PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE

SCHOOL OF ENGINEERING

FERROELECTRIC MEMORY AND ARCHITECTURE FOR DEEP NEURAL NETWORK TRAINING IN RESISTIVE CROSSBAR ARRAYS

CRISTÓBAL ALESSANDRI AMENÁBAR

Thesis submitted to Pontificia Universidad Católica de Chile and University of Notre Dame in partial fulfillment of the requirements for the Degree of Doctor in Engineering Sciences

Advisor:

ÁNGEL ABUSLEME

CHRISTIAN GUZMÁN

ALAN SEABAUGH

Santiago de Chile, April, 2019

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

FERROELECTRIC MEMORY AND ARCHITECTURE FOR DEEP NEURAL NETWORK TRAINING IN RESISTIVE CROSSBAR ARRAYS

CRISTÓBAL ALESSANDRI AMENÁBAR

Members of the Committee:

ÁNGEL ABUSLEME

CHRISTIAN GUZMÁN

ALAN SEABAUGH

CHRISTIAN OBERLI

PATRICK FAY

SIDDHARTH JOSHI

JUAN DE DIOS ORTÚZAR

Thesis submitted to Pontificia Universidad Católica de Chile and University of Notre Dame in partial fulfillment of the requirements for the Degree of Doctor in Engineering Sciences

Santiago de Chile, April, 2019

To Antonia

ACKNOWLEDGMENTS

I would like to thank my advisors Professor Alan Seabaugh, Professor Ángel Abusleme and Professor Dani Guzmán for their guidance and support throughout this thesis work. Special thanks to Ángel for introducing me to academic research and motivating me to pursue a PhD degree, and to Alan for giving me the advice, tools and freedom to successfully explore and develop my research path.

I would also like to thank Professors Siddharth Joshi, Patrick Fay, Suman Datta, Sharon Hu, Christian Oberli, Jaime Anguita and Marcelo Guarini for their support, insightful discussions and feedback throughout my PhD. I also thank Arman Kazemi, Sara Fathipour, Erich Kinder, Mina Asghari, Karla Gonzalez, Paolo Paletti, Enrique Álvarez, Diego Ávila and Matías Jara for many collaborations and discussions throughout these years. Special thanks to Pratyush Pandey, with whom I share many of the achievements presented in this thesis.

I also thank my parents for giving me an education and everything I ever needed to succeed. Last but not least, I thank my wife Antonia for selflessly joining me in this adventure and for her constant support.

This work was supported in part by the Center for Low Energy Systems Technology (LEAST), one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, by the National Science Foundations NSF-DMR-EPM under grant No. 1607935, by CONICYT-PCHA/Doctorado Nacional/2014-2114059, by CONICYT-FONDECYT project 1130334 and by the Notre Dame Center for Research Computing.

ABSTRACT

Deep neural networks (DNN) can perform cognitive tasks such as speech recognition and object detection with high accuracy. However, the computational cost to perform inference tasks with DNNs is a challenge for mobile applications, whereas the time and energy required to train DNN models can be prohibitive even at large data centers. The computational cost of Deep Neural Networks (DNN) is dominated by memory access and multiply accumulate operations. For this reason, it has been proposed to use resistive crossbar arrays to minimize data movement and perform efficient multiply accumulate operations. These architectures store the weight value in multilevel resistive memory elements, and perform matrix-vector multiplications in the analog domain. One of the main challenges of these architectures is the limited resolution and nonlinearity of resistive memories available today. In this thesis, this limitation is addressed in two ways: by developing a model to design and optimize multilevel memories based on ferroelectric materials, and by designing an architecture to mitigate the limitations of resistive crossbars for DNN training.

First, ferroelectrics are studied for multilevel memory devices in resistive crossbar arrays. Ferroelectrics are ceramic materials that can have two nonvolatile polarization states. In their polycrystalline form, these materials are composed of a multitude of grains with independent polarization states, allowing for dense, nonvolatile, multilevel memories compatible with standard semiconductor fabrication processes. However, modeling the dynamics of polycrystalline ferroelectrics is challenging due to the statistical variations in the composition of its grains. For this purpose, a model to extract the statistical properties of a ferroelectric film and a Monte Carlo simu-

lation that can describe and predict its polarization dynamics and variability were developed. This model provides the tools to characterize and optimize ferroelectric materials, and to design and evaluate devices, circuits and architectures for deep learning and other applications.

Secondly, architecture improvements to train DNN models in resistive crossbar arrays are presented. An accurate scheme for parallel weight update in resistive crossbar arrays is proposed and evaluated. By using pulse width- and frequency-modulated signals, the value of resistive elements in a crossbar array can be updated in parallel with higher accuracy than that of existing techniques based on stochastic multiplication. This scheme produces an unbiased multiplication with stochastic rounding, which is optimal for training neural networks with limited resolution. Finally, the mapping of DNN models to hardware with nonnegative weights is studied. To analyze different mapping schemes, a general vector-matrix multiplication is decomposed into a vector-matrix multiplication with nonnegative weight elements performed in a crossbar array, followed by a limited set of addition and subtraction operations described by a connection matrix. The mathematical conditions for the existence of such decomposition are derived and applied to fully connected and convolutional layers. Based on this analysis, an efficient mapping scheme is designed, which mitigates the effect of weight nonlinearity and limited resolution. These architectures are evaluated with low-level simulations of DNN training implemented in MATLAB and by extending the Keras open-source framework to incorporate nonideal weight elements and the connection matrix decomposition.

RESUMEN

Las redes neuronales profundas (DNN, por sus siglas en inglés) pueden realizar tareas cognitivas como el reconocimiento de voz y la detección de objetos con alta precisión. Sin embargo, el costo computacional para realizar tareas de inferencia con DNNs es un desafío para las aplicaciones móviles, mientras que el tiempo y la energía necesarios para entrenar los modelos pueden ser prohibitivos incluso en grandes centros de datos. El costo computacional de las redes neuronales profundas está dominado por multiplicaciones y acceso a memoria. Por esta razón, se ha propuesto utilizar matrices de elementos resistivos para minimizar el movimiento de datos y realizar multiplicaciones de manera eficiente en el dominio analógico. Uno de los principales desafíos de estas arquitecturas es la resolución limitada y la no linealidad de las memorias resistivas disponibles en la actualidad. En esta tesis, esta limitación se aborda de dos maneras: desarrollando un modelo para diseñar y optimizar memorias multiniveles basadas en materiales ferroeléctricos, y diseñando una arquitectura para mitigar las limitaciones de matrices resistivas para el entrenamiento de DNNs.

Primero, se estudian los dispositivos ferroeléctricos para implementar memorias multinivel. Los ferroeléctricos son materiales cerámicos que pueden tener dos estados de polarización no volátiles. En su forma policristalina, estos materiales se componen de una multitud de granos con estados de polarización independientes, lo que permite memorias densas, no volátiles y multinivel compatibles con los procesos estándar de fabricación de semiconductores. Sin embargo, modelar la dinámica de los ferroeléctricos policristalinos es un desafío debido a las variaciones estadísticas en la composición de sus granos. Para este propósito, se desarrolló un modelo para extraer

las propiedades estadísticas de un ferroeléctrico y una simulación de Monte Carlo que puede describir y predecir su dinámica de polarizacin y variabilidad. Este modelo proporciona las herramientas para caracterizar y optimizar materiales ferroeléctricos, y para diseñar y evaluar dispositivos, circuitos y arquitecturas para redes neuronales y otras aplicaciones.

En segundo lugar, se presentan mejoras en la arquitectura para entrenar modelos de redes neuronales en matrices resistivas. Se propone y evalúa un esquema preciso para la actualización de pesos en paralelo en una matriz resistiva. Al utilizar señales de ancho de pulso y modulación en frecuencia, el valor de los elementos resistivos puede actualizarse en paralelo con mayor precisión que las técnicas existentes basadas en la multiplicación estocástica. Este esquema produce una multiplicación con redondeo estocástico, que es óptimo para entrenar redes neuronales con resolución limitada. Finalmente, se estudia el mapeo de modelos de redes neuronales a hardware con pesos no negativos. Para analizar diferentes esquemas de mapeo, una multiplicación general de matrices se descompone en una multiplicacin de matrices con elementos no negativos realizados en una matriz resistiva, seguida de un conjunto limitado de operaciones de suma y resta descritas por una matriz de conexiones. Las condiciones matemáticas para la existencia de esta descomposicin se derivan y aplican a modelos de redes neuronales. Sobre la base de este análisis, se diseña un esquema de mapeo eficiente, que mitiga el efecto de la no linealidad y la resolución limitada de los elementos resistivos. Estas arquitecturas se evalúan con simulaciones implementadas en MATLAB y mediante la extensión del software de código abierto Keras para incorporar elementos de peso no ideal y la descomposición de la matriz de conexiones.

CONTENTS

Acknowledgments ii

Figures x

Tables xiii

Chapter 1: Introduction 1
1.1 Deep neural networks 1
1.2 Computational cost of DNN inference 7
1.3 Training a DNN 10
1.4 Hardware accelerators for DNNs 14
1.5 Resistive crossbar accelerators 17
1.6 Challenges to implement multilevel memory devices for training . . . 20
1.7 Objectives and hypothesis 21
1.8 Organization of this thesis 22

Chapter 2: Multilevel ferroelectric memory for resistive crossbar arrays 24
2.1 Ferroelectric polarization, hysteresis loops and partial polarization . . 24
2.2 Characterization of partial polarization of ferroelectric capacitors . . 27
2.3 Performance simulation in crossbar-based DNN accelerator 29
2.4 Polarization-based analog memory for resistive crossbar arrays 31
2.5 Conclusion 31

Chapter 3: Characterization and modeling of ferroelectric polarization reversal 37
3.1 Polarization reversal in a ferroelectric crystal 37
3.2 Nucleation-limited switching in polycrystalline ferroelectrics 39
3.3 Experimental results and parameter extraction 42
3.4 Study of thickness dependence in ferroelectric HZO capacitors 47
3.5 Conclusion 49

Chapter 4: Monte Carlo simulation of polarization dynamics in polycrystalline ferroelectrics 51
4.1 Revisit NLS model and nucleation rate 52
4.2 Monte Carlo simulation of polarization reversal 55
4.3 Monte Carlo simulation for arbitrary input waveforms 57

4.4 Accumulation and relaxation of the history-dependent switching rate 61
4.5 Model predictions . . . 64
4.6 Conclusion . . . 66

Chapter 5: Parallel weight update in resistive crossbar arrays by local modulation of pulse width and frequency . . . 68
5.1 Analysis of stochastic multiplication . . . 69
5.2 Multiplication by pulse width and frequency modulation . . . 71
5.3 Comparison of stochastic and rate-width multiplication . . . 73
5.3.1 Hardware resources . . . 73
5.3.2 Multiplication accuracy . . . 74
5.4 Performance evaluation in a resistive crossbar array . . . 75
5.5 Conclusion . . . 77

Chapter 6: Efficient mapping of neural network models to resistive crossbar arrays with limited weight resolution . . . 78
6.1 Prior approaches to map DNN model to resistive crossbar arrays . . . 78
6.2 Connection matrix decomposition for fully-connected layers in a resistive crossbar array . . . 81
6.2.1 Sufficient conditions for existence . . . 82
6.2.2 Implementation in a crossbar array . . . 84
6.3 Connection Matrix decomposition applied to convolutional layers . . . 87
6.4 Experimental validation with MNIST and CIFAR-10 datasets . . . 88
6.5 Evaluation with limited weight resolution and nonlinearity . . . 95
6.6 Conclusion . . . 99

Chapter 7: Conclusion . . . 100

Bibliography . . . 104

Appendix A: List of publications and patents . . . 118

Appendix B: US patent application number 16180453 . . . 120

Appendix C: Counter-based neural network architecture with rate-width multiplication . . . 149
C.1 Statistical analysis . . . 151
C.1.1 Stochastic pulse multiplication . . . 151
C.1.2 Rate-width multiplication . . . 153
C.2 Signed multiplication with dynamic fixed point precision . . . 154
C.3 Experimental evaluation . . . 156
C.4 Conclusion . . . 159

Appendix D: Reconfigurable electric double layer doping in an MoS_2 nanoribbon transistor 161

Appendix E: Optimal CCD readout by digital correlated double sampling . . . 168

FIGURES

1.1 Diagram of a neuron: the basic computational unit of a neural network 2
1.2 Logistic regression implemented with a single neuron with binary threshold activation function 3
1.3 Neural network formed by a sequence of two layers of neurons 4
1.4 Commonly used activation functions 5
1.5 Example 4-layer fully connected DNN 6
1.6 Exponential increase in DNN parameter number over time 9
1.7 Diagram of MAC operation and memory access 9
1.8 Temporal and spatial architecture paradigms for highly parallel operations 16
1.9 Memory hierarchy and MAC array in a DNN accelerator 16
1.10 Crossbar implementation of matrix-vector multiplication for acceleration of DNNs 17
1.11 Resistive processing unit operating principle 19

2.1 Ferroelectric permanent dipole moment in a noncentrosymmetric perovskite unit cell 24
2.2 Ferroelectric polarization and characteristic $P - V$ loop 26
2.3 Partial polarization in a polycrystalline ferroelectric film 26
2.4 Cross section of PZT and HZO ferroelectric capacitors 28
2.5 PZT and HZO polarization characteristic after wake up 28
2.6 Measurement protocol for ferroelectric partial polarization 33
2.7 Partial polarization measurements for PZT and HZO capacitors . . . 34
2.8 Baseline DNN training results 35
2.9 Weight update characteristic for ferroelectric and linear memory element 35
2.10 Simulated test error after training with ferroelectric memory element 36
2.11 Proposed RPU based on the partial polarization of a ferroelectric . . 36

3.1 Polarization reversal in a single ferroelectric crystal 38
3.2 Polarization reversal in a polycrystalline ferroelectric crystal 40

3.3 Cross section, $P-V$ loops and capacitance measurements of ferroelectric HZO sample . . . 43
3.4 Measurement protocol for polarization reversal . . . 44
3.5 Polarization reversal and fitted NLS model . . . 45
3.6 Extraction of the distribution of local field variations in MFM capacitor 46
3.7 Thickness dependence of polarization reversal in HZO capacitor . . . 48

4.1 Monte Carlo simulation of polarization reversal vs. NLS model . . . 56
4.2 Monte Carlo simulation of P–V loops vs. measurements . . . 59
4.3 Monte Carlo simulation of major and minor loops vs. measurements . 61
4.4 Monte Carlo simulation of pulsed polarization . . . 63
4.5 Comparison of single-grain and polycrystalline polarization dynamics 64
4.6 Predicted variability in small-area ferroelectric capacitors . . . 65

5.1 Stochastic multiplication . . . 70
5.2 Rate-Width multiplication . . . 71
5.3 Phase difference between the frequency- and width-modulated signals. 72
5.4 Hardware implementations of stochastic translators, width and frequency modulation. . . . 74
5.5 Comparison between rate-width and stochastic multiplication. . . . 75
5.6 DNN training with stochastic and rate-width multiplication. . . . 76

6.1 Prior approaches to map DNN model to crossbar array . . . 79
6.2 Diagram of connection matrix decomposition . . . 81
6.3 Diagram of connection matrix representation of prior approaches . . . 85
6.4 Crossbar implementation of double element, bias column and adjacent CM . . . 86
6.5 Convolutional layer implemented as matrix multiplications . . . 87
6.6 Connetion matrix decomposition applied to a convolutional layer . . . 89
6.7 Keras implementation of connection matrix . . . 91
6.8 Fully-connected network for MNIST dataset trained with different approaches . . . 93
6.9 Convolutional network for MNIST dataset trained with different approaches . . . 93
6.10 Convolutional network for CIFAR-10 dataset trained with different approaches . . . 94
6.11 Classification error of fully-connected network with quantized and nonlienar weights . . . 97

6.12 Classification error of convolutional network with quantized and nonlienar weights 98

C.1 Basic structure of multiplication by stochastic pulses and rate-width multiplication 150

C.2 Structure for vector-scalar multiplication and matrix-vector MAC. . . 151

C.3 Structure for vector outer product and matrix multiplication. 152

C.4 Phase difference between the frequency- and width-modulated signals. 153

C.5 Block diagram for signed multiplication with dynamic fixed point precision 155

C.6 Test error for 10-fold cross validation and different multiplication methods 159

TABLES

1.1 Six DNN applications that represent 95% of Google's TPU workload 8
1.2 Resistive processing unit requirements 20
4.1 HZO parameters extracted from polarization reversal measurements. . 54
6.1 Structure of DNNs used for validation 90
C.1 Parameters for baseline and reduced-precision DNN models. 158

CHAPTER 1

INTRODUCTION

The time and energy required to train a Deep Neural Network (DNN) could be dramatically reduced by architectures based on resistive crossbar arrays, which store the weight value in multilevel resistive memory elements and perform matrix-vector multiplications in the analog domain. One of the main challenges of these architectures is the limited resolution of resistive memories available today, as well as their asymmetric and nonlinear response to programming pulses applied during training.

In this thesis, these challenges are addressed from a device perspective by identifying, characterizing and modeling the potential of ferroelectrics for multilevel memory storage. From an application perspective, the impact of device nonidealities on the overall training performance is analyzed, and architectural approaches to mitigate their impact are proposed.

This chapter introduces basic concepts of DNNs that will be used throughout this thesis, and recent trends in the computational cost of DNNs are presented to motivate the need for application-specific hardware designed to efficiently implement DNNs. The use of resistive crossbar arrays to accelerate the training of DNN and their challenges are presented to motivate this thesis work.

1.1 Deep neural networks

Deep neural networks can perform cognitive tasks such as speech recognition and object detection with high accuracy [1]. Although neural networks have been

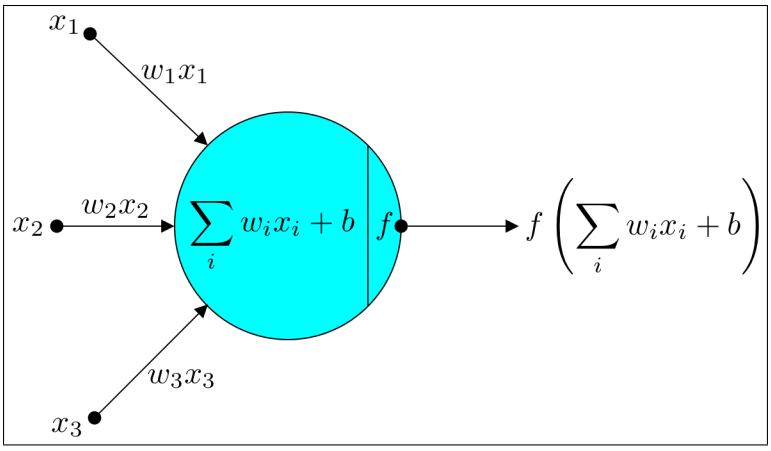

Figure 1.1. Diagram of a neuron: the basic computational unit of a neural network. The neuron computes a weighted sum of its inputs x_i and a bias b. The output is then computed by applying an activation function $f(z)$. Figure adapted from [4].

known for decades, their dramatic success in the last decade has been driven by the use of the backpropagation algorithm [2], the growth of computing power and the availability of large datasets [3]. These factors have made possible to implement and train large DNNs that outperform the state-of-the-art algorithms in tasks such as image recognition and natural language processing.

The basic computational unit in a neural network is the artificial neuron, shown in Fig. 1.1. An artificial neuron typically has multiple inputs and a single output. The neuron performs a weighted sum of its inputs, where w_i is the weight given to input x_i. A bias parameter b is added to the weighted sum, resulting in a scalar value given by

$$z = b + \sum_i w_i x_i. \tag{1.1}$$

The weights represent the connection strength given to each input, and are called synapses or synaptic weights in reference to biological neurons. The neuron then applies a transformation $f(z)$, called an activation function, to compute its output

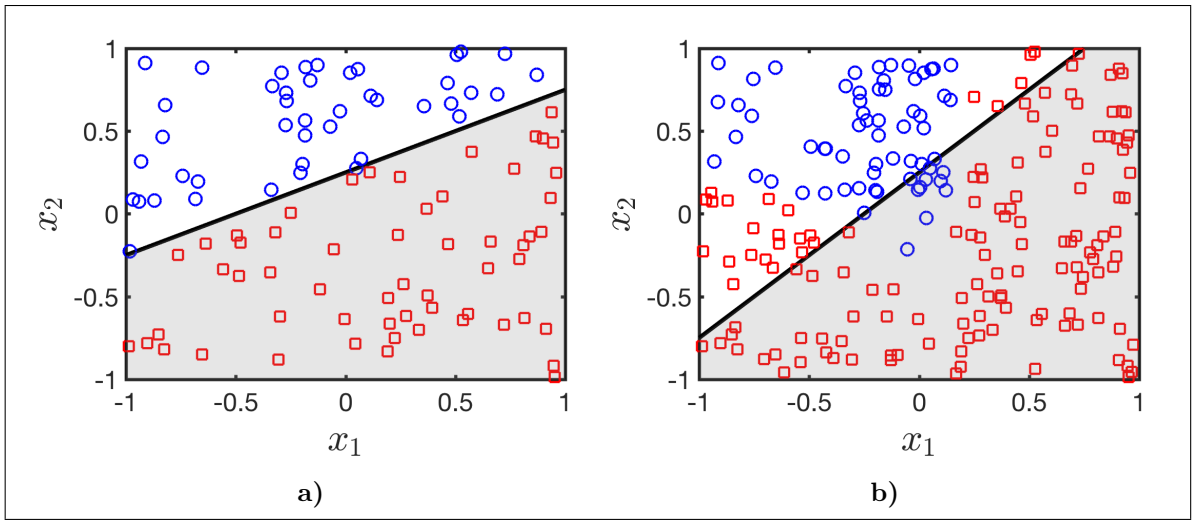

Figure 1.2. Logistic regression implemented with a single neuron with binary threshold activation function. Two inputs x_1 and x_2 define a plane, which is divided by the threshold $w_1x_1 + w_2x_2 + b = 0$. Colored symbols represent two data classes. (a) Linearly separable data can be classified correctly with a single neuron. (b) Data that is not linearly separable cannot be classified by a single neuron. Example adapted from [3]

as

$$y = f\left(b + \sum_i w_i x_i\right). \tag{1.2}$$

The activation function is typically a nonlinear function. The choice of the activation function and the length of the input vector are structural parameters, called hyperparameters to distinguish them from the weights and biases, which are tunable parameters.

To understand the representational capability of a neuron, consider a simple example with inputs x_1 and x_2, and a binary threshold activation function, defined as

$$f(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0 \end{cases} \tag{1.3}$$

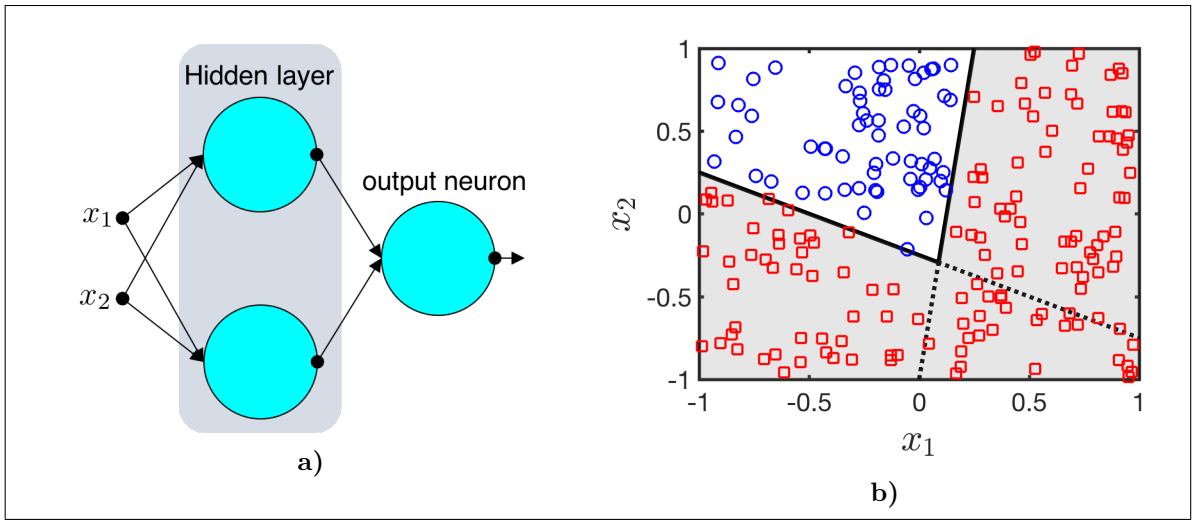

Figure 1.3. Neural network formed by a sequence of two layers of neurons. (a) The inputs x_1 and x_2 are fed to an intermediate layer of 2 neurons (hidden layer). The output of the hidden layer is fed to the output neuron, which performs the classification. (b) Boundary obtained with the neural network, which separates data with a nonlinear boundary. Example adapted from [3]

The plane defined by x_1 and x_2 is divided into 2 regions by the line $w_1x_1+w_2x_2+b=0$. The parameters w_1, w_2 and b determine the position of the threshold. With this simple model, the neuron can be used to perform a logistic regression. Consider the example shown in Fig. 1.2a, where 2 classes of data denoted by blue circles and red squares are distributed in the plane. Suppose we assign the label "1" to class circle, and the label "0" to class square. With the right set of parameters, the threshold defined by $z=0$ can effectively separate the 2 classes. The neuron can then perform an *inference* for an arbitrary input (x_1, x_2), for which it assigns either a 1 or a 0 label. The process of finding the parameters w_1, w_2, b that provide a good representation of the data is called *learning* or *training*.

Given that the argument of the activation function is a linear combination of the input vector, a single neuron can only classify data that are linearly separable [5]. Consider the case shown in Fig. 1.2b, which has a nonlinear boundary. There is no

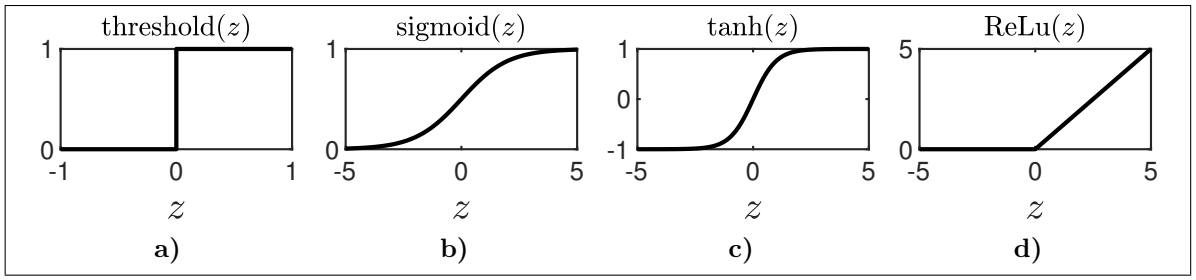

Figure 1.4. Commonly used activation functions: a) binary threshold, b) Sigmoid logistic function, c) hyperbolic tangent and d) Rectified linear unit.

set of parameters w_1, w_2, b that can correctly separate these data.

This limitation can be overcome by stacking multiple layers of neurons, as shown in Fig. 1.3a. The input features (x_1 and x_2) are fed to an intermediate layer of neurons, referred to as a hidden layer. The output of the hidden layer is fed to an output neuron that performs the classification. By tuning the parameters in the hidden layer, we can obtain nonlinear transformations of the input features that enable the output neuron to separate the data. Note that the activation function in the hidden layer needs to be nonlinear. Otherwise, the sequence of linear transformations performed at each layer would be reduced to a single linear transformation of the form in Eq. (1.1).

Figure 1.4 shows commonly used activation functions. The sigmoid function and the hyperbolic tangent are commonly used for classification in the output neuron. These functions have a continuous transition between two saturated values, as opposed to the threshold function we have been using so far. Sigmoid and hyperbolic tangent activations are also used for hidden layers, although the rectifier linear unit (ReLu) has gained popularity over the past years. The activation functions in Fig. 1.4 are scalar functions computed independently for each neuron in a layer. Other activation functions compute a vector operation of the neurons in a layer. For example, a softmax activation is similar to a sigmoid function with the output normalized to represent a probability distribution with multiple classes [6].

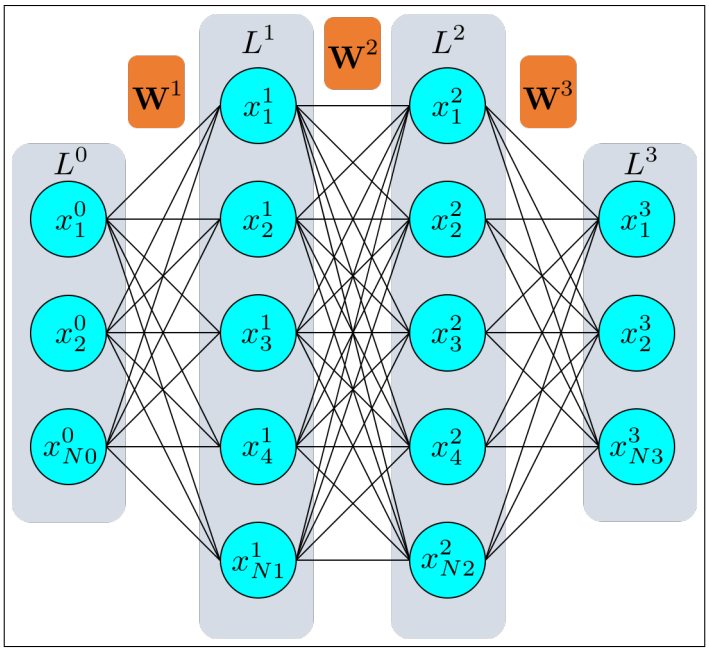

Figure 1.5. 4-layer DNN comprised of input layer with $N0$ neurons (circles), two hidden layers with $N1$ and $N2$ neurons and an output layer with $N3$ neurons. Bias units are not shown for simplicity. Layer numbers are indicated by superscript, whereas neuron rows are indicated by subscript. Weight matrices $\mathbf{W}^1$, $\mathbf{W}^2$, and $\mathbf{W}^3$ contain the weights that connect the respective layers.

Increasing the number of hidden layers allows the network to represent increasingly complex nonlinear functions, and this is one of the key principles of DNNs. Figure 1.5 shows a DNN example with 4 layers of neurons (L^i). The input layer L^0 represents the input vector, which is connected to the first hidden layer L^1 by a weight matrix $\mathbf{W}^1$. Each row j of $\mathbf{W}^1$ stores the weights that connect the input vector in L^0 to neuron j in L^1. Therefore, if layer 0 has N_0 neurons, and layer 1 has N_1 neurons, the weight matrix $\mathbf{W}^1$ has dimension $N_1 \times N_0$. Likewise, L^1 is connected to L^2 through the weights $\mathbf{W}^2$, and L^2 is connected to the output layer L^3 through $\mathbf{W}^3$. Each neuron in the output layer corresponds to one of the available classes, and the label assigned by the network corresponds to that of the neuron with the highest output. The DNN in Fig. 1.5 is one of the basic topologies used today, where all the

neurons in one layer are connected to all the neurons in the following layer, so it is called a fully-connected DNN. There are neither connections between neurons in the same layer, nor connections between neurons in nonconsecutive layers. The number of layers in the neural network is referred to as length or depth, whereas the number of neurons in a layer is referred to as the width of that layer.

