that the most common way to boost the output impedance of a current mirror was to add a cascode stage (Eq. 7). However, cascoding comes at the cost of increasing circuit area and reducing the maximum voltage of the neuron by an additional transistor threshold voltage (V_{th}). The cascoding solution worked extremely well for the typical-typical corner with no noise and no mismatch, allowing an approximately 1% change in current over the whole resistance range of the neuron, which resulted in a maximum shift of 12.8 mV of the output voltage when compared with an ideal current source. This means that neighboring popcount voltages nearer to 64 will be closer together due to the reduction in V_{DS} and therefore current, and those closer to 0 will be further apart due to more current per the resistance. I will talk more later about adjusting for these nonlinearities, but overall a shift of 0.2 mV per popcount allows the proper reading out the output voltage by an on-chip comparator.

$$R_{\text{out, cascode}} \approx g_m r_o^2 \tag{7}$$

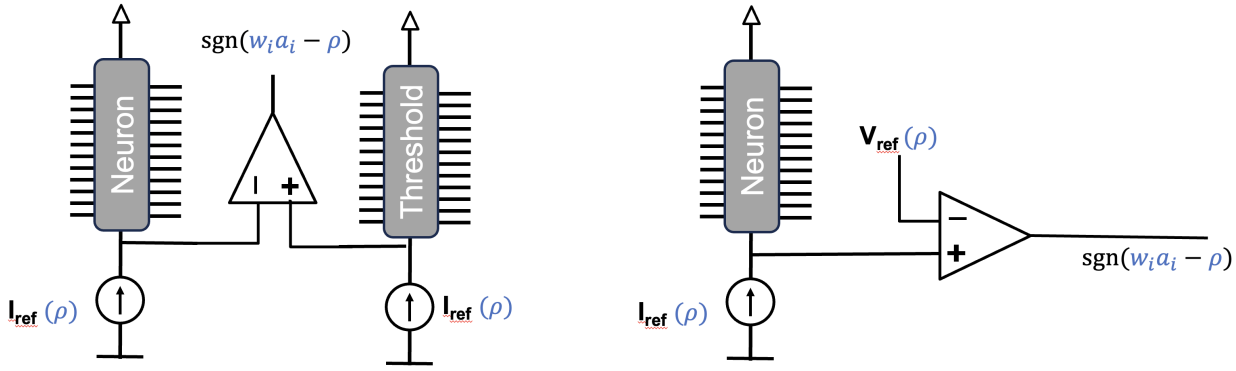
With high linearity across the entire range of impedances presented by the neuron, I began to move forward with the cascoded current mirror design. I performed corners simulations, which mildly reduced the performance in fast-slow, slow-fast, and slow-slow corners for NMOS-PMOS respectively, so things were looking promising. I even ran global Monte Carlo and added noise to my transient simulations which were well absorbed by the high impedance of the cascode mirror. However, it is when I began to run mismatch simulations that my hopes of an easy analog design were quickly dashed.

The mismatch between any two of my current mirrors was staggering: the variation between the currents had a mean of $\sim 0.25\mu$A, which led to a mean difference of ~ 20 mV between two columns with the same popcount. Without reducing this mismatch, it would have been impossible to differentiate neighboring popcounts with voltage differentials of ~ 3 mV as was predicted by the maximum read current of 3μA for the thin oxide transistors.

Therefore, I did some research and found an alternative topology which had the potential to solve my mismatch issue. The Wilson current mirror topology boasts a massively reduced mismatch between the original and mirrored currents [48]. Therefore, I implemented the Wilson mirror (Figure 23c), and tested its output impedance and mismatch. Unfortunately, its output resistance had significantly degraded compared to the the cascode despite using the same physical transistor dimensions.

To decide between these two current source topologies, I had to consider how it would be implemented

in the larger ADC structure. At this point in the design, there were two leading structures: fully-differential (Figure 24a) and one-sided (Figure 24b), just like the PCSA. The fully differential design was intriguing to me not only due to its symmetry, but due to the fact a second amplification stage could be placed before the comparator, to completely remove the "common-mode" $64 \cdot R_L \cdot I_d$ voltage signal on the output, effectively allowing one to multiply the TMR of the MRAM by some value α. However, this also magnifies input referred noise and any mismatch between the two mirrored current sources. In fact, the differential design had no possible way to adjust for mismatch. This lack of flexibility led me to choose the one-sided design, even though it requires a voltage reference, as one could theoretically adjust the threshold to compensate for either mismatch or finite output impedance of the current source.



(a) Differential current mirror

(b) One-sided current mirror

Figure 24: Two current-based ADC configurations: (a) differential current ADC, and (b) one-sided current ADC

3.4.2.3 Low-Dropout Regulators (LDOs)

The narrative is relatively unfulfilling, but it was a completely different form of current source which was eventually implemented into my final design: the low-dropout regulator (LDO). LDOs are most often used when one wishes to step down one on-chip voltage to another, with minimal headroom. The low-dropout means that the output voltage of the regulator can be extremely close to the supply voltage, up to $V_{dd} - V_{th}$ [49]. The output voltage of the LDO unit is regulated by closed loop feedback which flows from the output voltage, through a voltage divider, into the inverting input of an amplifier stage (Figure 25). When the output voltage is below the voltage reference, the feedback increase V_{GS} of the output transistor, increasing its current or lowering its output resistance; the opposite occurs if the output voltage is above the reference voltage. One potential way of visualizing the LDO is as a variable resistor, who changes its resistance such that the voltage divider created between it and the load results in an output voltage equal to the voltage reference.

When an LDO is connected to a known resistance value, it creates a current source with extremely low mismatch and high effective impedance do to its active feedback. This is a common methodology for creating extremely stable current sources, as the resistance of an integrated resistor varies less than the drain current of a transistor with a fixed V_{GS}. Figure 26 demonstrates how the LDO block is integrated to the circuit to create a stable current reference. Another major benefit of this design is that it does not

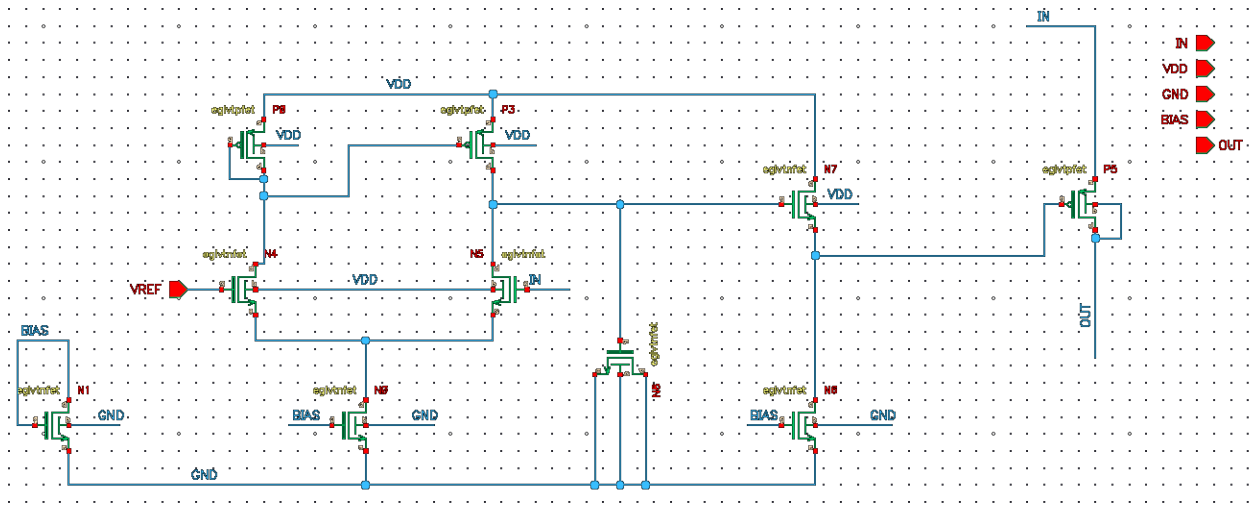


Figure 25: Screenshot of low-dropout regulator schematic taken from Cadence Virtuoso

require an on-chip current reference to mirror, rather it uses a voltage reference which is far easier to control, and can even come from an off-chip pad.

After sizing the and resistors transistors of the LDO, I was convinced that it really was the best option. Despite its larger size and static power consumption, it outperformed both the Wilson and cascode topologies in both output impedance and mismatch performance. However, the mismatch between the resistors of the columns of the device rendered still too large of a current signal compared to the signal between neighboring popcounts, preventing them from being recognized. Essentially, there was no current source that was ideal enough to be used in a fully differential design— therefore I came up with the idea to "trim" my threshold values per column to compensate in the one-sided current measurement topology.

3.4.3 Comparator

Before I talk in detail of the digital trimming methodology which allowed my design to be viable, I must first discuss the important element of the ADC which I have been ignoring: the comparator.

The role of the comparator is simple, when its positive voltage input terminal is higher in magnitude than its negative input terminal, the output of the comparator should go to V_{DD}. Similarly, when the input voltage relation is flipped, the comparator should go to GND. In some ways, a comparator can be thought of as an amplifier with such large gain, that its outputs always hit the voltage rails.

