A thesis submitted in partial satisfaction of the requirements for the degree of Master of Computer Science and Engineering in the Graduate School of the University of Aizu

Power-Aware Neuromorphic Systems with 3-D Stacking Synaptic Memory

by

Ngo-Doanh NGUYEN

March 2024

The thesis titled

Power-Aware Neuromorphic Systems with 3-D Stacking Synaptic Memory

by

Ngo-Doanh NGUYEN

is reviewed and approved by:

Chief referee

Associate Professor Date

Dang Nam Khanh

Professor Date

Ben Abdallah Abderazek

Senior Associate Professor Date

Okuyama Yuichi

THE UNIVERSITY OF AIZU

March 2024

Contents

List of Figures

List of Tables

List of Abbreviations

ADCs	Analog-to-Digital Converters
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
ASICs	Application-Specific Integrated Circuits
BER	Bit Error Rate
CAD	Computer Aided Design
CIFAR	Canadian Institue For Advanced Research
CMOS	Complementary Metal Oxide Semiconductor
CPUs	Center Processing Units
DNN	Deep Neural Network
DRAMs	Dynamic Random Access Memories
DVS	Dynamic Voltage Scaling
FinFET	Fin-shaped Field-Effect-Transistor
FPGAs	Field-Programmable Gate Arrays
GPUs	Graphic Processing Units
HBMs	High Bandwidth Memories
ICs	Integrated Circuits
IMC	In-Memory Computing
LIF	Leaky Integrated-and-Fire
LSBs	Least Significant Bits
MNIST	Modified National Institute of Standards and Technology dataset
MSBs	Most Signiificant Bits
NC	Neuromorphic Computing
NoC	Network-on-Chip
PEs	Processing Elements
ReLU	Rectified Linear Unit
SNM	Static Noise Margin
SNNs	Spiking Neural Networks
SOP	Synaptic OPeration
SRAMs	Static Random Access Memories
STDP	Spike-Timing-Dependent Plasticity
TCI	ThruChip Interface
TSVs	Through-Silicon Vias
VR	Voltage Regulator

List of Symbols

α	The ratio of hardware area between logic components and memory components
β	The ratio of the size between pull-up and pull-down transistors
C	The capacitance of the logic gates - A technology-dependent parameter
f_{sw}	The switching frequency of the neuromorphic systems
I_{leak}	The leakage current of the neuromorphic systems
K	Beltzman constant - A technology-dependent parameter
N	The number of transistors in the neuromorphic systems
P_{dyn}	The dynamic power of the neuromorphic systems
P_{leak}	The leakage power of the neuromorphic systems
P_{mem}	The power consumption of the memory blocks
P_{pe}	The power consumption of the processing elements
P_{total}	The total power consumption of the neuromorphic systems
V_{DD}	The supply voltage of the neuromorphic systems
Y_i	The yield rate of the i^{th} layer of the fabricated hardware
λ	The leaky value
$V_i(t)$	The membrane potential of i^{th} neuron at the t time step
$w_{i,j}$	The synaptic weight between the i^{th} neuron and the j^{th} neuron
$x_{i,j}$	The j^{th} pre-synaptic output spikes of the i^{th} neuron

The University of Aizu, March 2024

To My Dearest Mother

Acknowledgment

I would like to express my thanks and gratitude to Prof. DANG Nam Khanh for his support, encouragement, and guidance in achieving this degree. Also, I would like to thank Prof. Abderazek Ben Abdallah, and Prof. Okuyama Yuichi of The University of Aizu for taking the time to revise my thesis. I also would like to thank Dr. Akram Ben Ahmed of the Digital Architecture Research Center, National Institute of Advanced Industrial Sciences and Technology for his help on the paper's revision.

I want to thank all the members and my friends of the Smart Integration System Laboratory at Vietnam National University, Hanoi, and of the Adaptive Systems Laboratory at the University of Aizu. Their supportive words and encouraging messages kept me motivated to work harder and be a better researcher and person. Not to forget to appreciate the staff of the University of Aizu for their assistance.

Finally, my greatest gratitude to my mom, who has been strongly supporting me throughout my whole life. She has been inspiring and pushing me to achieve my goals.

Abstract

The combination of Spiking Neural Networks (SNNs) and 3-D Integrated Circuits (3-D ICs), so-called 3-D stacking neuromorphic systems, can be the most advanced architecture that inherits the benefits of both computing and interconnect paradigms to save power consumption while delivering optimal performance and accuracy. However, simply shifting the SNNs into the third dimension cannot fully exploit its potential or the benefits of 3-D structures. This chapter introduces the methodology to leverage the energy efficiency of SNNs. By stacking multiple layers of memory on top of logic circuits, the weights of SNNs can be split into several subsets. Each of them is placed in different isolated layers. Hence, various low-power techniques such as power gating, dynamic voltage frequency scaling, or dynamic voltage scaling can be easily applied to each separated layer depending on the importance of the training weight's bits. Although lowering supply voltage tends to reduce the accuracy of neural networks, with the support of 3-D structures, the performance of SNNs can be controlled or unchanged compared to normal operations when low-power techniques are used.

Chapter 1

Introduction

Edge devices embedding Artificial Intelligence (AI) have been an emerging computing paradigm recently [1]. However, embedding AI functions into these devices has a lot of challenges because of their resource intensity and power-hungry. As one of many solutions, SNNs show their potential for lightweight inferences compared to other neural network models [2–4]. Because, as a mimic of the biological brain, SNNs only transmit information using a sequence of spikes that are believed to be spatial and temporal sparse, which allows them to reduce energy significantly. Moreover, the computation involved in SNNs, especially with Integrate-and-Fire-like models, is comparatively simpler than the conventional neuronal network models. As a result, it reduces the power consumption and hardware area cost.

To exploit the great potential of SNNs, many researchers have investigated deploying these NC systems in recent years. These systems are usually implemented in specific hardware, such as Application-Specific Integrated Circuits (ASICs) or Field-Programmable Gate Arrays (FPGAs), to optimize power and area efficiency, and to perform computations in parallel. In practice, these neuromorphic systems have three main design approaches, which are: (1) *2-D IC-based digital hardware* [3, 4]; (2) *2-D IC-based analog mixed-signal hardware* [2, 5]; and (3) *3-D IC-based hardware* [6, 7]. The power consumption of the SNN architecture is similar to other conventional neural network architectures, which is the sum of power consumption by memory storage P_{mem} and power consumption by PEs P_{pe}. In practice, the power consumption from memory is usually dominant, which is about 75% of the total power [8]. It is because the neural network models often require millions of weights to acquire high accuracy and those weights are transferred back and forth in long-distance between memory and PEs. This leads to the huge size of memory, which prolongs the transferring distance and requires more power to transfer those weights in the conventional 2-D systems.

Nevertheless, as the era of Moore's Law for a single monolithic die nears its end, hardware architectures, particularly memory architectures, are undergoing a transition towards 3-D packages or 3-D ICs to enhance performance. The architecture of SNNs follows this trend as well [9]. On the other hand, with 3-D IC-based technologies, memories can be stacked to reduce the hardware footprint. However, we realize that instead of stacking memory banks, we can split the memory words and stack them above each other. In this case, each layer in 3-D memory will represent different levels of precision for synaptic weights, such as one, two, or multiple-bit precision. Consequently, the neuromorphic system can selectively deactivate the power supply of individual memory layers that contain the LSBs to conserve energy while still maintaining an acceptable level of accuracy. This is feasible because the absence of LSBs can be treated as a form of noise, and SNNs exhibit resistance to this type of fixed-pattern noise [10]. Based on this feature, in this thesis, we present a novel *in situ* dynamic quantization hardware architecture of a spiking computing processor using 3-D IC-based stacking memory. In our previous publications [11, 12], we have designed a 2-D SRAM-based neuromorphic core connected via 3-D Network-on-Chip (NoC), where the memory and the logic computations are placed at the same silicon layer. Based on our experiment, we found out that power consumption of the memory access occupies the major part of the whole system. With our previous architectures, it is difficult to isolate and optimize the power consumption of memory to reduce the overall power consumption of the system. Therefore, in this work, we present a new approach to dynamically reduce the power consumption of memory access with 3-D IC-based stacking memory and in-situ quantization. The main contributions of this document are summarized in the following:

- A novel low-power methodology to implement neuromorphic architectures with 3-D stacking synaptic memory, where the memory word is split into multiple subsets and placed in separate layers.
- With 3-D IC-based technologies, the under-voltage technique is applied separately to each memory layer in 3-D architecture based on the significant bits of synaptic weights. It aims to reduce overall power consumption with acceptable accuracy.
- Consequently, an in-situ dynamic quantization for synaptic weight is implemented in this work as the next level of undervolting. The weights are configured in the design phase and stay unchanged during inference. Therefore, the bit precision of synaptic weights is dynamically modified by removing completely the supply voltage of memory layer(s).
- A novel stacking memory mechanism that helps improve the yield rates by accepting imperfection at the top layers.

The rest of this document is organized as follows. Section 2 presents the related works. Section 3 introduces the methodology for 3-D IC-based implementation. The hardware architecture is shown in Section 4. In Section 5, the performance and power consumption of our spiking computing core in each supply voltage scenario are evaluated. Finally, we end the document with conclusions in Section 6.

Chapter 2

Legacies of the Past

2.1 Background

The high-level view of 3-D IC-based SNN architecture is shown in Fig. 2.1. Compared to other neural network models, information is encoded in Spiking Neural Networks (SNNs) using an encoding scheme. This information is then transmitted between neurons through trains of action potentials called spikes. Those spikes biologically are generated by the neuron's membrane potential reaching a certain threshold. They operate in a discrete-time domain, with each neuron sending and receiving spikes at specific times. As a result, it allows them to process temporal information, such as patterns and sequences, in a more natural way than traditional Artificial Neural Networks (ANNs). The most popular hardware model for simulating this behavior of biological neurons is the Leaky Integrated-and-Fire (LIF) because of its energy efficiency and capability of capturing the essential features of bio-information. Theoretically, LIF neuron operations are expressed in the following equation:

$$V_i(t) = V_i(t-1) + \sum_j w_{i,j} \times x_j(t-1) - \lambda \tag{2.1}$$

where $w_{i,j}$ is the synaptic weight between the i^{th} neuron and the j^{th} one. $V_i(t)$ is the membrane potential of i^{th} neuron at the t timestep and $x_{i,j}(t-1)$ is the j^{th} pre-synaptic output spike of the i^{th} neuron and the leaky value λ, respectively. This output of the i^{th} neuron is expressed with the equation below.

$$x_i(t) = \begin{cases} 1, \text{ if } V_i(t) \geq V_{th}, \\ 0, \text{ otherwise.} \end{cases} \tag{2.2}$$

Moreover, the neuromorphic systems are expected to be asynchronous and independent of neurons within the network. Therefore, the ability to learn the timing information is also crucial. In practice, there are two learning approaches, which are off-chip learning and on-chip learning. For the off-chip method, the popular one is the ANNs-to-SNNs conversion with a fully connected feed-forward neural network using the ReLU activation function [13]. It is usually trained in software using back-propagation with zero bias and then mapped into the LIF