More complex DNNs can have different topologies such as convolutional neural networks and recurrent neural networks, which are not introduced here. For an introduction to DNNs, the reader is referred to [7], whereas an in-depth review of deep learning can be found in [6].

1.2 Computational cost of DNN inference

The basic operation required for DNNs is the multiply accumulate (MAC) operation to compute the weighted sum in a neuron. Depending on the specific DNN topology, the MAC operation can be part of a matrix-vector multiplication, matrix-matrix multiplication or a convolution. Large DNN models have a large computational complexity due to the number of weights that need to be stored and MAC operations that are computed. Table 1.1 shows six DNN applications that represent 95% of the workload in Google's data centers running tensor processing units (TPU), a custom co-processor to accelerate inference in DNNs [8]. These DNN have between 5 million and 100 million parameters (weights and biases), which are determined by the number of layers, the type of layers and their width. The number of MAC operations can be roughly estimated by the number of operations performed for each weight byte, and exceeds a billion operations in most cases.

Furthermore, DNNs computational complexity and memory consumption has been growing exponentially [9]. Figure 1.6 shows the exponential increase in the number of parameters in popular DNN models for image recognition over the past 2 decades [9]. A similar trend was observed between the accuracy of different DNN

TABLE 1.1

SIX DNN APPLICATIONS THAT REPRESENT 95% OF GOOGLE'S TPU WORKLOAD

DNN name	Number of layers by type					Weights (millions)	Ops/ weight byte
	FC	Conv	Vector	Pool	Total		
MLP0	5				5	20	200
MLP1	4				4	5	168
LSTM0	24		34		58	52	64
LSTM1	37		19		56	34	96
CNN0		16			16	8	2888
CNN1	4	72		13	89	100	1750

Layers are fully connected (FC), convolutional (Conv), vector operations (Vector), and Pooling (Pool), which does nonlinear downsizing. The last two columns show the number of weights, and number of operations performed for each weight byte [8].

models and their size (number of parameters and operations) [9].

The energy cost of DNNs is directly related to the number of operations performed by the network, both due to the MAC operations and memory access. As depicted in Fig. 1.7, a MAC operation comprises one multiplication and one addition, and requires 3 inputs: the activation from a previous layer (x_i^{L-1}) and the filter weight (w_{ji}^{L}), which are the factors, and the partial sum that will be updated with the product. The updated sum is then written back to memory. Therefore, if no data is locally stored for reuse, every MAC operation would require 3 memory reads and 1 memory write, which severely impacts the throughput and the energy efficiency [10].

Over the past decade, the growth of DNNs has been largely driven by the computational power provided by graphics processing units (GPU) [1], which perform highly parallel MAC operations. However, this has also led to a dramatic increase in

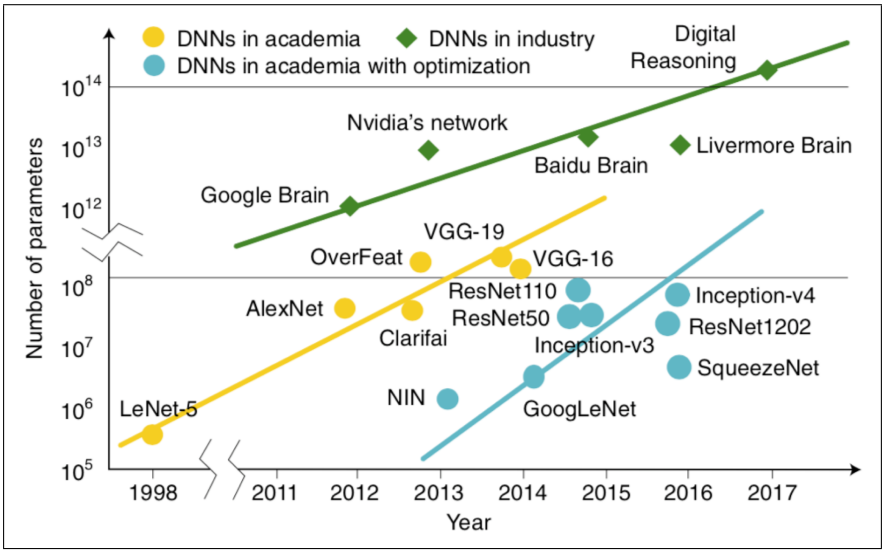

Figure 1.6. Exponential increase in DNN parameter number over time. DNNs with optimization are those designed for computational efficiency. Image reprinted from [9].

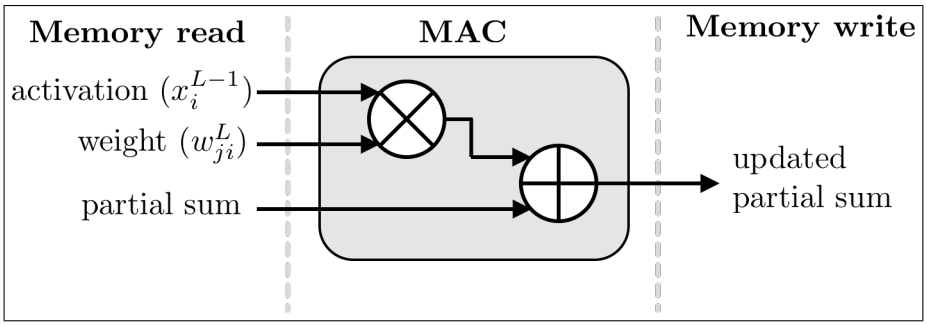

Figure 1.7. MAC operation comprises one multiplication and one addition, and reads 3 data from memory: the activation from a previous layer (x_i^{L-1}), the filter weight (w_{ji}^L), and the partial sum that will be updated with the product. The updated sum is then written back to memory. Figure adapted from [10].

their power consumption [11]. Given that memory access dominates the energy consumption, architectures with memory hierarchies that optimize data reuse for DNN requirements can achieve a better energy efficiency [10], and major improvements in cost-energy-performance must now come from domain-specific hardware [8].

1.3 Training a DNN

Training a DNN model is much more expensive than inference in terms of time and energy. In general, the training is performed by exposing the network to a large amount of labeled data. The DNN performs inferences on these data, and its predictions are compared against the actual labels to compute a measure of error, called a loss function. The parameters of the DNN are typically updated in an iterative manner using gradient descent. The gradients of the loss function with respect to each of the parameters (weights and biases) are computed through the backpropagation algorithm [2]. Although there can be some variations in the specific training algorithm depending on the DNN topology, it can be described in general by three steps: forward propagation, backpropagation and weight update. These steps will be explained in detail for the fully-connected DNN of Fig. 1.5. For other network topologies, the reader is referred to [6, 7, 12].

Consider a set of N labeled training examples $\{\mathbf{x}, \mathbf{y}\}_k,\ k = 1..N$, where $\mathbf{x}$ is the input vector and $\mathbf{y}$ is a vector containing the example's label. The length of the vector of labels $\mathbf{y}$ corresponds to the number of classes and is encoded as a one-hot vector. That is, a vector with a single "1" indicates the correct class, whereas all other elements are "0".

Forward propagation: this step is equivalent to performing an inference task, as was depicted in Section 1.1. The input vector $\mathbf{x}$ is applied at the input layer L^0, and each of the neurons in layer L^1 computes its weighted sum and applies the nonlinear activation function. The output of a neuron, its activation, is then input

to the following layer. This process is repeated until the output layer is reached. The neuron activation from Eq. (1.2) can be written in general as

$$x_j^L = f\left(b_j^L + \sum_{i=1}^{N_{L-1}} x_i^{L-1} w_{ji}^L\right) \tag{1.4}$$

where w_{ji}^L are the elements of the weight matrix $\mathbf{W}^L$, b_j^L are bias parameters and $f(z)$ is the nonlinear activation function. The initialization and range for these parameters will be discussed in Chapter 6. Expressed in vector notation,

$$\mathbf{x}^L = f\left(\mathbf{b}^L + \mathbf{W}^L \mathbf{x}^{L-1}\right) \tag{1.5}$$

Each of the output neurons represents one of the available classes, so the output of the DNN is a vector $\hat{\mathbf{y}}$ with the activations of these neurons. For example, if a sigmoid activation function is used in the output layer, each of the output neurons will have an activation ranging from 0 to 1, which represents the likelihood that the input corresponds to each of the classes. This vector will be used to compute a measure of error, as opposed to inference where a label would be assigned to the neuron with the highest activation.

Backpropagation: the loss function is computed as a measure of distance between the output vector $\hat{\mathbf{y}}$ and the correct label $\mathbf{y}$. The details of the choice of loss function are beyond the scope of this short introduction, and the reader is referred to [7] for details. With an appropriate choice of loss function, the gradients with respect to the activations of the output layer, $\boldsymbol{\delta}^{Lout}$, can be computed as [7]

$$\boldsymbol{\delta}^{Lout} = \hat{\mathbf{y}} - \mathbf{y}. \tag{1.6}$$

According to the backpropagation algorithm [2], the error in a layer L, $\boldsymbol{\delta}^L$, can be

computed in terms of the error in the following layer, $\boldsymbol{\delta}^{L+1}$, as

$$\boldsymbol{\delta}^{L} = \left\{ \left(\mathbf{W}^{L+1}\right)^{T} \boldsymbol{\delta}^{L+1} \right\} \odot f'(\mathbf{z}^{L}), \tag{1.7}$$

where $\left(\mathbf{W}^{L+1}\right)^{T}$ is the transpose of the weight matrix $\mathbf{W}^{L+1}$, $f'(\mathbf{z}^{L})$ is the derivative of the activation function in layer L, and $\odot$ is the elementwise vector multiplication (Hadamard product). Although this equation may seem complicated, note the similarity with Eq. (1.5). The term $\left(\mathbf{W}^{L+1}\right)^{T} \boldsymbol{\delta}^{L+1}$ is simply the weighted sum of the error in layer $L+1$ measured from layer L, and is computed in the same way as the forward propagation. Then, each neuron multiplies its weighted sum by the derivative of its activation function, in a similar fashion in which the activation function is applied during forward propagation.

Weight update: Once the backpropagation is completed, each layer of neurons has a vector of activations from the forward propagation $\mathbf{x}^{L}$ and a vector of backpropagated error $\boldsymbol{\delta}^{L}$. The weights and biases are updated according to the rule [6, 7]

$$w_{ji}^{L} \leftarrow w_{ji}^{L} - \eta \delta_{j}^{L} x_{i}^{L-1} \tag{1.8}$$

$$b_{j}^{L} \leftarrow b_{j}^{L} - \eta \delta_{j}^{L} \tag{1.9}$$

where w_{ji}^{L} is the weight that connects the neuron i in layer $L-1$ (preneuron) to the neuron j in layer L (postneuron), x_{i}^{L-1} is the preneuron activation, δ_{j}^{L} is the backpropagated error fed by the postneuron, and η is the learning rate. The learning rate is a global parameter that is typically adjusted by experimentation [6]. The update can be expressed in vector notation as

$$\mathbf{W}^{L} \leftarrow \mathbf{W}^{L} - \eta \boldsymbol{\delta}^{L} \otimes \mathbf{x}^{L-1} \tag{1.10}$$

$$\mathbf{b}^{L} \leftarrow \mathbf{b}^{L} - \eta \boldsymbol{\delta}^{L} \tag{1.11}$$

where $\otimes$ is the vector outer product.

The weight update is usually performed by averaging Eq. (1.8) from several examples. The term gradient descent is typically used when the weights are updated by averaging over all the training examples. When the weights are updated with a smaller number of examples, it is called stochastic gradient descent and the number of examples is referred to as the batch size. The extreme case where the weights are updated after every example is sometimes referred to as online learning. The training is performed by iterating several times over the full set of training examples, and each iteration is called a training epoch or simply an epoch.

Instead of computing one training example at a time, a fully-connected DNN can operate in batch mode by applying a matrix $\mathbf{X}$ with M columns of input vectors. Likewise, the vectors $\mathbf{z}$, $\mathbf{y}$ and $\boldsymbol{\delta}$ are converted into matrices with M columns $\mathbf{Z}$, $\mathbf{Y}$ and $\boldsymbol{\Delta}$. Equations (1.5), (1.6), (1.7), (1.10) and (1.11) are modified as:

$$\mathbf{X}^L = f\left(\mathbf{b}^L + \mathbf{W}^L \mathbf{X}^{L-1}\right) \tag{1.12}$$

$$\boldsymbol{\Delta}^{Lout} = \hat{\mathbf{Y}} - \mathbf{Y} \tag{1.13}$$

$$\boldsymbol{\Delta}^L = \left\{\left(\mathbf{W}^{L+1}\right)^T \boldsymbol{\Delta}^{L+1}\right\} \odot f'(\mathbf{Z}^L) \tag{1.14}$$

$$\mathbf{W}^L \leftarrow \mathbf{W}^L - \frac{\eta}{M} \boldsymbol{\Delta}^L \left(\mathbf{X}^{L-1}\right)^T \tag{1.15}$$

$$\mathbf{b}^L \leftarrow \mathbf{b}^L - \frac{\eta}{M} \sum_{rows} \boldsymbol{\Delta}^L. \tag{1.16}$$

The bias parameters remain as columns vectors, so it is assumed that the addition of a column vector to a matrix is performed columnwise. The sum in Eq. (1.16) represents a summation within the rows of $\boldsymbol{\Delta}^L$, which yields a column vector of the same dimensions as b_L.

Given that in general there is no ambiguity, the superscripts will be omitted in

the following chapters and the weight update rule will simply be expressed as:

$$w_{ji} \leftarrow w_{ji} - \eta x_i \delta_j. \tag{1.17}$$

1.4 Hardware accelerators for DNNs

In this section, a review of hardware accelerators for DNN is presented, highlighting the different architectural approaches and their main constraints. A more comprehensive review of DNN accelerators can be found in [3, 10, 13]. The accelerators presented here are designed and implemented with standard CMOS technology. The special case of hardware accelerators that leverages multilevel resistive memories to perform computation in the analog domain is presented in the following section.

The workload of DNN is dominated by MAC operations, which are executed within matrix multiplications or convolutions. These operations are repeated for every inference task, while only the inputs of the network change. For training, the same operations are repeated iteratively for several cycles. These operations govern all control flow and memory access patterns, and can be statically scheduled for efficient memory access and maximal parallelism [3, 13]. As a point of comparison, the memory hierarchy for general purpose computing is optimized to exploit spatial and temporal locality of data with different levels of cache memory for on-chip storage [14]. However, the memory access patterns can vary widely for different applications, and only a limited fraction of the memory access can be predicted during compilation. Therefore, the optimization relies on dynamic speculation techniques that are optimized to achieve a good overall performance for benchmarks with a wide variety of workloads [14]. On the other hand, the static scheduling of DNNs and the limited number of different operations they require offers a great potential for hardware accelerators [3]. Furthermore, DNN are robust to reduced-precision operations and approximations [15, 16], which allows for further optimization and trade-offs

between MAC resolution and DNN accuracy [10, 13].

In general, two paths to improve DNN performance can be identified: parallel, dense and low-energy MAC hardware; and an efficient memory hierarchy to minimize access to main memory.

As explained in [10], two architecture paradigms for highly parallel operations have been explored. Temporal architectures exploit vector operations with a centralized control unit and many arithmetic logic units (ALU), as depicted in Fig. 1.8a. This is the case of central processing units (CPUs) and GPUs. The data is always fetched directly from the memory hierarchy, with no internal data communication between the ALUs. In spatial architectures, also called systolic arrays, depicted in Fig. 1.8b, the ALUs form a processing chain so that they can pass data from one to another directly. In addition, the ALUs can have a local memory and control logic. Two representative implementations of this architecture can be found in [8, 17].

The memory hierarchy for a DNN accelerator is depicted in Fig. 1.9, showing representative values for storage density and the energy required to fetch data at different levels of the hierarchy [13]. The extent to which data can be reused at local hierarchy levels depends on the number of parameters in a model and the amount of intermediate data that is generated (partial sums, hidden layer activations, etc), and is ultimately limited by the size of the on-chip buffers. The size of local buffers is mainly limited by the chip area and the space occupied by the MAC array. For example, 24% of the die area in Google's TPU [8] is occupied by a systolic MAC array, 29% is used for a unified buffer for local activations and 6 % is used to accumulate partial sums. The remaining area is mainly occupied by input/output interfaces (including 2 ports for off-chip memory access), whereas the control logic only uses 2% of the overall die area. Increasing the on-chip memory size reduces the main memory access by leveraging data reuse. On the other hand, increasing the density of MAC units results in a higher throughput and potentially more data reuse, as

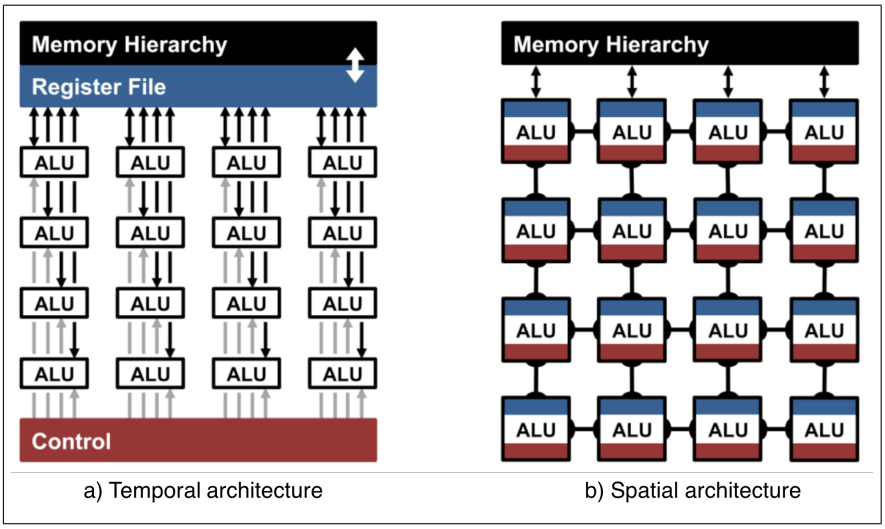

Figure 1.8. Architecture paradigms for highly parallel operations. a) Temporal architecture with a centralized control unit and many arithmetic logic units (ALU). Black arrows indicate that the register communicates directly with each ALU, without internal communication between ALUs. b) Spatial architecture consisting of an array of ALUs with local memory and control logic that can pass data from one to another directly. Figure reprinted from [10].

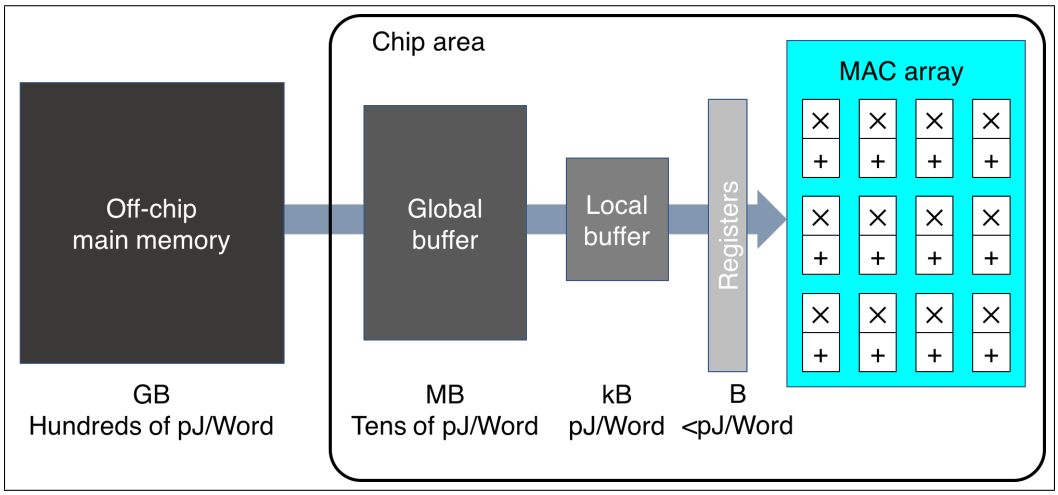

Figure 1.9. Memory hierarchy and MAC array in a DNN accelerator. The chip area is mainly occupied by the MAC array and different levels of local buffers in the memory hierarchy. Smaller buffers offer faster access time and lower energy, whereas off-chip memory access can be orders of magnitude more expensive in terms of time and energy. Figure adapted from [13].

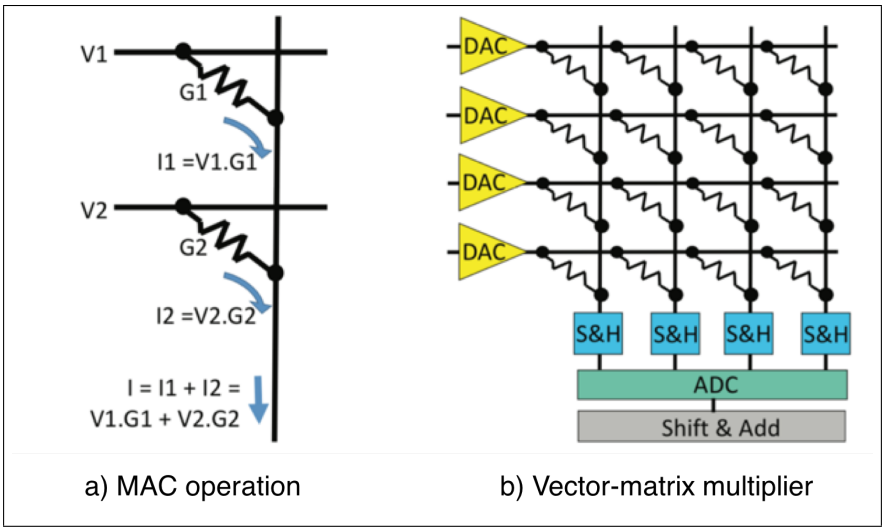

Figure 1.10. Crossbar implementation of matrix-vector multiplication for acceleration of DNNs. a) Input voltages are weighted by the conductance and integrated as a current to perform the equivalent of a weighted sum. b) Crossbar structure for matrix-vector multiplication. Additional circuits required are digital to analog converters (DAC), sample and hold (S&H), analog to digital converters (ADC), shift registers and addition. Figure reprinted from [18].

more operations can be performed simultaneously for a given weight or activation.

1.5 Resistive crossbar accelerators

Hardware accelerators based on resistive crossbar arrays have been proposed to implement efficient matrix-vector multiplication and minimize the movement of weights, projecting orders of magnitude of improvement in time and energy consumption over digital implementations [18–23]. The basic principle of resistive crossbar accelerators is shown in Fig. 1.10. A matrix-vector multiplication is computed by generating voltage signals from the rows of a crossbar array with resistive devices. The voltage inputs are weighted by the conductance and integrated as a current to perform the equivalent of a weighted sum. This implementation requires digital to analog converters (DAC), sample and hold circuits (S&H) and analog to digital

converters (ADC) to operate with analog signals. With this approach, a multilevel memory element is programmed with the weights of a DNN, and the crossbar structure can be employed to perform inference.

Additionally, it was shown that resistive crossbar arrays can be used to accelerate DNN training [20–22, 24, 25]. Gokmen and Vlasov [21] proposed the concept of resistive processing unit (RPU), a multilevel resistive element that can be programmed by voltage pulses. To understand the motivation for an RPU and its requirements, consider the schematic representation of the weight update rule in Eq. (1.8) applied in a crossbar array as shown in Fig 1.11a. Each weight in a crossbar structure receives an activation from the rows of the array (x_i), and a backpropagated error from the columns (δ_j), which are multiplied to update the weight value. The multiplication and weight update are traditionally computed in multipliers external to the crossbar array and then written to the corresponding resistive element. To perform the weight update locally, the inputs x_i and δ_j are translated into stochastic bit streams to perform a stochastic multiplication [21, 26], as shown in Fig. 1.11b). When there is a pulse coincidence, the weight is updated by a nominal value Δw, resulting in the update rule

$$w_{ji} \leftarrow w_{ji} \pm \Delta w \sum_{n=1}^{BL} A_i^n \wedge B_j^n, \tag{1.18}$$

where BL is the length of the stochastic bit stream, A_i^n and B_j^n are the values of the n-th bits, and $\wedge$ is the logic AND operation. A_i^n is asserted with probability $C_A x_i$, whereas B_j^n is asserted with probability $C_B \delta_j$. It can be shown that the stochastic multiplication produces, on average, the same result as direct multiplication [21].

To implement the weight update with an RPU, pulses of equal amplitude ($V_S/2$) and opposite polarity are applied at rows and columns. A pulse coincidence results in a V_S voltage at the device terminals, whereas a single pulse arrival only produces $V_S/2$, as shown in Fig. 1.11c. The RPU needs to have a nonlinear response to the applied voltages, so it changes its conductance for a coincidence of pulses and remains

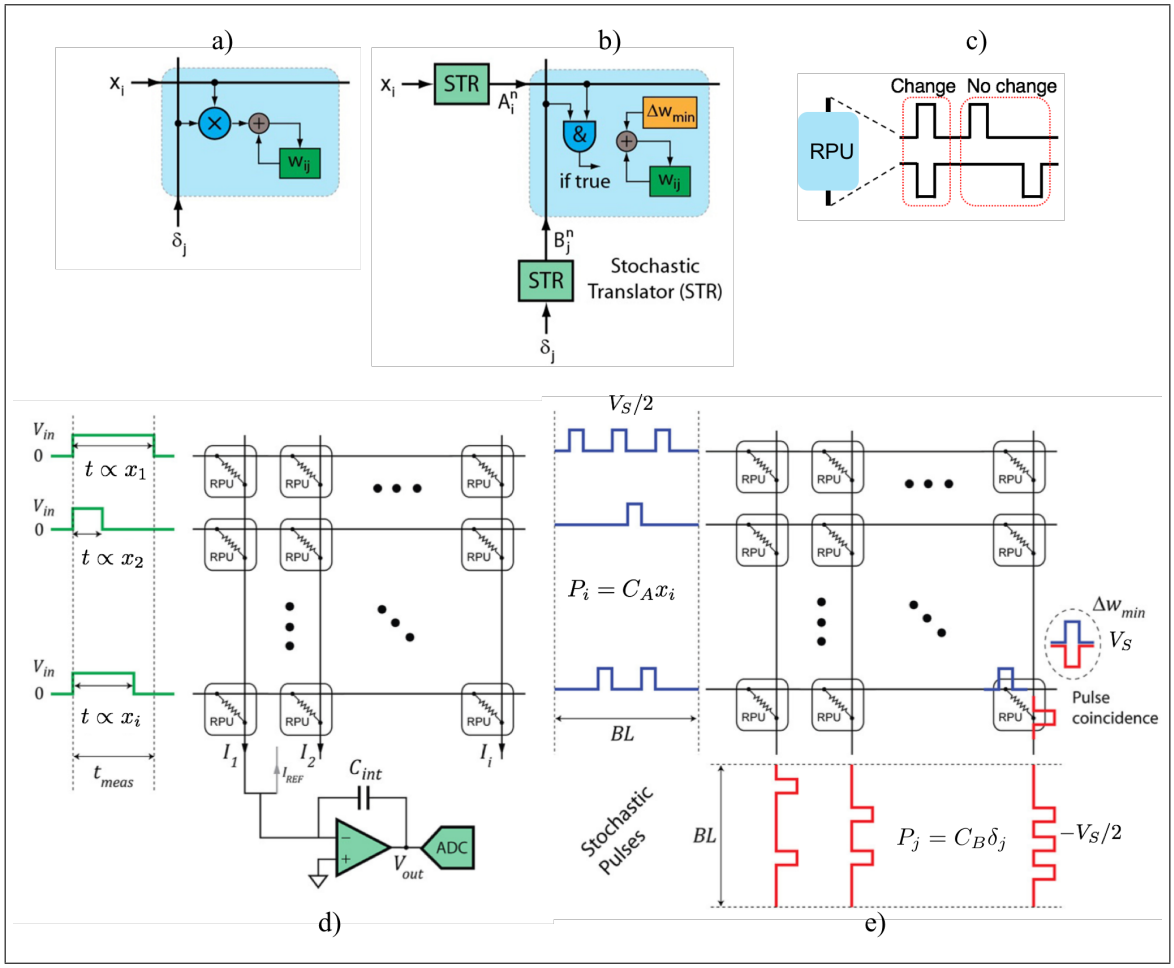

Figure 1.11. Resistive processing unit operating principle. a) Weight update rule applied in a crossbar array. Each weight in a crossbar structure receives an activation from the rows of the array (x_i), and a backpropagated error from the columns (δ_j), which are multiplied to update the weight value. b) Weight update performed locally by stochastic multiplication. The inputs x_i and δ_j are translated into stochastic bit streams and the weights are updated for each pulse coincidence. c) RPU implementation as a resistive element that changes its conductance by a nominal value Δw for each pulse coincidence and remains unchanged for single pulse arrivals. d) Forward propagation in a resistive crossbar array with RPUs implemented by a width-modulated pulse of amplitude $V_{in} < V_S/2$. e) Weight update step implemented by applying streams of pulses of amplitude $\pm V_S/2$ and opposite polarity. A pulse coincidence produces an amplitude V_S across the RPU. Figure reprinted from [21].

unchanged for single pulse arrivals. The sign of the multiplication is determined by the polarity of the pulses that are used during the update cycles [21, 22].

1.6 Challenges to implement multilevel memory devices for training

TABLE 1.2

RESITIVE PROCESSING UNIT REQUIREMENTS PROPOSED BY GOKMEN AND VLASOV [21]

Parameter	**Value**
Programming pulse	1 ns
Programming pulse ($\pm V_S$)	$\pm$1 V
Device area	0.04 μm^2
Average device resistance	24 MΩ
Maximum device resistance	112 MΩ
Minimum device resistance	14 MΩ
Resistance change at $\pm V_S$	100 kΩ
Resistance change at $\pm V_S/2$	$<$10 kΩ
Storage capacity	1000 levels

The requirements for an RPU to implement training with parallel weight update were outlined in [21], and are summarized in Table 1.2. The conductance (weight) needs to be updated for each pulse coincidence (resistance change at $\pm V_S$), whereas it has to remain unchanged when a single pulse arrives at either terminal (resistance

change at $\pm V_S/2$). Based on simulations of DNN training, Gokmen and Vlasov also showed that the symmetry of the RPU response to positive and negative weight updates was critical, requiring an asymmetry below 5% between conductance increments and decrements [21]. The importance of symmetric weight updates and other requirements for mutlilevel resistive elements have also been studied in [27–29].