3.4.3.1 Comparator Sizing

In order to reduce the power consumption of my comparator, I chose to use the strongARM comparator topology (Figure 27). The strongARM is a clocked and latched comparator which exhibits both high speed and minimal static power, due to its unique operation methodology [50]. When the clock is low, the circuit is precharged to V_{DD}, when the clock is high, whichever side of the differential pair (N7/N10) has a higher gate voltage sinks more current, causing its side to fall while the other rises, and then the cross coupled

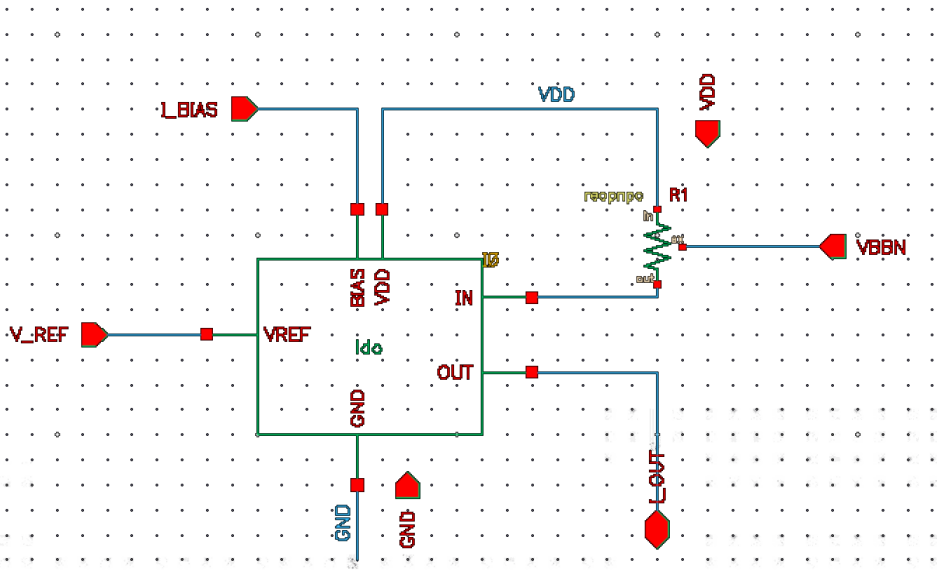


Figure 26: Schematic of the LDO block being used as a current source, using a fixed resistance as reference

inverter (P8 & N6 / P9 & N5) magnifies this feedback, bringing the less strongly driven side to V_{DD}, and the other to GND. Finally, to keep the output stable, the inverter output is sent to a latch, which holds its output until the next clock pulse is sent.

3.4.3.2 Thin vs. Thick Oxide Transistors

I performed my own optimization process using my geometric constraints and differential evolution to size the active stage of the strongARM for maximum sensitivity, eventually resulting in a width of 1μm and a length of 4μm. Despite this, I still could not get its sensitivity below 6 mV. This meant that my comparator could not differentiate two voltage signals within 6 mV of each other. Using thin oxide transistors with a $V_{DD} = 0.9V$, the maximum voltage that can be consistently regulated on the LDO output is $\sim 0.6V$. However, there is also a minimum voltage of $\sim 0.3V$ given by the current that still must flow through the column when the popcount is at its minimum value. The voltage per state is given by $\frac{V_{64}-V_0}{64} \approx 3$ mV. Because this is below the maximum sensitivity of the comparator, neighboring popcounts would be unable to be differentiated, even in the best case scenario.

Because of this, I took the advice of my advisors and switched from the thin oxide transistor to the thick oxide transistors. The thin oxide set of cells has a maximum V_{DS} of only $0.9V$, whereas the thick oxide can go up to $1.8V$. By using the thick oxide cells, the difference between the voltage of my maximum popcount and my minimum popcount could be more than doubled! However, this comes at the strong cost of significantly increasing power consumption, which scales with the square of V_{DD} for dynamic power, and linearly for static power. At this point, there was a significant bifurcation in the potential design of my chip. If I were to take the thin oxide route, I would have far lower power, but I would not be able to differentiate popcounts that are near to each other. At an algorithmic level, this can be significantly mitigated using hardware-aware training methodologies, and some noise can be beneficial for certain

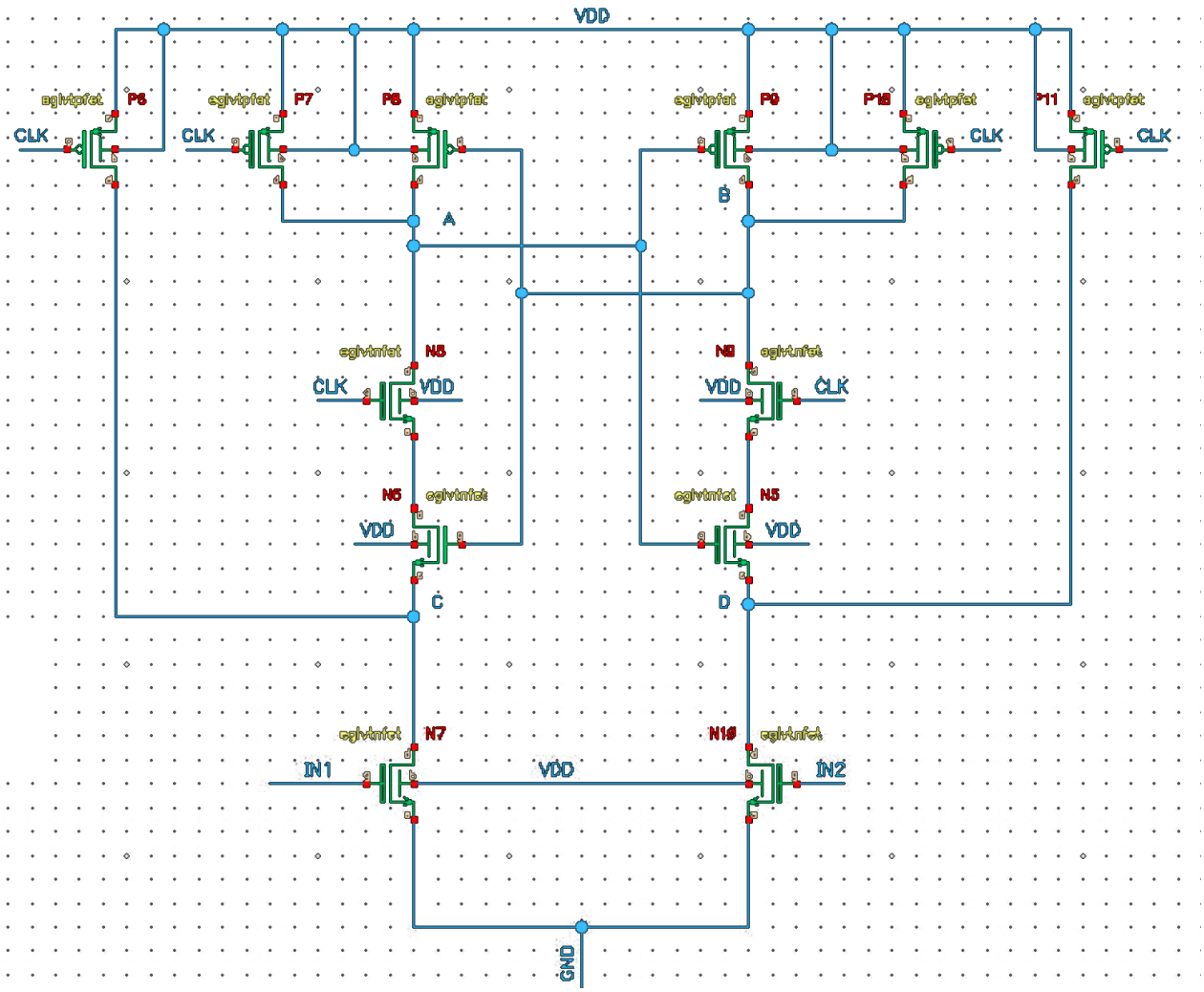


Figure 27: Screenshot of strongARM comparator schematic taken from Cadence Virtuoso

learning algorithms. I, however, took the thick oxide route, as I wanted to guarantee the functionality of my circuit and algorithm, even for neighboring popcounts at the cost of efficiency. If I were to do a second design, perhaps I would have taken the riskier choice, but for the first tapeout of this technology, it was better to be safe.

3.4.4 Digital-to-Analog Converter (DAC)

With my design moved and resized to use the thick oxide transistors, I could finally validate the robustness of my design against the my three sources of variation: noise, mismatch, and input positions. Mismatch existed not only between my resistors causing variations in drain current (I_d), but also existed within the width of the transistors inside the comparator itself, creating an additional offset which needed to be compensated to switch the output state compared to a perfectly matched comparator. It was for this reason that I had to perform a binary search to find the switching threshold between 0/1 for the comparator to measure the "true output" rather than just measuring the voltage generated at the current source. This way, for the given noise and mismatch seed, one would know with certainty which way the output would swing for the given threshold. Thus, I performed a massive simulation on my ideal ADC candidate to properly design my digital-to-analog converter (DAC).