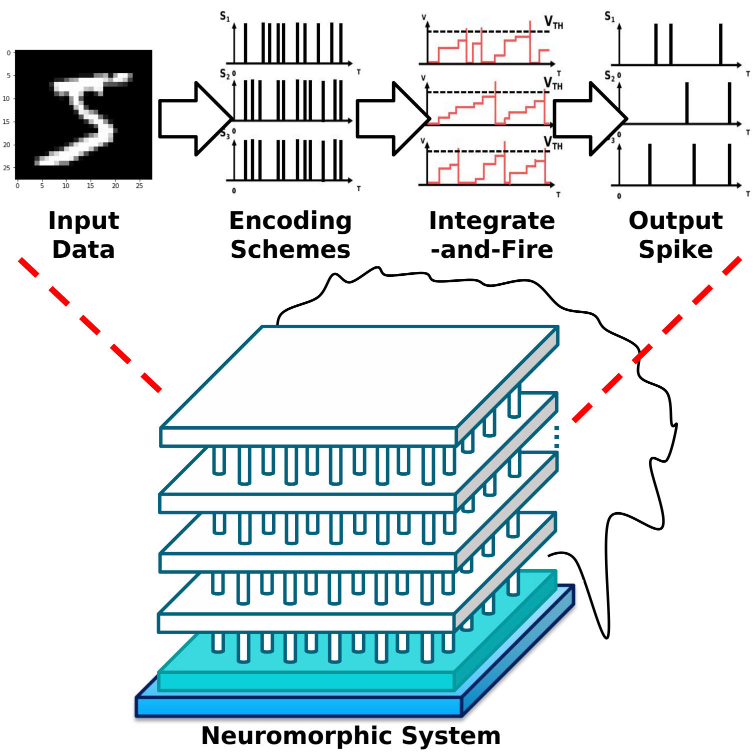

Figure 2.1: High-level view of the 3-D IC-based spiking neural network architecture.

network in a normalized way. For the on-chip method, the famous algorithm is the Spike-Timing-Dependent Plasticity (STDP) [14], an unsupervised learning algorithm with biological characteristics. It is based on synaptic plasticity to represent the relative difference in timing between the pre-synaptic spike and the post-synaptic one.

2.2 Neuromorphic Systems for Low-power Applications

As stated in the introduction, there are three different approaches to designing SNN hardware. The most widely used approach is the *2-D IC-based digital hardware*. Notable examples of this approach include Intel's Loihi [2] and IBM's TrueNorth [5]. Loihi utilizes the asynchronous Network-on-Chip (NoC) to represent the spike transmission of active synapses. Furthermore, Loihi's neurons are reconfigurable, allowing for the implementation of different neuron models and supporting adaptive bit-width operations (1-to-9 bits) for synapses. In the case of IBM's system, TrueNorth relies on fixed-bit-width weights for its LIF neuron cores. However, TrueNorth operates on a large-scale network with 1 million neuron cores, each having a 256×256 crossbar connecting pre-synaptic spike events to post-synaptic ones. In conclusion, TrueNorth stands out due to its ease of prototyping and system debugging. However, in terms of power consumption, it requires more power than the other two approaches (*2-D IC-based analog mix-signal hardware* and *3-D IC-based hardware*) when scaled to the identical fabrication technology [15].

Regarding the *2-D IC-based analog mixed-signal hardware*, this approach can accurately emulate the electrical behaviors of biological neurons while having lower power consumption than digital systems. A demonstration of such a system is NeuroGrid from Stanford University [3], which is based on the analog sub-threshold design. This system is capable of achieving real-time performance. NeuroGrid utilizes the NoC with a tree topology and multi-casting feature. Despite using older technology (180nm), NeuroGrid outperforms TrueNorth (28nm) in terms of energy efficiency, with an energy-per-operation result of 45pJ compared to 50pJ. Moreover, the analog mixed-signal approach can also match the capabilities of the digital system in cases of scalability and robustness, as demonstrated by Heidelberg University's BrainScaleS-2 architecture [4]. This system utilizes analog wafer-scale circuits and operates at a time scale 10.000× times faster than real-time biological processes. However, fabricating analog circuits has a higher complexity than digital circuits. The reason is that standard analog cells tend to require customization when shifting technology. Additionally, these systems pose challenges in terms of control and calibration, even when scalability is achieved. This is due to significant variations in analog circuit characteristics across different process technologies, temperatures, and voltage levels.

In terms of the *3-D IC-based hardware*, there is growing interest in the Loihi-2 architecture [7], which supports 3-D multi-chip scaling and represents the next generation of hardware architectures. NeuroSIM [6], a 3-D neuromorphic system, incorporates two-layer memristors as electronic synapses for SNNs. This integration leads to a 50% reduction in the hardware area, 1.48× times lower power consumption, and 2.58× times lower latency compared to traditional 2-D single-layer configurations. Another 3-D IC-based SNN architecture called MigSpike [12] is specifically designed for fault tolerance and reduces migration costs associated with remapping in NoC by a factor of 10.19× compared to 2-D approaches. Consequently, 3-D ICs offer significant advantages over the aforementioned approaches, including reduced hardware footprint, cost, and power consumption. It is reasonable to expect that a 3-D SNN system would provide even greater benefits in terms of power consumption and hardware area reduction for edge devices.

2.3 Neuromorphic Systems with Power-optimal Memories

Another way to improve the power efficiency of memory is to apply new technologies to restructure the memory cells such as *In-Memory Computing* (IMC), and *3-D stacking memory*. For instance, the emergence of IMC methods can be divided into analog IMC [16–18] and digital IMC [19–21]. Analog IMC may not be suitable for high-precision applications such AI because it has the disadvantage of low conversion accuracy limited by the low-cost analog-to-digital converters (ADCs), while digital IMC has the advantage of high computational accuracy. Moreover, the analog IMC is also vulnerable to noise caused by temperature, sneak currents, and many other sources of variations [22]. On the other hand, although the digital IMC has robustness and precision, it consumes more power compared to the analog IMC [23]. For the 3-D stacking memory in chips, there are several proposed works [24], [25] to shorten the data movements, which reduces power consumption. With a high bandwidth and a large capacity, 3-D stacking of SRAMs has drawn attention for being a large cache in CPUs and a large memory in DNN inference accelerators [26], [27]. The data communication between 3-D layers can be wired integration using through-silicon vias (TSVs) [24], [25] or a wireless integration using inductive coupling known as ThruChip Interface (TCI) [28]. However, despite these great benefits of 3-D stacking technology, the challenge of this approach is that it has a low yield rate and low reliability. In this thesis, to tackle one of these problems, we propose a 3-D architecture, which can improve the yield rate, by accepting defective layers while maintaining tolerable accuracy.

2.4 Neuromorphic Systems with Low-power Techniques

The *voltage scaling technique* is one of the famous techniques that are widely used for low-power systems. In fact, previous works proved that by applying the under-voltage technique power consumption related to memory could be greatly reduced. For example, Salami *et al.* [29] reduces power consumption by 39% on FPGA on-chip memories, Leng *et al.* [30] saves 20% of power in GPUs, and power consumption of DRAMs in [31] is dropped by 16%. In addition, Minerva [32] lowers the supply voltages of SRAMs to save a total of 2.7× power consumption. In order to accomplish the voltage transformation, the system is required to have an off-chip voltage regulator (VR) with a power switching technique [33], [34] or an on-chip one (i.e.: low-dropout VR [35], [36], switched capacitor VR [37], [38]). Moreover, the under-voltage technique could also be applied to internal components of FPGAs [39] or HBMs (High Bandwidth Memory) [40] to gain around 3× and 2.3× power efficiency, respectively. However, due to the supply voltage reduction, the noise margin of a memory cell is also reduced, which leads to an increase in the probability of errors such as read stability failure, write stability failure, or access time failure [41]. As a result, such small errors could lead to a huge impact on the accuracy of conventional 2-D neural network architectures [39]. This is because there is a chance that the MSBs of weights are affected by reducing the supply voltages of SRAMs. However, with 3-D technology, the weights can be split into multiple subsets placed in separate layers with isolated supply voltage, which is able to protect the memory layers containing MSBs and reduce the supply voltage of memory layers containing LSBs.

Chapter 3

Low-power Methodology for 3-D ICs

Before presenting the implemented architecture, in this section, we would like to illustrate the methodology of 3-D Stacking Synaptic Memory. To the best of our knowledge, this is the first work that utilizes both voltage scaling and power gating partially for memory without a significant drop in accuracy. It is because the prior works [39, 42–44] put all bits into the same voltage domain. As a result, the noise caused by dropping supply voltage to the subthreshold affects the meaningful active bits or MSBs. However, by taking advantage of 3-D ICs and multiple power rails through TSVs, we can isolate the meaningful active bits and the inactive bits into different layers. Hence, we can reduce the supply voltage below the subthreshold or completely power-gate the inactive bits without greatly affecting the final accuracy, unlike the prior works. Another difference between our work and the prior dynamic-voltage-scaling 3-D IC-based architecture [45] is that we also utilize the power-gating technique for the memory layers. Here, assuming that the synaptic weights consist of n-bit and are in fixed point and quantized from the floating point in the case of off-chip training. These bit configurations are unchanged after manufacturing.

3.1 Multiple-level Importance of Memory Weights

Conventionally, all bits are treated as same as each other regardless of their position in the weight. However, we can simply realize that in terms of value, they are not the same. Although spike neural network applications can be noise resilient, flipping bits due to undervolting or power gating still has different impacts on different positions of the bit. Assuming the weight of $n = 8$ bit: $NW[0 : 7] = 10101100$ with one signed bit and seven bits fractional, the differences in values are shown in Table 3.1. In summary, flipping bit in the LSBs has a lesser impact on the value of the weight itself.

Motivated by this, my methodology presents a method to allow power-reduction targeting LSBs. However, we can quickly notice that power-gating or voltage scaling for LSBs is mostly not possible with the native 2-D memory architecture. On the other hand, the 3-D architecture is different. It provides different power nets to each stacking layer. Therefore, the voltage-scaling and power-gating techniques could be applied to the memory layers consisting of LSBs to reduce power

Table 3.1: Difference between bit flipping positions

Value	**Original**	**Flipped bit position**			
		MSB	3^{rd} bit	5^{th} bit	LSB
Binary	10101100	00101100	10001100	10100100	10101101
Float	-0.34375	0.34375	-0.09375	-0.28125	-0.3515625
Diff. (%)	0 (0%)	+0.6875 (+200%)	+0.25 (+72.727%)	+0.0625 (+18.182%)	+0.0078125 (+2.273%)

consumption while maintaining acceptable accuracy.