Phase-change memory (PCM) [28, 30–32] and resistive random access memory (RRAM) [31–35] have been traditionally studied for multilevel weight elements. However, these materials exhibit highly asymmetric weight updates and abrupt transitions [31, 32]. There is ongoing research to improve the characteristics of RRAM and PCM, and several emerging material systems are also being explored [23, 32, 36–38]. Although ferroelecrics had been explored in the past for synaptic weight elements [39–46], the discovery of ferroelectricity in the CMOS-compatible hafnium oxide [47] has opened a new venue of research for neuromorphic and other applications [48–56].

Finally, several studies have focused on architectural considerations to mitigate nonidealities of the memory devices [20, 22, 24, 57–60]. Due to the fundamental trade-off between programming speed and retention time, it is challenging to realize nonvolatile memories with programming times in the nanosecond range and low programming voltages. For this reason, it has been proposed to use hybrid memory cells, where a slower, nonvolatile element is used for long-term retention and a faster element is used for frequent weight update [61–63]. These studies highlight the tight relation between devices and architectures, and the need to develop models to design peripheral circuits and architectures that address the limitations and trade-offs of the memory cells at the algorithm level.

1.7 Objectives and hypothesis

The general objective of this research has been to study and develop the use of ferroelectrics for multilevel memory elements for training deep neural networks

in resistive crossbar arrays. This objective was motivated by the hypothesis that ferroelectric partial polarization can be leveraged to develop high-density multilevel memories. In addition, architecture considerations were developed to reduce the memory requirements to train DNNs in resistive crossbar arrays.

The following objectives were proposed and achieved in the course of this research:

- Demonstrate the partial polarization of ferroelectric hafnium zirconate, a CMOS compatible ferroelectric material.
- Characterize and model polarization reversal in polycrystalline ferroelectrics to understand the physics and enable material optimization.
- Develop a model of the polarization dynamics in polycrystalline ferroelectrics for device and circuit design.
- Design and evaluate a scheme for accurate weight update in resistive crossbar arrays based on an approximate product obtained by local modulation of pulse width and frequency.
- Develop a hardware-efficient mapping between signed vector-matrix multiplications and its equivalent implementation in a crossbar array with nonnegative resistive elements.

1.8 Organization of this thesis

The first part of this thesis is dedicated to exploring the use of ferroelectrics for multilevel memory storage and its potential application in resistive crossbar arrays. Chapter 2 presents experimental results and analysis of partially polarized ferroelectrics for multilevel resistive memories for acceleration of deep neural networks. This chapter motivates the in-depth exploration of the polarization dynamics of ferroelectrics and the study of architecture approaches to improve the performance of resistive crossbar arrays with nonideal RPU cells. Chapter 3 presents the characterization and modeling of polarization reversal of ferroelecric hafnium zirconate under constant applied field. This is used to characterize the material system and extract relevant parameters. Then, Chapter 4 presents a general model for simula-

tion of ferroelectrics under arbitrary input waveforms, which is based on the material parameters extracted from the polarization reversal.

The second part of this thesis is dedicated to architecture considerations to improve the accuracy of DNN training in resistive crossbar arrays, focusing on the limited precision and other limitations of resistive weight elements. Chapter 5 presents a weight update scheme alternative to stochastic multiplication, which enables a parallel weight update with a simpler hardware implementation and lower variability. Chapter 6 presents an architecture to map DNN models efficiently to resistive crossbar arrays, considering that resistive weight elements are inherently non-negative and suffer from limited resolution and nonlinearity. A review of the main results and conclusions is presented in Chapter 7.

CHAPTER 2

MULTILEVEL FERROELECTRIC MEMORY FOR RESISTIVE CROSSBAR ARRAYS

This chapter introduces ferroelectrics (FE) and considers their use for dense multilevel memory storage. A behavioral simulation is presented to evaluate the potential performance of FEs for synaptic weight storage, and an FE-based memory device is proposed for operation in a resistive crossbar array.

2.1 Ferroelectric polarization, hysteresis loops and partial polarization

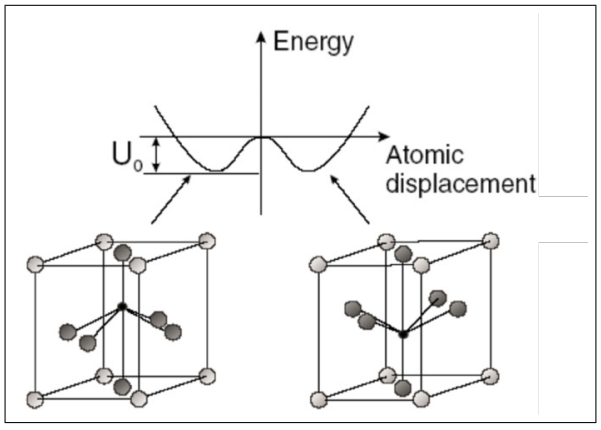

Figure 2.1. Ferroelectric permanent dipole moment in a noncentrosymmetric perovskite unit cell. The center atom can be at either of two stable positions, giving rise to a double-well potential as a function of the atomic displacement with respect to the center. Figure reprinted from [64].

Ferroelectrics materials exhibit a permanent dipole moment that can be configured by the application of an electric field. The ferroelectricity is originated by a noncentrosymmetric polar crystal phase [65], as depicted in Fig. 2.1. This results in a dipole moment that points either up or down, and can be switched by the application of an electric field. A single FE crystal tends to be in either of two stable states, where all of its unit cells are polarized in the same direction. In their polycrystalline form, an FE film is composed of a multitude of grains with independent polarization states, allowing for stable multilevel polarization of the FE film. More details about the polarization process will be given in Chapter 3.

The hysteretic polarization-voltage ($P-V$) loop depicted in Fig. 2.2 is a typical signature of ferroelectrics. Consider an FE capacitor with multiple grains, depicted by arrows separated by boundaries. When the FE is completely polarized in one direction, a saturation polarization $-P_S$ [C/cm^2] is obtained. Upon the application of a positive triangular pulse (Fig. 2.2a)), the FE polarization will be completely reversed, resulting in a $+P_S$ polarization, provided that the pulse is long and large enough. The polarization is then reversed back to $-P_S$ by applying a negative pulse. The current response during this process is shown in Fig. 2.2b), and has 2 components. The first component is the dielectric response due to the FE capacitance ($I = C\mathrm{d}V/\mathrm{d}t$), and results in a square waveform for the triangular pulse in Fig. 2.2a). The second component corresponds to the current peaks produced when the polarization switches. Fig. 2.2c) shows the current plotted as a function of voltage. By integrating this current, the plot of Fig. 2.2d) is obtained, which shows the hysteretic $P-V$ loop. The opening at 0 V corresponds to $2P_S$.

More interestingly, if the applied pulse is short enough, a polycrystalline FE can be polarized in an intermediate state, as shown in Fig. 2.3. By applying short pulses of varying width and amplitude, partially polarized states ranging from $-P_S$ to P_S can be obtained. This operation regime results in multilevel memory storage that

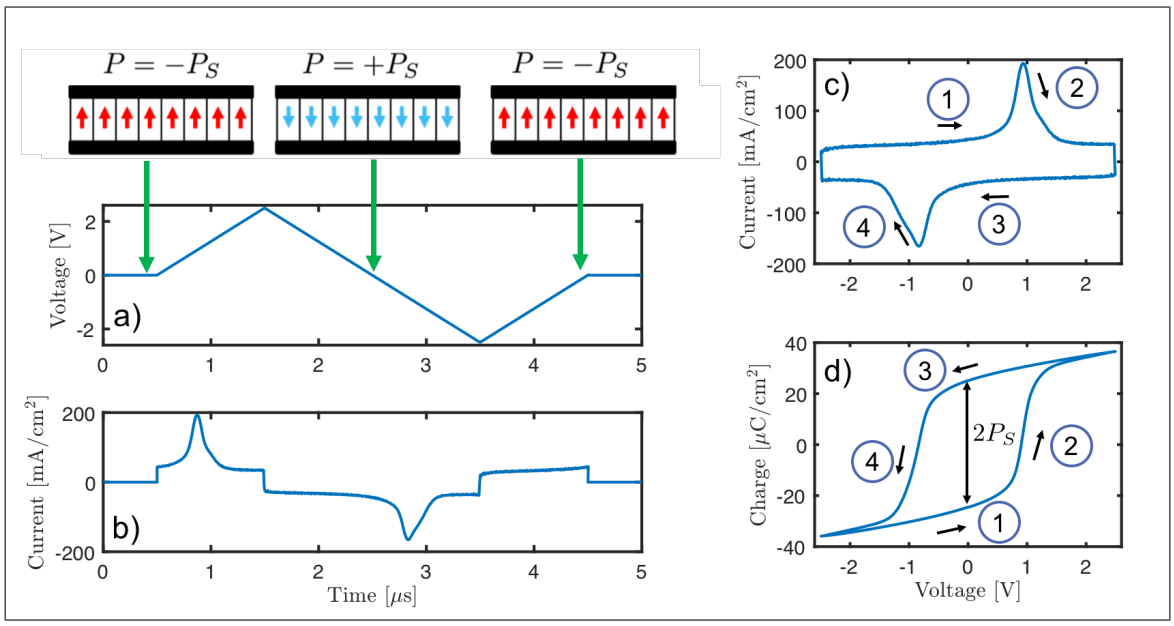

Figure 2.2. Ferroelectric polarization and characteristic $P-V$ loop. a) A triangular pulse is applied to switch the ferroelectric polarization between two saturated states $\pm P_S$. b) Current due to capacitance and FE polarization. c) Current-voltage characteristic. d) Polarization-voltage loop obtained by integrating the current response.

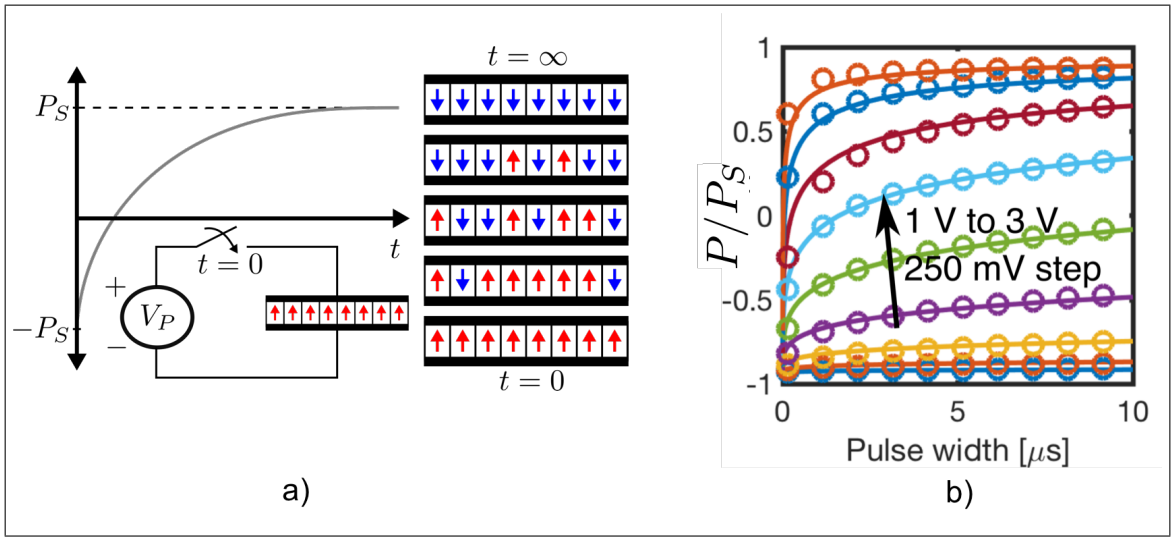

Figure 2.3. Partial polarization in a polycrystalline FE film. a) Starting from a fully polarized state $-P_S$, a positive voltage is applied at $t = 0$. The FE grains switch and increasingly align until the FE reaches the fully polarized state $+P_S$. b) Partially polarized states ranging from $-P_S$ to P_S can be obtained by applying short pulses of varying width and amplitude.

can be leveraged for several applications, and is the focus of this chapter.

2.2 Characterization of partial polarization of ferroelectric capacitors

To characterize the partial polarization of ferroelectrics, two FE capacitors were tested: a commercial 255-nm-thick lead zirconate titanate (PZT) capacitor [66], and a 12-nm-thick $Hf_{0.8}Zr_{0.2}O_2$ (HZO) capacitor, fabricated by Golnaz Karbasian of UC Berkeley [67]. The cross sections of the PZT and HZO capacitors are shown in Fig. 2.4. The HZO capacitor fabrication started by sputtering 40 nm TiN on a Si wafer. The HZO was deposited by atomic layer deposition (ALD) at 250 °C with tetrakis(ethylmethylamino)-hafnium and tetrakis(ethylmethylamino)-zirconium precursors, and water vapor as the oxidant [67]. The HZO was then capped with 30 nm of sputtered TiN and annealed at 500 °C for 30 s in N_2. The top TiN was etched and 60 μm-diameter dots were deposited by evaporation of Ni/Pd and lift-off. Finally, the HZO was etched and contacts to the back TiN were formed by evaporation of Ni/Pd and lift-off.

Electrical measurements were performed with a Keithley 4200 parameter analyzer with a 4225-PMU pulse measurement unit and two 4225-RPM remote preamplifiers. Before the partial polarization measurements, a triangular waveform with 4 ms period is applied for 100 cycles as a wake-up procedure, with 5 V amplitude for PZT and 3 V for HZO. The current-voltage and P–V characteristics are measured by applying the same waveform after wake-up and are shown in Fig. 2.5 for both HZO and PZT.

The measurement protocol designed to characterize the partial polarization is depicted in Fig. 2.6. A reset pulse of amplitude V_R polarizes the FE to the $-P_S$ state. A programming pulse of amplitude V_P and width T_P is applied to partially polarize the ferroelectric. The partial polarization is read out by applying two consecutive triangular pulses. The first pulse polarizes the capacitor back to the $-P_S$ state, and produces a current due to the linear capacitance and the polarization current

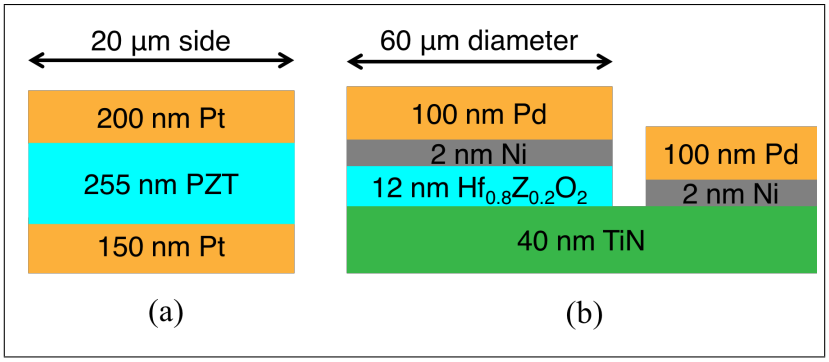

Figure 2.4. Cross section of (a) PZT and (b) HZO FE capacitors. HZO material provided by Sayeef Salahuddin and Golnaz Karbasian from UC Berkeley. Sample fabrication by Pratyush Pandey.

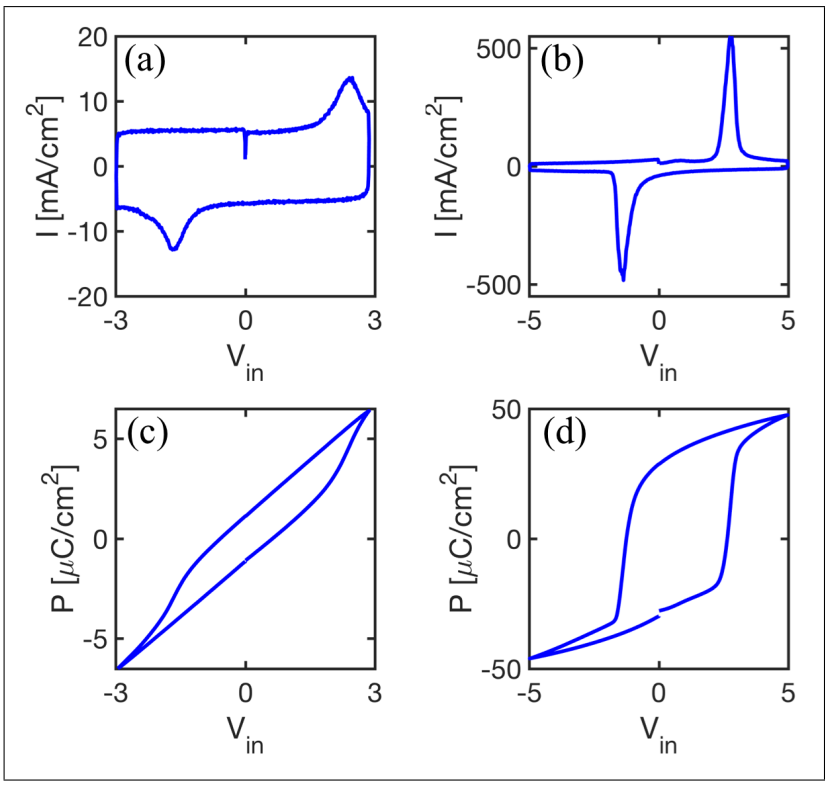

Figure 2.5. Ferroelectric polarization characteristics after wake up. Current-voltage characteristic for (a) HZO and (b) PZT. The P-V loop is obtained by integrating the current-voltage characteristic for (c) HZO and (d) PZT.

of the switched domains. The displacement current due to the linear capacitance alone is measured by the second pulse, where there is no polarization current. The current difference ΔI is integrated to calculate the partial polarization produced by the programming pulse ΔP.

Figure 2.7 shows the partial polarization measurements for PZT and HZO capacitors with pulse widths ranging from 200 ns to 200 μs. The reset and readout pulse amplitudes were 5 and 3 V for PZT and HZO, respectively. The partial polarization shows a nonlinear response to pulse amplitude, which can be leveraged to implement stochastic multiplication for parallel weight updates as described in Chapter 1.

2.3 Performance simulation in crossbar-based DNN accelerator

To evaluate the use of partially switched ferroelectrics as synaptic weight storage elements in crossbar-based accelerators, a behavioral simulation of neural network training was implemented in MATLAB. The simulation was implemented using the network in Fig. 1.5 and the MNIST dataset of handwritten digits [68]. The input layer size is 784 (28×28 pixels, normalized between 0 and 1), followed by two hidden layers with 256 and 128 neurons, and an output layer with 10 neurons for labels from 0 to 9. Sigmoid activation functions were used in hidden layers, softmax activations at the output and log-likelihood cost function [6, 7]. A baseline model was implemented using floating point precision and the ideal weight update

$$w_{ji} \leftarrow w_{ji} - \eta x_i \delta_j, \tag{2.1}$$

where w_{ji} is the weight that connects the neuron i in layer $L-1$ (preneuron) to the neuron j in layer L (postneuron). The baseline model achieves a 1.96% accuracy on the test set (Fig. 2.8), defined as the percentage of misclassified images from a validation set of 10,000 images [68].

For crossbar-based accelerators, the layers in the neural network are mapped to a crossbar structure with a multilevel memory element [21]. To simulate the stochastic weight update, introduced in Chapter 1, the update rule in Eq. (2.1) is replaced by

$$w_{ji} \leftarrow w_{ji} \mp \Delta w_0 N, \tag{2.2}$$

where N is the number of pulse coincidences obtained with stochastic multiplication and Δw_0 is the nominal weight update for each pulse [21]. For an ideal memory element, the weight value would increase/decrease linearly with the number of pulses until it reaches its maximum values $\pm w_{max}$, as shown in Fig. 2.9a). To a first order approximation, the FE polarization reversal can be modeled by an exponential settling. The update rule is modified to

$$w_{ji} \leftarrow w_{ji} \mp \Delta w_0 N \left(1 \pm \frac{w_{ji}}{w_{max}}\right) \tag{2.3}$$

and is shown in Fig. 2.9b). The number of levels required for the memory element can be roughly estimated by the largest Δw_0 and smallest w_{max} that can be simultaneously tolerated.

Figure 2.10 shows the performance of the neural network after 30 epochs (i.e. iterations of the 60,000 training images), measured as the percentage of mislabeled images on the test set. The performance is evaluated fo $\Delta w_0 = 0.001$, 0.01 and 0.1, and saturation values w_{max} ranging from 1 to 100. For $\Delta w_0 = 0.001$ and $w_{max} = 10$, the baseline error can be achieved, but the number of levels is prohibitive for practical implementations (20,000 levels). An error below 5% can be achieved with $\Delta w_0 = 0.01$ and $w_{max} = 2$ and 400 levels, whereas an error below 10% can be obtained with $\Delta w_0 = 0.1$ and $w_{max} = 2$ and only 40 levels. The FE area required can be roughly estimated by the grain size. Considering grain sizes with ~ 100 nm^2 area [69, 70], a FE could potentially store 10,000 levels in 1 μm^2.

2.4 Polarization-based analog memory for resistive crossbar arrays

Although HZO FE capacitors have a large storage density, many challenges still need to be addressed to make a useful memory device. To operate in a crossbar-based accelerator, the FE polarization needs to be read as a conductance state. This can be accomplished in a 2-terminal FE Tunnel Junction (FTJ), by measuring a change in tunneling current due to the ferroelectric polarization [71]. One of the challenges in this implementation is to optimize the asymmetric metal electrodes and scale the ferroelectric, so that the ON-state current is sufficiently large for fast readout.

Ferroelectric HZO can also be integrated into established CMOS process flows to form FE field-effect transistors (FeFET) [72]. In the FeFET, the polarization of the HZO capacitor modulates the charge in the semiconductor channel, and can be read as a shift in the threshold voltage. For a crossbar-compatible implementation, a two-terminal FeFET variation is proposed, depicted in Fig. 2.11. The device consists of an FE on top of a semiconductor channel with contacts at both ends (Fig. 2.11a)). An overlying top plate is connected to one of the contacts. The FE built-in nonlinearity is leveraged to alter the polarization by pulsing between the metal contacts at biases large enough to partially polarize the ferroelectric. The conductance state of the semiconductor channel is read out at a low bias. A vertical implementation is shown in Fig. 2.11b). The proposed device led to a US patent application sponsored by the Semiconductor Research Corporation (see Appendix B).

2.5 Conclusion

The polarization of ferroelectric PZT and HZO capacitors was studied, showing that they can be partially polarized with pulses down to nanosecond scales. These results represent the first measurements of partial polarization of FE HZO for multilevel memory storage and were presented at the 2017 Device Research Conference [73]. The

polarization response shows a highly nonlinear voltage dependence, which enables the use of stochastic multiplication for parallel weight update in resistive crossbar arrays. A neural network for classification of handwritten digits was simulated to provide a performance evaluation, showing the trade-off between accuracy and dynamic range. These findings led to a patent filing on a two-terminal multilevel memory for crossbar-based neural network accelerators proposed to access and program the FE memory state, included as Appendix B and filed on 5 November 2018.

During the course of this thesis work, many studies of partially polarized ferroelectrics have been presented in the literature [50, 52, 54, 63, 74–77], highlighting its potential for multilevel memories. Device and cell structures for neural network applications have been demonstrated and benchmarked [54, 63], and experimental studies of programming schemes to improve the linearity and symmetry of the polarization response have been presented [52, 54, 63, 74]. To design these devices and programming schemes, accurate and predictive models are required, which is the focus of the following chapters.

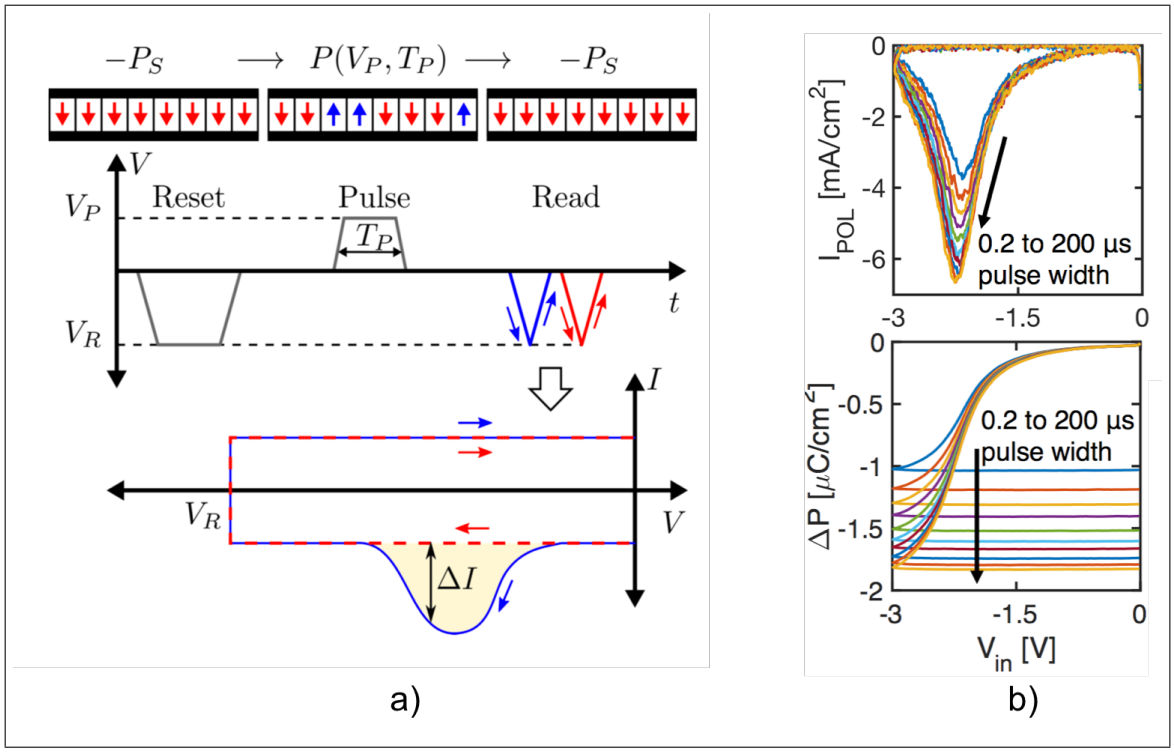

Figure 2.6. Measurement protocol for FE partial polarization. a) A reset pulse is applied to set the capacitor to the $-P_S$ state. A programming pulse (not to scale) of variable width and amplitude is applied to partially polarize the capacitor. The polarization is measured by two consecutive negative pulses, by which the displacement current is subtracted to obtain the polarization current. b) Example of polarization current (top) and partial $P-V$ loop (bottom) measured for different pulse widths. The opening of the partial $P-V$ loop corresponds to the polarization produced by the pulse.

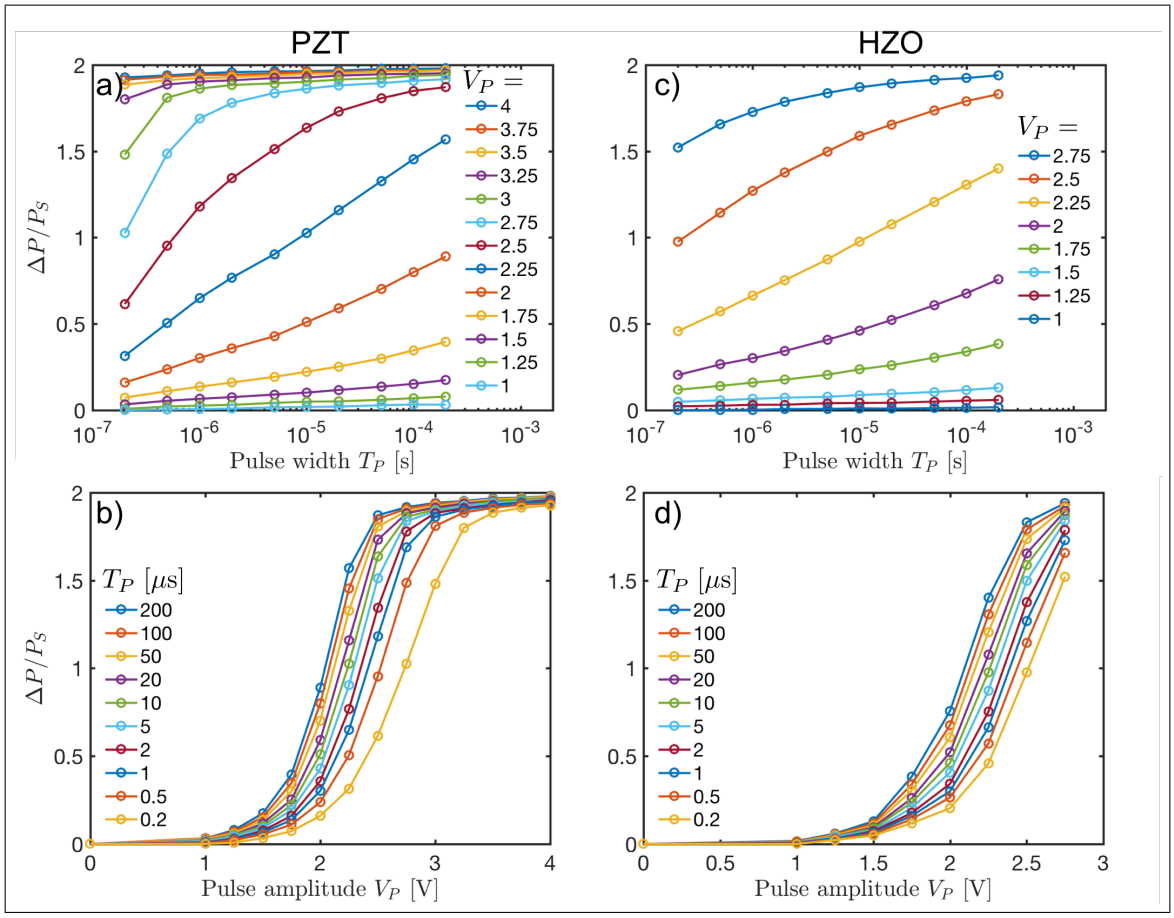

Figure 2.7. Partial polarization measurements as a function of pulse width for a) PZT and c) HZO capacitors. Partial polarization measurements as a function of pulse amplitude for b) PZT and d) HZO capacitors.