3.4.4.1 The Safe Zone

The results of this simulation are shown in Figure 28. I performed 100 binary searches to a 16 bit resolution to find the true comparator threshold considering mismatch, noise, and input variation for a popcount equal to 31 and for 32, for a total of 3,200 transient simulations. The top half of the figure shows the two histograms, with popcount 31 in blue and popcount 32 in orange. If the two histograms were overlapping, it would mean that there would be no possible voltage applied to the comparator as a threshold that could result in 100% accuracy for the design. However, because the two histograms are separated, if a DAC threshold were placed anywhere within that separation range, then the system would be able to accurately differentiate between 31 and 32 despite all sources of variation. If then the threshold were placed somewhere on the blue histogram, it would mean that some true popcount 31 states would be incorrectly identified as popcount 32 states by the ADC system. By measuring what percentage of the histogram lies above or below a theoretical DAC voltage, one can generate the accuracy as a function of the threshold without having to compute millions of simulations per threshold! I therefore call the region in green on Figure 28 the "Safe Zone", and the goal of a DAC is to be able to provide a voltage within the safe zone for any column and any algorithmic threshold desired.

3.4.4.2 Levels

The most important design choice for the DAC is the number of voltage levels available. As the DAC will be composed of a voltage ladder and an analog multiplexer (see 3.4.4.3), the analog multiplexer is controlled by a binary string therefore the number of available levels should be a power of 2. The minimum number of levels for the design would be 64, corresponding to the number of possible popcounts for the design.

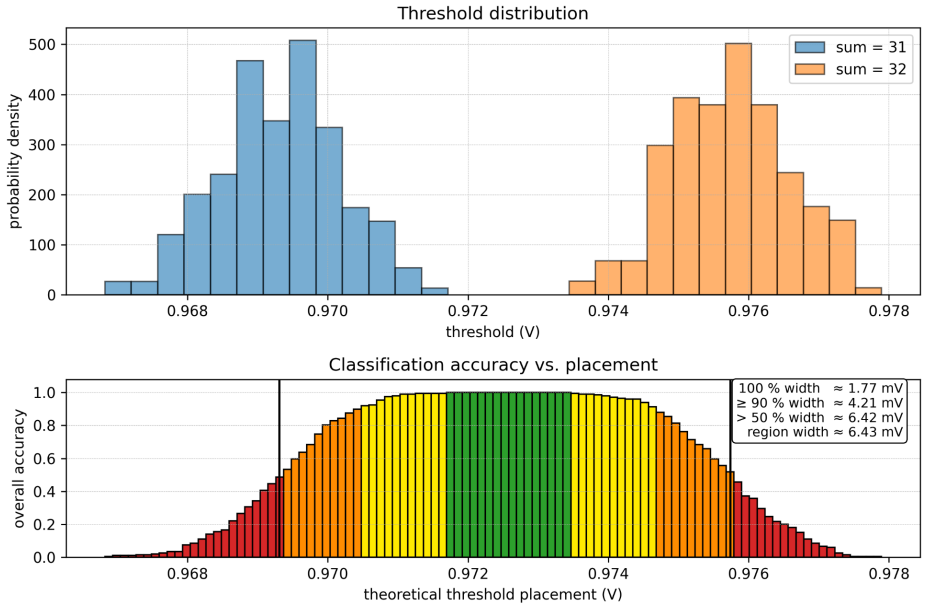


Figure 28: (top) Comparator threshold distribution between popcount = 31 (blue) and popcount = 32 (orange), with noise, mismatch, and input variation enabled (bottom) inference accuracy as a function of DAC threshold. 100% accuracy is indicated as green, ≥ 90% as yellow, ≥ 50% as orange, and < 50% as red.

However, this does not give any margin of error. The sixty four levels must correspond with all of the safe zones for each popcount differentiation. While the LDO current characteristic is much more linear than the Wilson mirror, the current still changes as the impedance of the column changes. Therefore, the spacing between popcounts should change slightly over the course of the design. This means that if one were to only place 64 levels linearly spaced in the design, only a subset of them could potentially be aligned to the safe zones. One potential solution would be to use an unequal spacing of the resistors in the DAC ladder in order to better match the current characteristic of the source, however this still leaves one susceptible to mismatch if one of the current source resistors is different from typical. Therefore, if we increase the number of states, perhaps we can "trim" the offset voltage digitally to adjust for mismatch.

In Figure 29, we see how the number of levels affects the accuracy, when performing digital trimming to match the safe zone at popcount = 32. The top portion of the figure shows the relationship between the threshold and popcount for 100 seeds. The relation is primarily linear. However, there exists an offset from the mean due to mismatch within the comparator, and there is a deviation in slope due to mismatch of the current source. An ideal trimming setup would be able to adjust for both slope and offset, but here I focus just on offset for now. Because the DAC can only adjust in broad increments equivalent to the 64 levels, very few of the popcount voltage relationships are able to be adjusted into the safe zone (middle of the figure), resulting in low accuracy (bottom of the figure). However, when the number of levels is increased to 256 (Figure 30), one is able to properly offset the threshold to align with the safe zone for popcount 32.

Figure 31 compares the derived accuracy for 64, 128, and 256 level DACs. While 128 levels performs reasonably well, it is unable to reach 100% accuracy for popcount 32. This makes sense from an intuitive level, as the width of the safe zone is 1.77 mV, and 64, 128, and 256 levels result in voltage resolutions of

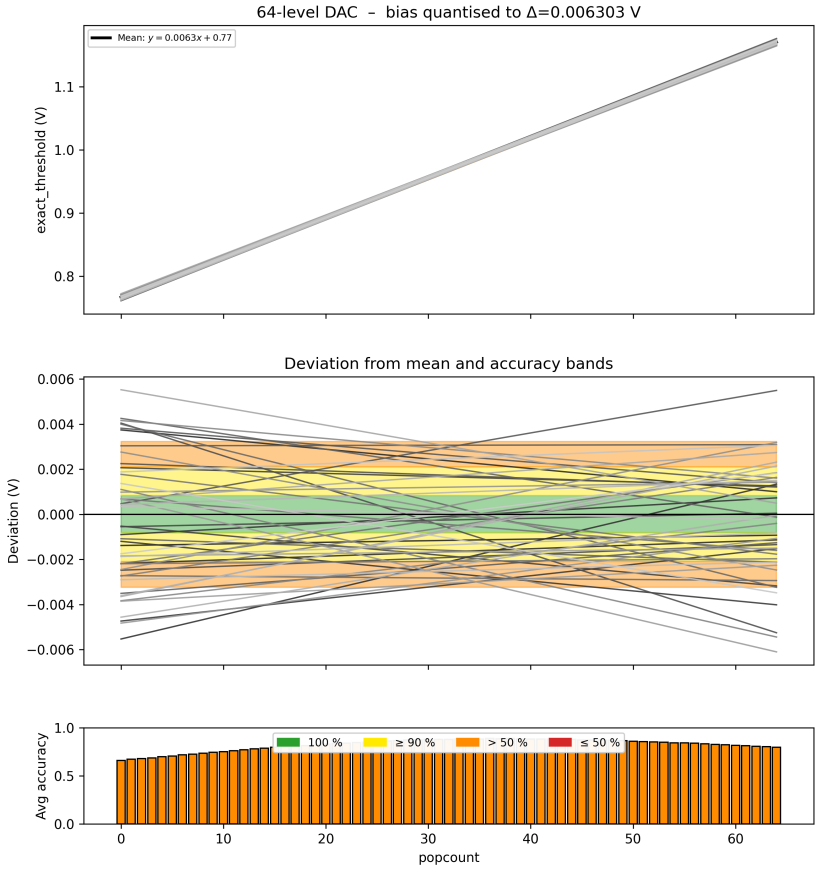


Figure 29: Deviation from ideal popcount / threshold relationship for 64 level DAC (top) Unmodified linear popcount / threshold relationships (middle) deviation from average relationship, colored by the accuracy which would result as a function of its distance from the midline (bottom) derived accuracy as a function of popcount, by counting the number of lines belonging to the middle green area