3.2 3-D Architectures and Methodologies

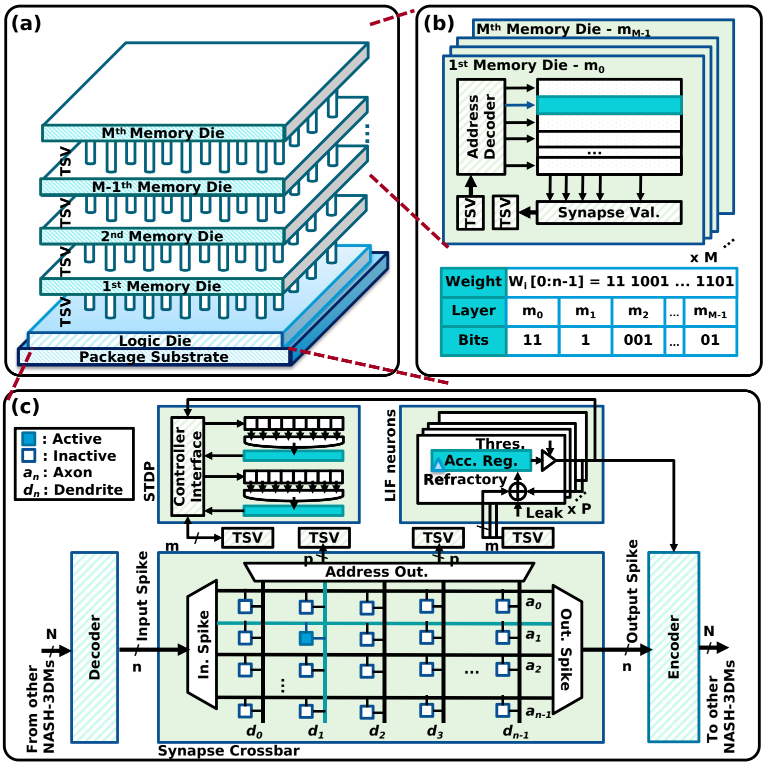

Figure 3.1: The overview hardware architecture of NASH-3DM with 3-D IC-based stacking memory. (a) The hardware contains P Leaky Integrate-and-Fire (LIF) cores and M memory layers stacked on top of it. (b) The bit distribution in M stacking memory layers. (c) The hardware architecture of each LIF core at the logic die.

Fig.3.1 illustrates the architectural overview of our NASH-3DM hardware. Here, we show the NASH-3DM contained P LIF neurons with M stacking memory layers. All neurons or processing elements are placed at the bottom layer (logic die) and the stacked layers (memory die) contain only synaptic memory. The synaptic weights are partitioned into M memory layers, with data transmission via Through-Silicon Vias (TSVs). It is important to highlight that the number of

LIF neurons and memory layers are customizable parameters that can be adjusted during the design phase.

Each neuron inside NASH-3DM has its address decoder and encoder inside to update the synaptic weights correctly. They act as the receiver and transmitter for messages in the network. The output spike of LIF neurons to the next ones could either be in the same NASH-3DM or other NASH-3DMs. On the contrary, the input spike received from the previous neurons triggers the crossbar to attach the corresponding weights from memory layers via TSVs for the LIF function. Each LIF neuron contains one STDP for self-learning and self-updating synaptic weights over operating time.

Let's assume the SNN system uses n-bit weight format for design which stays unchanged after manufacturing. Rather than consolidating one or multiple n-bit weights within a single memory word, our approach involves dividing each p-bit weight into a collection of subset bits $\{m_0, m_1, ...m_{M-1}\}$, where m_i represents subset i and M denotes the total number of subsets. Notably, m_0 represents the subset with the highest significance, while m_{M-1} corresponds to the subset with the lowest significance. The strategy for *in-situ* low-power structure is acquired by the three following modes (I, II, III), which represent the corresponding low-power techniques. However, by reducing the power supply of the memory blocks, the bits of the stored weights may be flipped if the power-supply reduction reaches the near threshold voltage. Therefore, the accuracy may be affected, as shown in Table 3.1. In summary, we define those three modes for easier mentioning in the explanation and evaluation.

- *Normal power mode:* The neuromorphic systems operate without power-gating or voltage-scaling.
- *Low-power mode I:* Voltage-scaling is applied to the neuromorphic systems.
- *Low-power mode II:* Power-gating is applied to the neuromorphic systems.
- *Low-power mode III:* Both voltage-scaling and power-gating are applied to the neuromorphic systems.

If the system is currently at low-power mode and the *normal power mode* is detected, the system gradually restores the supply voltage to every inactive memory layer. The order will be bottom-up, which starts from MSBs among all inactive bits. One of the drawbacks of splitting memory weights is having smaller memory cells which lead to lower density and high power consumption. However, we could solve this issue by merging multiple adjacent weights into a single memory cell [5, 46]. Notably, we utilize multiple power rails for every memory layer to change their power supply. Hence, it is the hardware overhead compared to the traditional voltage scaling. However, our hardware architecture is implemented in 3-D and every memory layer has the same hardware area. As a result, compared to the implementation in 2-D architecture, there is no overhead in hardware footprint. Another concern of this method is that the number of combinations for configuring and deciding low-power mode for each layer is huge. As a result, a standalone optimization algorithm is required to decide the best operating mode in a specific situation.

In the exemplary model as in Fig. 3.1(b), we divide those $n = 8$-bit weights into M separated memory layers. The synaptic weights can be split unevenly into these layers. In addition, the LSBs are on the top memory layer(s) and the MSBs are on the bottom. By separating the bits of synaptic weights into different layers, our hardware architecture is capable of power-gating the top memory layer(s) to act as reducing the bit precision of SNN (called *in-situ* dynamic quantization). The LSBs will be treated as all zero in the processing elements. Consequently, this leads to a significant reduction in overall power consumption while maintaining a graceful level of accuracy. It is suitable for edge devices when their battery or power source almost runs out. This happens by taking advantage of the noise and bit-loss resilience of SNN, which other neural network models usually lose their accuracy sharply because of the operating-bit reduction. Moreover, with the separating structure, this approach has two other benefits. First, the quantization can be operated after manufacturing and without any interruptions in the system operations. Hence, in the case of the power supply reaching a certain low-level threshold, the system could switch to the low-power mode, which reduces a small fraction of accuracy, to increase the operation time. Second, unlike *ex-situ* quantization, the LSBs can be refilled and reattached if necessary during the operations. It is important because the power supply can be also dynamically adjusted or recharged at run time.

3.3 Power Efficiency with Dynamic 3-D Stacking Synaptic Memory

The power consumption of our hardware is similar to other conventional neural network architectures, which is the sum of power consumption by memory storage P_{mem} and power consumption by PEs P_{pe}. In practice, the power consumption from memory is usually dominant, which is about 75% of the total power [8]. It is because the neural network models often require millions of weights to acquire high accuracy and those weights are transferred back and forth in long-distance between memory and PEs. This leads to the huge size of memory, which prolongs the transferring distance and requires more power to transfer those weights in the conventional 2-D systems. However, as mentioned above, the 3-D design of memory-on-logic brings the two most benefits: distance reduction, and footprint reduction, for neural network models in general, and SNNs in particular.

On the other hand, the power consumption of CMOS-based circuits could be further expressed as P_{total}, a sum of two components, the dynamic power P_{dyn} (or active power) and the leakage power P_{leak} (or static power).

$$P_{total} = P_{leak} + P_{dyn} \tag{3.1}$$

Furthermore, those two power consumptions are mathematically represented by the following equations:

$$P_{dyn} = C \times f_{sw} \times V_{DD}^2 \tag{3.2}$$

$$P_{leak} = K \times N \times I_{leak} \times V_{DD} \tag{3.3}$$

where C is the capacitance of the gates, a technology-dependent parameter,

V_{DD} is the supply voltage, and f_{sw} is the switching frequency of hardware systems. Furthermore, K denotes a technology-dependent parameter, N is the number of transistors, and I_{leak} is the leakage currents of circuits. These equations clearly show that power consumption could be significantly reduced by adjusting the supply voltage. In the case of dynamic power, Eq. 3.2 expresses the power reduction in quadratic-fold when scaling down the supply voltage. Moreover, the dynamic power consumption could be further reduced with the power-gating technique, which completely removes the supply voltage. It can only happen in our 3-D hardware architecture because of the multiple-layer memory and the noise resilience of SNNs. Likewise, the leakage power consumption is also reduced linearly, as shown in Eq. 3.3, by implementing the same techniques. Each technique applied to the hardware architecture is explained in the following subsections.

3.3.1 Partial Voltage-scaling for 3D Stacking Synaptic Memory

In this subsection, the power efficiency and the Bit Error Rate (BER) of voltage scaling for stacking synaptic memories in our hardware are analyzed. In addition, since the synaptic memory of our hardware is implemented using SRAM models, the analysis will focus on the BER of SRAM cells. The BER of an SRAM cell is the probability that the Static Noise Margin (SNM) appears to be close to zero [47, 48]. Assuming that SNM has a normal distribution, the BER of an SRAM cell is analytically expressed by the following equation:

$$BER = f(SNM) = \frac{1}{\sqrt{2\pi}\sigma_{SNM}} \exp - \frac{(SNM - \mu_{SNM})^2}{2\sigma_{SNM}^2} \tag{3.4}$$

where σ_{SNM} is the standard deviation of SNM and μ_{SNM} is the mean value of SNM. In practice, these two values vary from one technology to another. It is because SNM depends on the threshold voltage V_T, the supply voltage V_{DD}, and the ratio β, which vary depending on the doping profile, the manufacturing process, and the transistor sizing [49]. Fig. 3.2 shows the BER of 45-nm 6T SRAM with multiple supply voltages near the threshold region. According to Seevinck *et al.* [49], the SNM is estimably calculated by the following equation:

$$SNM = V_T - \left(\frac{1}{k+1}\right)\left[\frac{V_{DD} - \frac{2r+1}{r+1}V_T}{1 + \frac{r}{k(r+1)}} - \frac{V_{DD} - 2V_T}{1 + k\frac{r}{q} + \sqrt{\frac{r}{q}\left(1 + 2k + \frac{r}{q}k^2\right)}}\right] \tag{3.5}$$

where $r = \beta_p/\beta_a$ is the ratio of β between pull-up transistors and access transistors and $q = \beta_d/\beta_a$ is the ratio of β between pull-down transistors and access transistors. k is calculated by the following Eq. 3.6.

$$k = \left(\frac{r}{r+1}\right)\left(\sqrt{\frac{r+1}{r+1-V_s^2/V_r^2}} - 1\right) \tag{3.6}$$

where $V_s = V_{DD} - V_T$ and $V_r = V_s - \left(\frac{r}{r+1}\right)V_T$ [49]. As a result, the BER of an SRAM cell from a specific technology can be approximately obtained. In practice,

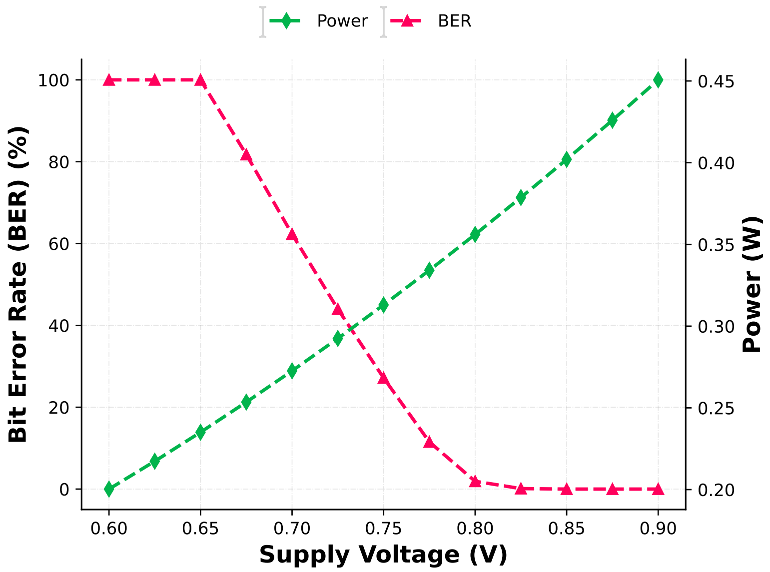

Figure 3.2: The bit error rate vs. power consumption of memory (45-nm 6T SRAM cell) at near-threshold voltage.