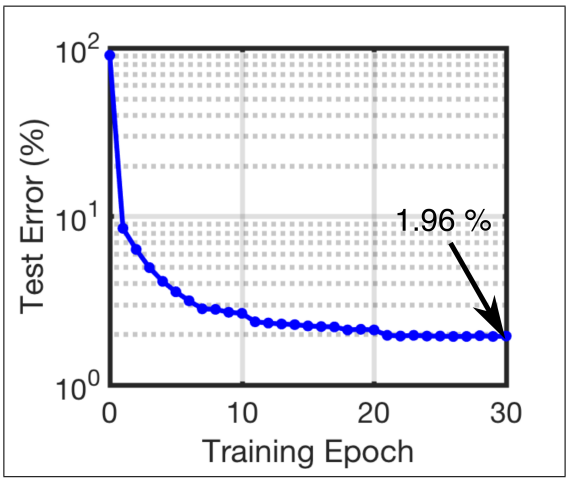

Figure 2.8. Baseline DNN training results obtained using the ideal update rule from Eq. (2.1) and η =0.01, 0.005 and 0.0025 for epochs 1 to 10, 11 to 20 and 21 to 30 respectively. A 1.96% error is achieved on the test set.

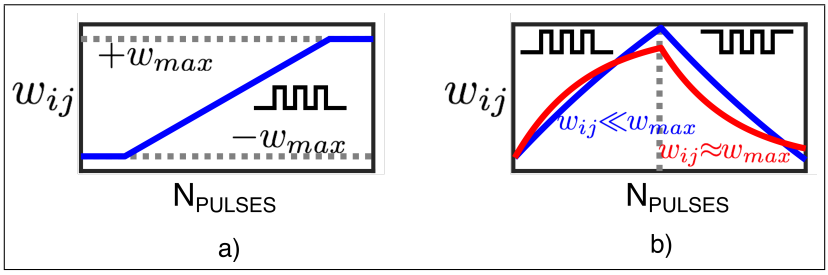

Figure 2.9. Weight update characteristic for a) linear memory element with symmetric increase/decrease and b) FE memory element.

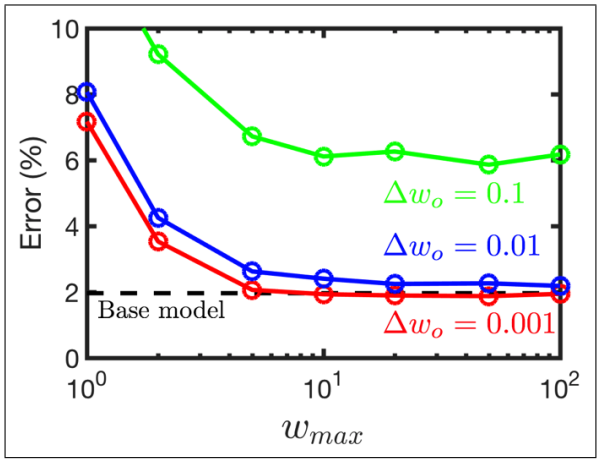

Figure 2.10. Simulated test error after 30 epochs with the weight update rule in Eq. (2.3) for different values of Δw_o.

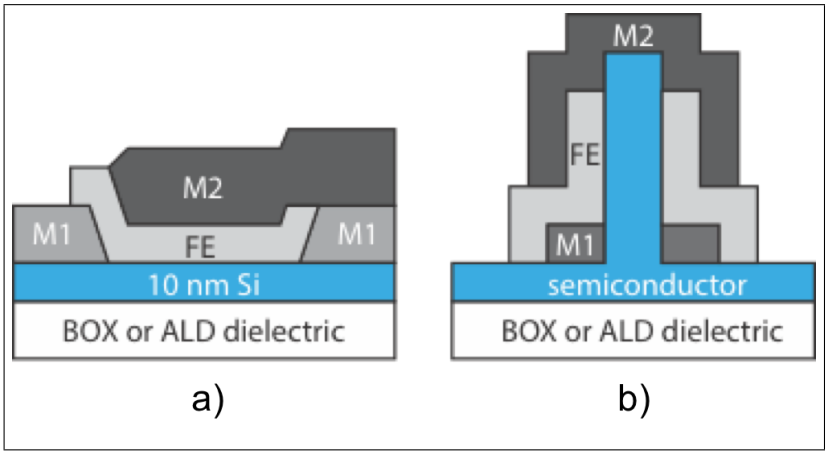

Figure 2.11. Proposed RPU based on the partial polarization of a ferroelectric. a) Planar implementation and b) vertical implementation.

CHAPTER 3

CHARACTERIZATION AND MODELING OF FERROELECTRIC POLARIZATION REVERSAL

The discovery of ferroelectricity in the CMOS-compatible HfO_2 material system [47] has led to a variety of applications including memory [78, 79], steep slope transistors [53, 56], and neuromorphic computing [52, 73]. Several studies have analyzed the effect of growth and annealing conditions on the FE properties of HZO, such as the Zr concentration [67, 80], electrode material [81, 82] and annealing temperature [83, 84]. These studies usually focus on properties that can be directly measured from the P-V loops, such as the remanent polarization and endurance, but provide limited insight into the FE dynamics or speed limitations, a subject that is widely debated [85, 86].

In this chapter, the polarization reversal of ferroelectric HZO is analyzed to show that the measurements are well described by a nucleation limited switching model, which enables extraction of the minimum switching time and the probability distribution of local electric field variations in the polycrystalline film. This characterization framework is shown to be useful to quantify, compare and optimize the switching dynamics of polycrystalline FEs.

3.1 Polarization reversal in a ferroelectric crystal

To understand the polarization dynamics in a polycrystalline FE, it is necessary to understand the processes that govern the polarization reversal in a single FE crystal, as shown in the diagram in Fig. 3.1. Consider an FE crystal that is completely

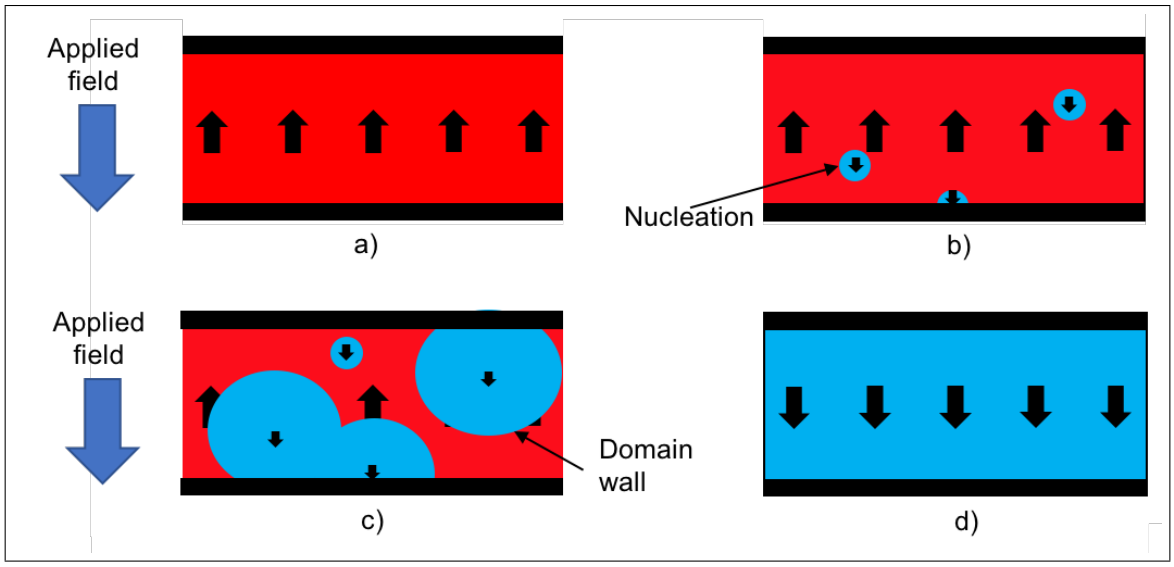

Figure 3.1. Polarization reversal in a single ferroelectric crystal. a) Starting from a fully polarized state, an electric field that opposes its polarization is applied. b) The polarization reversal starts with the nucleation of domains with opposite polarization with a minimum volume V_c. c) Domain wall expansion: nucleated domains grow until d) an homogeneous polarization is obtained, which is aligned with the external field.

polarized in one direction. When an external field that opposes this polarization is applied, the dipole moments will tend to align to the external field. Polarization reversal typically happens in two steps due to the energy barrier between two polarization states. First, small volumes of reversed polarization are nucleated with a certain rate per unit volume. For these nuclei to be stable, they need to be above a certain critical size V_c. Once a domain above this critical size is nucleated, it starts growing by domain wall expansion. The FE polarization is reversed by a combination of domain nucleation and domain wall expansion until the entire crystal has aligned to the external applied field.

The physics-based Kolmogorov-Avrami-Ishibashi (KAI) model [87] describes the

polarization reversal dynamics in an infinite FE crystal by

$$Q(t) = 1 - e^{-A(t)}, \tag{3.1}$$

where the dimensionless factor $Q(t)$ represents the switched fraction of the FE and $A(t)$ corresponds to Avrami's extended volume [87]. Assuming a constant nucleation rate and domain wall velocity, $A(t)$ takes the form $(t/t_o)^n$, where t_o is a bias-dependent time constant and n is the Avrami exponent given by the dimensionality of the domain growth (i.e. 1, 2 or 3). The assumption of constant wall velocity is not necessarily accurate and is a source of noninteger values of dimensionality that result from least-square fitting of experimental data [88]. Considering the polarization reversal from $-P_S$ to P_S, the polarization can be expressed in terms of $Q(t)$ as

$$P(t) = -P_S + 2P_S Q(t), \tag{3.2}$$

where $\Delta P(t) = 2P_S Q(t)$ is the polarization change due to the switched fraction $Q(t)$.

3.2 Nucleation-limited switching in polycrystalline ferroelectrics

In thin FE films, the grain size is comparable to the film thickness [89], and the KAI model is no longer accurate due to the contribution of multiple grains and interaction with grain boundaries [90]. The Nucleation-Limited Switching (NLS) model was proposed to account for multigrain polarization by characterizing the thin film as an ensemble of elementary regions that switch independently with a distribution of switching time constants [91]. The nucleation-limited switching can be described as shown in Fig. 3.2. The FE film is composed of a set of grains, which are fully polarized and point in the same direction. Upon the application of an external field that opposes its polarization, the polarization reversal will start

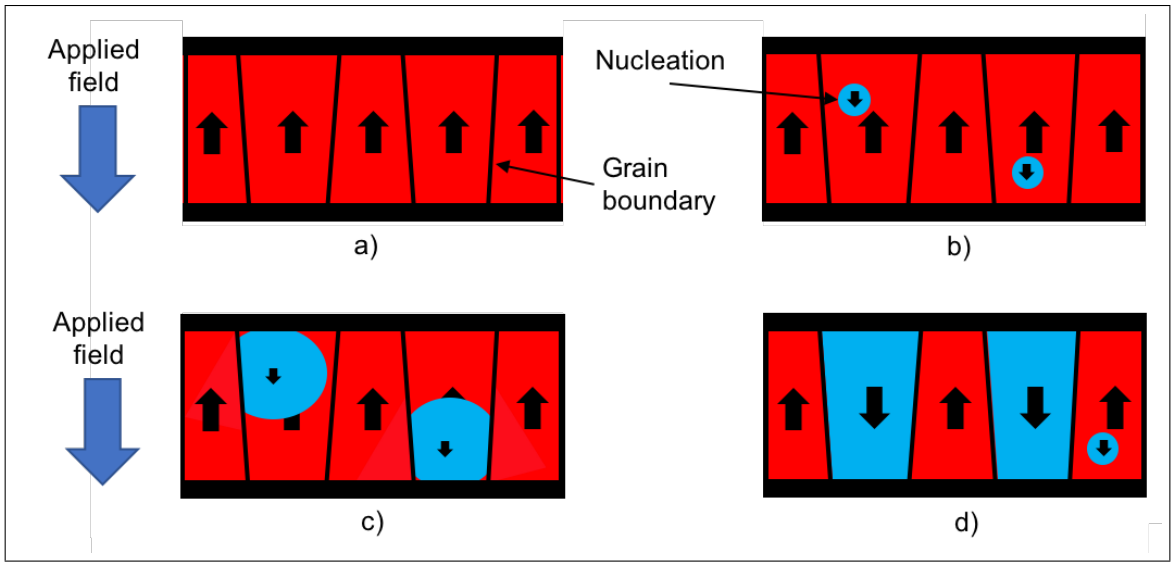

Figure 3.2. Polarization reversal in a polycrystalline FE crystal. a) Starting from a fully polarized state, where all grains are polarized in the same direction, an electric field that opposes its polarization is applied. b) As in an infinite crystal, the polarization reversal start with domain nucleation. c) Domain wall expansion stops at grain boundaries. d) Grains continue to switch independently by domain nucleation.

by domain nucleation, as in the case of a single crystal. However, it is assumed that domain wall motion stops at the grain boundary and does not propagate to an adjacent grain. Therefore, each grain in the FE film switches independently when a domain of reversed polarization is nucleated within its boundaries, but it is not affected by the switching of adjacent grains.

The NLS theory presented in [91] also assumed that the wait time for a nucleation event is much larger than the time needed for the expanding domain to occupy the entire grain. Under this assumption, the switching of a grain occurs instantaneously when a nucleation event occurs within its boundaries. Considering a constant nucleation rate $1/\tau$, the switching of a grain can be modeled as a Poisson process, where

the cumulative distribution function (CDF) of the switching time t_S is

$$P(t_S < t|\tau) = 1 - \exp\left(-\frac{t}{\tau}\right). \tag{3.3}$$

The average polarization of the film is given by

$$P(t) = -P_S + 2P_S \int_0^\infty P(t_S < t|\tau)\theta(\tau)\mathrm{d}\tau, \tag{3.4}$$

where $\theta(\tau)$ represents a distribution of switching times. This distribution of switching times was later attributed to variations in the local electric field when a uniform external field is applied, due to impurities or crystal defects [92], or the intrinsic inhomogeneity of the FE film [93]. In addition, as will be discussed in Chapter 4, the assumption of a constant nucleation rate can be inaccurate for thin-film FEs, and Eq. 3.3 was generalized to a stretched exponential with parameter β [92, 93]. Taking these modifications into account, the field-dependent NLS model can be summarized as follows:

The switching of a single elementary region is described by a stretched exponential with parameter β [92, 93]

$$p(t, \tau) = 1 - \exp\left\{-\left(\frac{t}{\tau}\right)^\beta\right\}. \tag{3.5}$$

The characteristic switching time τ is a function of the local field E and an activation field E_a, expressed by the empirical relation [88, 93]

$$\tau(E_a, E) = \tau_\infty \exp\left\{\left(\frac{E_a}{E}\right)^\alpha\right\}, \tag{3.6}$$

where τ_∞ is the time constant obtained for an infinite applied field, and α is an empirical parameter. Assuming an inhomogeneous and field-independent dielectric

permittivity, the local electric field value is expressed as $E = \eta E_{ext}$, where E_{ext} is the constant applied field and η is a random variable with a probability density function (PDF) $f(\eta)$ and unity mean, defined in the $[0, \infty)$ interval [93]. As a result, the polarization reversal from $-P_S$ to $+P_S$ is computed as

$$P(E_{ext}, t) = -P_S + 2P_S \int_0^\infty p(t, \tau(E_a, \eta E_{ext})) f(\eta) d\eta \tag{3.7}$$

With this mathematical formulation, the FE film is characterized by the parameters P_S, E_a, β, α, τ_∞ and the probability density function $f(\eta)$.

3.3 Experimental results and parameter extraction

Although it has been shown that the polarization reversal of FE HfO_2 occurs in the nucleation limited regime [48, 50, 94], only one prior study reports parameter extraction by applying an NLS model to Al-doped HfO_2 [77]. In this section, the field-dependent NLS model is applied to characterize an 8 nm thick $Hf_{0.5}Zr_{0.5}O_2$ FE capacitor, fabricated by doctoral student Pratyush Pandey. The cross section and transmission electron microscopy (TEM) of the capacitor are shown in Fig. 3.3a) and b). The device fabrication started by sputtering 65 nm W on a Si wafer. The HZO was deposited by atomic layer deposition at 300 °C with tetrakis-(ethylmethylamino)-hafnium and tetrakis-(ethylmethylamino)-zirconium precursors, and water vapor as the oxidant. The HZO was then capped with 65 nm of sputtered W and annealed at 500 °C for 30 s in N_2. Then, 60 μm-diameter dots were deposited by shadow mask evaporation of Ti/Pd. Finally, the top W and HZO were etched using the Ti/Pd electrodes as a hard mask. A Hf:Zr ratio of 1:1 was verified with energy dispersive X-ray linescans.

Electrical measurements were performed with a Keithley 4200 parameter analyzer with a 4225-PMU pulse measurement unit and two 4225-RPM remote preamplifiers.

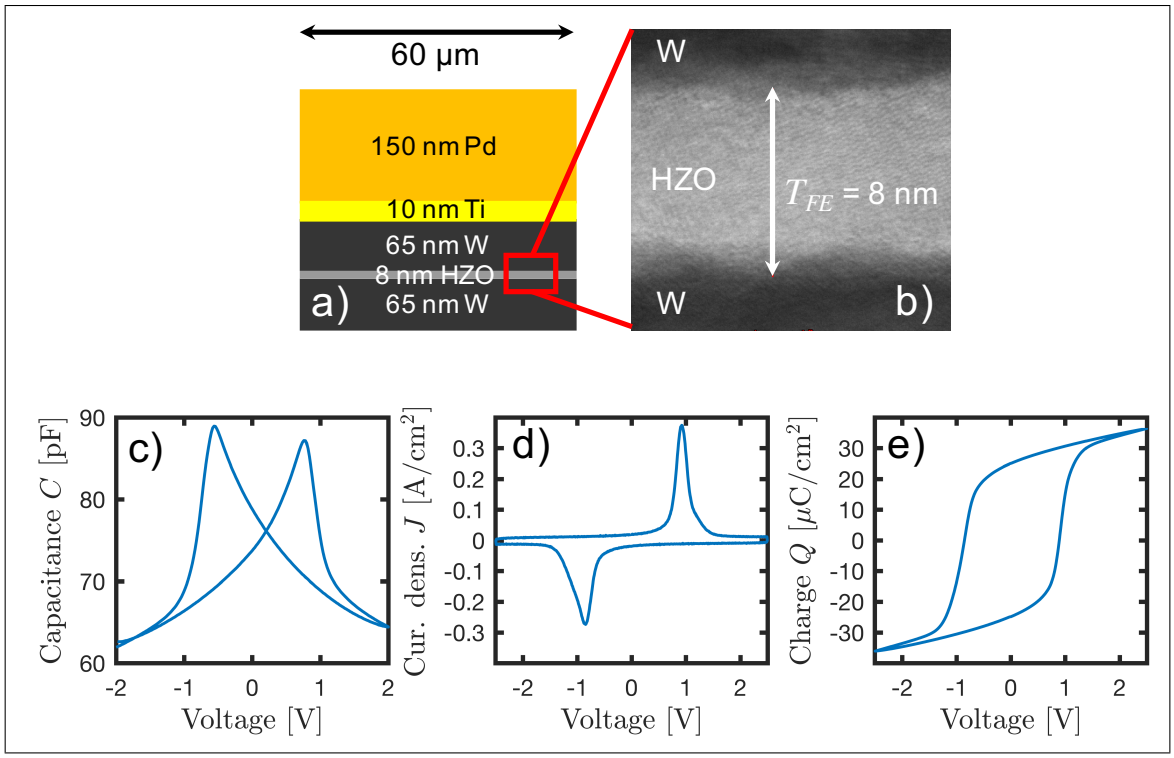

Figure 3.3. a) Cross section and b) TEM of 8 nm thick $Hf_{0.5}Zr_{0.5}O_2$ FE capacitor with 60 μm-diameter top electrode. c) Capacitance-Voltage measurements with 1 V/s sweep rate and 30 mV, 100 kHz AC signal. d) Current-voltage characteristic of $Hf_{0.5}Zr_{0.5}O_2$, measured with a 2.5 V triangular waveform with 4 ms period. e) P-V loop obtained by integrating the current-voltage characteristic.

The experimental setup and the capacitor diameter were designed so that the partial polarization measurements are not limited by RC delays. The combined resistance of the 50 Ω output resistance of the remote amplifier and the series resistance of the probes was measured to be 54 Ω, which with the measured capacitance (Fig. 3.3c)) results in a time constant in the order of 5 ns. A 2.5 V triangular waveform with 4 ms period was applied for 500 cycles for wake up. The current-voltage and P-V characteristics are shown in Fig. 3.3d) and e), measured after wake up. The measurement sequence performed to characterize the polarization reversal is depicted in Fig. 3.4. Conditioning pulses of amplitude $V_R = 2.5$ V are applied to reset the FE. The pro-

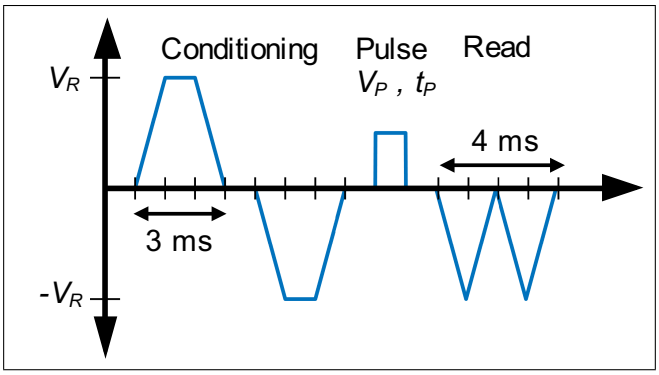

Figure 3.4. Measurement protocol for polarization reversal. Conditioning pulses of amplitude $V_R = 2.5$ V are applied to reset the FE. A programming pulse of varying width and amplitude is applied, and the polarization is measured by two pulses of amplitude $V_R = 2.5$ V.

gramming pulse width (t_P) was stepped from 200 ns to 7.6 ms in increments of 1.5×, then the amplitude (V_P) was stepped from 0.8 V to 2 V in increments of 100 mV. The polarization is measured by two consecutive negative pulses of amplitude $V_R = 2.5$ V and 1 ms rise time. The first pulse polarizes the capacitor back to the $-P_S$ state, and produces a current due to the linear capacitance and the polarization current of the switched domains. The displacement current due to the linear capacitance alone is measured by the second pulse, where there is no polarization current. The current difference is integrated to calculate the partial polarization. The pulse amplitude V_P is translated to field E_{ext} by dividing by the film thickness (T_{FE}).

Figure 3.5 shows the polarization reversal measurements (dots) and the fitted model (solid lines) as a function of pulse width and pulse amplitude. In addition, polarization measurements with 2.5 V pulses (diamonds) are shown to verify that the conditioning and read pulses produce a saturated polarization.

The distribution of local field variations $f(\eta)$ was extracted from measurements with the method presented in [93]. The logarithmic derivative of the polarization with respect to the applied field exhibits a maximum at a certain field E_{max} that

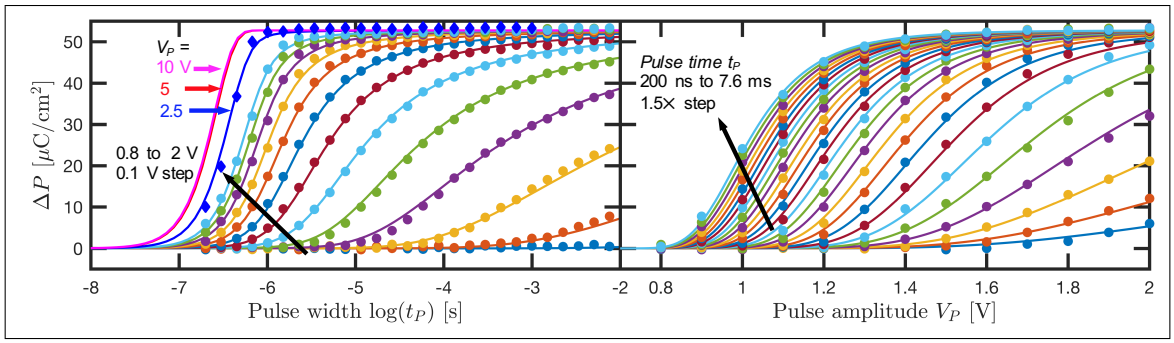

Figure 3.5. Measured partial polarization (dots) vs. pulse width (left) and pulse amplitude (right) show close agreement with model (solid line) over 5 decades of pulse times. Extracted parameters are $P_S = 26.4\ \mu\text{C/cm}^2$, $\tau_\infty = 236$ ns, $E_a = 2.42$ MV/cm, $\alpha = 3.73$ and $\beta = 2.06$. Measurements with 2.5 V pulse amplitude (diamonds) were not used for parameter extraction, which demonstrates the predictive capability of the model.

depends on the pulse time t_P, as shown in Fig. 3.6a). When the x-axis is normalized by E_{max} for each pulse time t_P, the derivatives overlap into a master curve $\Phi(x)$ (Fig. 3.6b)). The distribution $f(\eta)$ is obtained from the master curve as

$$f(\eta) = \frac{1}{\eta}\Phi\left(\frac{1}{\gamma\eta}\right), \tag{3.8}$$

where γ is a proportionality constant derived from the condition of unity mean [93]. As shown in Fig. 3.6c), the data is well described by a generalized beta distribution of type 2, whose PDF is

$$GB2(\eta|a,b,p,q) = \frac{\frac{|a|}{b}\left(\frac{\eta}{b}\right)^{ap-1}}{B(p,q)\left(1+\left(\frac{\eta}{b}\right)^{a}\right)^{p+q}}, \tag{3.9}$$

where $B(p,q)$ is the beta function. The distribution parameters are $a = 9.0986$, $b = 1.3935$, $p = 1.1101$ and $q = 15.197$. The parameters P_S, E_a, β, α and τ_∞ were then extracted by performing a least square fit of Eq. (3.7) with the polarization reversal data. Alternatively, once the analytic form of the distribution is known, its

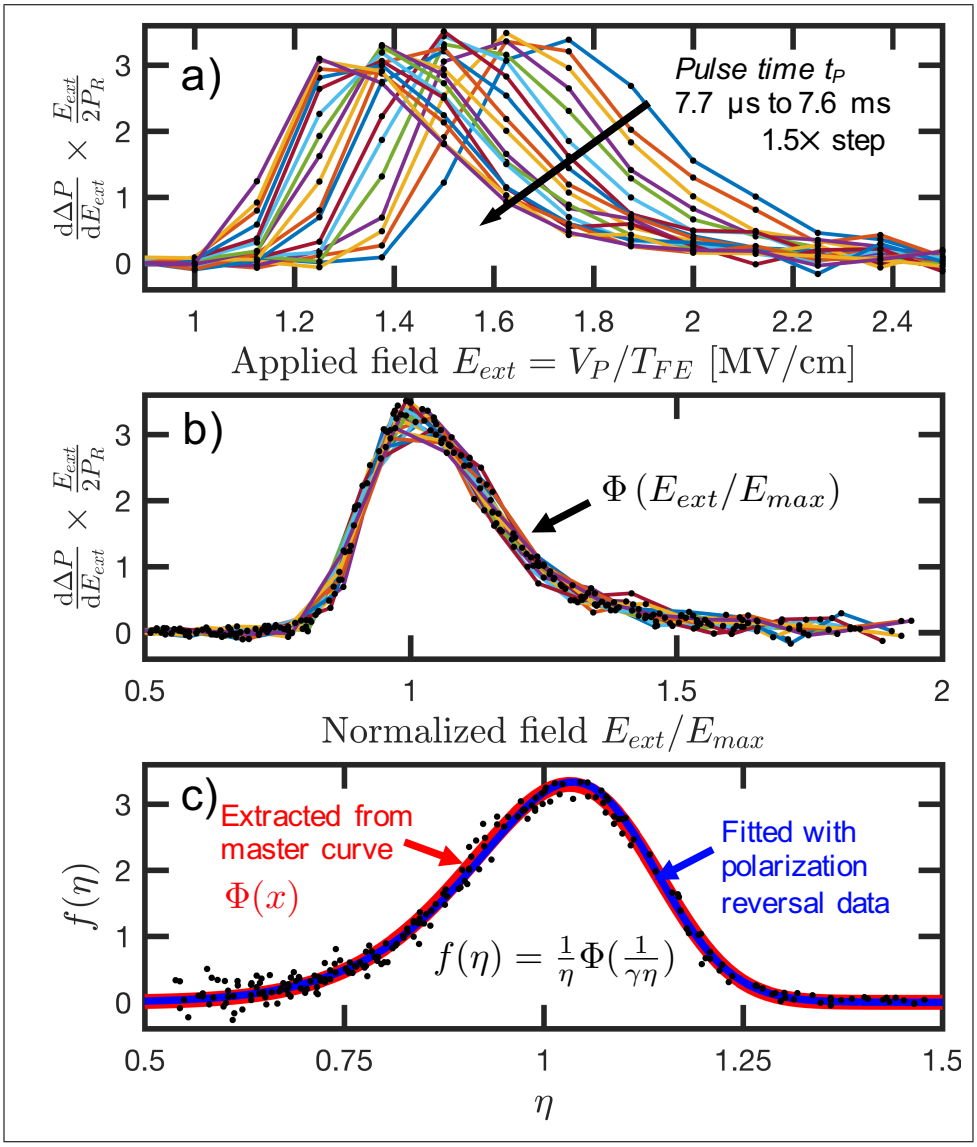

Figure 3.6. Extraction of the distribution of local field variations $f(\eta)$ [93]. a) Derivatives of the polarization with respect to applied field. b) The derivatives overlap into a master curve $\Phi(x)$ when the x-axis is normalized by the field at which the derivatives are peaked. c) Distribution of local fields obtained from $\Phi(x)$ (red), with the proportionality constant γ derived from condition of unity mean. The same distribution was obtained by fitting the distribution parameters directly from the polarization reversal data (blue).

parameters can be extracted with the least square fit directly from the polarization reversal data. The resulting distribution is not sensitive to the extraction method, as shown in Fig. 3.6c), and the fitted parameters vary by less than 1%. Furthermore, the fitted model is able to predict the measured polarization reversal with 2.5 V pulses, which was not used for parameter extraction.

According to Eq. (3.6), as the applied field increases, τ asymptotically reaches its minimum value $\tau_\infty = 236$ ns, which imposes a hard limit on the switching speed. This limitation is shown in Fig. 3.5 by extrapolating the pulse amplitude to 2.5, 5 and 10 V. The relative speed at which the FE grains switch is determined by the spread of the distribution of local field variations: grains at the higher end of the distribution have a smaller time constant and will switch sooner than those with smaller values of η. Therefore, a large variance in the local field distribution favors partial polarization, but is not desirable for fast transitions between saturated states ($\pm P_S$). Furthermore, a narrow distribution is needed for memory writing schemes that leverage the nonlinearity of the FE response to applied field [79].