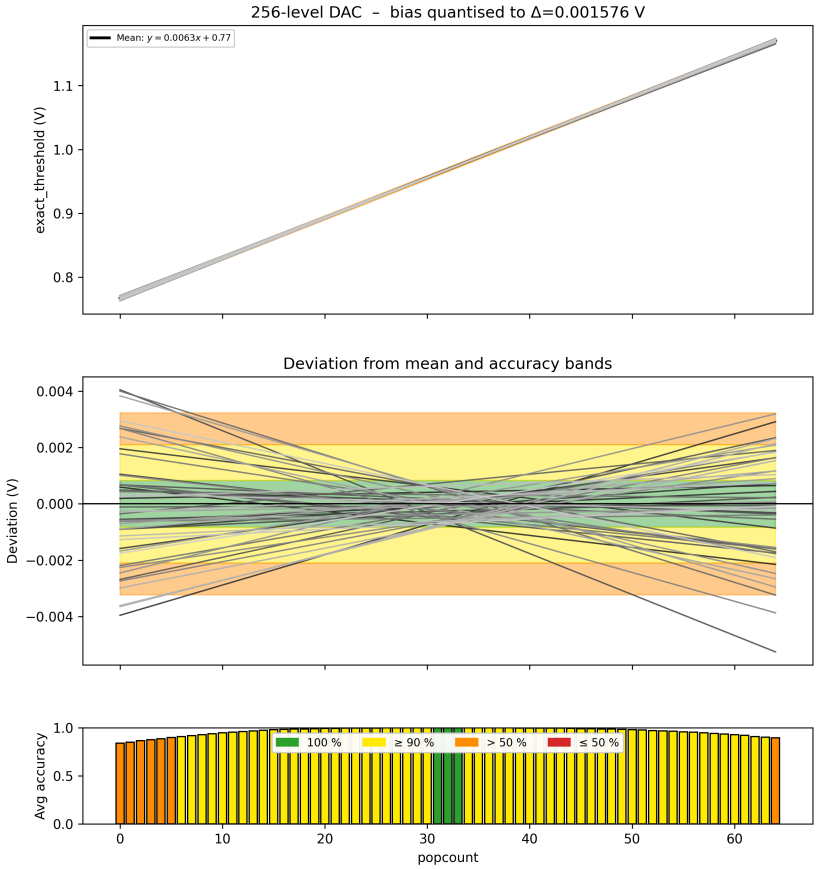


Figure 30: Deviation from ideal popcount / threshold relationship for 256 level DAC (top) Unmodified linear popcount / threshold relationships (middle) deviation from average relationship, colored by the accuracy which would result as a function of its distance from the midline (bottom) derived accuracy as a function of popcount, by counting the number of lines belonging to the middle green area

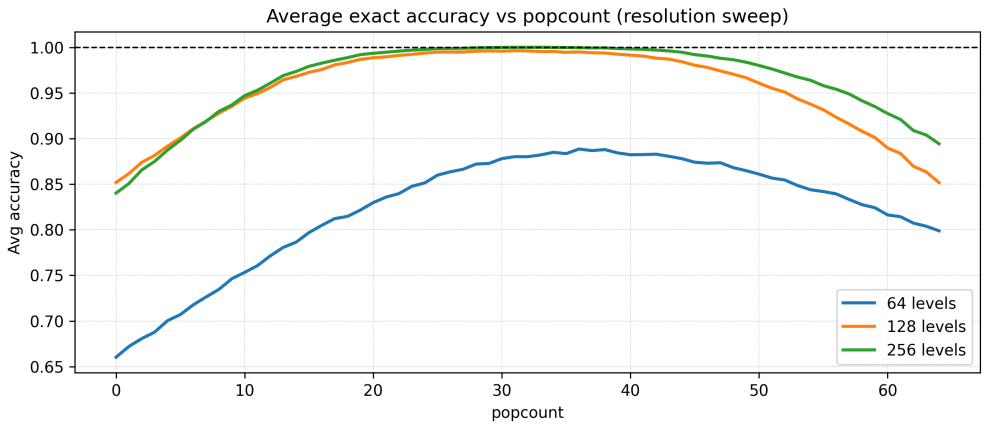


Figure 31: Computed accuracy as a function of threshold, for 64 levels (blue), 128 levels (orange) and 256 levels (green)

7.03 mV, 3.51 mV, and 1.75 mV respectively. Because the minimum resolution of the DAC is smaller than the width of the safe zone, this allows for one to trim the threshold in order to obtain 100% accuracy at popcount 32.

I will discuss the digital trimming circuit more in 3.5.1, but a 256 level DAC should be sufficient to achieve 100% accuracy not only for popcount 32, but for every popcount by performing a tuning process prior to inference.

3.4.4.3 Circuit

Finally, here I will discuss the circuit implementation of the DAC, which consists of two parts: the resistive ladder and the analog multiplexer.

The resistive ladder was the simplest to design. I used the highest sheet resistance resistor available in my PDK, and connected 258 resistors in series (Figure 32). The additional two resistors were applied to the top and the bottom of the ladder, such that the input of the ladder is not directly applied to the output levels. To determine the resistance of each resistor, I simply looked at the output range of the comparator inputs (0.75 to 1.2 V), and selected a reasonable current of less than 1μA. From there, I derived the total resistance, and found that selecting a resistance of $2k\Omega$ per resistor would result in a total current of less than 1μA. Choosing an even lower current would reduce power consumption, but the physical size of the resistors would grow to be significant, and the ladder would become susceptible to transient currents changing its nominal voltage levels.

Then, to design the multiplexer, I used my transmission gate that was used in the design of the enable cell, and cascaded them together from 2×1, to 4×2, to 4×2, and eventually 256×8. The design of the 2×1 multiplexer (Figure 33 is extremely similar to the enable cell, however, the select lines are differentially encoded; it is impossible to put the multiplexer in the high impedance state.

With the DAC completed, the ADC conversion unit was finalized, thereby finishing the analog design of the entire array. It was certainly the most difficult part of the design.

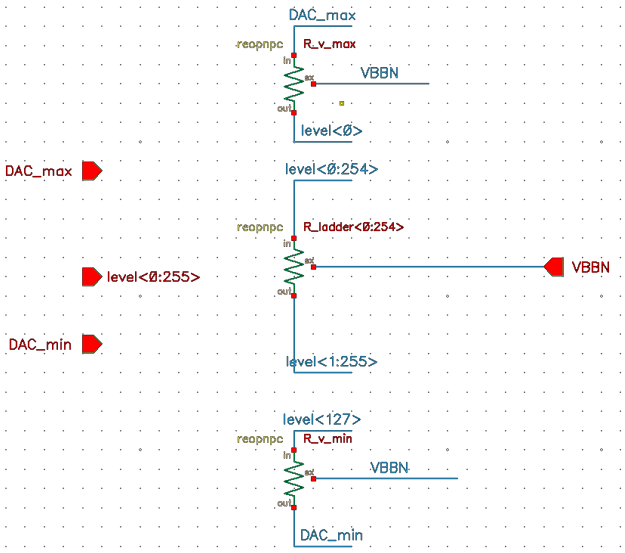


Figure 32: Screenshot of resistive ladder schematic taken from Cadence Virtuoso

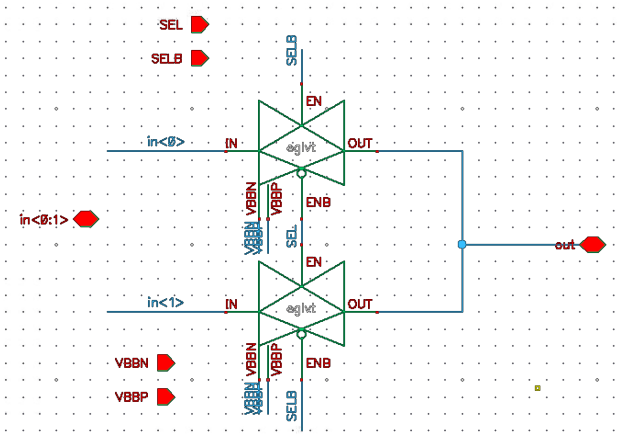


Figure 33: Screenshot of 2x1 analog multiplexer schematic taken from Cadence Virtuoso

3.5 Digital Design

The digital design of the chip is still ongoing, as we would like to choose the exact functionality / features of the chip nearer to the end of the design processes. Nonetheless, two digital components of the chip will be necessary and have been designed at the block level: first the digital trimming block, and secondly the serial programming interface (SPI).

3.5.1 Digital Trimming

To activate the preactivation of the neuron, the threshold is placed at one of 64 levels, corresponding to the potential popcounts of the neuron. However, as we have previously seen, the thresholds often need to be adjusted from the theoretical value. Thus, I propose here four potential ways to implement the design of digital trimming.

The first is to only store the DAC selection signal in a single 8 bit register. This is the simplest option, but the least versatile. One will have to retune the exact threshold in order to change the algorithmic threshold each time. Fortunately, the threshold does not change often, but the tuning process does take time. The second option is to encode the algorithmic weight in a 6 bit register, then to encode the offset to that weight in a 8 bit register, which gets added/subtracted from the algorithmic weight to result in the exact offset. Then after tuning for popcount 32, the system retains its accuracy as the algorithmic threshold changes, just like in Figure 30. A third possible solution would be not only to save this offset, but also a modification to the slope of each of the popcount voltage characteristics, by including a multiplier on-chip as well. This way, one can adjust the bias and the slope of all the characteristics to not have to tune the neuron more than once, and to maintain 100% accuracy for all popcounts. A final way would be to tune for each of the 64 popcounts, then use a lookup table to output the exact threshold, but this would be area and energy inefficient.

3.5.2 Serial Programming Interface (SPI)

To create the serial programming interface (SPI), I will devote four pads of my chip, one for the clock (SCLK), one for the controller output peripheral input (COPI), one for the controller input peripheral output (CIPO), and another for chip select (CS). This will allow a microcontroller to interface with my chip by enabling the chip select, sending data using the COPI, and receiving data using the CIPO, all controlled with the SCLK.