Reviriego *et al.* [47] evaluated the BER of SRAM cells approximately around 3.99×10^{-2} and 2.29×10^{-3} at the half of normal supply voltage, $0.4V$, at 16nm CMOS and FinFET technologies, respectively. This BER usually accumulates over time which steadily causes the collapse of memory. This is because the conventional architecture does not support partial undervolting or power-gating the memory. However, our hardware architecture takes advantage of 3-D design to separate the MSBs and LSBs of synaptic weights. Since the MSBs are kept at a different layer with full-voltage protection, the collapse of all memories does not happen. As a result, with the noise resilience of SNNs, the accuracy of our hardware only suffers a fraction of loss, yet its energy efficiency can gain up to twice or threefold depending on the dropping voltage.

In the examplary model shown in Fig. 2.1, our hardware has $M = 4$ memory layers, $\{m_0, m_1, m_2, m_3\}$. Therefore, the total power consumption of the memory P_{mem} could be expressed as the following equation:

$$P_{mem} = \sum_{i=0}^{M-1} P_{m_i} \tag{3.7}$$

where P_{m_i} represents the power consumption of the i^{th} memory layer. In addition, each memory layer has its own dynamic power consumption and leakage power consumption, as shown in Eq. 3.2 and Eq. 3.3, respectively. Assuming that the supply voltages in all four memory layers are the same voltage, V_{DD}, in the *normal power mode*. With the voltage-scaling, those four memory layers then have their specific supply voltages, $\{V_{m_0}, V_{m_1}, V_{m_2}, V_{m_3}\}$. Combining with Eq. 3.1, the power consumption reduction of memory using undervolting could be expressed as the following equation:

$$P'_{mem} = \sum_{i=0}^{M-1} \left(C_i \times f_{sw_i} \times V_{m_i}^2 + K_i \times N_i \times I_{leak_i} \times V_{m_i} \right) \tag{3.8}$$

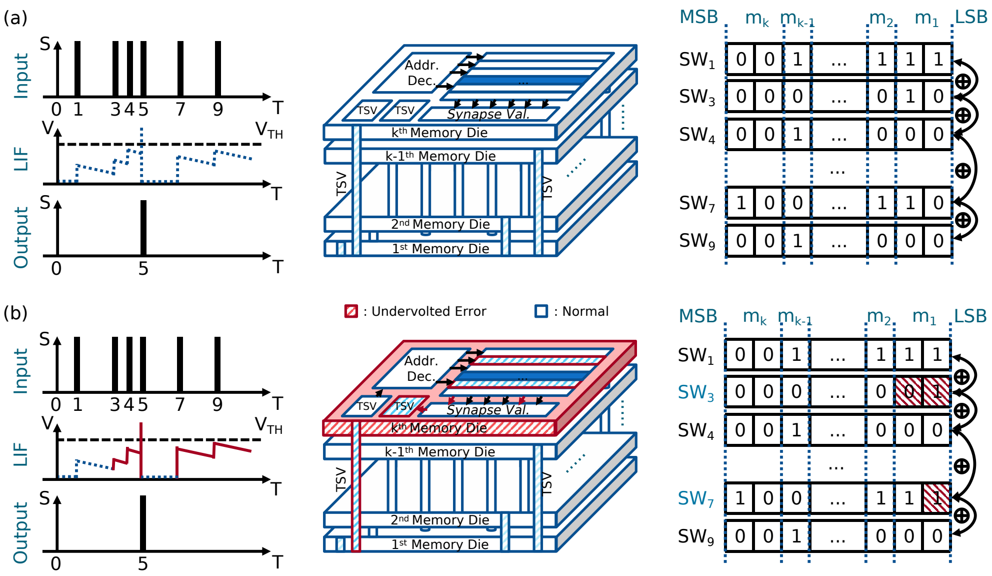

Figure 3.3: Example of 8-bit synaptic weights' operation with undervolting memory layer(s). (a) The operation of our hardware under normal conditions. (b) The operation of our hardware with undervolting for the top memory layer.

where P'_{mem} is the power consumption of all four memory layers when the undervolting is implemented. As a result, the ratio between the power consumption of the undervolting hardware and the power consumption of the normal hardware is approximately equal to the following equation:

$$\frac{P'_{mem}}{P_{mem}} = \frac{\sum_{i=0}^{M-1}\left(C_i \times f_{sw_i} \times V_{m_i}^2 + K_i \times N_i \times I_{leak_i} \times V_{m_i}\right)}{C \times f_{sw} \times V_{DD}^2 + K \times N \times I_{leak} \times V_{DD}} \tag{3.9}$$

To illustrate the power mode I, Fig. 3.3 shows our hardware with undervolting only for the top memory layers and provides the normal supply voltage for the remaining memory layers. In detail, Fig. 3.3(a) shows the normal LIF operation without voltage scaling, and Fig. 3.3(b) demonstrates the LIF operations with the effect of voltage scaling at near-threshold voltage. Here, the red-square areas are the flip-bits due to undervolting. As a result, the flip-bit fault only causes the error in LSBs of synaptic weights and the output spike will not be affected. We first assume that the supply voltage of the top memory layers is reduced by half and there are four stacked memory layers. The total C is $6nF$, $K = 1$, the total number of transistors is 10^9, the normal voltage supply is $1.1V$, and the leakage current is $I_{leak} = 50pA$. Hence, our hardware, which has a switching frequency of $50MHz$, theoretically could save about 17.92% power consumption on the memories while the accuracy of our hardware drops insignificantly because of the noise resilience of SNNs. The drop in accuracy will be later evaluated. In practice, it could extend approximately the operating time of edge devices by 20%, which is in a power-hungry situation without changing its neural network model and hardware components. Moreover, the accuracy is only trade-offed by a marginal volume.

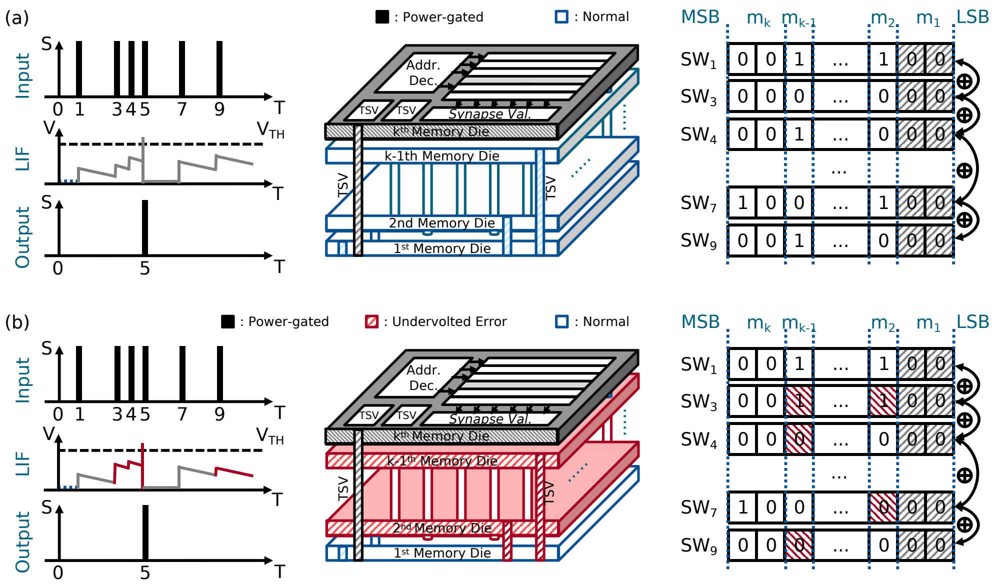

Figure 3.4: Example of 8-bit synaptic weights' operation with undervolting and power-gating memory layer(s). (a) The operation of our hardware with power-gating the top memory layer. (b) The operation of our hardware with power-gating the top memory layer and undervolting two memory layers.

3.3.2 Power-gating for 3D Stacking Synaptic Memory

With the power-gating, our hardware proceeds the *in situ* synaptic weight quantization by turning the memory layer(s) off if the *low-power mode II* is detected and turning it on if the *normal power mode* is detected. Therefore, the alternation of the total power consumption is from the memory. For example, with the n-bit synaptic memory from the architecture in Fig. 2.1, we can define the total power consumption of synaptic memories based on Eq. 3.1.

$$P_{mem} = P_{mem_{leak}} + P_{mem_{dyn}} \tag{3.10}$$

where $P_{mem_{leak}}$ is the leakage power of synaptic memories and $P_{mem_{dyn}}$ is the power consumption of synaptic memories from switching activities. Assuming that the power supply is divided equally into synaptic memories. Hence, when one or more memory layers consisting of t LSBs, are turned off, the power consumption of synaptic memories theoretically reduces by t/n.

$$P'_{mem} = \frac{n-t}{n} \times (P_{mem_{leak}} + P_{mem_{dyn}}) \tag{3.11}$$

This is because all the memories in the layers are unified and have the same switching activities when the input spike event occurs. With $n = 8$ as in Fig. 2.1, the expected power reductions are 25% and 50%, for $t = 2$ and $t = 4$, respectively. Therefore, for each possible value of t, we can define a power-aware mode. In addition, we can also use the voltage-scaling technique for the non-power-gated memory layer(s) to further decrease the overall power consumption. In this case, the system enters the *low-power mode III*.

Fig. 3.4 shows the example of both *low-power mode II* and *low-power mode III*. With the power-gated top layer, the LSBs of synaptic weights are treated as zeros.

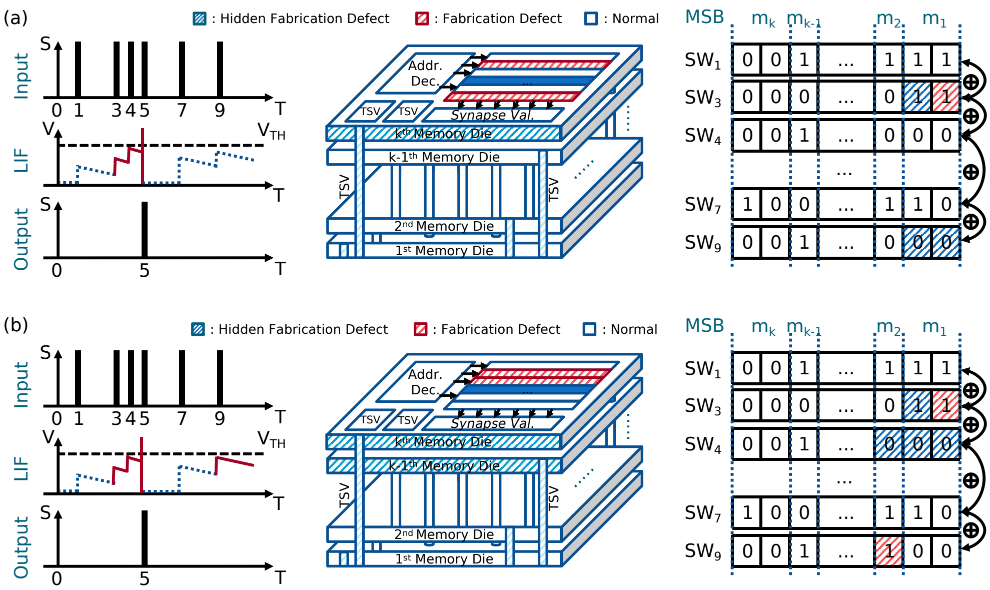

Figure 3.5: Example of 8-bit synaptic weights' operation with fabrication defects. (a) The operation of our hardware with the top memory layer defected by fabrication. (b) The operation of our hardware with two upper memory layers defected by fabrication.