3.4 Study of thickness dependence in ferroelectric HZO capacitors

To study the thickness dependence of the NLS parameters, another set of FE W/HZO/W capacitors were fabricated with HZO thickness being 8.3, 10.6, and 15 nm (fabrication by doctoral student Pratyush Pandey). Polarization reversal measurements were carried out by applying the measurement protocol in Fig. 3.4. The pulse width was stepped from 200 ns to 10 ms in increments of 1.5×, then amplitude was stepped in increments of 100 mV. Reset and read amplitude of 2.5, 3 and 3.5 V were used for 8.3, 10.8 and 15 nm capacitors, respectively. The complete procedure was repeated 3 times for each sample to verify that no significant aging effects are observed. The polarization reversal measurements and the fitted NLS model are shown in Fig. 3.7a). Note that due to the form of Eq. (3.6), a distribution of local

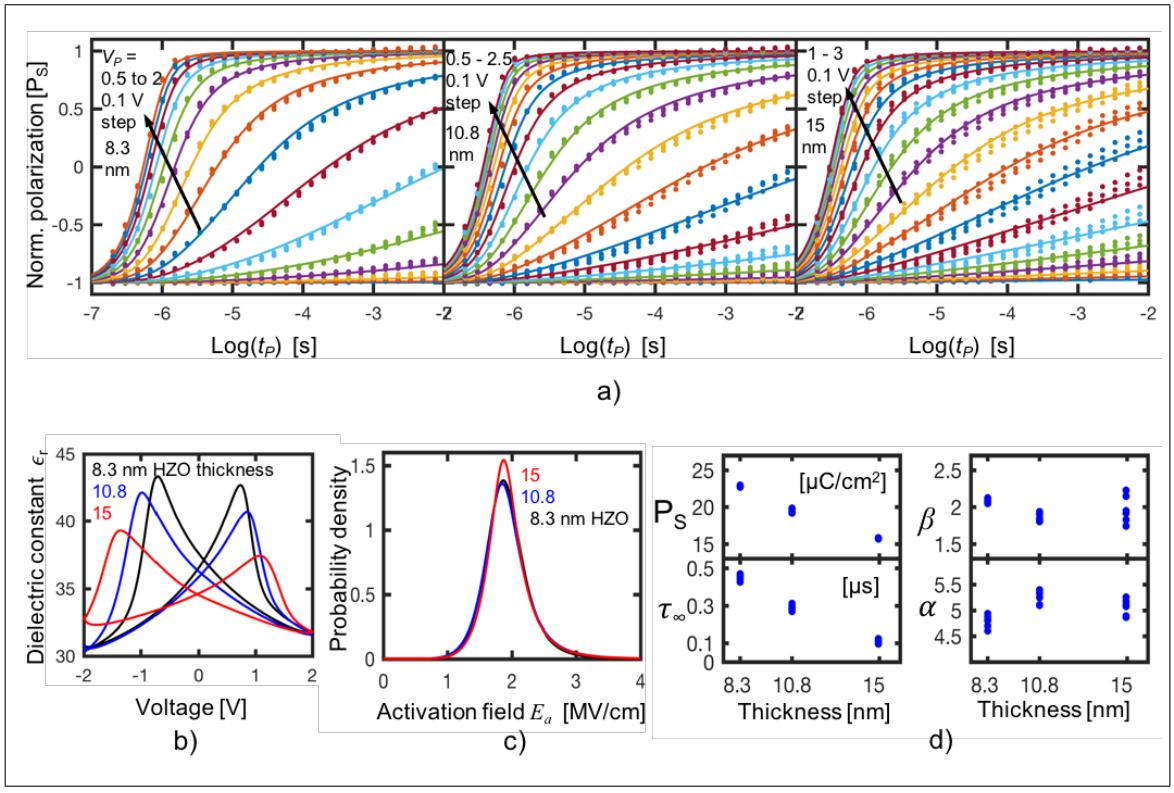

Figure 3.7. a) Partial polarization data for 3 runs (dots) show close agreement with fitted NLS model (solid line) over 5 decades. b) Measured dielectric constant from capacitance-voltage data (0.2 V/s sweep rate and 25 mV, 100 kHz AC). c) Extracted distributions of activation field reflect minor variations in the statistical properties with film thickness. d) Extracted parameters: P_S and τ_∞ decrease by 0.6× and 0.25× respectively for thickness from 8.3 to 15 nm, whereas α and β show variations below 10% for different samples and thickness.

field variations is mathematically equivalent to a distribution of effective activation fields $E'_a = E_a/\eta$ with probability density

$$g(E'_a) = \frac{\eta^2}{E_a} f(\eta), \tag{3.10}$$

which is also a generalized beta distribution of type II. The definition of an effective activation field can represent other physical phenomena, in addition to local field variations, that may cause variations in the switching time constants. With this approach, the activation field is incorporated into the distribution, and the FE parameters are P_S, τ_∞, α and β, in addition to the distribution of activation fields. The extracted distributions and parameters are shown in Fig. 3.7c) and Fig. 3.7d), respectively. As previously shown [84], the remanent polarization decreases with increasing film thickness without significant change in the extracted activation field distributions. The extracted minimum switching time constant τ_∞ also decreases with film thickness, reaching a minimum value in the order of 100 ns, imposing a hard limit on the switching speed of these FEs. Note that this limitation is not universal and can vary significantly for different film compositions and growth conditions.

3.5 Conclusion

The polarization reversal dynamics of polycristalline HZO was characterized and modeled. The results show that the field-dependent NLS model provides a comprehensive description of the polarization reversal for varying pulse amplitudes and pulse width spanning over 5 decades. The extracted probability distribution characterizes the local electric field variations in the FE film, and a minimum switching time constant of 100 ns was obtained for these deposition conditions and electrodes. This characterization framework provides the tools to quantify, compare and optimize the switching dynamics and the nonlinear response of HZO films. These results were

published in IEEE Electron Device Letters in November 2018 [95].

CHAPTER 4

MONTE CARLO SIMULATION OF POLARIZATION DYNAMICS IN POLYCRYSTALLINE FERROELECTRICS

NLS models [91–93] provide an accurate description of the polarization reversal dynamics of FE thin films. However, NLS models are limited as they are polarization reversal models, and can only describe the switching dynamics of an FE starting from a fully polarized state and under the application of a constant field. To design devices and circuits that leverage FE polarization, reliable and predictive models of the FE polarization dynamics are needed. Describing the switching behavior of thin-film polycrystalline FEs is complicated by the fact that they are composed of a multitude of grains having different switching thresholds, the distribution of which is highly dependent on the growth conditions. Therefore, to predict the time evolution of a FE film it is necessary to keep track of the configuration of switched grains.

Prior dynamic models have been based on the static Preisach model [96–98], which approximate the $P-V$ hysteresis loops by a hyperbolic tangent function, while the dynamic component is included by using equivalent circuits having either fixed or bias dependent time constants [98]. Due to these approximations, these models do not keep track of the distributions of switched grains, and resort to interpolation and scaling of parameters to replicate the history dependence of partially polarized FEs [96, 97].

The field-dependent NLS model characterizes the FE film as an ensemble of elementary regions that switch independently with a distribution of field-dependent time constants, effectively coupling the distribution of switching thresholds and the switch-

ing dynamics. These models have been experimentally validated in FE HfO_2 [95, 99], lead zirconate titanate [91–93, 100] and other material systems [100]. In this chapter, we present a Monte Carlo simulation approach that describes the dynamic, history-dependent switching of a polycrystalline FE. In this framework, the field-dependent NLS model is generalized for use with arbitrary input waveforms. After a parameter extraction from polarization reversal measurements, the model is able to accurately predict the dynamical behavior of FE hafnium zirconate (HZO) under various applied waveforms without further parameter tuning, showing the predictive capability of the model.

4.1 Revisit NLS model and nucleation rate

In the NLS model, it is assumed that the switching of a grain occurs once a domain of reversed polarization is nucleated, and the wait time for the first nucleation event is much larger than the time needed for a nucleated domain wall to expand and occupy the entire grain. The NLS theory presented in [91] originally assumed that nucleation events occur spontaneously at a constant rate $1/\tau$, so the switching of a grain was modeled as a Poisson process, where the cumulative distribution function (CDF) of the switching time t_S is

$$P(t_S < t|\tau) = 1 - \exp\left(-\frac{t}{\tau}\right). \tag{4.1}$$

However, according to classical nucleation theory, the nucleation rate is not constant [101]. Domain nucleation occurs in a series of stages, starting with an incubation period where small clusters with reversed FE polarization continuously form and decompose, the distribution of which evolves over time until a quasi-steady-state distribution is reached. During this period, the nucleation rate increases monotonically until it becomes almost constant [101]. The assumption of constant nucleation

rate was originally introduced as a special case to model the polarization reversal in an infinite crystal [87], where multiple nucleation events occur until the FE volume has reversed its polarization. In this regime, the incubation period could be safely ignored, but it can be the dominant factor in a polycrystalline FE where the switching time is determined by the first nucleation event.

Based on experimental results, Eq. (4.1) was generalized to a stretched exponential with parameter β [92, 93], which can be interpreted as a Weibull process [102] where the CDF for the switching time is given by

$$P(t_S < t|\tau, \beta) = 1 - \exp\left[-\left(\frac{t}{\tau}\right)^{\beta}\right]. \tag{4.2}$$

This results in a time-dependent switching rate

$$r(t) = \frac{\beta}{\tau}\left(\frac{t}{\tau}\right)^{\beta-1}, \tag{4.3}$$

as opposed to a constant nucleation rate. Note that for $\beta = 1$, this reduces to a Poisson process with constant rate $1/\tau$. With $\beta > 1$ a monotonically increasing nucleation rate is obtained, which provides an approximation for the FE nucleation during the incubation period.

As described in Chapter 3, the time constant τ is a function of applied field E_{FE} and an activation field E_a

$$\tau(E_a, E_{FE}) = \tau_\infty \exp\left[\left(\frac{E_a}{E_{FE}}\right)^{\alpha}\right]. \tag{4.4}$$

The polarization reversal is computed as

$$P(E_{FE}, t) = -P_S + 2P_S \int_0^\infty P(t_S < t|\tau(E_a, E_{FE}), \beta) g(E_a) dE_a. \tag{4.5}$$

TABLE 4.1

HZO PARAMETERS EXTRACTED FROM POLARIZATION REVERSAL MEASUREMENTS.

Parameter	Value
P_R	22.9 μC/cm^2
τ_∞	387 ns
α	4.11
β	2.07
a	12.1
b	1.79 MV/cm
p	0.691
q	0.633

where $g(E_a)$ is the distribution of activation fields, whose probability distribution is given by

$$GB2(\eta|a,b,p,q) = \frac{(|a|/b)\,(\eta/b)^{ap-1}}{B(p,q)\,[1+(\eta/b)^a]^{p+q}}, \tag{4.6}$$

where $B(p,q)$ is the beta function.

The FE parameters extracted from polarization reversal measurements are shown in Table 4.1. An offset voltage of $V_{OS} = 80$ mV was measured from $P{-}V$ loops, such that $V_{FE} = V_A + V_{OS}$, where V_A is the applied voltage and V_{FE} is the actual voltage across the FE. This offset was considered during parameter extraction and applied to all simulations. The field at the FE is computed as $E_{FE} = V_{FE}/T_{FE}$.

4.2 Monte Carlo simulation of polarization reversal

For the Monte Carlo simulation, a set of N grains $g^{(i)}, i \in (1, N)$ is initialized by sampling values of activation fields $E_a^{(i)}$ from the distribution $g(E_a')$. The parameters P_S, β, α, and τ_∞ are common to all the FE grains. Each FE grain can have one of two possible orientations, corresponding to a positive or negative polarization state ($s^{(i)} = \pm 1$), and the time evolution of each grain is governed by Eq. (4.2) and (4.4).

For a polarization reversal simulation from $-P_S$ to P_S, all grains are initialized to the state $s^{(i)} = -1$. Under a constant applied field, a grain $g^{(i)}$ has a fixed time constant $\tau^{(i)}$ given by Eq. (4.4). The simulation is performed by dividing the time into discrete time intervals and computing the probability of transition for each unswitched grain according to Eq. (4.2). This is expressed as the probability that the switching time t_S is in the time interval $[t, t + \Delta t]$, given that the grain has not switched until t,

$$P^{(i)}(t_S < t + \Delta t | t_S > t) = 1 - \exp\left[\left(\frac{t}{\tau^{(i)}}\right)^\beta - \left(\frac{t + \Delta t}{\tau^{(i)}}\right)^\beta\right]. \tag{4.7}$$

For each grain, the switching probability is evaluated as a Bernoulli trial with probability $P^{(i)}$, and the state $s^{(i)}$ is updated to $+1$ in case of success. The total polarization due to the orientation of the FE grains is computed as

$$P_{FE}(t) = \frac{P_S}{N} \sum_{i=1}^{N} s^{(i)}(t). \tag{4.8}$$

The Monte Carlo simulation for polarization reversal is summarized in Algorithm 1. Note that this model does not consider a distribution of grain sizes and orientations. Although these effects can easily be added to the simulation, further experimental work is required to characterize these effects and their correlation with the FE parameters and the activation fields.

Algorithm 1 Monte Carlo polarization reversal

Instantiate FE:

Define parameters $\{P_S,\ \beta,\ \alpha,\ \tau_\infty\}$

Sample N activation fields $E_a^{(i)}$ from $g(E_a')$

Initialization: for grains $g^{(i)}, i \in (1, N)$

$s^{(i)} \leftarrow -1$

$\tau^{(i)} \leftarrow \tau_\infty \exp\left[\left(E_a^{(i)}/E_{FE}\right)^\alpha\right]$

Simulation: for timestep $[t, t+\Delta t]$ and grains $g^{(i)}, i \in (1, N)$

if $s^{(i)} = -1$

$P^{(i)} \leftarrow 1 - \exp\left[(t/\tau^{(i)})^\beta - ((t+\Delta t)/\tau^{(i)})^\beta\right]$

if Bernoulli$(P^{(i)}) = 1$

$s^{(i)} \leftarrow 1$

end if

end if

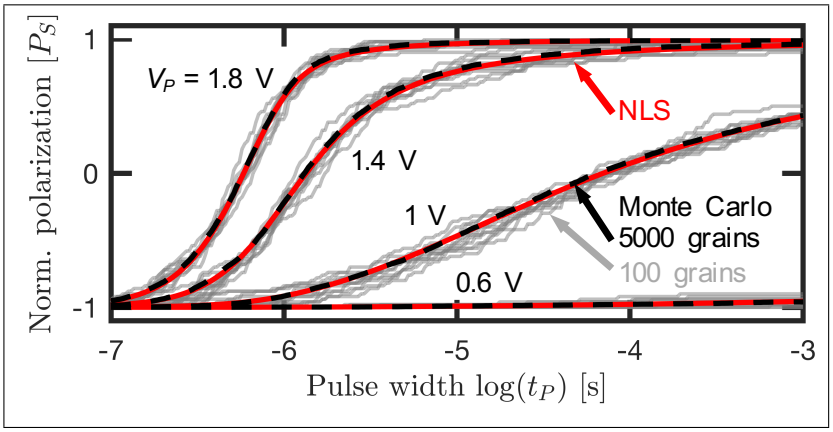

Figure 4.1. Polarization reversal simulation with NLS model (dashed red lines) and Monte Carlo simulation with 5000 grains (black) are indistinguishable. Monte Carlo simulations with 100 grains (gray) show variation around the mean value (10 repetitions).

Figure 4.1 shows Monte Carlo simulations of polarization reversal and the analytic polarization reversal computed with the NLS model with the same parameters (Table 4.1). A Monte Carlo simulation with 5000 grains is indistinguishable from the NLS model, whereas 10 runs with 100 grains show variability around the mean value.

Note that, as shown in Eq. (4.7), the switching probability has an accumulation effect over time, even for a constant applied field. Therefore, the state of a grain is not only determined by its polarization $s^{(i)} = \pm 1$, but also depends on the accumulated stimuli t/τ.

4.3 Monte Carlo simulation for arbitrary input waveforms

For an arbitrary field $E_{FE}(t)$, the time constant $\tau^{(i)}$ is a function of time, so the accumulated stimuli (previously computed as t/τ), is obtained by integrating the instantaneous values of $1/\tau$. This accumulated stimuli is defined as history parameter $h^{(i)}(t)$, which is computed as

$$h^{(i)}(t) = \int_{t_o}^{t} \frac{dt'}{\tau\left(E_{FE}(t'), E_a^{(i)}\right)}, \tag{4.9}$$

where t_o indicates the moment at which the stimuli to switch the grain starts. The switching rate is expressed as

$$r^{(i)}(t) = \frac{\beta}{\tau^{(i)}(t)} \left(h^{(i)}(t)\right)^{\beta-1}, \tag{4.10}$$

which results in a switching probability

$$P^{(i)}(t_S < t + \Delta t | t_S > t) = 1 - \exp\left[\left(h^{(i)}(t)\right)^{\beta} - \left(h^{(i)}(t + \Delta t)\right)^{\beta}\right]. \tag{4.11}$$

The Monte Carlo simulation is performed as outlined in Algorithm 2. After instantiating a FE with N grains, the state of each grain is initialized by defining its

Algorithm 2 General Monte Carlo simulation

Instantiate FE:
Define parameters $\{P_S,\ \beta,\ \alpha,\ \tau_\infty\}$
Sample N activation fields $E_a^{(i)}$ from $g(E_a')$
Initialization: for grains $g^{(i)}, i \in (1, N)$
$s^{(i)} \leftarrow 1$ **or** $s^{(i)} \leftarrow -1$
$h^{(i)} \leftarrow 0$
Simulation: for timestep $[t, t + \Delta t]$ and grains $g^{(i)}, i \in (1, N)$
if $s^{(i)}E(t) < 0$
$\tau^{(i)} \leftarrow \tau_\infty \exp\left[\left(E_a^{(i)}/|E(t)|\right)^\alpha\right]$
$h_{new}^{(i)} \leftarrow h^{(i)} + \Delta t/\tau^{(i)}$
$P^{(i)} \leftarrow 1 - \exp\left[(h^{(i)})^\beta - (h_{new}^{(i)})^\beta\right]$
$h^{(i)} \leftarrow h_{new}^{(i)}$
if Bernoulli$(P^{(i)}) = 1$
Update $s^{(i)}$
$h^{(i)}$ relaxation
end if
end if

polarization $s^{(i)} = \pm 1$ and setting the history parameter to 0. Note that only a scalar value $h^{(i)}$ is stored for each grain, and updated during the simulation. Given that the FE switching can occur in both directions (i.e. from 1 to -1 or from -1 to 1), it is first verified that a grain is not already aligned with the external field. For the grains that are not aligned with the external field, the history parameter is updated to compute the switching probability, which is evaluated as a Bernoulli trial and the state of the grain is updated in case of success. Finally, the history parameter is updated when a grain switches according to a given relaxation rule, which needs to be determined. For a first approximation, two possible cases are evaluated: reset $h^{(i)}$ to 0 after a grain has switched, or keep its current value.

The experimental protocol in Fig. 4.2(a) was applied to validate the Monte Carlo simulation and evaluate the relaxation condition for $h^{(i)}$. Starting with the FE fully polarized to the $+P_S$ state that has been resting for a minute, a double triangular waveform is applied. The first pulse completely polarizes the FE to the $-P_S$ state,

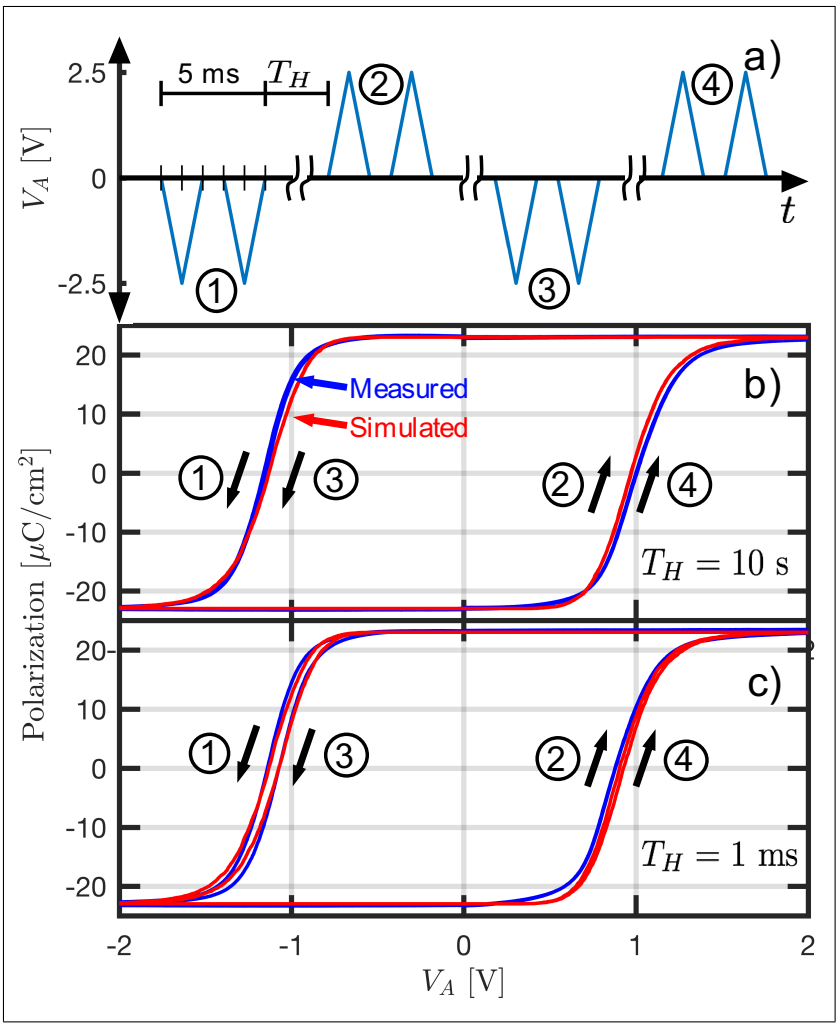

Figure 4.2. a) Experimental protocol to measure P–V loops. A double triangular waveform V_A is applied: the first triangle produces a current due to the linear capacitance and the polarization reversal. The displacement current due to the linear capacitance alone is measured by the second triangle, where there is no polarization current. A hold time T_H is applied between polarization pulses. Measured and simulated P–V loops with b) 10 s hold time and c) 1 ms hold time.

whereas the second pulse is used to measure and subtract the current due to the dielectric response and leakage. After a hold time T_H, a double triangular waveform of opposite polarity is applied to polarize the FE to the $+P_S$ state. After another hold time T_H, the procedure is repeated. The measured polarization response is plotted over the applied voltage in Fig. 4.2(b) with a 10 s hold time between pulses, which shows that transitions 1 and 3 (from $+P_S$ to $-P_S$) follow the same trajectories. Likewise, transitions 2 and 4 (from $-P_S$ to $+P_S$) also overlap. When the history parameter is reset after a grain switches (i.e. $h^{(i)} = 0$), the Monte Carlo simulation closely matches the experiment, shown with red lines in Fig. 4.2(b). When the hold time is reduced to 10 ms, a different behavior is observed. The first transition from $-P_S$ to P_S follows the same path as the case with a 10 s hold time, given that the initial condition is the same. However, subsequent transitions occurs at a lower voltage (earlier in time), as shown in Fig. 4.2(c). This apparent speed-up has been observed in similar experiments, and could be related to the distribution of clusters after a grain switches [103]. A simulation performed for the extreme case, where $h^{(i)}(t)$ is not reset between transitions, produces a similar behavior (red lines in Fig. 4.2(b)).

Having verified that the Monte Carlo model closely matches measurements of saturated $P-V$ loops, the model predictions were evaluated for minor loops. Figures 4.3(a) and (b) show experimental and simulated data taken with a triangular waveform of varying amplitude. Under these conditions, the dielectric response is not cancelled as in Fig. 4.2, so the total FE charge is modeled as

$$Q_{FE}(t) = P_{FE}(t) + \epsilon_{FE}E(t), \tag{4.12}$$

where ϵ_{FE} is the permitivitty of the FE film. For this simulation, $h^{(i)}(t)$ was not reset between transitions as in Fig. 4.2(c). The Monte Carlo simulation accurately

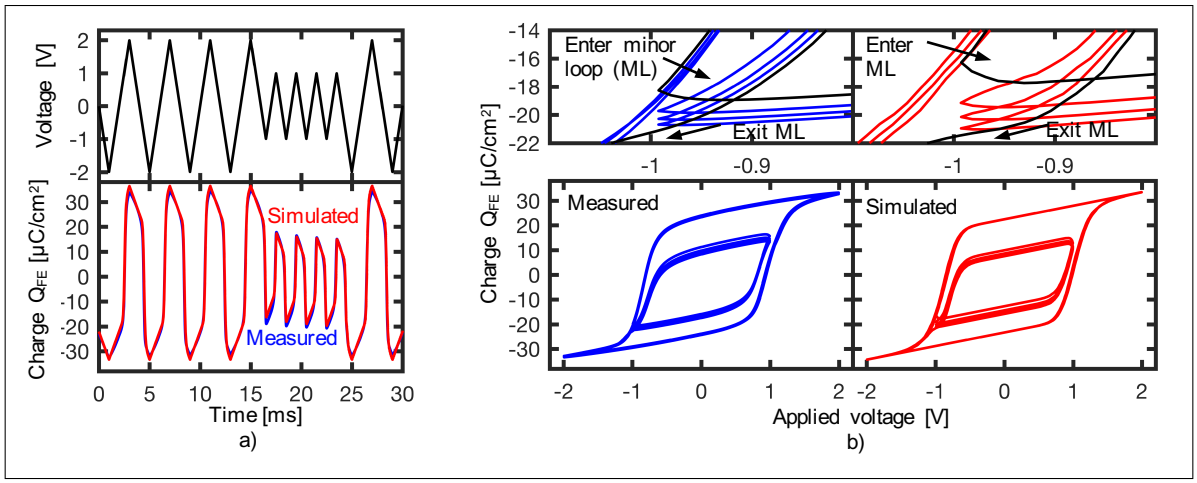

Figure 4.3. Experimental validation of Monte Carlo simulation framework. a) Measured and simulated polarization vs. time for an 8.3 nm HZO capacitor with a triangular input waveform of varying amplitude. b) Measured and simulated major and minor loops obtained from a) with detail of the transition between minor loops and major loops.

predicts the behavior of the FE as it enters and exits the minor loops, as well as the drifting of the minor loops with field cycling. Small differences between the measured and simulated characteristics occur in part due to the assumption of a constant FE capacitance, whereas the measured capacitance exhibits the well-known butterfly shape [95].

4.4 Accumulation and relaxation of the history-dependent switching rate

Based on experimental results, it has been observed that resetting the history parameter when a grain switches works well when a long resting period is applied between pulses. For shorter resting periods or for periodic stimuli, not resetting $h(t)$ produces a close match with experimental measurements, although this extreme case results in a continuously increasing rate that will slowly depart from experiments. Therefore, a more general reset condition would be to set $h^{(i)}$ to a certain reset value h_S that represents the distribution of clusters immediately after a grain switches,

which may be a function of the history parameter before switching and the grain parameters. In addition, a relaxation rule for the history parameter could be incorporated when there is no applied field or when the grain is already aligned with the external field. Such effects could be incorporated to the simulation as depicted in Algorithm 3, although its functional form remains to be determined. Note that other effects such as FE wake-up and fatigue shall not be modeled by the history parameter, but by variations in the FE parameters and the distribution activation fields.

The measurement protocol in Fig. 4.4 was applied to better understand the timescale of the relaxation behavior. Starting with a FE fully polarized in the $-P_S$ state, either a single pulse of varying width or a train of pulses with equivalent accumulated pulsed time are applied. The width-modulated pulse ranges from 1 to 20 μs. The train of pulses has a constant pulse width of 1 μs, with off time between pulses t_{OFF} of either 1 or 10 μs. Amplitudes of 1, 1.25 and 1.5 V are applied for both the width-modulated pulse and the train of pulses.

The Monte Carlo simulation was implemented according to Algorithm 3, by applying a simple relaxation rule during the off time between pulses, defined as

$$h^{(i)} \leftarrow h^{(i)} \times \gamma(t_{OFF}). \tag{4.13}$$

By setting $\gamma = 0.55$ for a 1 μs off time between pulses, and $\gamma = 0.3$ for 10 μs off time, the simulation closely matches the experiment for pulses of 1, 1.25 and 1.5 V amplitude. It is proposed that further investigation of the dynamics of formation and decomposition of clusters in the incubation period will lead to a direct relation between the switching rate and the underlying distribution of clusters, in order to define improved accumulation and relaxation equations.

Algorithm 3 Monte Carlo simulation with proposed relaxation

Instantiate FE:
Define parameters $\{P_S,\ \beta,\ \alpha,\ \tau_\infty\}$
Sample N activation fields $E_a^{(i)}$ from $g(E_a')$
Initialization: for grains $g^{(i)}, i \in (1, N)$
$s^{(i)} \leftarrow 1$ **or** $s^{(i)} \leftarrow -1$
$h^{(i)} \leftarrow 0$
Simulation: for timestep $[t, t + \Delta t]$ and grains $g^{(i)}, i \in (1, N)$
if $s^{(i)}E(t) < 0$
$\tau^{(i)} \leftarrow \tau_\infty \exp\left[\left(E_a^{(i)}/|E(t)|\right)^\alpha\right]$
$h_{new}^{(i)} \leftarrow h^{(i)} + \Delta t/\tau^{(i)}$
$P^{(i)} \leftarrow 1 - \exp\left[(h^{(i)})^\beta - (h_{new}^{(i)})^\beta\right]$
$h^{(i)} \leftarrow h_{new}^{(i)}$
if Bernoulli$(P^{(i)}) = 1$
Update $s^{(i)}$
$h^{(i)} \leftarrow h_S$
end if
else
Relax $h^{(i)}$ // when $s^{(i)}E(t) \geq 0$
end if

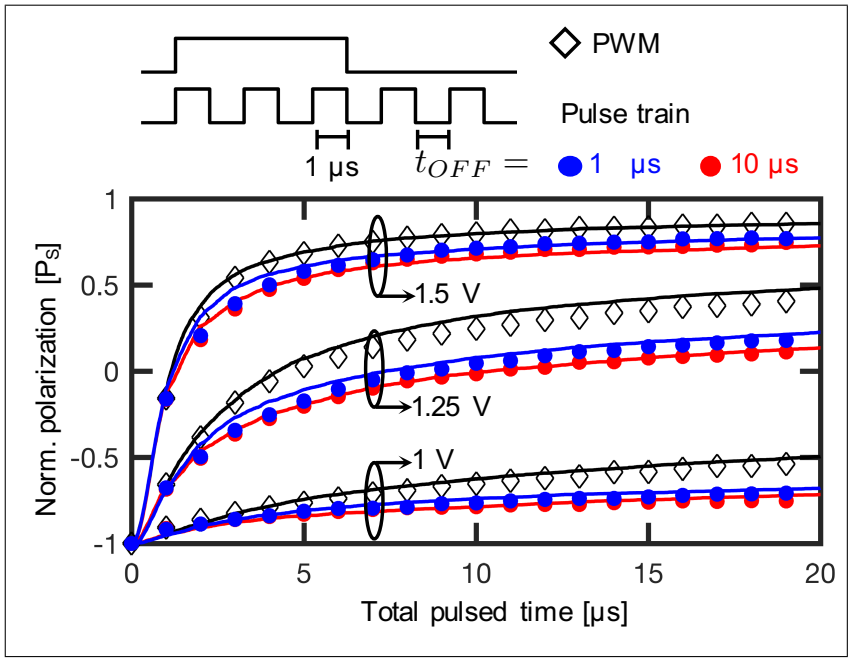

Figure 4.4. Measured (markers) and simulated (solid lines) polarization obtained by pulse width modulation (diamonds) and a train of pulses (dots) with equivalent accumulated time.