I plan to use a control word containing 70 bits: five for the register address, one to encode read/write, and 64 bits for the data stored within the register. My chip needs fewer than 32 total registers to operate the chip including those needed for the control circuitry and the ADC thresholds / output registers. When the control word is read into the chip, (Figure 34) the SPI peripheral reads the word, and writes or reads the selected register, and outputs the contents of the register on the following clock pulses for read. The figure is a simple potential solution, and for the final design I plan to use a preexisting SPI library to create the SystemVerilog, to then synthesize and place my cells outside of the analog area of my design.

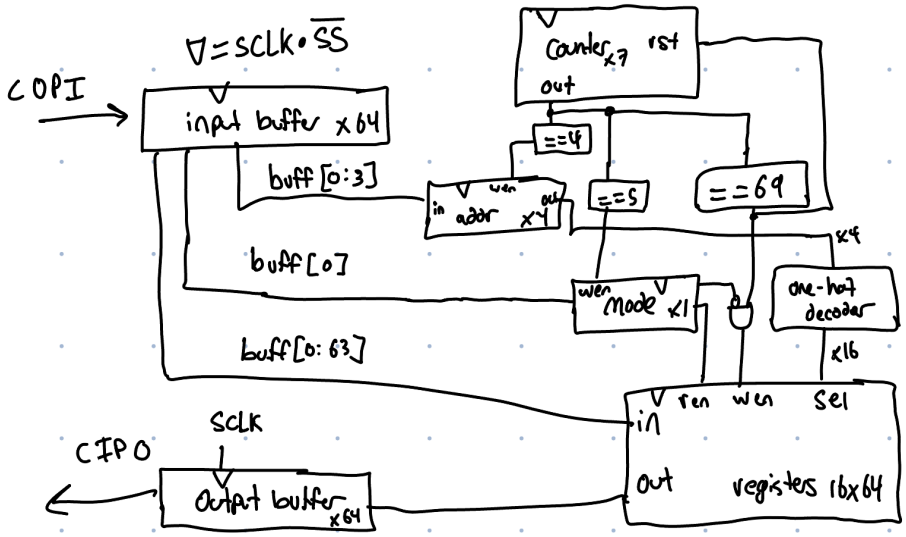


Figure 34: Hand-drawn block diagram for serial programming interface

4 Results

4.1 Performance Comparison

With all of the analog design for the chip complete, I evaluate its performance in comparison to the state-of-the-art paper from [45]. To perform a single inference operation (for a single column), it takes 100 fJ of energy in a period of 0.7 ns (Figure 35). This means an average power of 143 μW per column, and 9.1 mW for the entire array. However, we have 5.851 TOPs (tera operations per second) using the notation methodology of [45], where each multiplication counts as an operation to the sum (one operation per cell). Dividing by the power gives an estimated 643 TOPs/W, higher than the maximum predicted by Samsung's paper of 474 TOPs/W, at much higher speeds of operation (~ GHz instead of ~ MHz) by avoiding the slow (yet efficient) time-to-digital methodology. It is very likely that this efficiency will continue to diminish when more digital components with static power are added to the circuit, and after parasitic extraction from my layout, when additional capacitances will reduce the maximum operating speed of the circuit. However, this work clearly shows that my design is at least competitive with the state of the art preexisting solutions.

4.2 Layout Footprint

Finally, here I would like to report on the estimated area of my completed array. I have since started the layout process for my design, and have created the folded cell shown in Figure 36. It has a width of 2.27 μm and a height of 3μm. While it does not currently pass design rule check (DRC) due to a licensing issue with dual patterning, it is layout versus schematic (LVS) clean, and it provides a reasonable estimate to the final dimensions and form of the cell and the whole array. Therefore, the total array should measure an area of 145 μm by 192 μm, for an area of 27,840 μm^2. I am still currently working on the remainder of the

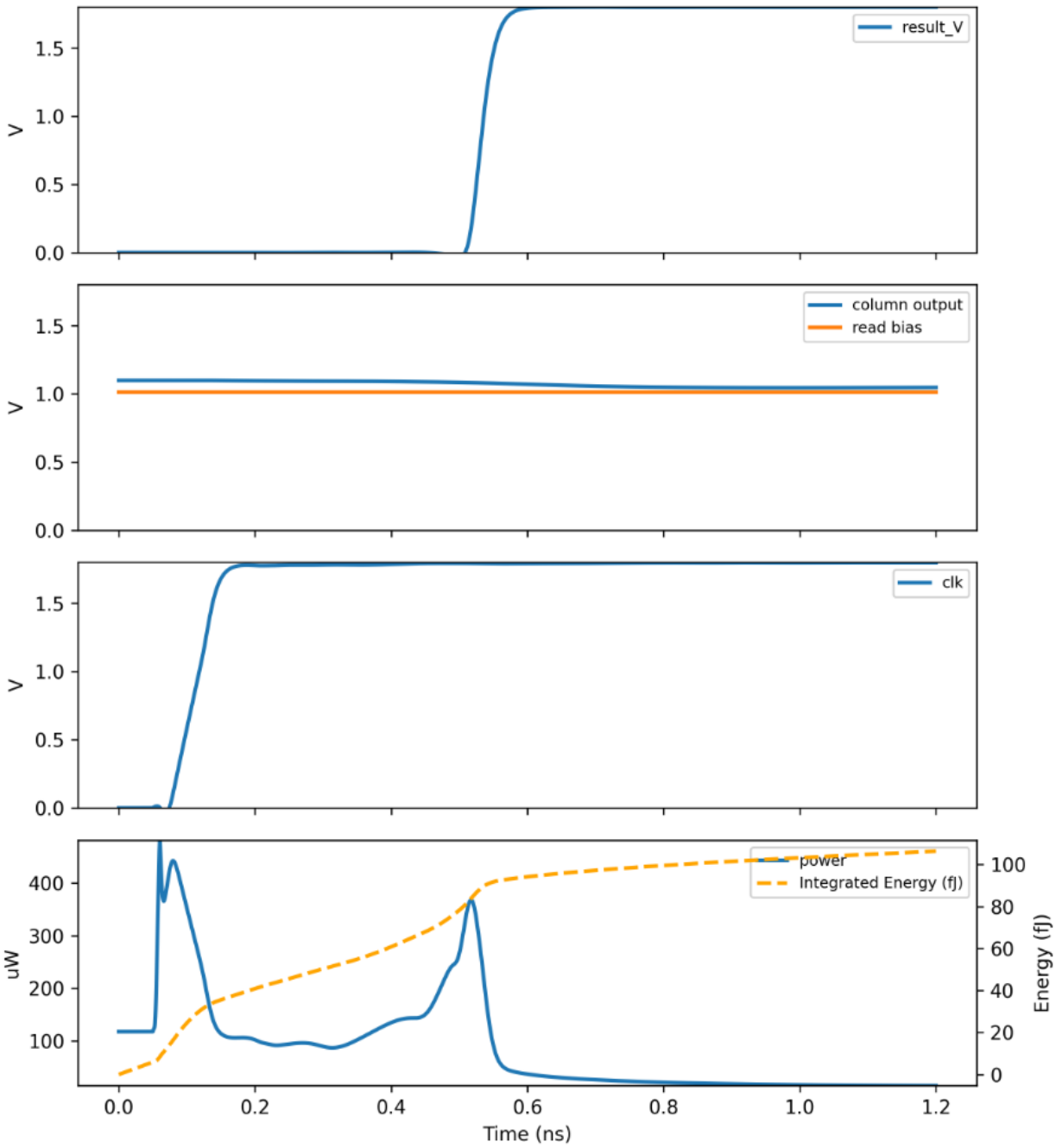


Figure 35: Transient simulation results for a single column inference operation (top) voltage characteristic for the result of the operation (top middle) voltage for the output of the neuron (blue) and the DAC voltage (orange) (bottom middle) clock voltage (bottom) power (blue) and integrated energy (orange) plot

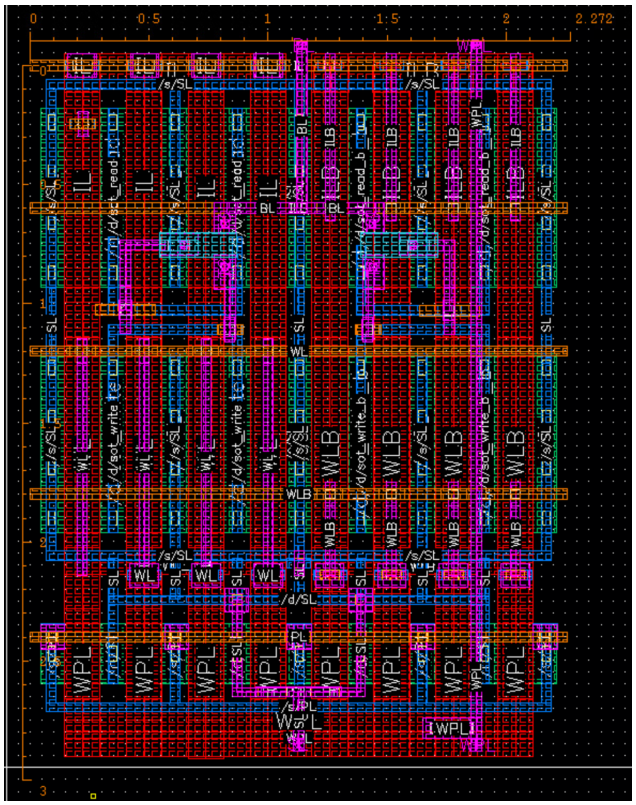


Figure 36: Screenshot of layout for the array cell from Cadence Virtuoso LayoutXL

layout, but it will be completed according to the Gantt chart in order to tapeout come mid-December.