It leads to a slight decrease in the value of synaptic weights but our architecture still receives the correct output spike, as shown in Fig. 3.4(a). On the other hand, in the **low-power mode III** (Fig. 3.4(b)), the synaptic weights in undervolted layers are randomly flipped because of the lack of supply voltage. It also leads to a transformation in the output value of the LIF neuron but the output spike is still correct. It is because the memory layer containing MSBs is untouched. However, the number of untouched MSBs also needs to be considered for the correctness of the SNN model. Despite the noise resilience of SNNs, further dropping the power supply out of the remaining memory layers will cause the spiking computing core to collapse, unable to operate correctly. The evaluation section will demonstrate the experimental results for each operating power-aware mode.

3.3.3 Improving the yield rate by accepting LSBs layers' defects

As we mentioned earlier in the introduction, the low yield rate is one of the most critical issues in stacking 3-D IC-based technology. Assuming the yield rate for a single layer (die) is $Y_{1\_layer} < 1.0$, the yield rate of D layers is Y_{D_layers} which is much smaller than Y_{1_layer}. It is because each layer has its defection and, by stacking multiple layers, the defect probability increases exponentially since we do not know the die quality before stacking. This yield rate can be represented by the following equation:

$$Y_{D\_layers} = \prod_{i=0}^{D-1} Y_i \tag{3.12}$$

where D is the number of layers and Y_i is the yield rate of the i^{th} layer. For example, assuming that all layers have the same yield rate, $Y_{layer} = 0.9$ and the stacked layer is $D = 4$. Therefore, the actual yield rate of the 3D-stacked chip is reduced to 0.6561 and the defect rate is increased to 0.3439.

In detail, the defective layer will cause errors in the logic functions of transistors, which are usually stuck-bit or bridging faults. Without the correctness of logic functions, the fabricated chip cannot work as designed. However, in our architecture, we split the memory and stack them on top of processing elements. As a result, the yield rate of the second layer onward can be categorized generally into two types, which are for the control-logic region in memory, Y_{logic}, and the memory cell region, Y_{mem}.

$$Y_{layer} = Y_{layer_{logic}} \times Y_{layer_{mem}} \tag{3.13}$$

Moreover, the memory cell region takes the most area in memory. On the other hand, in our architecture, fabrication defects in memories are considered noises, as shown in Fig. 3.5. The LIF operations with the defects of the top memory layer and the two upper memory layers are presented in Fig. 3.5(a) and Fig. 3.5(b), respectively. Assuming that we have stuck-at defects in the memory cells of the top layer(s), the bit values at defected regions always stay at $'0'$ or $'1'$. With the noise resilience of SNNs, the output spike is still correct even with defective synaptic weights. We assume that the defects that appeared in the wafer have a uniform distribution. Therefore, the probability that the defects occur in memory is equal to the ratio of hardware area between logic components and memory components multiplied by the yield rate. Assuming that this ratio is approximately one-ninth (α= 1/9) and the total number of layers is $D = 5$. We can have the actual yield rate if we accept defects in $T = 2$ upper memory layers as follows:

$$Y_{D\_layers} \approx \prod_{i=1}^{D-T-1} Y_{layer_i} \prod_{j=D-T}^{D-1} \left[1 - \frac{\alpha}{1+\alpha}(1 - Y_{layer_j})\right] \tag{3.14}$$

Substituting numbers into the equation, the actual yield rate is $Y_{actual} \approx 0.7145$, not 0.5904, which leads to an improved overall yield rate. Therefore, we can accept the manufacturing defects to improve the overall yield rate while reducing a fraction of accuracy.

Chapter 4

Evaluation

4.1 Evaluation Methodology

The proposed hardware architecture was implemented in Verilog-HDL, synthesized, and evaluated with commercial CAD tools from Cadence and Synopsys (Cadence Innovus, Synopsys Design Compiler, PrimeTime, Custom Compiler, HSPICE). The physical design of our hardware is implemented with the NANGATE 45-nm library [50] and NCSU FreePDK3D45 TSV [51]. The system memory is 6T SRAM generated from OpenRAM [52] and its BER characteristic, when undervolting is applied, is calculated from Python based on Eq. 3.4 and is evaluated by HSPICE. In order to evaluate the transformation of power consumption and accuracy, we implemented our hardware as a neuromorphic core with $M = 4$ memory layers stacked on top of $L = 48$ LIF modules. The SNN model embedded into the hardware is configured with a neural network of three layers *(784:48:10)* for the MNIST dataset. We also evaluate the hardware system with the VGG16 model under the CIFAR-10 dataset [53]. Since the hardware design for VGG16 is not available in this work, we estimate the energy consumption via CACTI SRAM's model [54]. The images were encoded into spikes using the rate-coding scheme under the Poisson distribution. In addition, the synaptic weights are trained as $n = 8$-bit values for MNIST, and $n = 16$-bit values for CIFAR-10. They are split equally into four memory layers of the hardware, which is two bits per layer. Please take note that the configurations of the SNN model and our hardware architecture can also be modified into different ones during the design phase.

First, for the *low-power mode I*, we examine the Signal Noise Margin (SNM) of SRAM cells at near-threshold supply voltages to extract the BER or probability of faults according to materials presented in previous works [47–49]. The BER is exported through Monte Carlo simulations with PrimeSim HSPICE and mathematical calculation at multiple supply voltages. After that, we insert the faults according to the extracted probabilities into synaptic weights trained from the software model. The position of faults is distributed randomly using the Monte Carlo simulation again with uniform distribution. Because we implement the hardware with four memory layers, the undervolting evaluation is then categorized into four settings. The modified synaptic weights are then loaded into hardware to evaluate the power consumption and the accuracy of the SNN model affected by undervolting.

Second, the transformation of power consumption and accuracy at *low-power mode II* are evaluated. Similar to the *low-power mode I*, the power-gating hardware also has four settings to inspect. However, the accuracy of our hardware is broken when the supply voltage of the third memory layer is turned off. Therefore, in this thesis, the evaluation only covers three settings which are: normal setting without power-gating any layers, power-gating one layer, and power-gating two layers. In this case, our hardware treats the bit values of synaptic weights as zero(s) and uses them to perform LIF computations. Similarly, the switching activities of power-gating hardware are then loaded into Synopsys PrimeTime to extract power consumption. Third, the *low-power mode III* are evaluated. Because of the time-consuming simulation, we only pick one case out of all combinations to evaluate the power-accuracy transformation. Finally, we evaluate the hardware complexity and compare our system with other works [2, 5, 11, 46, 55–58].

4.2 Undevolting Hardware (Low-power Mode I)

As shown in Fig. 4.1, the evaluation of power transformation and accuracy transformation are taken with supply voltages from $0.7V$ to $0.85V$ with downing $0.025V$ per step. Particularly, Fig. 4.1(a) is the evaluation of accuracy transformation, Fig. 4.1(b) is for energy transformation, and the BER of our SRAM is shown in Fig. 4.1(c). According to the NANGATE 45-nm library [50], the voltage threshold of a transistor is around $0.65V$. As a result, we evaluate the transformation from $0.7V$ to $0.85V$ to capture the best affective region of SNM in the 6T SRAM. Here, the bit order of synaptic weights, as mentioned in Chapter 3, is that the memory layer m_0 contains the MSBs and the memory layer m_3 contains the LSBs. Furthermore, we synchronize all four memory layers ($\{m_0, m_1, m_2, m_3\}$) with the same supply voltage ($V_{m_0} = V_{m_1} = V_{m_2} = V_{m_3} = V_{DD}$). Please take note that the supply voltages could be independent of each memory layer.

Fig. 4.1 shows that the energy per prediction could be reduced $1.4\times$ times when scaling down the supply voltage to $0.85V$ all four memory layers compared to the scaling down of only one memory layer, m_3. However, with the supply voltage going down, which is near to threshold voltage region, the BER of SRAMs starts to increase exponentially. For example, when undervolting only the memory layer m_3, the BER is approximately 0.00029 and 0.001557 at a supply voltage of $0.825V$ and $0.7V$, respectively. The numbers increase to 0.00116 and 0.623 when undervolting to all four memory layers. However, the accuracy of our hardware greatly reduces when undervolting is applied to the third memory layer m_1 $(0.75 - 0.8V)$. It is because the MSBs of synaptic weights start to be affected. In this case, the average accuracy drops from 92.38% to 49.74% with the supply voltage at $0.8V$ and $0.75V$, respectively. In addition, the accuracy swing ($Max_{Accuracy} - Min_{Accuracy}$) also increases greatly, which is from $6.7\%\Big|_{V_{DD}=0.8V}$ to $43.12\%\Big|_{V_{DD}=0.75V}$.

To illustrate the transformation of accuracy under the voltage-scaling, Fig. 4.2 shows the accuracy of our hardware per time step, up to 350-time steps. As seen in Fig. 4.2, the average accuracy in all four undervolting modes at a supply voltage of $0.825V$ is around 92%. The noticeable transformation is that the accuracy significantly swings when undervolting all four memory layers. This is because the MSBs of synaptic weights are affected. However, the BER of SRAMs at

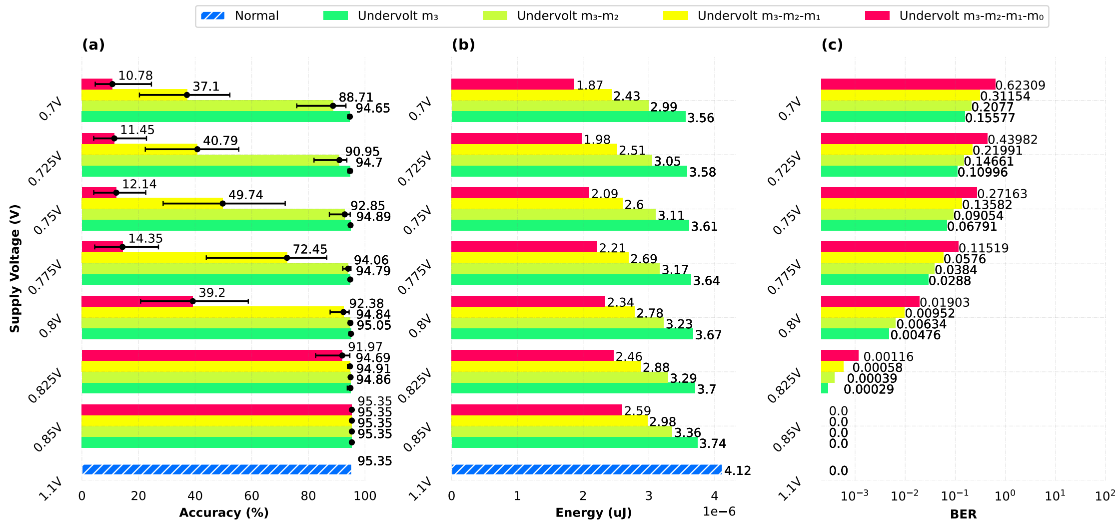

Figure 4.1: The transformation of BER and accuracy and energy with undervolting memory layer(s). (a) Accuracy when undervolting each combination of memory layer(s). (b) Energy when undervolting each combination of memory layer(s). (c) BER when undervolting each combination of memory layer(s).