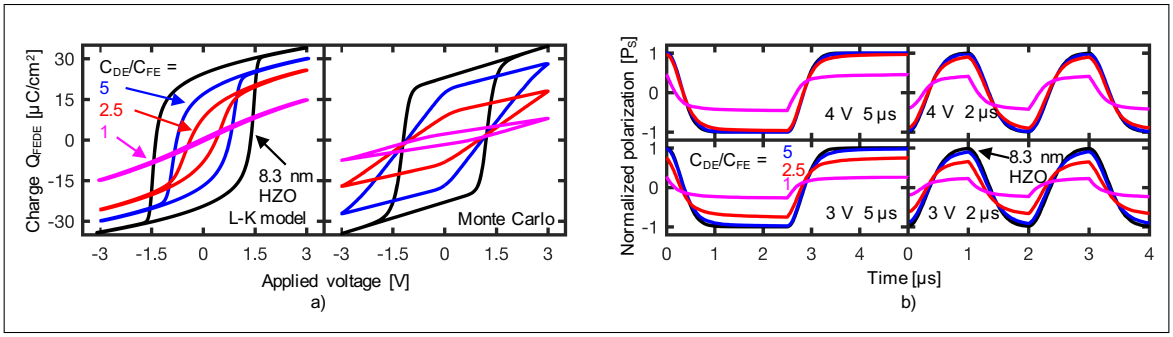

Figure 4.5. (a) Simulation of ferroelectric-dielectric $P-V$ loops with L-K model for single-grain FE and Monte Carlo simulation of polycrystalline FE. (b) Monte Carlo simulation of polarization vs. time of a FE capacitor and FE-DE structures with different dielectric capacitance, programmed with square waveforms of amplitudes 3 and 4 V with 2 μs and 5 μs period.

4.5 Model predictions

Ferroelectric-dielectric (FE-DE) stacks are integral to many proposed FE devices, in both memory and logic [56]. The Monte Carlo simulation framework was applied to model these structures and understand the key differences between polycrystalline FE films and a single-grain FE described by the Landau-Khalatnikov (L-K) model [98]. Figure 4.5(a) shows simulated $P-V$ loops for an 8.3 nm HZO capacitor and a FE-DE stack of an 8.3 nm HZO film and a series dielectric with different capacitance ratios C_{DE}/C_{FE}. The $P-V$ loops are simulated with a triangular waveform of 4 ms period and 3 V amplitude. According to the L-K model, adding a series capacitor results in a decreased switching voltage with an abrupt transition, suggesting that the programming voltage of a FE-DE stack can be lower than that of a FE capacitor. However, this behavior is not observed with a polycrystalline FE [104]. As shown in the Monte Carlo simulation in Fig. 4.5(a), the switching starts at a lower voltage due to the depolarizing field of the DE, but the transition is not abrupt. The depolarizing field of the DE aids switching only when the magnitude of FE polarization is decreasing (i.e. from $\pm P_S$ to 0), but opposes the switching when its magnitude

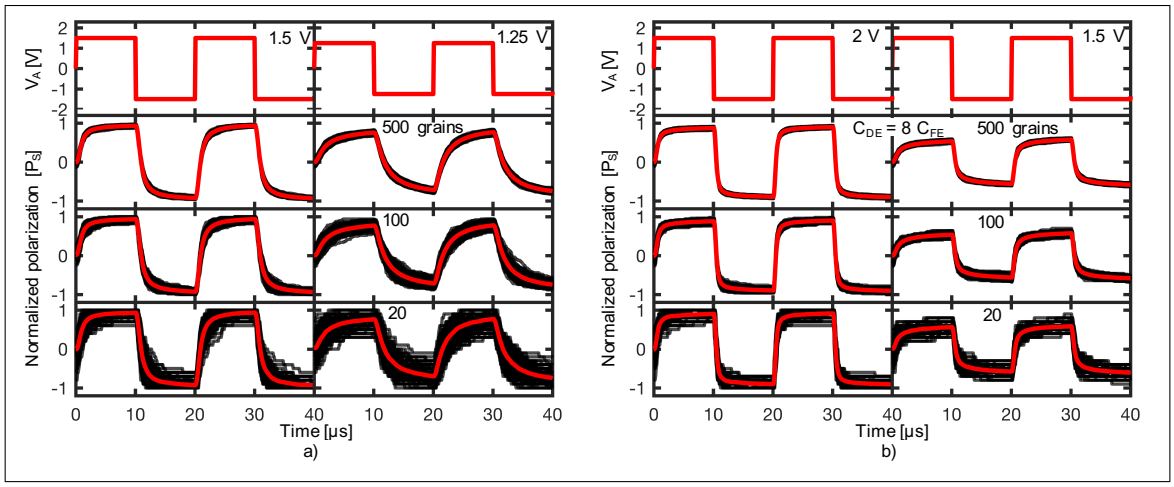

Figure 4.6. Simulated device-to-device variations of 200 devices (black) with 500, 100 and 20 grains for (a) 8 nm-thick FE and (b) FE-DE capacitor with $C_{DE} = 8C_{FE}$. With 20 grains, the memory window of the FE is reduced by 50% with respect to the mean value (red) for a 1.5 V programming voltage, and is completely lost with 1.25 V. With the same number of grains, the FE-DE requires a programming voltage above 1.5 V to a obtain a memory window.

is increasing (i.e. from 0 to $\pm P_S$). Therefore, as the DE capacitance decreases (DE thickness increases), fewer FE grains switch under the same programming conditions. Figure 4.5(b) shows Monte Carlo simulations of the polarization vs. time for the same FE and FE-DE capacitors when a square programming waveform is applied, with 2 and 5 μs period and amplitudes of 3 and 4 V. Irrespective of the pulse duration, the FE-DE starts switching earlier than the FE, but takes a longer time to settle. As the DE capacitance decreases, the switched polarization is reduced due to the effect of the depolarizing field.

Finally, the Monte Carlo modeling approach allows for the investigation of the effects of device-to-device variability due to the grains having a distribution of activation fields. Figure 4.6(a) shows simulated device-to-device variations of an 8.3 nm FE capacitor initialized with 500, 100 and 20 grains, programmed with a square waveform with 20 μs period. For each case, the simulation is repeated 200 times

and plotted with black lines, whereas the red line shows the mean value of all simulations. With a 1.5 V programming amplitude, a $2P_S$ memory window is obtained for 500 grains, which is reduced by approximately 50% for 20 grains. For a 1.25 V programming voltage, the memory window collapses with 20 grains. Figure 4.6(b) shows device-to-device variations of a FE-DE stack with $C_{DE} = 8C_{FE}$ under the same conditions. In this case, a 1.5 V programming voltage produces a memory window close to P_S for 500 grains, which close to 0 with 20 grains. The programming voltage needs to be increased to 2 V to obtain similar memory windows than a FE with 1.5 V programming voltage.

It is important to emphasize that this is a model for polycrystalline FEs in a nucleation-limited regime. A fundamental assumption of nucleation-limited models is that the nucleation time dominates the polarization dynamics, whereas the transient of domain growth within a grain is negligible and is considered to occur instantaneously. Therefore, this model cannot accurately describe the transient behaviour of a single-grain FE.

4.6 Conclusion

A Monte Carlo simulation framework, capable of predicting the dynamic, history-dependent response of a FE under arbitrary input waveforms was developed. After a parameter extraction procedure from polarization reversal measurements, the proposed model was used predict the polarization response of an HZO FE capacitor under different experimental conditions with the same set of parameters. The model was applied to characterize the dynamic response of FE-DE bilayer structures, showing that the response of polycrystalline FE is significantly different than that of single-grain FE. With this proposed model, the reduction in memory window due to device variability was quantified, both for FE capacitors and FE-DE stacks. Finally, an accumulation effect that leads to grain switching was studied and modeled for the

first time by a history parameter. This effect is in agreement with classical nucleation theory, and further theoretical and experimental study is suggested to establish a direct relation between the history-dependent switching probability and the underlying distribution of clusters during the incubation period of domain nucleation. This Monte Carlo simulation framework was presented at the 2018 International Electron Devices Meeting [104].

CHAPTER 5

PARALLEL WEIGHT UPDATE IN RESISTIVE CROSSBAR ARRAYS BY LOCAL MODULATION OF PULSE WIDTH AND FREQUENCY

The parallel weight update operation proposed in [21] relies on the use of stochastic multiplication to compute an approximated weight update locally at the memory element. As described in Chapter 1, the weight update operation is computed as

$$w_{ji} \leftarrow w_{ji} - \eta x_i \delta_j, \tag{5.1}$$

where w_{ji} is the weight that connects the output from neuron i in layer $L-1$ to neuron j in layer L. In a resistive crossbar array implementation, w_{ji} corresponds to the weight that connects row i to column j. For the weight update operation, the inputs x_i (from the rows) and δ_j (from the columns) are translated into stochastic bit streams to perform a stochastic multiplication [21, 26]. When there is a pulse coincidence, the weight is updated by a nominal value Δw, resulting in the update rule

$$\begin{aligned} N &= \sum_{n=1}^{N_{BL}} A_i^n \wedge B_j^n \\ w_{ji} &\leftarrow w_{ji} \pm \Delta w N \end{aligned} \tag{5.2}$$

where N is the number of pulse coincidences, N_{BL} is the length of the stochastic bit stream, A_i^n and B_j^n are the values of the n-th bits, and $\wedge$ is the logic AND operation.

It can be shown that the stochastic multiplication produces, on average, the same result that direct multiplication [21]. However, this approach has significant noise

that introduces errors in the weight updates. Combined with the limited resolution achievable in a resistive weight element, the overall accuracy of DNN training can be significantly impacted. In this chapter, an alternative approximate multiplication is proposed by using local modulation of pulse width and frequency to perform an accurate, parallel weight update. A statistical analysis of both multiplication mechanisms is presented, showing the superior accuracy of the proposed method. A behavioral simulation of DNN training is presented to evaluate the impact of the different multiplication approaches as a function of the number of levels in the resistive element.

5.1 Analysis of stochastic multiplication

Consider the diagram in Fig. 5.1, which represents the stochastic multiplication of two positive factors x and δ. These factors are encoded as bit streams of length N_{BL}, where each bit has a probability of being asserted given by

$$x \longrightarrow P(A_k = 1) = C_A x \leq 1 \tag{5.3}$$

$$\delta \longrightarrow P(B_k = 1) = C_B \delta \leq 1, \tag{5.4}$$

where C_A and C_B are proportionality constants. Given that the probability cannot exceed 1, values of $C_A x$ or $C_B \delta$ larger than 1 are truncated. The number of pulse coincidences N is computed as

$$N = \sum_{k=1}^{N_{BL}} A_k \wedge B_k, \tag{5.5}$$

which is a sequence of N_{BL} Bernoulli trials with probability $P(A_k \wedge B_k) = x\delta C_A C_B$. This results in a binomial distribution $B(N_{BL}, p)$, where $p = x\delta C_A C_B$. The mean

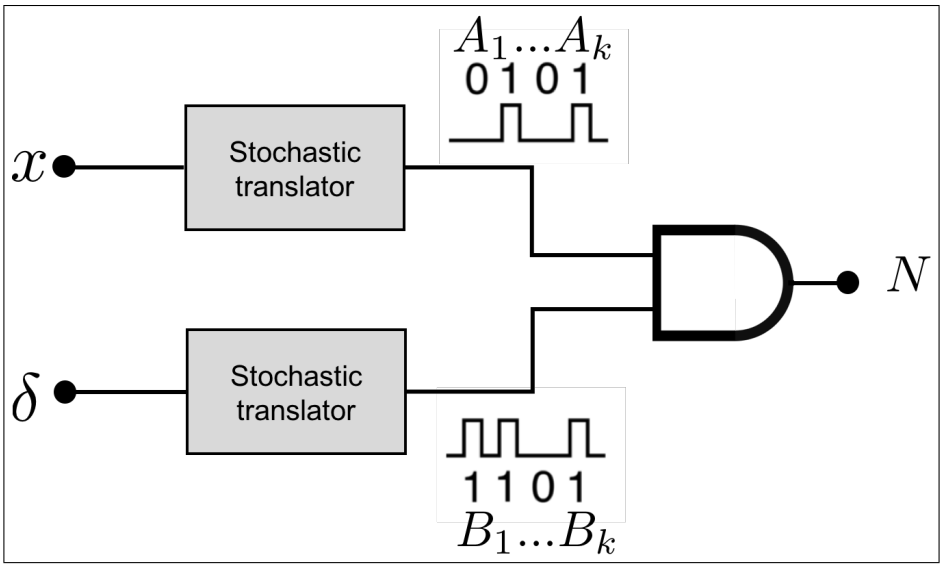

Figure 5.1. Stochastic multiplication.

number of pulse coincidences N is given by

$$\mu_N \quad = \quad N_{BL} C_A C_B x \delta, \tag{5.6}$$

which is proportional to the desired product $\gamma = x \times \delta$. Replacing the mean value in Eq. (5.2),

$$w_{ji} \quad \leftarrow \quad w_{ji} \pm \Delta w N_{BL} C_A C_B x \delta, \tag{5.7}$$

which results in a learning rate $\eta = \Delta w N_{BL} C_A C_B$. The variance of the number of pulse coincidences is given by

$$\sigma_N^2 \quad = \quad N_{BL} C_A C_B x \delta \left(1 - C_A C_B x \delta\right). \tag{5.8}$$

Considering pulses with period T_c and frequency $f_c = 1/T_c$, the time required to perform the stochastic multiplication is $N_{BL} T_c$.

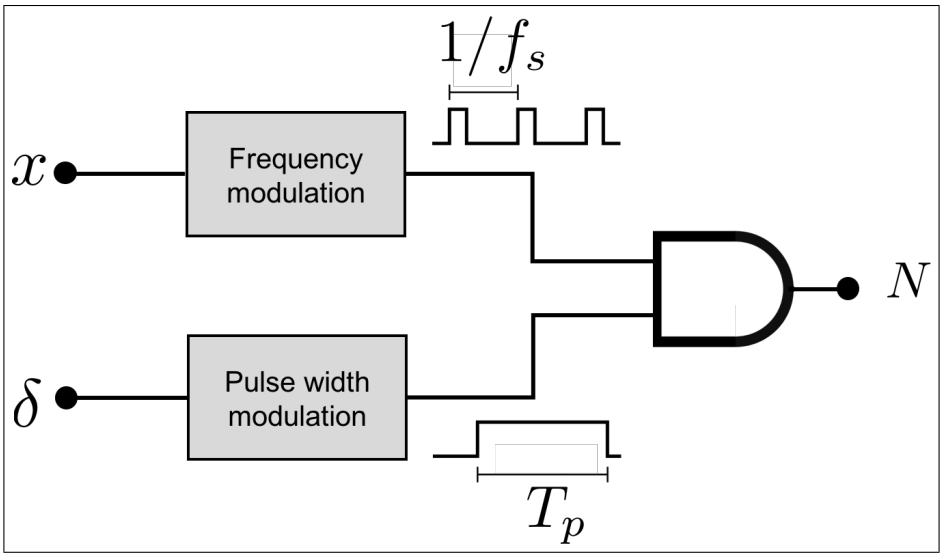

Figure 5.2. Rate-Width multiplication.

5.2 Multiplication by pulse width and frequency modulation

Consider now the diagram in Fig. 5.2, which represents the proposed multiplication for weight update operation. One of the factors is encoded as a frequency modulated signal

$$x \quad \longrightarrow \quad f_s = xC_A f_c, \tag{5.9}$$

where f_c is the maximum frequency and C_A is a proportionality constant. As in the case of stochastic multiplication, values of xC_A larger than 1 are truncated. The other factor is encoded as a pulse-width-modulated signal

$$\delta \quad \longrightarrow \quad T_p = \delta C_B T_{max}, \tag{5.10}$$

where T_{max} is the maximum pulse width and C_B is a proportionality constant. Values of δC_B are also truncated to 1. To impose the same timing constraints than in

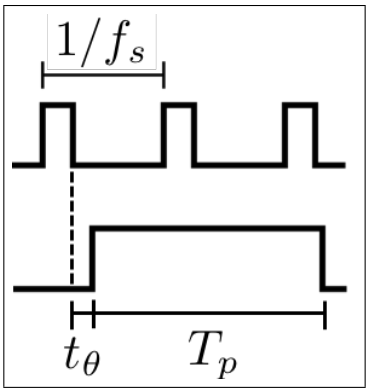

Figure 5.3. Phase difference between the frequency- and width-modulated signals.

stochastic multiplication, $T_{max} = N_{BL}T_c$.

To compute the number of pulse coincidences, consider the diagram in Fig. 5.3. The time between the last falling edge of the frequency-modulated signal, and the rising edge of the width-modulated signal is defined as t_θ. Consider first the case where $t_\theta = 0$. The number of pulse coincidences during T_p is given by

$$N = \lfloor T_p f_s \rfloor = \lfloor x\delta C_A C_B f_c T_{max} \rfloor = \lfloor x\delta C_A C_B N_{BL} \rfloor, \tag{5.11}$$

where $\lfloor \ \rfloor$ is the integer part, or floor, operation. The number of pulse coincidences is proportional to the desired product, rounded down to an integer.

Consider now the general case where t_θ is in the range $[0, 1/f_s)$. The number of pulse coincidences is given by

$$N = \lfloor (T_p + t_\theta) f_s \rfloor = \lfloor x\delta C_A C_B N_{BL} + t_\theta f_s \rfloor = \lfloor x\delta C_A C_B N_{BL} + \theta \rfloor, \tag{5.12}$$

where θ is in the range $[0, 1)$. For $\theta = 0.5$, N is proportional to the desired product rounded to the nearest integer. If the phases of both signals are not synchronized, such that θ is a uniform random variable in the range $[0, 1)$, N will be proportional

the desired product with stochastic rounding [78]. The mean and variance are given by

$$\mu_N = x\delta C_A C_B N_{BL} \tag{5.13}$$

$$\sigma_N^2 = \Delta(1-\Delta), \tag{5.14}$$

where $\Delta = x\delta C_A C_B N_{BL} - \lfloor x\delta C_A C_B N_{BL} \rfloor$. For example, 3.5 will be rounded up with probability 0.5, whereas 3.2 will be rounded up with probability 0.2.

This multiplication mechanism can also be used as a stand-alone architecture for DNN training or inference, as has been recently demonstrated [105, 106]. This approach was also studied to implement a counter-based DNN and is presented in Appendix C.

5.3 Comparison of stochastic and rate-width multiplication

5.3.1 Hardware resources

The stochastic multiplication is typically implemented in the digital domain by a random number generator and a magnitude comparator [108, 109], as shown in Fig. 5.4a). The random number generator is typically implemented by linear feedback shift registers (LFSR) [108, 110], which require a sequence of B registers with feedback taps to generate pseudo-random numbers in the range $[1, 2^B-1]$ [109]. The magnitude comparator is equivalent to a full adder of B bits [111].

The pulse width and frequency modulation can be implemented in several ways, and two specific implementations that allow for a direct comparison with a stochastic translator are presented. The pulse width modulation can be implemented by a simple counter, as shown in Fig. 5.4b). Therefore, the pulse width modulation requires a similar hardware than the LFSR alone. The frequency modulation can be implemented by the direct-digital frequency synthesizer (DDS) proposed in [107], shown

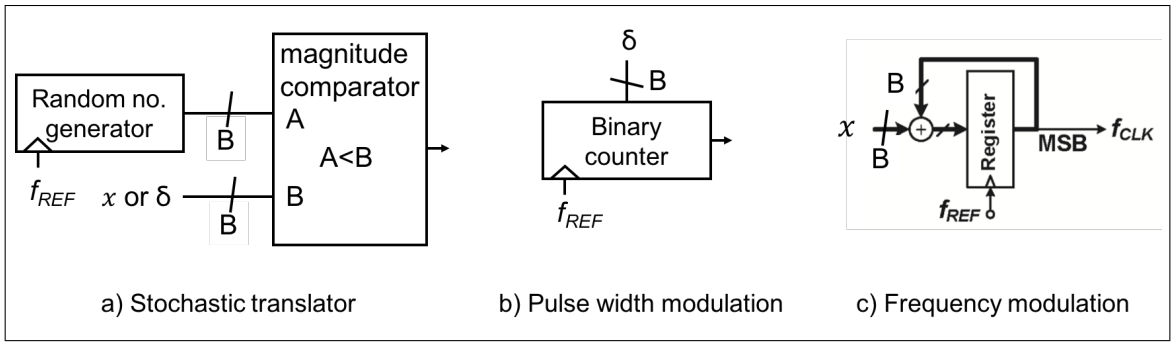

Figure 5.4. Hardware implementations of a) stochastic translators, b) pulse width modulation and c) frequency modulation. Figure reprinted from [107]

in Fig. 5.4c). The DDS is comprised of a full adder and a register of B bits. The DDS output (the most significant bit of the register) is given by $f_{clk} = f_{REF}x/2^B$ for x in the range $[0, 2^B - 1]$ [107].

Therefore, the hardware required to implement both pulse width and frequency modulation is comparable to that of a single stochastic translator. Furthermore, pulse-width-modulated signals are also used in the forward and backpropagation cycles in a resistive crossbar array [24], so no additional hardware is actually required for its implementation.

5.3.2 Multiplication accuracy

To visually compare the different multiplication methods, a simulation was performed by sampling 10^5 values for x and δ uniformly between 0 and 2. Values higher than 1 are truncated to 1. The proportionality constants C_A and C_B are set to 1. The number of pulse coincidences are computed for rate-width multiplication with aligned phases ($t_\theta = 0$) and unsynchronized phases ($\theta\ U(0, 1)$). For stochastic multiplication, the number of pulse coincidences are drawn from a Binomial distribution with N_{BL} trials and probability $x\delta$. Simulations were repeated for $N_{BL} = 10$ and $N_{BL} = 20$. It is shown that rate-width multiplication with aligned phases rounds down, whereas

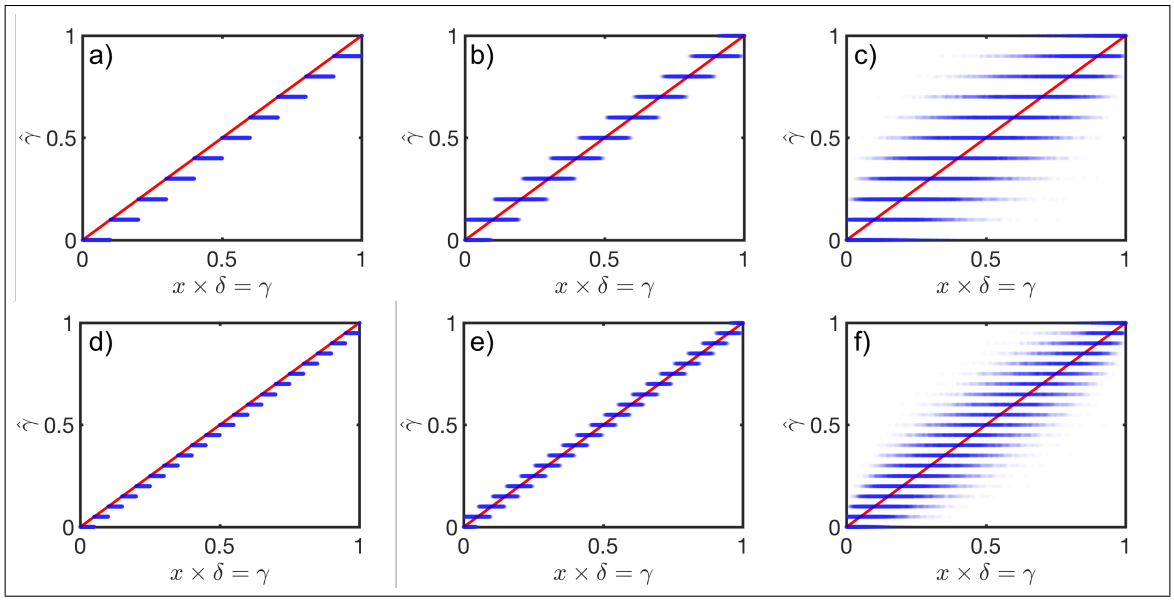

Figure 5.5. Comparison of approximated multiplication methods for $x \times \delta = \gamma$ with $C_A = C_B = 1$. The $y-$axis shows $\hat{\gamma} = N/N_{BL}$. Solid red lines show the ideal case $\gamma = \hat{\gamma}$. Cases with $N_{BL} = 10$ for a) rate-width multiplication with $t_\theta = 0$ (b) rate-width multiplication with unsynchronized phases (c) stochastic multiplication. Cases with $N_{BL} = 20$ for d) rate-width multiplication with $t_\theta = 0$ (e) rate-width multiplication with unsynchronized phases (f) stochastic multiplication.

the case with random phases produces stochastic rounding. Stochastic multiplication has a visibly larger variability, with its maximum at $\gamma = 0.5$.

5.4 Performance evaluation in a resistive crossbar array

Given that neural networks can be tolerant to noise, it is not guaranteed that a lower-variance multiplication will improve the accuracy obtained after training. To evaluate the impact of the proposed weight update scheme, a behavioral simulation of neural network training was implemented in MATLAB. The simulation was implemented using a fully-connected network with the MNIST dataset of handwritten digits [68]. The network has 784 input neurons, hidden layers with size 256 and 128, and an output layer with 10 elements for labels from 0 to 9. ReLU activations were

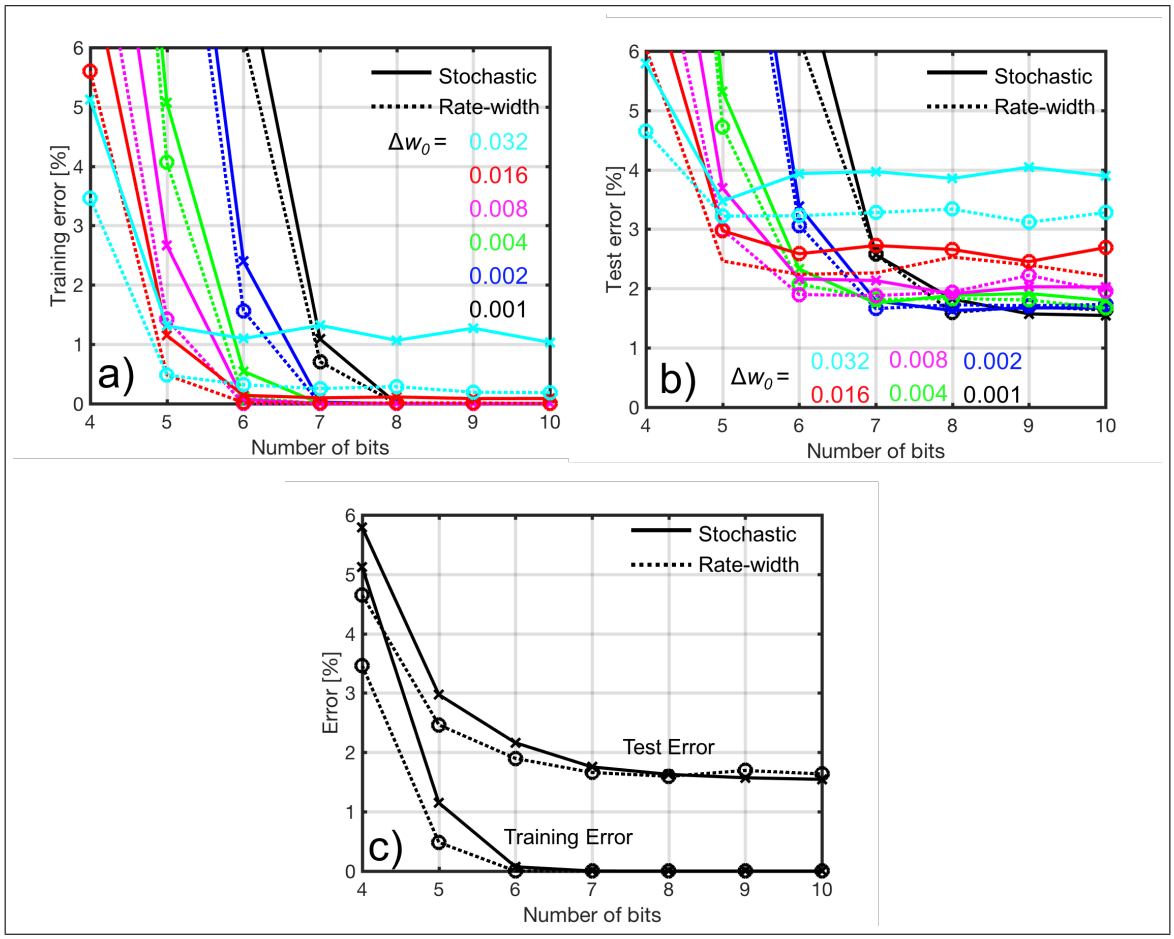

Figure 5.6. DNN training with stochastic and rate-width multiplication. a) Training for different values of the minimum weight update Δw_o as a function of the number of bits. b) Test error for different values of the minimum weight update Δw_o as a function of the number of bits. c) Training and test error obtained by selecting the best value of Δw_o at each resolution B

used for hidden layers and softmax activation at the output.

The weight update is performed according to Eq. (5.2) for stochastic multiplication and rate-width multiplication with stochastic rounding, using $N_{BL} = 10$ for both cases. It is assumed that the weight elements increase/decrease linearly with a minimum update Δw_o and saturate at values $\pm\Delta w 2^{B-1}$, where 2^B represents the number levels. Figure 5.6 shows the training and test errors obtained after 30 training cycles for different values of Δw_o as a function of the number of bits B. Figure 5.6c) shows the best training and test errors obtained by selecting the best value of Δw_o at each resolution B. It is shown that a DNN trained with rate-width achieves lower train and test error than the equivalent network trained with stochastic multiplication. The difference becomes larger as the resolution of the weights is reduced. Only for resolutions above 7 bits, the test error is slightly lower for stochastic multiplication. This effect is caused by the regularization effect of noise during training [112], but is only useful when the training accuracy is not limited by the resolution of its weights.