5 Conclusion

This thesis report has described most of the design process for a neural network inference chip, starting from the array structure, to the control circuitry, the analog-to-digital converter, and finally ending with the digital design. In order to tape-out in December, the analog layout must be finished, followed by the remainder of the digital design and then digital layout, along with pad placement and input/output (I/O) design. Per the Gantt chart, things are on-track, and my design should be sent to be manufactured come this December. The chip performs at an estimated efficiency of 643 TOPs/W, while outperforming the state of the art in accuracy, and hopefully efficiency post parasitic extraction.

5.1 Future Work

In order to stick with the schedule defined by the Gantt chart and tapeout date, it is too late to make large modifications to the schematic, especially as the layout process has already begun. However, if I were to get access to a second tapeout (if the first were successful), I would likely make several modifications to my design. First, knowing the real TMR for the process would allow me to make more risky design choices. I would swap the large LDO unit for a smaller cascode design using my voltage trimming method to adjust

for the higher mismatch currents. I would also like to switch to a thin oxide design, trading accuracy in precise popcount determination for significantly reduced power consumption.

5.2 Impact

As mentioned in the introduction, SOT-MRAM-based inference arrays, like the one designed here, offer a promising path towards more energy-efficient neuromorphic computing. By enabling analog in-memory computation, they could significantly reduce the environmental impact associated with machine learning models and computing in general. I hope that this project results in a successful chip, that proves the viability of the technology and creates opportunity for various spin-off projects and other designs.

References

[1] N. Randewich and C. Mehta, "Ai mentions rise in s&p 500 earnings calls as firms tout technology," *Reuters*, Feb. 2024, Accessed 2025-09-06. [Online]. Available: https://www.reuters.com/technology/ai/ai-mentions-rise-sp-500-earnings-calls-2024-02-15/.

[2] "Generative ai and copyright: Part 3 — copyright law and policy," U.S. Copyright Office, Tech. Rep., May 2025, Accessed 2025-09-06. [Online]. Available: https://copyright.gov/policy/artificial-intelligence/.

[3] OpenAI. "Introducing chatgpt." (Nov. 2022), [Online]. Available: https://openai.com/index/chatgpt/.

[4] "Lacking job security, filipino call centre workers face ai threat," *Context (Thomson Reuters Foundation)*, 2024, Accessed 2025-09-06. [Online]. Available: https://www.context.news/business-digital-rights/lacking-job-security-filipino-call-centre-workers-face-ai-threat.

[5] "Is ai killing graduate jobs?" *Financial Times*, Jul. 2025, Accessed 2025-09-06. [Online]. Available: https://www.ft.com/content/.

[6] A. Günlü, T. Oral, S. Yoo, and S. Chung, "Reliability and validity of the problematic tiktok use scale among the general population," *Frontiers in Psychiatry*, vol. 14, 2023. DOI: 10.3389/fpsyt.2023.1068431. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyt.2023.1068431/full.

[7] P. Galanis *et al.*, "The tiktok addiction scale: Development and validation," *AIMS Public Health*, 2024. DOI: 10.3934/publichealth.2024061. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/publichealth.2024061?viewType=HTML.

[8] R. I. for the Study of Journalism. ""ai-generated slop" is quietly conquering the internet. is it a threat to journalism—or a problem that will fix itself?" Accessed 2025-09-06. (2024), [Online]. Available: https://reutersinstitute.politics.ox.ac.uk/news/ai-generated-slop-quietly-conquering-internet-it-threat-journalism-or-problem-will-fix-itself.

[9] Federal Trade Commission, *Ftc announces exploratory challenge to prevent the harms of ai-enabled voice cloning*, Nov. 2023. [Online]. Available: https://www.ftc.gov/news-events/news/press-releases/2023/11/ftc-announces-exploratory-challenge-prevent-harms-ai-enabled-voice-cloning.

[10] "Un report urges stronger measures to detect ai-driven deepfakes," *Reuters*, Jul. 2025, Accessed 2025-09-06. [Online]. Available: https://www.reuters.com/business/un-report-urges-stronger-measures-detect-ai-driven-deepfakes-2025-07-11/.

[11] "Ai use rising in influence campaigns online, but impact limited — us cyber firm," *Reuters*, Aug. 2023, Accessed 2025-09-06. [Online]. Available: https://www.reuters.com/technology/ai-use-rising-influence-campaigns-online-impact-limited-us-cyber-firm-2023-08-17/.

[12] K. Hill, "They asked an a.i. chatbot questions. the answers sent them spiraling," *The New York Times*, Jun. 2025. [Online]. Available: https://www.nytimes.com/2025/06/13/technology/chatgpt-ai-chatbots-conspiracies.html.

[13] N. Jegham *et al.*, *How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference*, 2025. [Online]. Available: https://arxiv.org/abs/2505.09598.

[14] K. G. A. Ludvigsen, "The carbon footprint of gpt-4," *Medium (TDS Archive)*, Jul. 2023. [Online]. Available: https://medium.com/data-science/the-carbon-footprint-of-gpt-4-d6c676eb21ae.

[15] OpenAI, J. Achiam, *et al.*, *Gpt-4 technical report*, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774.

[16] U.S. Environmental Protection Agency, *Greenhouse gas equivalencies calculator: Calculations and references*, Accessed 2025-09-06, 2024. [Online]. Available: https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator-calculations-and-references.

[17] P. Mathur, *Entergy receives regulatory approval for investments to support meta's louisiana data center*, https://www.reuters.com/business/energy/entergy-receives-regulatory-approval-investments-support-metas-louisiana-data-2025-08-20/, Reuters. Reporting by Pranav Mathur; Editing by Vijay Kishore, Aug. 2025. (visited on 09/06/2025).

[18] A. Shehabi, S. J. Smith, A. Hubbard, *et al.*, "2024 united states data center energy usage report," Lawrence Berkeley National Laboratory, Berkeley, CA, Tech. Rep. LBNL-2001637, Dec. 2024. [Online]. Available: https://eta-publications.lbl.gov/sites/default/files/2024-12/lbnl-2024-united-states-data-center-energy-usage-report_1.pdf.

[19] U.S. Department of Energy, Grid Deployment Office, *Clean energy resources to meet data center electricity demand*, Cites EPRI estimate that data centers could consume up to 9% of U.S. electricity by 2030, 2024. [Online]. Available: https://www.energy.gov/gdo/clean-energy-resources-meet-data-center-electricity-demand.

[20] U.S. Energy Information Administration, *Electric power annual, table 1.2: Summary statistics for the united states, 2013–2023 (sales to ultimate customers)*, Total U.S. electricity sales in 2020 were 3,717,674 GWh, 2024. [Online]. Available: https://www.eia.gov/electricity/annual/table.php?t=epa_01_02.html.

[21] J. Wang *et al.*, "Tam: A computing-in-memory scheme using tandem array within stt-mram for multi-bit analog mac operations," in *Proceedings of DATE 2023*, 2023. [Online]. Available: https://past.date-conference.com/proceedings-archive/2023/DATA/408.pdf.

[22] J. G. Koomey, S. Berard, M. Sanchez, and H. Wong, "Implications of historical trends in the electrical efficiency of computing," *IEEE Annals of the History of Computing*, vol. 33, no. 3, pp. 46–54, 2011. DOI: 10.1109/MAHC.2010.28. [Online]. Available: https://www.computer.org/csdl/magazine/an/2011/03/man2011030046/13rRUwjoNz5.

[23] A. Prieto, B. Prieto, J. J. Escobar, and T. Lampert, "Evolution of computing energy efficiency: Koomey's law revisited," *Cluster Computing*, vol. 28, no. 1, p. 42, Oct. 2024. DOI: 10.1007/s10586-024-04767-y.

[24] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, 2019. DOI: 10.1145/3282307. [Online]. Available: https://cacm.acm.org/research/a-new-golden-age-for-computer-architecture/.

[25] N. S. Kim, T. Austin, D. Blaauw, *et al.*, "Leakage current: Moore's law meets static power," *Computer*, vol. 36, no. 12, pp. 68–75, 2003. DOI: 10.1109/MC.2003.1250885. [Online]. Available: https://web.eecs.umich.edu/~manowar/publications/power-COMPUTER-Dec-2003.pdf.

[26] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Computer Architecture News*, vol. 23, no. 1, pp. 20–24, 1995. DOI: 10.1145/216585.216588. [Online]. Available: https://libraopen.lib.virginia.edu/downloads/4b29b598d.

[27] T. Dillinger, "In-memory computing for low-power neural network inference," *SemiWiki*, Jul. 17, 2020, Accessed: 2025-09-07. [Online]. Available: https://semiwiki.com/semiconductor-manufacturers/tsmc/287672-in-memory-computing-for-low-power-neural-network-inference/.

[28] ŌURA. "Oura's ongoing commitment to health data privacy." (Aug. 2025), [Online]. Available: https://ouraring.com/blog/health-data-privacy/.