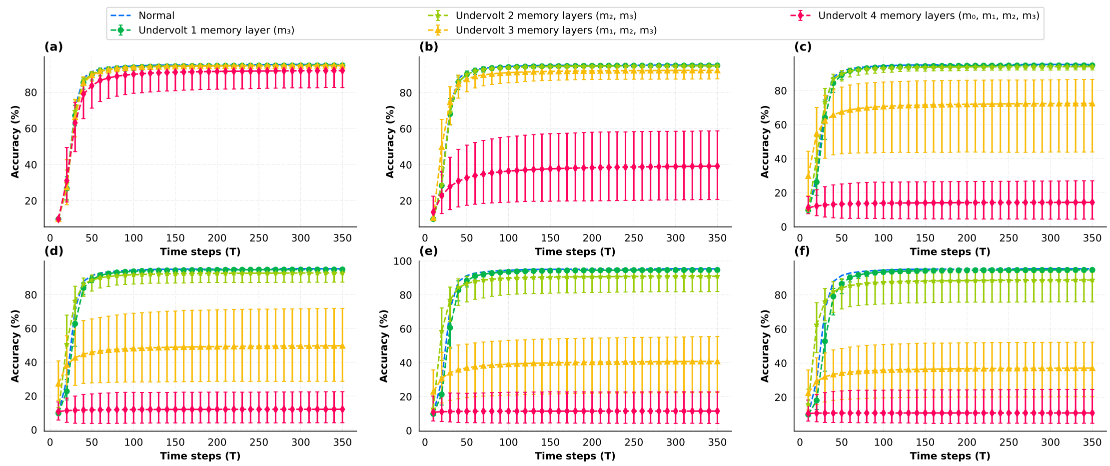

Figure 4.2: Accuracy with undervolting memory layer(s) in every time step. (a) $V_{DD} = 0.825V$; BER = 0.00116. (b) $V_{DD} = 0.8V$; BER = 0.01903. (c) $V_{DD} = 0.775V$; BER = 0.11519. (d) $V_{DD} = 0.75V$; BER = 0.27163. (e) $V_{DD} = 0.725V$; BER = 0.43982. (f) $V_{DD} = 0.7V$; BER = 0.62309.

this supply voltage is low (0.00116). Therefore, the number of modified synaptic weights is low and the worst case for accuracy is around 82.58%. With the supply voltage scaling down, the average accuracy curves of undervolting three memory layers and undervolting all memory layers are steadily dropped, while the ones from undervolting two memory layers and undervolting one memory layer are only changed slightly. Consequently, undervolting memory layers containing LSBs can lead to achieving high energy efficiency while maintaining acceptable accuracy.

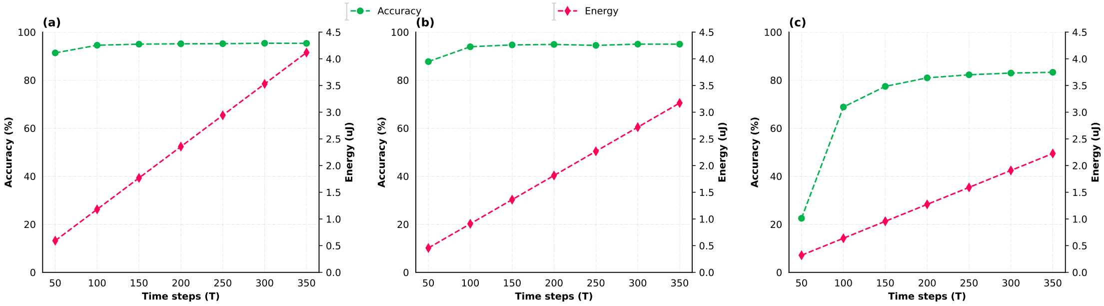

Figure 4.3: Accuracy and Energy Consumption of our hardware in different power-gating modes. a) Trade-off Accuracy vs. Energy at normal operations (no power-gating). b) Trade-off Accuracy vs. Energy when power-gating m_3 c) Trade-off Accuracy vs. Energy when power-gating m_3, m_2.

Table 4.1: The settings for the evaluation of low-power mode II.

Name	**Setting II-1**	**Setting II-2**	**Setting II-3**
Defination	Normal operation	Power-gating one memory layer	Power-gating two memory layers
Power-gated layer	-	m_3	m_2, m_3
# Active bits	8 bits	6 bits	4 bits

4.3 Power-gating Hardware (Low-Power Mode II)

In this section, we evaluate the power transformation and accuracy transformation of our hardware when power-gating the memory layer(s). Our hardware architecture can gain power efficiency by power-gating the memory layers containing LSBs depending on the power situation. Moreover, with the proposed architecture, the *in-situ* dynamical quantization for synaptic weights was achieved without modifying the hardware components. Therefore, we evaluate with two factors: (1) the accuracy when removing the LSBs by power-gating memory layer(s) and (2) the energy efficiency when power-gating. In this thesis, we evaluate the accuracy of our hardware and its energy consumption in three operation settings, as shown in Table 4.1.

As shown in Fig. 4.3, the accuracy of our power-gated hardware at the 350^{th} computing time-step reaches 95.32%, 94.98%, and 83.28% for each power setting, respectively. This is a very strong indicator that we may be able to offer low-power modes in the trade-off of accuracy loss. In fact, at the 100^{th} computing time-step, the accuracy of our system drops to 94.49%, 93.96%, and 68.71% in each power-gating setting. The accuracy of 4-bit synaptic operations (Fig. 4.3(c)), when applying the setting II-3, loses about 15% compared to the 8-bit operations (Fig. 4.3(a)). On the other hand, the accuracy is only reduced slightly by 1% when applying the setting II-2 (Fig. 4.3(b)). Here, we can observe that power consumption could be also reduced greatly with the right time step while maintaining a reasonable accuracy. In terms of energy, this reduction in computing

Table 4.2: The settings for the evaluation of low-power mode III.

Name	**Setting III-1**	**Setting III-2**	**Setting III-3**
Defination	Undervolting two memory layers	Power-gating one memory layer, Undervolting two memory layers	Power-gating two memory layers, Undervolting two memory layers
Power-gated layer	-	m_3	m_2, m_3
Under-volted layer	m_3, m_2	m_1, m_2	m_0, m_1
Supply Voltage $\{V_{m_0}$; V_{m_1}; V_{m_2}; $V_{m_3}\}$	$1.1V$; $1.1V$; $[0.675{-}0.8V]$; $[0.675-0.8V]$	$1.1V$; $0.8V$; $[0.675{-}0.8V]$; $0V$	$0.825V$; $[0.675{-}0.8V]$; $0V$; $0V$
# Active bits	8 bits	6 bits	4 bits

time-step leads to a reduction in energy per prediction and energy per Synaptic OPeration (SOP). For the total energy consumption per time-step with the same bit-width synaptic operation, it increases from the 50^{th} time-step to the 350^{th} one approximately by 7× fold.

4.4 Undervolting and Power-gating Hardware (Low-Power Mode III)

In this section, we investigate the power-accuracy transformation of our hardware when mixing the voltage-scaling and power-gating techniques for memory layer(s). For the power-gating, the supply voltage of the power-gated memory layer is treated as zero. In this thesis, we have four stacked memory layers. Therefore, the configuration of supply voltage for each layer is $\{V_{m_0}, V_{m_1}, V_{m_2}, V_{m_3}\}$. Due to the time-consuming simulation, we chose to evaluate only three settings out of all combinations with 1,000 tests from the Monte-Carlo simulation each. The configurations are defined in Table4.2 and its evaluation is illustrated in Fig.4.4.

As shown in Fig. 4.4(a), the average accuracy of setting III-1 in 1,000 tests at the supply voltage $V_{DD} = 0.8V$ is similar to the normal operation of our hardware and this accuracy reduces by 1-2% per undervolting step. In the worst test, the accuracy drops about 20% compared to the one at the normal operation condition. However, the energy efficiency gains 25%. The energy continues to drop when power-gating is applied to the top layer and undervolting two middle layers (Fig. 4.4(b)). Compared to the normal operation, it is reduced by half yet the average accuracy only reduces slightly. The only noticeable concern is that the range of accuracy is expanded, and the worst accuracy is 55.27% (dropped about 40% of accuracy compared to the normal operation). As we continue to drop the supply voltage (Fig. 4.4(c)), the accuracy swings stronger. Consequently, the worst accuracy is 22.76% at $V_{m_1} = 0.675V$ and $V_{m_0} = 0.825V$. However, at $V_{m_1} = 0.8V$, we can see that the energy is reduced four times compared to the normal operation while reducing 6.57% in accuracy.

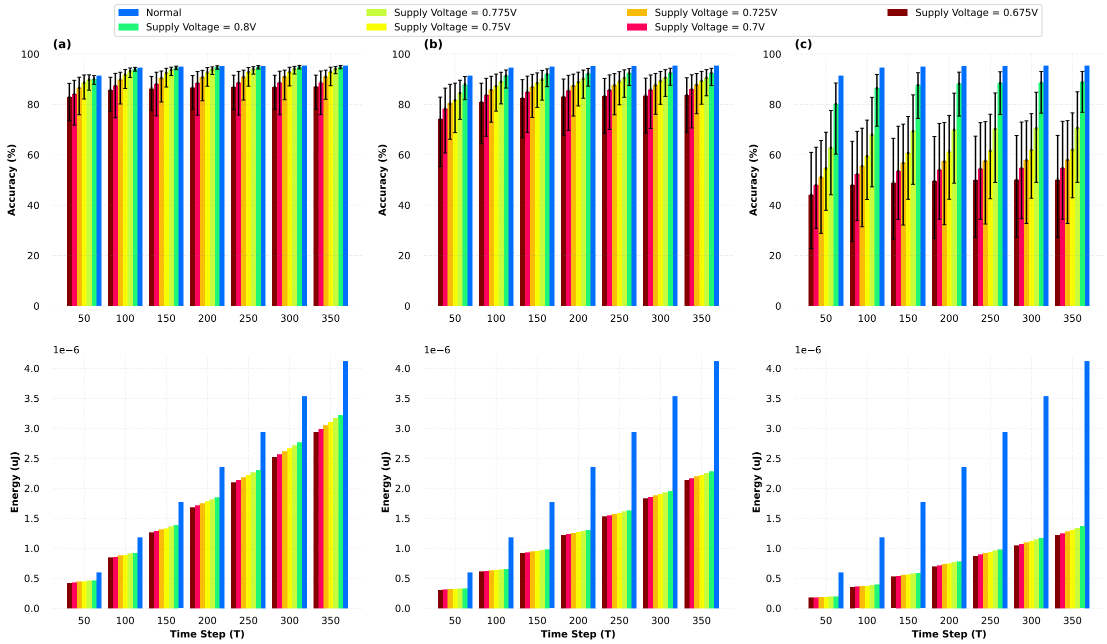

Figure 4.4: The evaluation of accuracy and energy with both power-gating and undervolting. The supply voltage of the power-gated layer is treated as zero. a) Accuracy transformation and Energy transformation with setting III-1. b) Accuracy transformation and energy transformation with setting III-2. c) Accuracy transformation and energy transformation with setting III-3.