5.5 Conclusion

An accurate scheme for parallel weight update in resistive crossbar arrays is proposed and evaluated. By using pulse width- and frequency-modulated signals, the value of resistive elements in a crossbar array can be updated in parallel with higher accuracy than existing techniques based on stochastic multiplication. This scheme produces an unbiased multiplication with stochastic rounding, which is optimal for training neural networks with limited resolution. Furthermore, the pulse width and frequency modulation can be implemented with fewer hardware resources than stochastic translators. It is shown that a DNN trained with rate-width multiplication achieves lower train and test error than the equivalent network trained with stochastic multiplication.

CHAPTER 6

EFFICIENT MAPPING OF NEURAL NETWORK MODELS TO RESISTIVE CROSSBAR ARRAYS WITH LIMITED WEIGHT RESOLUTION

DNN accelerators based on multiplication performed in the analog domain using resistive elements can potentially reduce the time and energy consumption of DNN training by orders of magnitude [21]. However, these implementations naturally perform multiplication with nonnegative weights, given that the resistance can only have positive values. The DNN simulations presented in Chapters 2 and 5 have been performed with signed weight elements, which implicitly assumes that there is a mechanism to map the nonnegative conductance to a signed value. In this chapter, the mapping of DNN models to hardware with nonnegative weights is studied. To analyze different mapping schemes, a general vector-matrix multiplication is decomposed into a vector-matrix multiplication with nonnegative elements performed in a crossbar array, followed by a limited set of addition and subtraction operations described by a connection matrix. The mathematical conditions for the existence of such decomposition are derived and applied to fully connected and convolutional layers. Based on this analysis, an efficient mapping scheme is designed, which mitigates the effect of weight nonlinearity and limited resolution without additional overhead.

6.1 Prior approaches to map DNN model to resistive crossbar arrays

There are two approaches typically used to map a DNN model to a reistive crossbar array. The first case, shown in Fig. 6.1a), uses two resistive elements to represent each weight [18, 22, 28, 59, 114, 115]. With this approach, the output from two

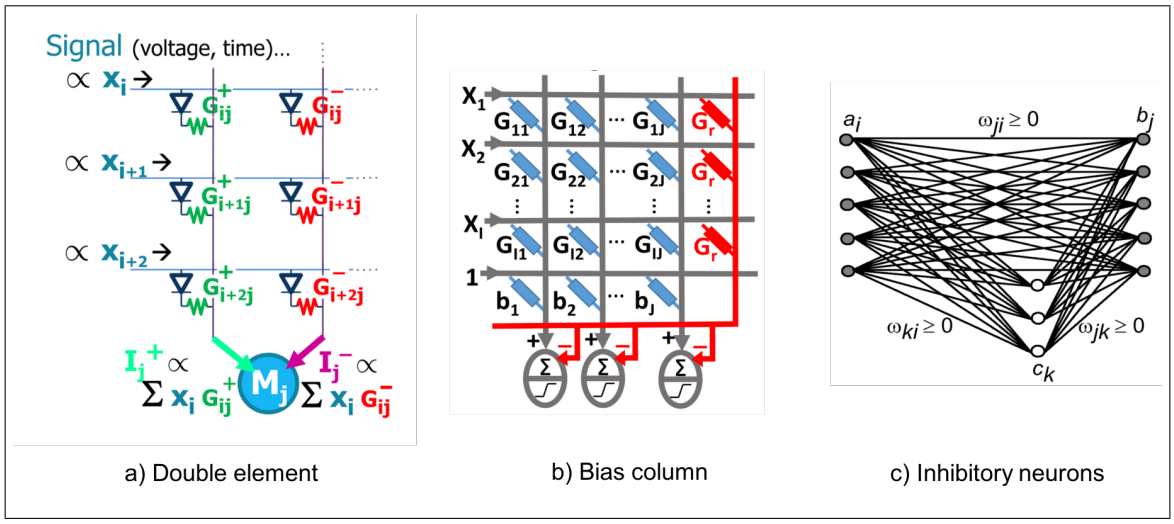

Figure 6.1. Prior approaches to map DNN model to crossbar array. a) Using two resistive elements to represent each weight. Figure reprinted from [22]. b) Using a single column of resistive elements as reference. Figure reprinted from [59]. c) General approach to implement a DNN with nonnegative weights by using inhibitory neurons. Figure reprinted from [113].

columns of the crossbar array are subtracted to compute the signed weighted sum of a single neuron, which results in roughly 2× area and power consumption of the crossbar array. The second case, shown in Fig. 6.1b), uses a single bias column that is fixed in the middle value of the resistive element [57, 59, 116]. The output from this column is subtracted from to the output of all other columns to compute the signed weighted sum of each neuron. In both cases, a vector-matrix multiplication with nonnegative weights is performed in the crossbar array, followed by a combination of the outputs from its columns to obtain an equivalent vector-matrix multiplication with signed weights. In both cases it can be shown that an equivalent signed multiplication is obtained, and it is straightforward to map a DNN model to its hardware equivalent.

A more general approach to implement a DNN with nonnegative weights is the use of inhibitory neurons [113], as shown in Fig. 6.1c). In this case, an additional set of neurons are computed from the input, which are then subtracted from the output neurons to produce a signed weighted sum. This general approach provides a wider range of combinations that can produce an equivalent signed multiplication, although it is not straightforward to map a DNN model to its hardware equivalent. More importantly, the double element and bias column approaches can be seen as particular cases of inhibitory neurons.

This can be extended to decompose a signed vector-matrix multiplication into a vector-matrix multiplication with nonnegative elements followed by a connection matrix, which represents the combination of the outputs from columns of the crossbar array. With this formulation, more efficient mapping schemes can be explored and evaluated. The analysis is presented first for the case of matrix multiplication in fully connected layers, and then extended to convolutional layers.

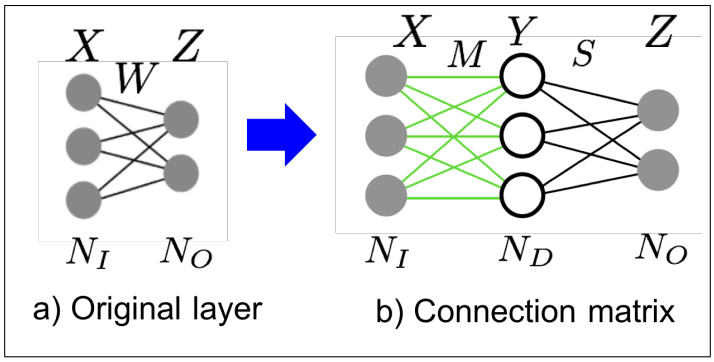

Figure 6.2. Diagram of connection matrix decomposition. The original fully-connected layer is decomposed as a sequence of a nonnegative layer with weights M and N_D elements, followed by a connection matrix S.

6.2 Connection matrix decomposition for fully-connected layers in a resistive cross-bar array

A fully connected layer with N_O elements receives an input X with dimensions $[N_I, B_S]$, where B_S is the batch size and N_I is the number of elements from the previous layer, as shown in Fig. 6.2a). The output of the layer is computed as

$$Z = f\left(WX + b\right), \tag{6.1}$$

where Z has dimension $[N_O, B_S]$, W is the weight matrix with dimensions $[N_O, N_I]$, b is the bias vector, and $f()$ represents the activation function of the layer.

To perform the multiplication with nonnegative weights only, the weight matrix W is decomposed into a matrix with nonnegative elements M, followed by a connection matrix S, as shown in Fig. 6.2b). For this purpose, a dummy layer Y with N_D elements is defined, so that

$$\begin{aligned} Y &= MX \\ Z &= f\left(SY + b\right), \end{aligned} \tag{6.2}$$

where M is a matrix of nonnegative elements with dimensions $[N_D, N_I]$ and S is a connection matrix of dimensions $[N_O, N_D]$.

From the algorithm perspective, it is necessary to find sufficient conditions for S that guarantee that the decomposition exists, such that a model that was trained with signed weights can be mapped to an implementation in a resistive crossbar array. More importantly, this guarantees that a model trained with the connection matrix decomposition has the equivalent capacity of the original model. From the hardware perspective, it is desired that the connection matrix is "simple", so it does not introduce significant overhead. Furthermore, it is desirable to minimize the number of weight elements in the resistive crossbar array required to represent the equivalent signed matrix, which is determined by the size of the dummy layer.

6.2.1 Sufficient conditions for existence

It is sought to find sufficient conditions for a matrix S, such that any arbitrary matrix W can be decomposed as

$$SM = W, \quad M \geq 0, \tag{6.3}$$

where $M \geq 0$ means that all elements of M are nonnegative. Considering W with dimensions $N_O \times N_I$, S has dimensions $N_O \times N_D$ and M has dimensions $N_D \times N_I$. This problem can be formulated independently for each column of W and M, expressed as:

$$Sm_k = w_k, \quad m \geq 0 \tag{6.4}$$

where m_k and w_k are the k-est columns of M and W with $k = 1..N_I$.

A necessary condition for the existence of a solution to Eq. (6.4) is that w_k is in

the column space of S. This is true for any arbitrary w_k if and only if

$$\text{rank}(S) = N_O. \tag{6.5}$$

This implies that the N_O original neurons are computed from linearly independent combinations of the N_D neurons in the dummy layer. When $N_D = N_O$, there is a unique solution to the system $Sm_k = w_k$, but in general it is not guaranteed to be nonnegative.

In addition to the necessary condition for the existence of a solution, a sufficient condition for the existence of a nonnegative solution is that there exists a vector x_h in the null space of S with strictly positive elements. This condition guarantees that any particular solution x_p to the system $Sm_k = w_k$ can be shifted as $x'_p = x_p + \alpha x_h$ to be nonnegative. The conditions are summarized as:

$$\begin{aligned} &1. \quad \text{rank}(S) = N_O \\ &2. \quad \text{There exists } \; x_h > 0 \; \in \mathbb{R}^{N_D}, \text{ such that } \; Sx_h = 0. \end{aligned} \tag{6.6}$$

If these conditions are met, any fully-connected layer with weights W can be mapped to a connection matrix layer. Furthermore, a fully-connected can be trained as a connection matrix layer in hardware with nonnegative weights. Once the layer has been trained, the equivalent weight matrix is simply obtained as

$$W = SM. \tag{6.7}$$

Finally, note that $N_D = \text{rank}(S) + \text{nullity}(S)$, and $\text{nullity}(S) \geq 1$ given that there is at least one element (x_h) in the null space of S. Therefore, the dummy layer N_D has at least one extra element than the original layer N_O, which results in one extra column in the crossbar array. A particular case that satisfies condition 2 is $x_h = \mathbf{1}$,

which implies that the elements of the rows of S add up to 0. Therefore, a solution is guaranteed if the original neurons are computed by a balanced combination of inhibitory (negative) and excitatory (positive) inputs from the dummy layer.

6.2.2 Implementation in a crossbar array

The connection matrix defines how the neurons from the dummy layer are combined as an input to the original neurons, which can be seen as a generalization of previous architectures, as depicted in Fig. 6.3. The original layer (Fig. 6.3a) can be implemented by using two weight elements to compute each of the original weights, represented by a connection matrix as shown in Fig. 6.3b, where $N_D = 2N_O$. As an alternative, the bias column approach can also be mapped to a connection matrix as shown in Fig. 6.3c) where $N_D = N_O + 1$. In both cases, the connection matrix represents only addition and subtraction operations (i.e. with only -1, 0 and $+1$ elements), so condition 2 is satisfied given that every neuron receives an even number of positive and negative inputs.

The crossbar array implementation for both approaches in shown in Fig. 6.4. The outputs of the columns are combined by addition operations (top) described by the connection matrix S (bottom). Note that the connection matrix defines a small set of subtraction operations, whereas most of its elements are zeros. Although the dummy layer Y is defined for the purpose of the analysis, it is not implemented in hardware as an additional crossbar array, given that the elements from the columns are combined by simple adders after the crossbar array. Moreover, the intermediate tensor Y needs not be stored after the forward propagation cycle, given that there is no activation in the dummy layer and the connection matrix is not updated during training. During back-propagation, the gradients are simply propagated to the corresponding columns of the crossbar array.

It is straightforward to see that both the double element and bias column ap-

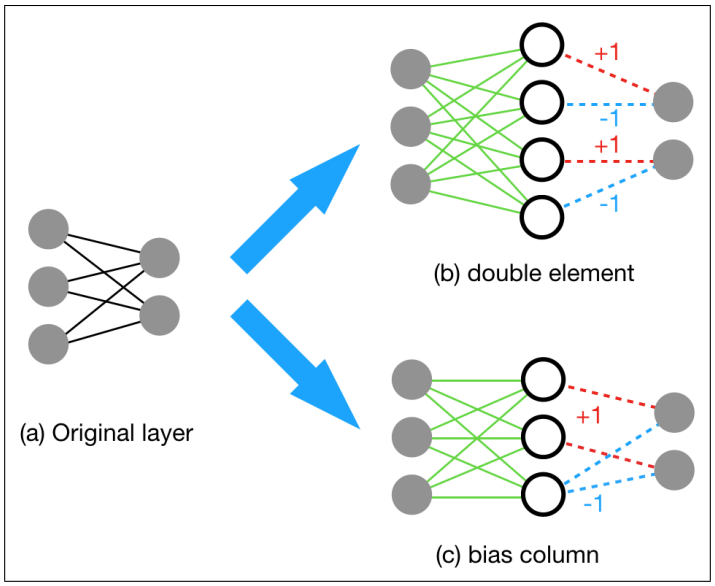

Figure 6.3. Diagram of connection matrix representation of prior approaches. a) Original layer. b) Double element implementation represented by a connection matrix with $N_D = 2N_O$. c) Bias column implementation represented by a connection matrix with $N_D = N_O + 1$.

proaches satisfy the conditions stated in Eq. (6.6) and a single addition is computed for each of the original neurons. However, the double element approach has $N_D = 2N_O$ columns, whereas the bias column has the minimum number of columns, $N_D = N_O + 1$. Therefore, in terms of the number of resistive elements and the overhead of the connection matrix, the bias column approach uses the minimum hardware resources. However, if the conductance values are limited in the range $M = [0, G_{max}]$, the weights of the bias column will be in the range $W = [-G_{max}/2, G_{max}/2]$, given that the bias column is fixed to $G_{max}/2$. For the double element approach, the range of weights will be $[-G_{max}, G_{max}]$ at the expense of using twice as many weight elements. In addition, computing the weights by the difference of adjacent elements is expected to result in a better tolerance to device nonlinearity and process variations.

A connection matrix is proposed based on these considerations, which computes the original neurons as a combination of adjacent columns with alternating signs as

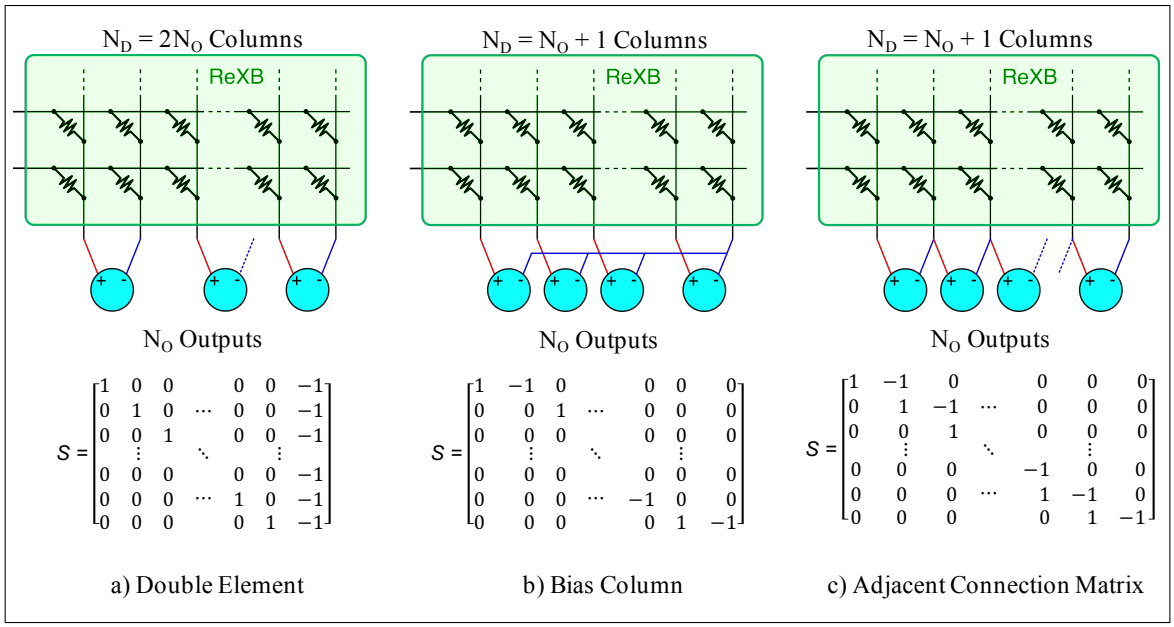

Figure 6.4. Crossbar implementation of a) double element, b) bias column and c) adjacent connection matrix.

shown in Fig. 6.4c), which will be referred to as Adjacent Connection Matrix (ACM). As opposed to the double element case, the dummy layer has $N_D = N_O + 1$ elements to use the minimum amount of resistive elements, so each column in the crossbar arrays is used for more than one of the original neurons. The hardware complexity due to the subtractions is the same as in the bias column case, with the difference that adjacent columns are subtracted instead of having a single reference column subtracted from all other columns in the array. As will be shown in Section 6.4, the three cases produce equivalent results when trained without any constraint in the resolution of the resistive elements, given that all cases satisfy the conditions in Eq. (6.6). The advantages of the proposed approach will become apparent in Section 6.5, where limited resolution and nonlinearity restrictions are applied during training.

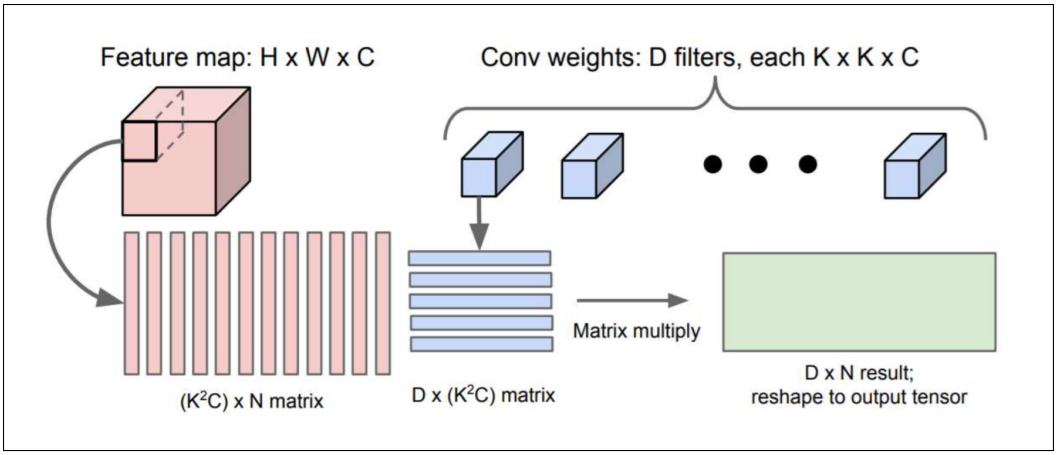

Figure 6.5. Convolutional layer implemented as a sequence of matrix multiplications. Each kernel is mapped to a row vector of dimensions K^2C, which results in a matrix with D rows of kernels. The input image is partitioned into slices of dimensions $K \times K \times C$, which correspond to one step of the spatial convolution. The $N = H \times W$ slices are transformed into a vector of size K^2C and sequentially multiplied to the matrix of kernels. The output is finally reshaped to $H \times W \times D$. Figure reprinted from [4]

6.3 Connection Matrix decomposition applied to convolutional layers

In convolutional layers, typically used for image recognition, a set of filters are applied to an input image to capture spatial information that is otherwise lost in fully-connected layers. A brief introduction of convolutional neural networks can be found in [3, 10], whereas a detailed treatment can be found in [6]. Consider an image of dimensions $H \times W \times C$, where H is the height, W is the width and C is the number of channels. A filter, also called kernel, of dimensions $K \times K \times C$ is applied by a 2-dimensional convolution across the height and width of the image, producing an output *feature map* of dimensions $H \times W \times 1$ (it is assumed that padding is used at the edges of the image for simplicity). By applying D filters, an output of dimensions $H \times W \times D$ is obtained. The parameters of the layer correspond to the $K \times K \times C \times D$ elements of the filters, in addition to the biases ($H \times W \times D$, one for each output neuron).

A convolutional layer with D kernels can be implemented as a sequence of matrix multiplications as shown in Fig. 6.5, with a procedure called *im2col* [4]. Each kernel is mapped to a row vector of dimensions K^2C, which results in a matrix with D rows of kernels. The input image is partitioned into slices of dimensions $K \times K \times C$, which correspond to one step of the spatial convolution. The $N = H \times W$ slices are transformed into vectors of size K^2C and sequentially multiplied by the matrix of kernels. The output is finally reshaped to $H \times W \times D$.

Once the convolution has been mapped to a matrix multiplication, it can be implemented and trained in a crossbar array [24]. Furthermore, the conditions in Eq. (6.6) can be directly applied to map the vector-matrix multiplications obtained with *im2col* approach to a multiplication with nonnegative elements followed by a connection matrix, as shown in Fig. 6.6. The matrix of kernels is mapped to a crossbar array with K^2C rows and $D + 1$ columns followed by a connection matrix to perform an equivalent matrix multiplication. Therefore, the hardware implementation is equivalent to that of a fully-connected layer. Finally, this sequence of operations is equivalent to applying $D + 1$ kernels with nonnegative weights, followed by D filters of size $1 \times 1 \times D + 1$. Each of these filters is defined by a row of the connection matrix. This abstraction will be useful to simulate the behavior of the connection matrix in convolutional layers.

6.4 Experimental validation with MNIST and CIFAR-10 datasets

To evaluate the proposed connection matrix and validate the theoretical analysis, a set of simulations were implemented using the Keras [117] open-source framework. Different networks were tested with the MNIST [68] and CIFAR-10 [118] datasets for image recognition. The networks that were evaluated are described in Table 6.1. Max pooling and dropout [112] are standard layers in convolutional networks, but their details are not essential for the implementation of the connection matrix. An

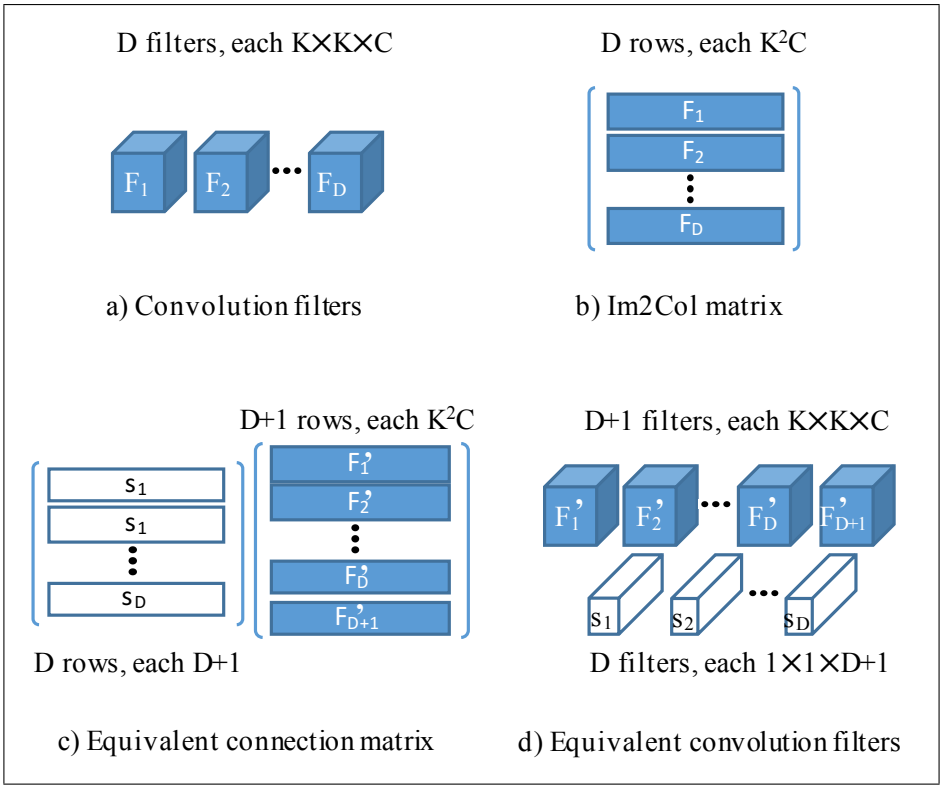

Figure 6.6. Connetion matrix decomposition applied to a convolutional layer. a) Original convolutional filters. b) Convolutional filters mapped to a weigh matrix with the *im2col* approach to implement convolution as vector-matrix multiplication. c) Weight matrix implemented with connection matrix decomposition and nonnegative filters F'. d) Mapped back to convolutional domain, the connection matrix decomposition is equivalent to $D+1$ nonnegative filters followed by $1 \times 1 \times D+1$ convolutions defined by the rows of the connection matrix.

TABLE 6.1

STRUCTURE OF DNNS USED FOR VALIDATION

Layer	Size	Activation
Input	784	
Dense_1	256	ReLu
Dense_2	128	ReLu
Dense_3	10	Softmax

a) MNIST fully-connected

Layer	Size	Activation
Input	(28,28) x 1	
Conv_1	(3,3) x 32	ReLu
Conv_2	(3,3) x 64	ReLu
MaxPool_1	(2,2)	
Dropout_1		
Dense_1	128	ReLu
Dropout_2		
Dense_2	10	Softmax

b) MNIST convolutional

Layer	Size	Activation
Input	(32,32) x 3	
Conv_1	(3,3) x 32	ReLu
Conv_2	(3,3) x 32	ReLu
MaxPool_1	(2,2)	
Dropout_1		
Conv_3	(3,3) x 64	ReLu
Conv_4	(3,3) x 64	ReLu
MaxPool_2	(2,2)	
Dropout_2		
Dense_1	512	ReLu
Dropout_3		
Dense_2	10	Softmax

c) CIFAR-10 convolutional

interested reader is referred to [6].

Keras is a modular, high-level library for Tensorflow [119], Google's deep learning platform. Keras has a set of defined layers, among them convolutional and fully connected layers. In simple terms, a DNN model in Keras is defined by specifying an input size, a sequence of layers and a loss function associated with the output layer. The training is handled by an optimizer, which computes the error from labelled training data, backpropagates the gradients and updates the weights. The parameters in a Keras layer are associated with an initializer object, which specifies the initial conditions of the parameters. In addition, a constraint object can be associated with the parameters.

The connection matrix is implemented in Keras as shown in Fig. 6.7. Consider a layer from a model with signed weights, which will be referred to as baseline.

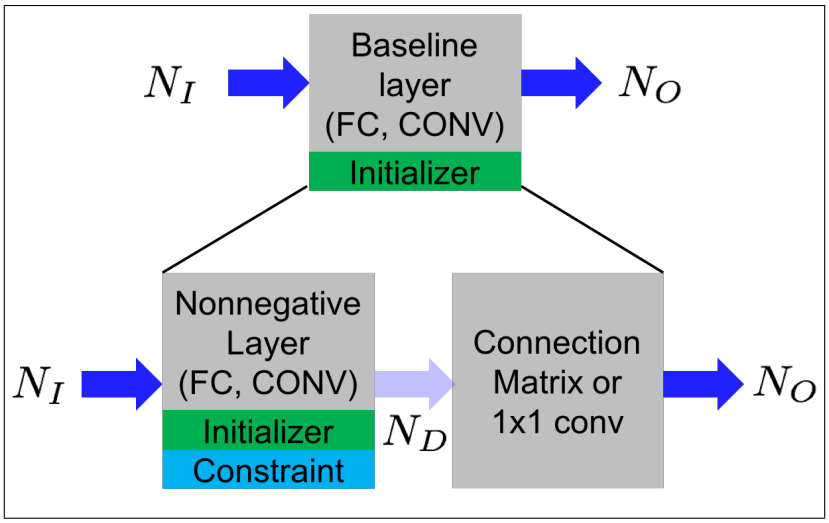

Figure 6.7. Keras implementation of connection matrix. A layer from the baseline model is implemented by a nonnegative layer followed by a connection matrix.

This layer can be either fully-connected or convolutional, and has an initialization object but no constraints. The connection matrix equivalent is defined as two layers. The first layer is of the same type, but with N_D outputs and nonnegative weights implemented by a constraint object. The second layer has non-trainable weights corresponding to the connection matrix for a fully-connected layer, or a $1 \times 1 \times N_D$ convolution for a convolutional layer. The biases and activations are applied at the second layer. Unless otherwise specified, the models were trained with stochastic gradient descent with a batch size of 128 images.

In this section, the networks are trained with floating point precision to validate that the different approaches produce equivalent results. The effect of limited weight resolution and nonliniearity will be analyzed in the following section. Four cases are evaluated:

1. Baseline model: original network trained with signed weights.
2. Double element (DE): network implemented with the connection matrix defined in Fig. 6.4a).

3. Bias column (BC): network implemented with the connection matrix defined in Fig. 6.4b).
4. Adjacent connection matrix (ACM): network implemented with the proposed connection matrix, defined in Fig. 6.4c).

Unless otherwise specified, the weights are initialized by sampling from a normal distribution with zero mean according to He normal initialization [120]. For the nonnegative layer, the weights are initialized with the same variance, but centered around a positive value, so the left-side of the distribution is not truncated. When the weight resolution is restricted, the weights are initialized around the middle value of their range. For the simulations presented in this section, the weight resolution is not limited, so the weights are initialized around 10, an arbitrary value that has no effect on the simulation results.

Figure 6.8 shows the training and test errors of the fully-connected network in Table 6.1a) for two different learning rates. Figure 6.8 shows the training and test errors of the convolutional network in Table 6.1b) with and without dropout. It is shown that all models trained with the connection matrix decomposition are equivalent to the baseline model trained with signed weights.