[29] Ō. Support. "How oura protects your data." (Apr. 2025), [Online]. Available: https://support.ouraring.com/hc/en-us/articles/360025586673-How-Oura-Protects-Your-Data.

[30] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st MCC Workshop on Mobile Cloud Computing (MCC '12)*, 2012, pp. 13–16. DOI: 10.1145/2342509.2342513. [Online]. Available: https://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf.

[31] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016. DOI: 10.1109/JIOT.2016.2579198. [Online]. Available: https://cse.buffalo.edu/faculty/tkosar/cse710_spring20/shi-iot16.pdf.

[32] G. Simsek, "Edge computing and the future of the cloud," *Software Engineering Daily*, Sep. 14, 2018, Accessed: 2025-09-07. [Online]. Available: https://softwareengineeringdaily.com/2018/09/14/edge-computing-and-the-future-of-the-cloud/.

[33] "Irds 2022 executive summary," IEEE International Roadmap for Devices and Systems (IRDS), Tech. Rep., 2022. [Online]. Available: https://irds.ieee.org/images/files/pdf/2022/2022IRDS_ES.pdf.

[34] R. Carter, J. Mazurier, L. Pirro, *et al.*, "22nm fdsoi technology for emerging mobile, internet-of-things, and rf applications," in *2016 IEEE International Electron Devices Meeting (IEDM)*, 2016. DOI: 10.1109/IEDM.2016.7838029.

[35] C. Suárez Segovia, "Electrical and physicochemical characterization of metal gate processes for work function modulation and reduction of local VTH variability in 14FDSOI technologies," Feb. 4, 2016.

[36] P. Flatresse, *Process and design solutions for exploiting fd-soi technology towards energy efficient socs*, ISLPED 2014 presentation, 2014. [Online]. Available: https://islped.org/2014/files/ISLPED2014_FD_SOI_Philippe_Flatresse_STMicroelectronics.pdf.

[37] H. Fatemi, A. B. Kahng, H. Lee, and J. Pineda de Gyvez, "Heuristic Methods for Fine-Grain Exploitation of FDSOI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2860–2871, Oct. 2020, ISSN: 1937-4151. DOI: 10.1109/TCAD.2019.2935053. [Online]. Available: https://ieeexplore.ieee.org/document/8796415/?arnumber=8796415 (visited on 09/28/2024).

[38] H. Yoda, "MRAM Fundamentals and Devices," in *Handbook of Spintronics*, Y. Xu, D. D. Awschalom, and J. Nitta, Eds., Dordrecht: Springer Netherlands, 2016, pp. 1031–1064, ISBN: 978-94-007-6892-5. DOI: 10.1007/978-94-007-6892-5_39. [Online]. Available: https://doi.org/10.1007/978-94-007-6892-5_39 (visited on 09/29/2024).

[39] user. "[Future Semiconductor Technology] The Present of the Next-Generation Ultra-Low-Power MRAM Technology - SK hynix Newsroom." (Jan. 14, 2021), [Online]. Available: https://news.skhynix.com/future-semiconductor-technology-the-present-of-the-next-generation-ultra-low-power-mram-technology/ (visited on 09/28/2024).

[40] D. Apalkov, A. Khvalkovskiy, S. Watts, *et al.*, "Spin-transfer torque magnetic random access memory (STT-MRAM)," *J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, 13:1–13:35, May 29, 2013, ISSN: 1550-4832. DOI: 10.1145/2463585.2463589. [Online]. Available: https://dl.acm.org/doi/10.1145/2463585.2463589 (visited on 09/29/2024).

[41] K. Garello, "Spin-orbit torque switching of magnetic tunnel junctions for memory applications," presented at the ICSM, 2023.

[42] "Approximation by superpositions of a sigmoidal function | Mathematics of Control, Signals, and Systems." (), [Online]. Available: https://link.springer.com/article/10.1007/BF02551274 (visited on 09/29/2024).

[43] V. Maiorov and A. Pinkus, "Lower bounds for approximation by MLP neural networks," *Neurocomputing*, vol. 25, no. 1, pp. 81–91, Apr. 1, 1999, ISSN: 0925-2312. DOI: 10.1016/S0925-2312(98)00111-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231298001118 (visited on 09/29/2024).

[44] T. Simons and D.-J. Lee, "A Review of Binarized Neural Networks," *Electronics*, vol. 8, no. 6, p. 661, 6 Jun. 2019, ISSN: 2079-9292. DOI: 10.3390/electronics8060661. [Online]. Available: https://www.mdpi.com/2079-9292/8/6/661 (visited on 09/29/2024).

[45] S. Jung, H. Lee, S. Myung, *et al.*, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, no. 7892, pp. 211–216, Jan. 2022, ISSN: 1476-4687. DOI: 10.1038/s41586-021-04196-6. [Online]. Available: https://www.nature.com/articles/s41586-021-04196-6 (visited on 06/03/2024).

[46] P. Zhou, J. A. Smith, L. Deremo, S. K. Heinrich-Barna, and J. S. Friedman. "Synchronous Unsupervised STDP Learning with Stochastic STT-MRAM Switching." (Dec. 10, 2021), [Online]. Available: http://arxiv.org/abs/2112.05707 (visited on 09/29/2024), pre-published.

[47] W. A. Borders, A. Madhavan, M. W. Daniels, *et al.*, "Measurement-driven neural-network training for integrated magnetic tunnel junction arrays," *Physical Review Applied*, vol. 21, no. 5, p. 054 028, May 14, 2024. DOI: 10.1103/PhysRevApplied.21.054028. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.21.054028 (visited on 09/29/2024).

[48] N. Pilli, A. Kumar, S. Singh, and X. Xiong, "An inductor-less, discontinuous current source gate driver for sic devices," *IEEE Access*, vol. PP, pp. 1–1, Mar. 2019. DOI: 10.1109/ACCESS.2019.2904085.

[49] A. Paxton, "Ldo basics: Dropout," Texas Instruments, Tech. Rep. SSZTAC2, 2017. [Online]. Available: https://www.ti.com/document-viewer/lit/html/SSZTAC2.

[50] B. Razavi, "The StrongARM Latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, Spr. 2015, ISSN: 1943-0582. DOI: 10.1109/MSSC.2015.2418155. [Online]. Available: http://ieeexplore.ieee.org/document/7130773/ (visited on 06/19/2025).

Acronyms

GND	ground. 17, 21, 30, 31
V_{DD}	supply voltage. 17
V_{DS}	transistor drain to source voltage. 26–28, 31
V_{th}	transistor threshold voltage. 28
ADC	analog-to-digital converter. 4, 6, 15, 16, 18, 21, 24–26, 29, 30, 33, 37, 39, 51, 52
AI	artificial intelligence. 5
CiM	compute-in-memory. 6, 14
CIPO	controller input peripheral output. 39
CMOS	complementary metal-oxide semiconductor. 6, 9
COPI	controller output peripheral input. 39
CS	chip select. 39
DAC	digital-to-analog converter. 15, 33–37, 39, 41
DRC	design rule check. 40
FBB	forward body biasing. 10
FD	fully-depleted. 9
FD-SOI	fully-depleted silicon-on-insulator. 8
FinFET	fin field-effect transistor. 9
GAA	gate all-around. 9
I/O	input/output. 42
LDO	low-dropout regulator. 4, 29–31, 34, 42
LLM	large language model. 5
LVS	layout versus schematic. 40
LVT	low voltage threshold. 20
MOS	metal-oxide semiconductor (transistor). 49

MRAM magnetoresistive random access memory. 6, 8
MTJ magnetic tunnel junction. 11

NMOS n-channel MOS. 10, 20
NN neural network. 13

PD partially-depleted. 9
PMOS p-channel MOS. 10, 20, 26, 28

RAM random access memory. 11
RBB reverse body biasing. 10
ReRAM resistive RAM. 14

SLVT super-low voltage threshold. 16, 20
SOT spin–orbit torque. 8
STT spin-transfer torque. 11

TDC time-to-digital conversion. 15
TMR tunnel magnetoresistance. 11

A Employer Description

The Centre de Nanosciences et de Nanotechnologies (C2N) is a research center located in Palaiseau as a joint unit between Centre National de la Recherche Scientifique (CNRS) and the Université Paris-Saclay. C2N's research spans material science, nanophotonics, nanoelectronics, and microsystems, all enabled by its state-of-the-art 2900 m^2 cleanroom.

Within C2N, I worked with the IntegNano laboratory, co-headed by Damien Querlioz and Liza Herra Diez. The IntegNano laboratory focuses on energy-efficient artificial intelligence systems, enabled by emerging devices and physics such as filament memristors and spintronics.

B Gantt Chart

The Gantt chart in Figure 37 outlines the process of design from the start of the thesis period until the tapeout date in mid-December. Working backwards, I expect to need two or so weeks to package and document everything before the design is closed and sent to the foundry. Further, I expect to need at least two weeks to integrate my digital and analog components together on the same GDS file, while also adding pads. Then, I expect the digital design to take around a month, which can be done in parallel to the tapeout of my device. Then looking further back into what I have already accomplished, I spent almost two months testing and designing the various ADC methodologies, and before that I had spent two weeks to design the control circuitry and to fix the leakage problem of my 2A internship. Thus, I have created the Gantt chart corresponds to the full process of my chip design.