Table 4.3: The accuracy and the yield of our hardware with two upper defected memory layers (The normal accuracy = 95.35%).

Yield Rate per Layer	**Avg. Acc.**	**Min. Acc.**	**Max. Acc.**	**Avg. Acc. Loss**	**Normal Yield**	**Yield Improv.**
$Y_1 = 0.999$	94.97%	94.45%	95.38%	0.38%	0.995	0.9968 (+0.18%)
$Y_2 = 0.99$	94.71%	93.25%	95.45%	0.64%	0.951	0.9683 (+1.73%)
$Y_3 = 0.9$	93.85%	91.38%	95.05%	1.70%	0.5905	0.7145 (+12.40%)

4.5 Accuracy with defected memory layers

As explained in Section 3.3.3, the defective memory caused by fabrication is treated as noise for our proposed architecture and we accept these manufacturing defects to increase the yield rate. In this section, we evaluate the accuracy of our design with three different yield rates in one wafer, which are $Y_1 = 0.999$, $Y_2 = 0.99$, and $Y_3 = 0.9$. With the assumption in Section 3.3.3, the defects that appeared in the wafer have a uniform distribution. Therefore, we insert the stuck-bits events into memory with the corresponding probabilities to evaluate the trade-off between accuracy and yield rate. In this case, the yield rate improvement is calculated based on Eq. 3.14.

Table 4.3 shows the accuracy of our hardware over 1,000 Monte-Carlo simulation tests. In each yield rate, we evaluate the accuracy with $M = 4$ stacking memory layers and one computing layer, which represents our evaluated architec-

Table 4.4: Hardware complexity of the proposed architecture.

Technology		$45nm$
Frequency		$100MHz$
# LIF		48 LIFs
# Stacking Memory		4 layers
# bit of Synaptic Weights		8 bits
Bit Configuration in Memory Layer		2-2-2-2
Gate Count	Total	$809.98KGEs$
	Memory Blocks	$791.76KGEs$
	Crossbar & Address Decoder	$9.68KGEs$
	LIFs	$8.52KGEs$

ture. Overall, the average accuracy in all cases drops by $0.38\% - 1.7\%$ compared to the accuracy in normal conditions (95.35%). In addition, the result in the worst case drops 3.97%, which we could consider accepting the manufacturing defect to increase the yield rate. Furthermore, in some cases, the stuck-bit event even leads to an increase in the accuracy of our hardware, which is maximally about 0.1%. In conclusion, the yield rate of the 3-D stacked chip is recently low (e.g.: $Y = 0.5904$ when $D = 5$ and $Y_{layer} = 0.9$). On the other hand, our architecture is able to improve this yield rate by 12.40% with the acceptance of defective memory layers. The trade-off comes with a reduction of about 1.7% in accuracy.

4.6 Hardware Complexity and Comparison

As shown in Table 4.4, the area cost of our synthesized hardware is about $809.98KGEs$ at the operating frequency of $100MHz$. In detail, the synaptic SRAM-based memory occupies the largest part of the hardware area, which is around 97% because it is necessary to store a large number of synaptic weights for high accuracy. For the rest, the processing elements and control units occupy about 3% of the total area of our hardware.

Table 4.5 represents the comparison results between our work and other existing works [2,5,11,46,58], which are all based on the MNIST benchmark. In terms of accuracy, the result shows that our system has an accuracy of 95.32% in normal conditions. Furthermore, we pick two other configurations (case 1 and case 2), which use undervolting and power-gating for memory layers. The configurations of supply voltage for each memory layer are: case 1 is $\{V_{m_0} = 1.1V; V_{m_1} = 1.1V; V_{m_2} = 0.8V; V_{m_3} = 0.8V\}$, case 2 is $\{V_{m_0} = 0.825V; V_{m_1} = 0.8V; V_{m_2} = 0V; V_{m_3} = 0V\}$, and case 3 is $\{V_{m_0} = 0.825V; V_{m_1} = 0.8V; V_{m_2} = 0.8V; V_{m_3} = 0V\}$. As shown in Table 4.5, in case 2, with the operation of 4-bit synaptic weights, the accuracy drops by 6.58% compared to the normal operation (8-bit). However, this accuracy is similar to the works of *Kim et al.* [57] and *ODIN* [46], which also operates at 4-bit synaptic weight precision.

In terms of power, we compare our work with others using the energy per synaptic operation parameter. Due to the gap in technology, we use the well-known scaling equation from *Stillmaker et al.* [59] to scale down the 14-nm technology node. As shown in Table 4.5, our hardware consumes $244.28pJ$, $191.46pJ$, and $81.16pJ$ at the 45-nm technology node in three cases for 350 time-steps, re-

Table 4.5: Comparison results between the proposed architecture and existing works.

Parameters	TrueNorth [5]	Loihi [2]	ODIN [46]	Karimi *et al.* [58]	This work					
					Normal Case	Case 1	Case 2	Normal Case	Case 1	Case 3
Benchmark	MNIST	MNIST	MNIST	MNIST	MNIST (784:48:10)			CIFAR-10 (VGG16)*		
Accuracy (%)	91.94	96	84	99.2	95.35	94.84	88.77	91.38	91.26	69.50
Neuron Model	IF	DenMem	LIF & Izhikevicz	LIF	LIF					
Synaptic Weight Storage	1-bit SRAM	1-to-9-bit SRAM	4-bit SRAM	CTT twin-cell	8-bit SRAM			16-bit SRAM		
Interconnect	2-D	2-D	2-D	2-D	3-D					
Implementation	Digital	Digital	Digital	Mix-signal	Digital			Software simulation		
Learning Rule	Un-supervised	On-chip STDP	On-chip Stochastic SDSP	Off-chip	Off-chip					
Technology	28nm	14nm FinFET	28nm FD-SOI	22nm FD-SOI	45nm					
Supply Voltage	0.7-1.05V	0.5-1.2 V	0.55-1 V	0.8 V	0.65V - 1.1V					
Energy per SOP (pJ)	26 (0.775V)	23.6 (0.75V)	8.4	8	244.28 (1.1V)	191.46[1]	81.16[2]	475.20 (1.1V)	372.13[1]	205.55[3]
Energy per SOP (pJ) (in 14nm)	4.902	23.6	1.078	4.32	14.02 (1.1V)	10.98[1]	4.65[2]	27.27 (1.1V)	21.35[1]	11.79[3]

Case 1: $\{V_{m_0} = 1.1V;\ V_{m_1} = 1.1V;\ V_{m_2} = 0.8V;\ V_{m_3} = 0.8V\}$ (Low-power Mode I)
Case 2: $\{V_{m_0} = 0.825V;\ V_{m_1} = 0.8V;\ V_{m_2} = 0V;\ V_{m_3} = 0V\}$ (Low-power Mode III)
Case 3: $\{V_{m_0} = 0.825V;\ V_{m_1} = 0.8V;\ V_{m_2} = 0.8V;\ V_{m_3} = 0V\}$ (Low-power Mode III)

spectively. After scaling down to the 14-nm technology, our energy per synaptic operation achieves the values, which accordingly are $14.02pJ$, $10.98pJ$, and $4.65pJ$. Furthermore, we also evaluate our methodology with the 16-bit VGG-16 using the CIFAR-10 dataset. As shown in Table 4.5, the accuracy only drops slightly by 0.12% while the energy per SOP decreases significantly by 21.68% in case 1. However, in the case 3, despite the energy reduction of 56.74%, the accuracy is also reduced seriously by 21.88%.

In conclusion, these results show that our architecture with 3-D stacking memory has an advantage in terms of reducing energy consumption when applying voltage-scaling and power-gating techniques for memory layers. For the MNIST dataset, switching from the normal mode to the low-power mode I, the accuracy drops by 0.51% to trade-off the energy reduction of 21.62%. When our hardware switches to the low-power mode III, the accuracy drops by 6.58% to reduce the energy consumption by 66.77%. In the case of the CIFAR-10 dataset, with the software simulation, the accuracy also drops by a small fraction (0.12%) to reduce 21.68% energy per synaptic operation when switching from the normal mode to the low-power mode I. Moreover, at the low-power mode III, the accuracy decreases by 21.88% saving 56.74% of energy consumption.

Chapter 5

Impacts of Low-power 3-D IC-based Methodology

In this section, we provide some discussions related to the limitations of our work and potential solutions. First, besides the reliability issue of stacking layers, Through-Silicon-Via's (TSVs) reliability is also one of the major concerns. There are numerous works on dealing with TSV defects by using redundancies. Therefore, these techniques can be embedded into our architecture to deal with TSV defects. Unlike TSV defects which can be dealt with by using redundancies, defects on stacking memory dies are mostly unrepairable; therefore, we focus on this type of defects in this work.

Second, thermal dissipation is another critical issue of 3-D ICs as stacking multiple layers prevents the heat transmission to the heatsink. Although the thermal issue is still an open problem in this work, by lowering the power consumption; our work has the potential to alleviate this issue of 3-D ICs.

Third, as we show in the evaluation section there are numerous combinations of different voltages and power gating. Also, the scaling step of the voltage can also be adjusted which leads to more voltages being chosen. Moreover, the splitting method of the memory can be also different between designs (i.e., 16-bit can be 4×4bit or 2×8bit or 8×2bit) or can be asymmetric (i.e., 8-bit can be two subsets of $3 + 5$bit or $4 + 4$bit or $5 + 3$bit) to isolate the meaningful bits and to reduce the power of inactive bits. Because of this, it is not possible to cover all possible cases to specify the standard of faulty-energy-accuracy trade-off. Hence, our picks of configuration in the comparison in Table 4.5 may be suboptimal. To solve this issue, one of the methods is to perform an optimization process (i.e. Genetic Algorithm or Particle Swarm Optimization). However, in combination with the Monte-Carlo simulation, as we have shown in the evaluation, the number of searching values can be overwhelming.

Fourth, although our work focuses on SRAM which is easily accessible, there is a possibility to apply our methodology to advanced memory technologies (eDRAM, STT-RAM, ...). In fact, this could be even more power efficient as non-volatile memories are more efficient in terms of power and can retain their value after the power gating period.

Fifth, our work focuses on an array of LIF array; however, this method can also be applied for large-scale Network-on-Chip-based architecture [12]. As each NoC core can be undervolted and power-gated separately, this could open a more fine-grained control for the system. Furthermore, the power of spike generation and spike transmission are two other factors that can affect the power consumption of the chip and must be considered in the future.

Sixth, our work utilizes multiple power rails through TSVs to supply power for every memory layer, which is dependent on an off-chip voltage regulator. However, an on-chip voltage regulator can also be implemented into the neuromorphic systems for better scalability. In this case, the hardware overhead is also needed to consider when applying multiple supply voltages for every memory layer. For example, the hardware area of the voltage regulator in [36] is around $0.375\mu m^2$ ($0.111\mu m^2$ without wired area) with the UMC 1.1V 40-nm CMOS technology. Hence, by putting this regulator into our memory layer under 45-nm CMOS technology and ignoring the wired area, the hardware overhead is mathematically about 27.05%, where the total area of memory blocks in one memory layer without wired is $0.337\mu m^2$. As a result, it could add up to a significant hardware area for voltage scaling in every memory layer. However, the hardware footprint is unchanged compared to the traditional 2-D DVS one. It is because our hardware architecture is implemented in 3-D and every memory layer has the same hardware area.

Although there are several drawbacks in this work, the proposed methodology and its implemented architecture have shown the potential to be able to reduce power consumption with graceful performance degradation.

Chapter 6

Conclusion

In this Master's study, we have proposed a methodology to split and stack the synaptic memories for low-power operation. With the 3-D technology, the memory can be isolated into different layers, which allows the possibility to separately control the supply voltage of each layer. As a result, the proposed architecture can apply the voltage-scaling technique and also further turn on/off the power supply of one or multiple layer(s) inside it to save the overall energy consumption. In addition, by splitting the synaptic weights into multiple memory layers, the accuracy can be maintained by protecting the memory layer(s) containing the MSBs while dropping the supply voltage of the memory layer(s) containing LSBs. Our future works will extend this work into a very large-scale system using Network-on-Chips with an optimal power-saving strategy.

However, our proposed work still has some drawbacks. First, the combination of splitting weight bits is sub-optimal. This is because there are many combinations to divide synaptic weights. To address this problem, our future works will investigate optimization algorithms such as Genetic Algorithms or Particle Swarm Optimization. Second, adding more low-power techniques (e.g., voltage-scaling, clock-gating) can further improve this work. Hence, one of our future works will be a combination of our quantization with lowering the memory voltage to further reduce the power consumption and the integration of large-scale systems using Network-on-Chips.

Third, the 3-D stacking SRAMs used in this thesis are only one of many memory types and our proposed architecture is able to work with any type of memory. In the future, we will investigate and implement other memory technologies such as 2-D SRAM, ReRAM, and go on, to evaluate their power consumption in our system. Furthermore, we would like to investigate our yield improvement mechanism as a fail-safe future with the help of fault detection or testing.

References

[1] M. Merenda and *et al.*, "Edge Machine Learning for AI-Enabled IoT Devices: A Review," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2533

[2] M. Davies and *et al.*, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[3] B. V. Benjamin and *et al.*, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[4] W. Guo and *et al.*, "Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems," *Frontiers in Neuroscience*, vol. 15, 2021.

[5] F. Akopyan and *et al.*, "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[6] H. An and *et al.*, "Three-Dimensional Neuromorphic Computing System With Two-Layer and Low-Variation Memristive Synapses," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 3, pp. 400–409, 2022.

[7] G. Orchard and *et al.*, "Efficient Neuromorphic Signal Processing with Loihi 2," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 2021, pp. 254–259.

[8] R. V. W. Putra and *et al.*, "EnforceSNN: Enabling resilient and energy-efficient spiking neural network inference considering approximate DRAMs for embedded systems," *Frontiers in Neuroscience*, vol. 16, 2022.

[9] A. Ben Abdallah and K. N. Dang, "Toward Robust Cognitive 3D Brain-Inspired Cross-Paradigm System," *Frontiers in Neuroscience*, vol. 15, 2021.

[10] T. Wunderlich and *et al.*, "Demonstrating Advantages of Neuromorphic Computation: A Pilot Study," *Frontiers in Neuroscience*, vol. 13, 2019.

[11] O. M. Ikechukwu and *et al.*, "On the Design of a Fault-Tolerant Scalable Three Dimensional NoC-Based Digital Neuromorphic System With On-Chip Learning," *IEEE Access*, vol. 9, pp. 64 331–64 345, 2021.

[12] K. N. Dang and *et al.*, “MigSpike: A Migration Based Algorithms and Architecture for Scalable Robust Neuromorphic Systems,” *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 602–617, 2022.

[13] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[14] M. Rahimi Azghadi and *et al.*, “Spike-Based Synaptic Plasticity in Silicon: Design, Implementation, Application, and Challenges,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 717–737, 2014.

[15] S. Furber, “Large-scale neuromorphic computing systems,” *Journal of Neural Engineering*, vol. 13, 08 2016.

[16] E. Lee and *et al.*, “A Charge-Domain Scalable-Weight In-Memory Computing Macro With Dual-SRAM Architecture for Precision-Scalable DNN Accelerators,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3305–3316, 2021.

[17] M. E. Sinangil and *et al.*, “A 7-nm Compute-in-Memory SRAM Macro Supporting Multi-Bit Input, Weight and Output and Achieving 351 TOPS/W and 372.4 GOPS,” *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, 2021.

[18] S. Jain, L. Lin, and M. Alioto, “±CIM SRAM for Signed In-Memory Broad-Purpose Computing From DSP to Neural Processing,” *IEEE Journal of Solid-State Circuits*, vol. 56, no. 10, pp. 2981–2992, 2021.

[19] H. Kim and *et al.*, “A 16K SRAM-Based Mixed-Signal In-Memory Computing Macro Featuring Voltage-Mode Accumulator and Row-by-Row ADC,” in *2019 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2019, pp. 35–36.

[20] A. Agrawal and *et al.*, “X-SRAM: Enabling In-Memory Boolean Computations in CMOS Static Random Access Memories,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4219–4232, 2018.

[21] W. Simon and *et al.*, “A Fast, Reliable and Wide-Voltage-Range In-Memory Computing Architecture,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[22] M. Hu and *et al.*, “Dot-Product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-Vector Multiplication,” in *Proceedings of the 53rd Annual Design Automation Conference*, ser. DAC ’16. New York, NY, USA: Association for Computing Machinery, 2016.

[23] M. R. Haq Rashed and *et al.*, “Hybrid Analog-Digital In-Memory Computing,” in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021, pp. 1–9.

[24] K. Cho and *et al.*, “SAINT-S: 3D SRAM Stacking Solution based on 7nm TSV technology,” in *IEEE Hot Chips Symposium*, 2020, pp. 1–13.

[25] N.-D. Nguyen and *et al.*, "An In-Situ Dynamic Quantization With 3D Stacking Synaptic Memory for Power-Aware Neuromorphic Architecture," *IEEE Access*, vol. 11, pp. 82 377–82 389, 2023.

[26] M. Evers and *et al.*, "The AMD Next-Generation "Zen 3" Core," *IEEE Micro*, vol. 42, no. 3, pp. 7–12, 2022.

[27] K. Ueyoshi and *et al.*, "QUEST: Multi-Purpose Log-Quantized DNN Inference Engine Stacked on 96-MB 3-D SRAM Using Inductive Coupling Technology in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 186–196, 2019.

[28] K. Shiba and *et al.*, "A 96-MB 3D-Stacked SRAM Using Inductive Coupling With 0.4-V Transmitter, Termination Scheme and 12:1 SerDes in 40-nm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 2, pp. 692–703, 2021.

[29] B. Salami and *et al.*, "Comprehensive Evaluation of Supply Voltage Underscaling in FPGA on-Chip Memories," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2018, pp. 724–736.

[30] J. Leng and *et al.*, "Safe limits on voltage reduction efficiency in GPUs: A direct measurement approach," in *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2015, pp. 294–307.

[31] K. K. Chang and *et al.*, "Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, jun 2017.

[32] B. Reagen and *et al.*, "Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 267–278.

[33] L. Di and *et al.*, "Power switch characterization for fine-grained dynamic voltage scaling," in *2008 IEEE International Conference on Computer Design*, 2008, pp. 605–611.

[34] Z. Bai and *et al.*, "A Cascaded Multilevel Battery Energy Storage Based Parallel Dynamic Voltage Compensator for Medium Voltage Industrial Distribution Systems," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2023.

[35] N. Adorni and *et al.*, "A 10-mA LDO With 16-nA IQ and Operating From 800-mV Supply," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 2, pp. 404–413, 2020.

[36] C.-H. Huang and W.-C. Liao, "A High-Performance LDO Regulator Enabling Low-Power SoC With Voltage Scaling Approaches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 5, pp. 1141–1149, 2020.

[37] P. H. McLaughlin and *et al.*, "A Monolithic Resonant Switched-Capacitor Voltage Regulator With Dual-Phase Merged-LC Resonator," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 12, pp. 3179–3188, 2020.

[38] D. Lutz and *et al.*, "12.4 A 10mW fully integrated 2-to-13V-input buck-boost SC converter with 81.5% peak efficiency," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, 2016, pp. 224–225.

[39] B. Salami and *et al.*, "An Experimental Study of Reduced-Voltage Operation in Modern FPGAs for Neural Network Acceleration," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2020, pp. 138–149.

[40] S. S. N. Larimi and *et al.*, "Understanding Power Consumption and Reliability of High-Bandwidth Memory with Voltage Underscaling," *CoRR*, vol. abs/2101.00969, 2021.

[41] S. Mukhopadhyay and *et al.*, "Statistical design and optimization of SRAM cell for yield enhancement," in *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, 2004, pp. 10–13.

[42] R. G. Dreslinski and *et al.*, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[43] J. Zhang and *et al.*, "Thundervolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Learning Accelerators," in *Proceedings of the 55th Annual Design Automation Conference*, 2018.

[44] P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "GreenTPU: Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[45] J. Zhao and *et al.*, "An energy-efficient 3D CMP design with fine-grained voltage scaling," in *2011 Design, Automation & Test in Europe*, 2011, pp. 1–4.

[46] C. Frenkel and *et al.*, "A 0.086-mm2 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28nm CMOS," *IEEE Transactions on Biomedical Circuits and Systems*, pp. 1–1, 2018.

[47] P. Reviriego and *et al.*, "Error-Tolerant Data Sketches Using Approximate Nanoscale Memories and Voltage Scaling," *IEEE Transactions on Nanotechnology*, vol. 21, pp. 16–22, 2022.

[48] P. Royer and M. López-Vallejo, "Using pMOS Pass-Gates to Boost SRAM Performance by Exploiting Strain Effects in Sub-20-nm FinFET Technologies," *IEEE Transactions on Nanotechnology*, vol. 13, no. 6, pp. 1226–1233, 2014.

[49] E. Seevinck and *et al.*, "Static-noise margin analysis of MOS SRAM cells," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, pp. 748–754, 1987.

[50] N. Inc. Nangate Open Cell Library 45 nm. [Online]. Available: http://www.nangate.com/

[51] N. E. D. Automation. FreePDK3D45 3D-IC Process Design Kit. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK3D45

[52] M. R. Guthaus and *et al.*, "OpenRAM: An open-source memory compiler," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1–6.

[53] A. Krizhevsky, "Learning multiple layers of features from tiny images," Canadian Institute for Advanced Research, Tech. Rep., 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[54] N. Muralimanohar and *et al.*, "CACTI 6.0: A tool to model large caches," *HP laboratories*, vol. 27, p. 28, 2009.

[55] J. Schemmel and *et al.*, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 1947–1950.

[56] J.-s. Seo and *et al.*, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *2011 IEEE Custom Integrated Circuits Conference (CICC)*, 2011, pp. 1–4.

[57] J. K. Kim and *et al.*, "A 640M pixel/s 3.65mW sparse event-driven neuromorphic object recognition processor with on-chip learning," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, 2015, pp. C50–C51.

[58] M. Karimi and *et al.*, "Ctt-based scalable neuromorphic architecture," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 1–1, 2023.

[59] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180nm to 7nm," *Integration*, vol. 58, pp. 74–81, 2017.