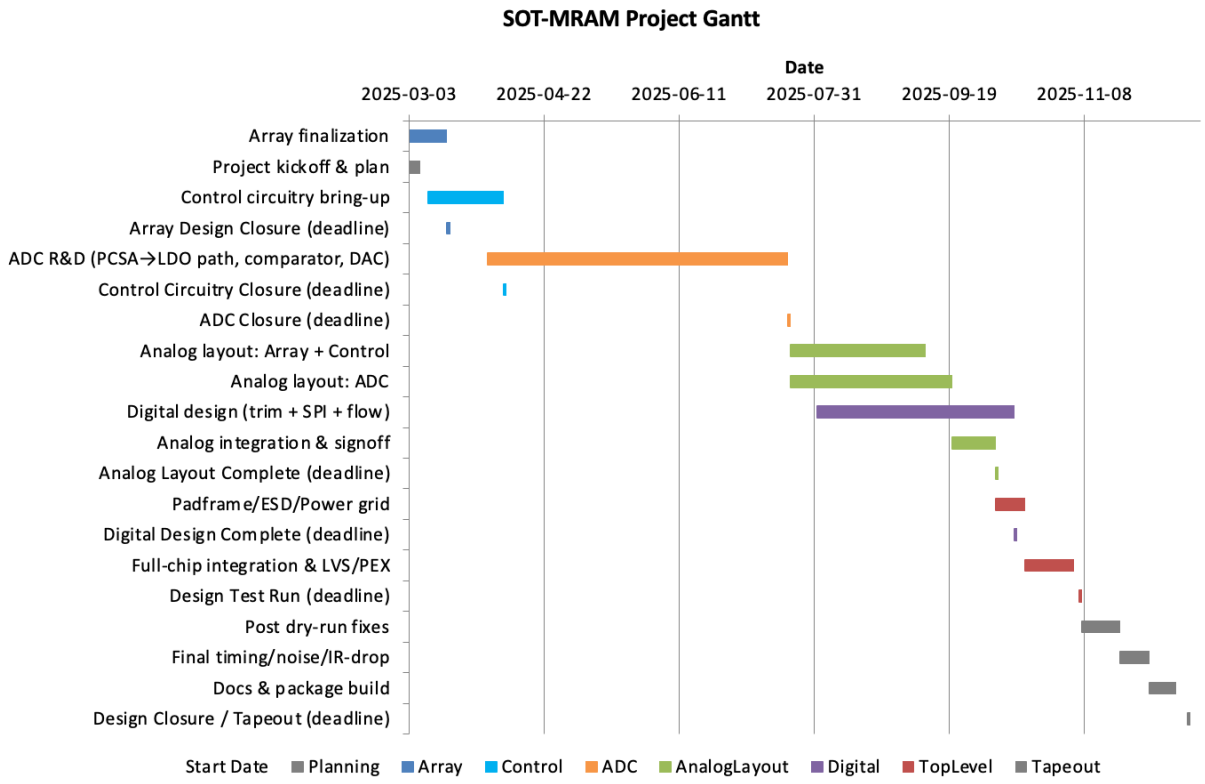


Figure 37: Gantt chart for the 3A internship. The various tasks are shown on the left side, whereas the estimated time for each task is represented by the length of the respective bar. The project begins with planning (grey), then the finalization of the array (dark blue), the control circuitry (light blue), then the ADC (orange), analog layout (green), digital design and layout (purple), top-level design (pads/IO in red), and finally the packing of the chip/cleaning of files for tapeout (grey)

C Summaries

C.1 English

In this 3A thesis report, I describe the process of designing a novel inference chip, which combines a novel spin-orbit torque (SOT) magnetic random access memory (MRAM) with a 22 nm fully-depleted silicon-on-insulator (FD-SOI) transistor process to perform inference operations in-memory. I begin by motivating the need to improve the energy efficiency of artificial intelligence systems to reduce their negative societal effects as they become more prominently used. Then, I describe the basics of the technologies used and of neural networks, to understand the technical goals of the project. The report then details the design results of my 2A internship, before discussing the array design, the control circuitry, the analog-to-digital conversion, and the digital design necessary. Several key decisions were made regarding circuit topologies, circuit components, and cell sizing in order to create a chip that minimizes the risks associated with a new process, hopefully resulting in a chip that works properly when tested. The final design achieves an estimated 643 TOPs/W pre-extraction, which is competitive with other state of the art designs. Well optimized SOT-MRAM inference arrays could continue to reduce the environmental impacts

of machine-learning, and computing in general.

C.2 French

Dans ce rapport de thèse 3A, je décris le processus de conception d'une nouvelle puce d'inférence, qui combine une mémoire magnétique à accès aléatoire à couple spin-orbite (SOT-MRAM) avec un procédé de transistors FD-SOI (Fully-Depleted Silicon-On-Insulator) de 22 nm afin d'effectuer des opérations d'inférence directement en mémoire. Je commence par motiver la nécessité d'améliorer l'efficacité énergétique des systèmes d'intelligence artificielle afin d'en réduire les effets sociétaux négatifs à mesure qu'ils se généralisent. Je décris ensuite les bases des technologies utilisées et des réseaux de neurones, afin de comprendre les objectifs techniques du projet. Le rapport détaille ensuite les résultats de conception de mon stage 2A, avant d'aborder la conception de l'array, les circuits de commande, la conversion analogique-numérique et la conception numérique nécessaires. Plusieurs décisions clés ont été prises concernant les topologies de circuits, les composants et le dimensionnement des cellules, afin de minimiser les risques associés à un nouveau procédé, dans l'espoir d'aboutir à une puce qui fonctionne correctement lors des tests. La conception finale atteint une estimation de 643 TOPs/W en pré-extraction, ce qui est compétitif par rapport aux conceptions à l'état de l'art. Des réseaux d'inférence SOT-MRAM bien optimisés pourraient continuer à réduire les impacts environnementaux de l'apprentissage automatique et de l'informatique en général.

C.3 Italian

In questo rapporto di tesi 3A descrivo il processo di progettazione di un nuovo chip di inferenza, che combina una memoria magnetica ad accesso casuale a coppia spin-orbita (SOT-MRAM) con un processo a transistor FD-SOI (Fully-Depleted Silicon-On-Insulator) a 22 nm per eseguire le operazioni di inferenza direttamente in memoria. Inizio motivando la necessità di migliorare l'efficienza energetica dei sistemi di intelligenza artificiale per ridurne gli effetti sociali negativi, via via che il loro utilizzo si diffonde. Descrivo quindi le basi delle tecnologie impiegate e delle reti neurali, per comprendere gli obiettivi tecnici del progetto. Il rapporto illustra poi i risultati di progettazione del mio tirocinio 2A, quindi discute la progettazione dell'array, dei circuiti di controllo, della conversione analogico-digitale e della progettazione digitale necessaria. Sono state prese diverse decisioni chiave riguardo alle topologie circuitali, ai componenti e al dimensionamento delle celle, al fine di minimizzare i rischi associati a un nuovo processo, con l'auspicio di ottenere un chip che funzioni correttamente in prova. Il progetto finale raggiunge una stima di 643 TOPs/W in pre-estrazione, competitiva rispetto ad altri progetti allo stato dell'arte. Array di inferenza SOT-MRAM ben ottimizzati potrebbero continuare a ridurre gli impatti ambientali del machine learning e dell'informatica in generale.

D Summary Sheet

1. Benjamin Wallace Walker
2. Phelma International Nanotech Program
3. A.Y. 2024-2025 (A3)
4. Spintronic on-chip learning with Bayes' Rule, from 03/03/25 to 12/09/25, with 2 week interruption
5. 

Centre de Nanosciences et de Nanotechnologies, 10 Boulevard Thomas Gobert 91120 Palaiseau
6. Damien Querlioz, `damien.querlioz@universite-paris-saclay.fr`
7. Jonathan Miquel
8. During this 6 month internship, the student will explore the intersection between spintronic devices and Bayesian learning. The operation and switching of spintronic devices such as SOT-MRAM are inherently stochastic, and to overcome these limitations typical applications require increased current and therefore energy to ensure reliable operation. However, there exist specific problems where stochasticity is advantageous, such as Bayesian learning and reasoning. Therefore, during this internship, the student will explore the intersection between the physics of spintronic devices such as SOT-MRAM and Bayesian neural networks. By designing systems across both levels of abstraction, the student will be able to leverage the unique mechanics of spintronic systems to create novel learning circuits and systems with enhanced energy efficiency, a requirement essential for the edge-computing systems of the 21st century. *NOTE: The actual contents of the thesis focus more on the continuation of the design from the 2A period, rather than focusing on spintronic learning. Some learning algorithms were explored during the internship, but were out of the scope of the report, as most of the time was spent doing chip design.*
9. I used servers provided by C2N and collaborated with several other researchers (Jean-Michel Portal, Kamel Harabi) and PhD students (Akib Iftakher, Théo Ballet) for their advice and support during the course of the internship.