The network in Table 6.1c) was trained with the CIFAR-10 dataset for the baseline and ACM models under different conditions. Figure 6.10a) shows the training without dropout, whereas Figure 6.10b) shows the training with dropout rates of 0.25, 0.25 and 0.5 for dropout_1, dropout_2 and dropout_3. Figure 6.10c) shows the training results with data augmentation [121], where the training images are randomly shifted horizontally and vertically by $\pm$ 10% to improve generalization. Finally, Figure 6.10d) shows the training results with both data augmentation and dropout. These experiments show that a network trained with the connection matrix decomposition produces equivalent results than the baseline models. Furthermore, the training and test errors follow similar trajectories as a function of the number of epochs. This is consistently observed for different networks and training conditions

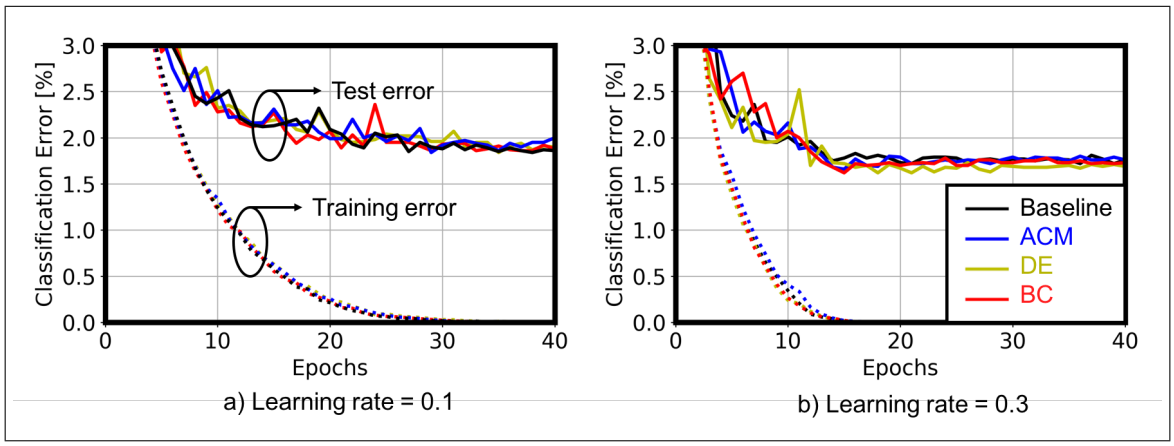

Figure 6.8. Fully-connected network for MNIST dataset trained with signed weights (baseline), adjacent connection matrix (ACM), double element (DE) and bias column (BC) for learning rates of a) 0.1 and b) 0.3. Training error is shown with dashed lines and test error is shown in solid lines.

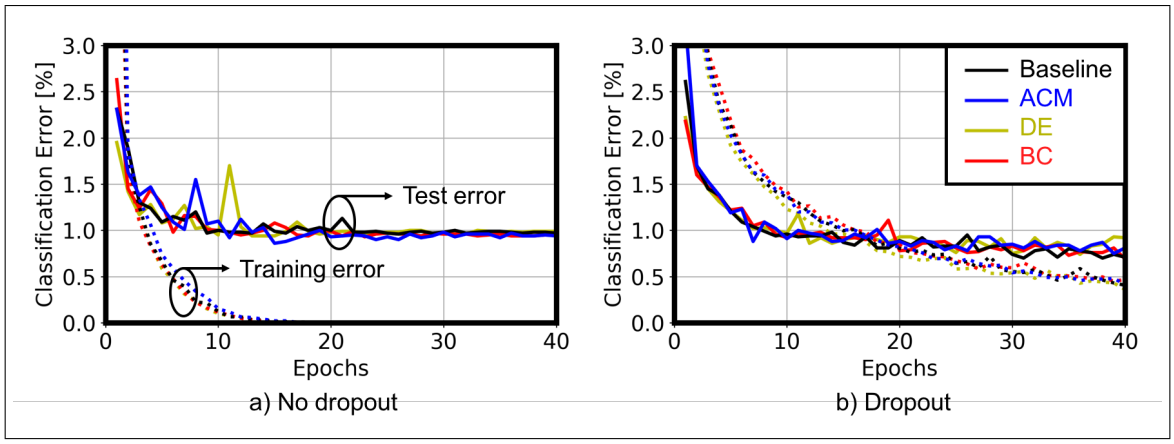

Figure 6.9. Convolutional network for MNIST dataset trained with signed weights (baseline), adjacent connection matrix (ACM), double element (DE) and bias column (BC). Training was performed with a 0.1 learning rate and a) no dropout or b) dropout with rates of 0.25 in dropout_1 and 0.5 in dropout_2. Training error is shown with dashed lines and test error is shown in solid lines.

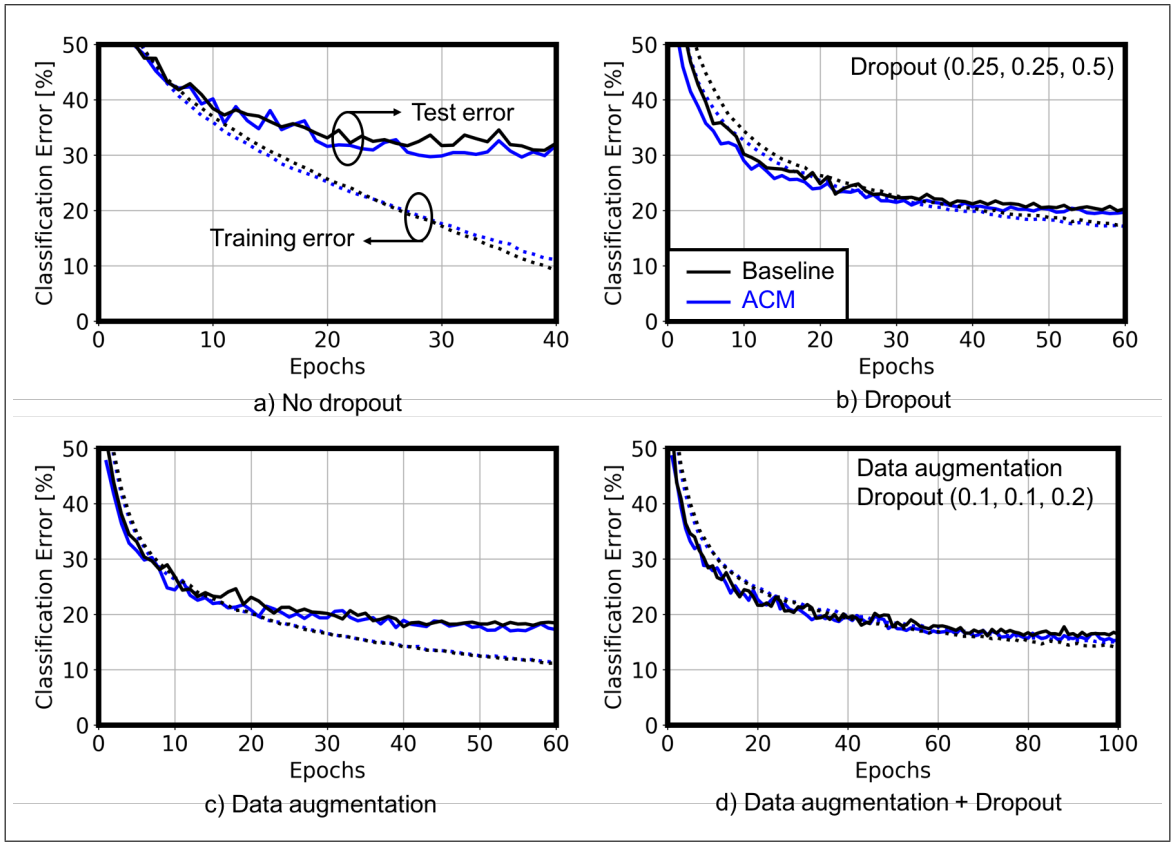

Figure 6.10. Convolutional network for CIFAR-10 dataset trained with signed weights (baseline) and adjacent connection matrix (ACM). Training was performed with a) no dropout, b) dropout with rates of 0.25, 0.25 and 0.5, c) data augmentation and d) data augmentation and dropout with rates of 0.1, 0.1 and 0.2. Training error is shown with dashed lines and test error is shown in solid lines.

for two datasets, which validates the proposed connection matrix decomposition and the conditions derived in Eq. (6.6).

6.5 Evaluation with limited weight resolution and nonlinearity

After validating the connection matrix approach and verifying that it produces equivalent results as the baseline model, the effect of limited weight precision for the different mapping schemes is studied. The simulations were implemented in Keras by a custom constraint object that receives the gradients computed by the optimizer Δ_{IDEAL}, and performs a constrained weight update operation. Two quantized weight updates were evaluated. The first case is a quantized weight element with linear and symmetric weight updates, defined by the expressions

$$\Delta_Q \leftarrow w_0 \times \text{Round}\left\{\Delta_{IDEAL}/w_0\right\} \tag{6.8}$$

$$w \leftarrow w + \Delta_Q \tag{6.9}$$

$$w \leftarrow \text{Clip}\left\{w : \min = 0, \max = w_0 2^B\right\}, \tag{6.10}$$

where Δ_Q is the quantized gradient, w_0 is the minimum weight step and B is the number of bits. The Round{} function produces either rounding to the nearest integer or stochastic rounding. The Clip{} function limits the weight value to the specified min and max values. The second case is a nonlinear weight element defined by the weight update operations

$$\Delta_Q \leftarrow w_0 \times \text{Round}\left\{\Delta_{IDEAL}/w_0\right\} \tag{6.11}$$

$$w \leftarrow w + \Delta_Q \left(1 - \frac{w}{w_0 2^B}\right) \tag{6.12}$$

$$w \leftarrow \text{Clip}\left\{w : \min = 0, \max = w_0 2^B\right\}, \tag{6.13}$$

which produces a nonlinear weight update with symmetric increase/decrease steps, similar to the requirements outlined in [122] and some RPU demonstrations [54, 63, 123].

Figure 6.11 and Figure 6.12 show the training results for the fully connected network in Table 6.1a) and the convolutional network in Table 6.1b), respectively. The test and training errors are shown as a function of the number of bits B for the quantized and nonlinear weight updates. For the quantized training without stochastic rounding, the error of the double element case is equivalent to the error of the bias column shifted by 1 bit. This is expected, given that the double element basically has twice the number of elements, which results in an extra bit of resolution. The error in the adjacent connection matrix is in-between the bias column and the double element, while using the same hardware resources than the bias column. When stochastic rounding is applied, the differences become smaller, but follow the same trends. For the case of nonlinear training, the differences between the double element and bias column become larger than 1 bit. Due to the nonlinear weight update, the weights tend not to be around their middle value, so subtracting the fixed bias column results in a larger performance degradation than combining trainable adjacent elements. As before, the error of the adjacent connection matrix is in-between the error of double element and bias column schemes. However, when stochastic rounding is applied, the performance gain of the adjacent connection matrix with respect to the bias column becomes more significant. These results show that the adjacent connection matrix achieves better accuracy than the bias column in every case, while using the same hardware resources. The largest difference is obtained for nonlinear weights with stochastic rounding, with a gain of up to 2 bits for the convolutional network.

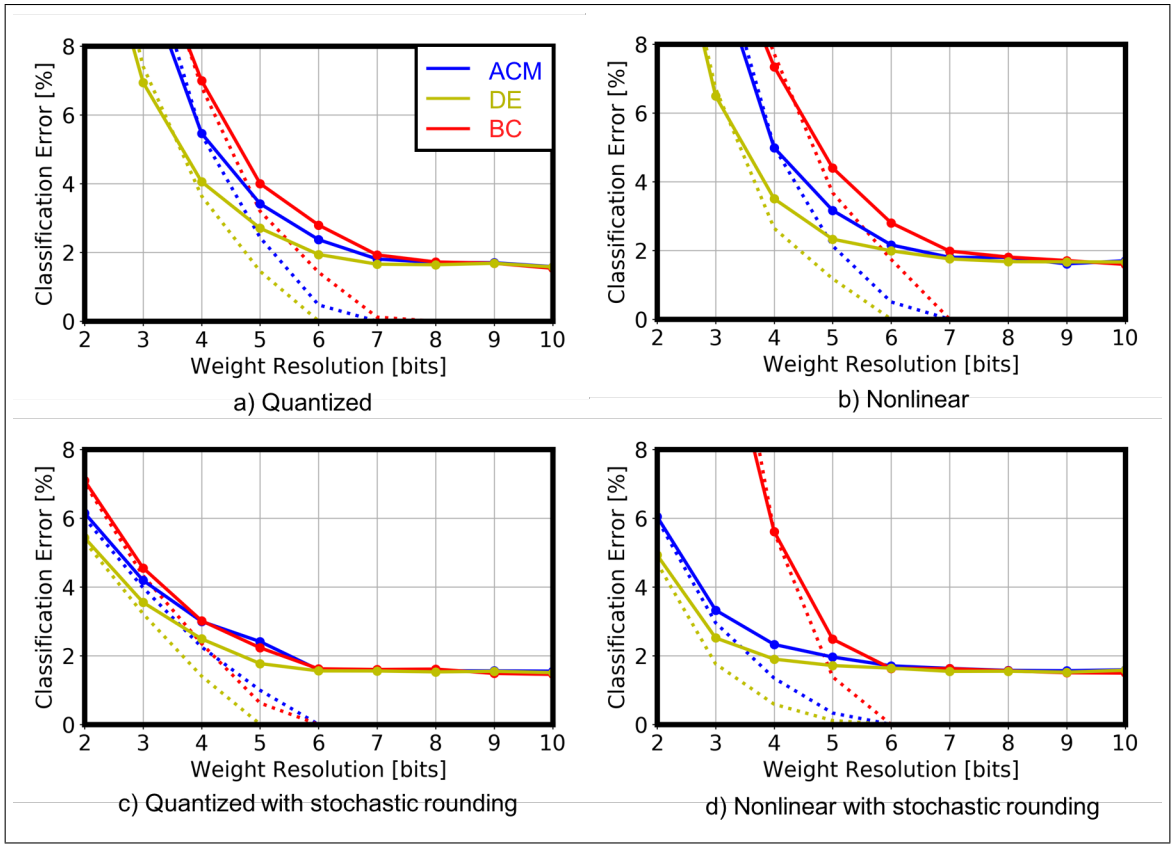

Figure 6.11. Classification error of fully-connected network trained with the MNIST dataset with quantized and nonlienar weights. Training error is shown with dashed lines and test error is shown in solid lines.

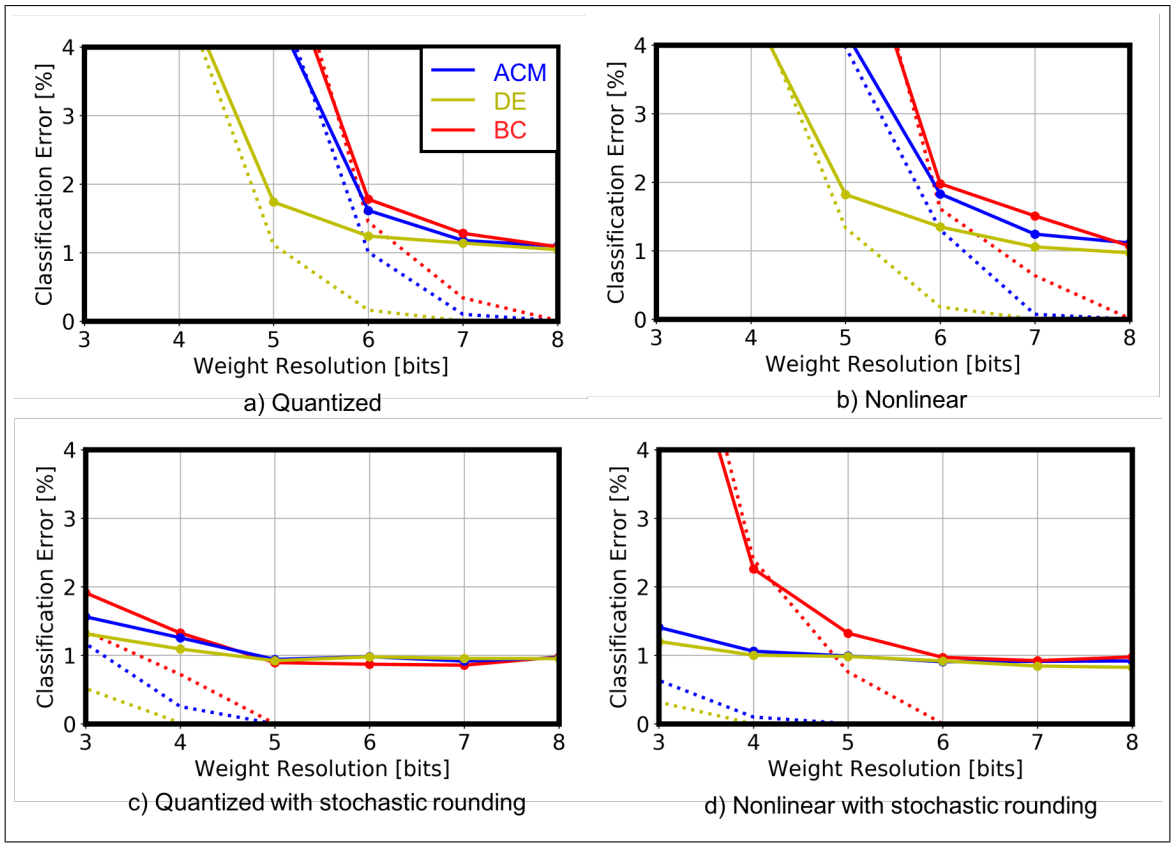

Figure 6.12. Classification error of convolutional network trained with the MNIST dataset with quantized and nonlienar weights. Training error is shown with dashed lines and test error is shown in solid lines.

6.6 Conclusion

A mathematical framework to analyze the mapping of neural network layers to implementations with nonnegative weights was developed. By decomposing a general vector-matrix multiplication into a vector-matrix multiplication with nonnegative elements followed by a limited set of addition and subtraction operations described by a connection matrix, mapping schemes were evaluated and compared. The mathematical conditions for the existence of such decomposition were derived and applied to fully connected and convolutional layers implemented in resistive crossbar arrays. Based on this analysis, it was determined that a crossbar array requires a minimum of one additional column to represent an equivalent signed matrix multiplication, which can be achieved with one subtraction operation for each neuron. Finally, a connection matrix that mitigates the effect of reduced weight resolution and nonlinearity while using minimum resources was proposed. Experimental results were presented by training different networks with MNIST and CIFAR-10 datasets to validate the connection matrix decomposition. Finally, training experiments with limited resolution and nonlinearity were presented, showing that the adjacent connection matrix effectively mitigates the effect of weight nonidealities without hardware overhead.

CHAPTER 7

CONCLUSION

In this thesis, the challenges of implementing DNN training accelerators with resistive crossbar arrays were addressed in two ways. First, a ferroelectric-based memory was proposed and a model to design and optimize multilevel memories based on ferroelectric materials was developed. Second, an architecture to mitigate the effect of weight nonidealities and improve the accuracy of the parallel weight update operation was designed.

The first approach to this problem was to investigate the use of ferroelectrics for multilevel memory storage. The polarization of ferroelectric PZT and HZO capacitors was studied, showing that they can be partially polarized with pulses down to nanosecond scales. These results were the first measurements of partial polarization of FE HZO for multilevel memory storage and were presented at the 2017 Device Research Conference [73]. The polarization response shows a highly nonlinear voltage dependence, which enables the use of stochastic multiplication for parallel weight update in resistive crossbar arrays. A neural network for classification of handwritten digits was simulated to provide a performance evaluation, showing the trade-off between accuracy and dynamic range. These findings led to a patent application on a two-terminal multilevel memory for crossbar arrays proposed to access and program the FE memory state.

After identifying ferroelectrics as a promising material for multilevel memory applications, the polarization reversal dynamics of polycrystalline HZO was characterized and modeled. The results show that the field-dependent NLS model provides a

comprehensive description of the polarization reversal for varying pulse amplitudes and pulse width spanning over 5 decades. The extracted probability distribution characterizes the activation field variations in the FE film, and a minimum switching time constant of 100 ns was obtained for the deposition conditions and electrodes that were measured. This characterization framework provides the tools to quantify, compare and optimize the switching dynamics and the nonlinear response of HZO films. These results were published in IEEE Electron Device Letters in November 2018 [95].

Based on the parameters obtained from polarization reversal, a Monte Carlo simulation framework, capable of predicting the dynamic, history-dependent response of a FE under arbitrary input waveforms was developed. This framweork was presented at the 2018 International Electron Devices Meeting and a manuscript has been submitted to IEEE Transactions on Electron Devices. It was shown that after a well-defined parameter extraction procedure, the proposed model can predict the polarization response of an HZO ferroelectric capacitor under different experimental conditions with the same set of parameters. The model was applied to characterize the dynamic response of FE-DE bilayer structures, showing that the response of polycrystalline FE is significantly different than that of single-grain FE. With this proposed model, the reduction in memory window due to device variability was quantified, both for FE capacitors and FE-DE stacks. Finally, an accumulation effect that leads to grain switching was studied and modeled for the first time by a history parameter. This effect is in agreement with classical nucleation theory, and further theoretical and experimental study is suggested to establish a direct relation between the history-dependent switching probability and the underlying distribution of clusters during the incubation period of domain nucleation.

From the architecture perspective, an accurate scheme for parallel weight update in resistive crossbar arrays was proposed and evaluated. By using pulse width- and

frequency-modulated signals, the value of resistive elements in a crossbar array can be updated in parallel with higher accuracy than existing techniques based on stochastic multiplication. This scheme produces an unbiased multiplication with stochastic rounding, which is optimal for training neural networks with limited resolution. Furthermore, the pulse width and frequency modulation can be implemented with fewer hardware resources than stochastic translators. It was shown that a DNN trained with rate-width multiplication achieves lower train and test error than the equivalent network trained with stochastic multiplication.

Finally, a mathematical framework to analyze the mapping of neural network layers to implementations with nonnegative weights was developed. By decomposing a general vector-matrix multiplication into a vector-matrix multiplication with nonnegative elements followed by a limited set of addition and subtraction operations described by a connection matrix, different mapping schemes can be evaluated and compared. The mathematical conditions for the existence of such decomposition were derived and applied to analyze fully connected and convolutional layers implemented in resistive crossbar arrays. Based on this analysis, it was determined that a crossbar array requires a minimum of one additional column to represent an equivalent signed matrix multiplication, which can be achieved with one subtraction operation for each neuron. A connection matrix that mitigates the effect of reduced weight resolution and nonlinearity was proposed and experimental results were presented to validate the connection matrix decomposition by training different networks with MNIST and CIFAR-10 datasets. Finally, training experiments with limited resolution and nonlinearity showed that the adjacent connection matrix effectively mitigates the impact of weight nonidealities without hardware overhead.

Overall, this thesis work has been driven by a comprehensive understanding of the application domain and the interaction between devices, circuits and architectures. This allowed me to identify, characterize and model a promising material system for

multilevel memory storage and design the architecture based on the characteristics and limitations of the memory devices. Furthermore, the model that I developed will benefit several applications for ferroelectrics that are currently being studied, which also require accurate and predictive models for circuit design.

BIBLIOGRAPHY

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 05 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

2. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 10 1986. [Online]. Available: http://dx.doi.org/10.1038/323533a0

3. B. Reagen, R. Adolf, P. Whatmough, G.-Y. Wei, and D. Brooks, "Deep learning for computer architects," *Synthesis Lectures on Computer Architecture*, vol. 12, no. 4, pp. 1–123, 2017. [Online]. Available: https://doi.org/10.2200/S00783ED1V01Y201706CAC041

4. F.-F. Li, A. Karpathy, and J. Johnson. Stanford CS class CS231n: Convolutional neural networks for visual recognition. [online] http://cs231n.stanford.edu/.

5. M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.

6. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

7. M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

8. N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, "In-datacenter performance analysis of a tensor processing unit," *SIGARCH Comput. Archit. News*, vol. 45, no. 2, pp. 1–12, Jun. 2017. [Online]. Available: http://doi.acm.org/10.1145/3140659.3080246

9. X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, pp. 216–222, 2018. [Online]. Available: https://doi.org/10.1038/s41928-018-0059-3

10. V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec 2017.

11. S. Mittal and J. S. Vetter, "A survey of methods for analyzing and improving GPU energy efficiency," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 19:1–19:23, Aug. 2014. [Online]. Available: http://doi.acm.org/10.1145/2636342

12. V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *ArXiv e-prints*, Mar. 2016.

13. M. Verhelst and B. Moons, "Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to iot and edge devices," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 4, pp. 55–65, Fall 2017.

14. J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th ed., ser. The Morgan Kaufmann Series in Computer Architecture and Design. Morgan Kaufmann, 2011.

15. S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 1737–1746. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045303

16. P. Merolla, R. Appuswamy, J. V. Arthur, S. K. Esser, and D. S. Modha, "Deep neural networks are robust to weight binarization and other non-linear distortions," *CoRR*, vol. abs/1606.01981, 2016. [Online]. Available: http://arxiv.org/abs/1606.01981

17. Y. H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 367–379.

18. A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 14–26.

19. J. Hasler and H. Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers in Neuroscience*, vol. 7, p. 118, 2013. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2013.00118

20. G. W. Burr, P. Narayanan, R. M. Shelby, S. Sidler, I. Boybat, C. di Nolfo, and Y. Leblebici, "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Comparative performance analysis (accuracy, speed, and power)," in *2015 IEEE International Electron Devices Meeting (IEDM)*, Dec 2015, pp. 4.4.1–4.4.4.

21. T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Frontiers in Neuroscience*, vol. 10, p. 333, 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2016.00333

22. P. Narayanan, L. L. Sanches, A. Fumarola, R. M. Shelby, S. Ambrogio, J. Jang, H. Hwang, Y. Leblebici, and G. W. Burr, "Reducing circuit design complexity for neuromorphic machine learning systems based on non-volatile memory arrays," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

23. S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, Feb 2018.

24. T. Gokmen, M. Onen, and W. Haensch, "Training deep convolutional neural networks with resistive cross-point devices," *Frontiers in Neuroscience*, vol. 11, p. 538, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2017.00538

25. T. Gokmen, M. J. Rasch, and W. Haensch, "Training LSTM networks with resistive cross-point devices," *Frontiers in Neuroscience*, vol. 12, p. 745, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2018.00745

26. C. Merkel and D. Kudithipudi, "A stochastic learning algorithm for neuromemristive systems," in *2014 27th IEEE International System-on-Chip Conference (SOCC)*, Sept 2014, pp. 359–364.

27. M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1864–1878, Oct 2014.

28. G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Nov 2015.

29. S. Agarwal, S. J. Plimpton, D. R. Hughart, A. H. Hsia, I. Richter, J. A. Cox, C. D. James, and M. J. Marinella, "Resistive memory device requirements for a

neural algorithm accelerator," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 929–938.

30. B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla, A. G. Schrott, R. S. Shenoy, B. N. Kurdi, C. H. Lam, and D. S. Modha, "Nanoscale electronic synapses using phase change devices," *J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, pp. 12:1–12:20, May 2013. [Online]. Available: http://doi.acm.org/10.1145/2463585.2463588

31. D. Kuzum, S. Yu, and H.-S. P. Wong, "Synaptic electronics: materials, devices and applications," *Nanotechnology*, vol. 24, no. 38, p. 382001, 2013. [Online]. Available: http://stacks.iop.org/0957-4484/24/i=38/a=382001

32. G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, L. L. Sanches, I. Boybat, M. L. Gallo, K. Moon, J. Woo, H. Hwang, and Y. Leblebici, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017. [Online]. Available: http://dx.doi.org/10.1080/23746149.2016.1259585

33. D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 05 2008. [Online]. Available: http://dx.doi.org/10.1038/nature06932

34. H. . P. Wong, H. Lee, S. Yu, Y. Chen, Y. Wu, P. Chen, B. Lee, F. T. Chen, and M. Tsai, "Metal–oxide RRAM," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, June 2012.

35. Z. Xu, A. Mohanty, P.-Y. Chen, D. Kadetotad, B. Lin, J. Ye, S. Vrudhula, S. Yu, J. sun Seo, and Y. Cao, "Parallel programming of resistive cross-point array for synaptic plasticity," *Procedia Computer Science*, vol. 41, pp. 126 – 133, 2014, 5th Annual International Conference on Biologically Inspired Cognitive Architectures, 2014 BICA. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914015397

36. J. Shen, J. Cong, D. Shang, Y. Chai, S. Shen, K. Zhai, and Y. Sun, "A multilevel nonvolatile magnetoelectric memory," *Scientific Reports*, vol. 6, pp. 34 473 EP –, 09 2016. [Online]. Available: http://dx.doi.org/10.1038/srep34473

37. B. Rajendran and F. Alibart, "Neuromorphic computing based on emerging memory technologies," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 2, pp. 198–211, June 2016.

38. Y. van de Burgt, E. Lubberman, E. J. Fuller, S. T. Keene, G. C. Faria, S. Agarwal, M. J. Marinella, A. Alec Talin, and A. Salleo, "A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing," *Nat Mater*, vol. 16, no. 4, pp. 414–418, 04 2017. [Online]. Available: http://dx.doi.org/10.1038/nmat4856

39. H. Ishiwara, "Proposal of adaptive-learning neuron circuits with ferroelectric analog-memory weights," *Japanese Journal of Applied Physics*, vol. 32, no. 1S, p. 442, 1993. [Online]. Available: http://stacks.iop.org/1347-4065/32/i=1S/a=442

40. D. Lee, S. M. Yang, T. H. Kim, B. C. Jeon, Y. S. Kim, J.-G. Yoon, H. N. Lee, S. H. Baek, C. B. Eom, and T. W. Noh, "Multilevel data storage memory using deterministic polarization control," *Advanced Materials*, vol. 24, no. 3, pp. 402–406, 2012. [Online]. Available: http://dx.doi.org/10.1002/adma.201103679

41. A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier, "A ferroelectric memristor," *Nat Mater*, vol. 11, no. 10, pp. 860–864, 10 2012. [Online]. Available: http://dx.doi.org/10.1038/nmat3415

42. M.-H. Kim, G. J. Lee, C.-M. Keum, and S.-D. Lee, "Voltage-readable nonvolatile memory cell with programmable ferroelectric multistates in organic inverter configuration," *Organic Electronics*, vol. 14, no. 5, pp. 1231 – 1236, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566119913000797

43. J. W. Jang, S. Park, G. W. Burr, H. Hwang, and Y. H. Jeong, "Optimization of conductance change in $Pr_{1-x}Ca_xMnO_3$-based synaptic devices for neuromorphic systems," *IEEE Electron Device Letters*, vol. 36, no. 5, pp. 457–459, May 2015.

44. J. Lee, A. J. J. M. van Breemen, V. Khikhlovskyi, M. Kemerink, R. A. J. Janssen, and G. H. Gelinck, "Pulse-modulated multilevel data storage in an organic ferroelectric resistive memory diode," *Scientific Reports*, vol. 6, pp. 24 407 EP –, 04 2016. [Online]. Available: http://dx.doi.org/10.1038/srep24407

45. I. Katsouras, K. Asadi, W. A. Groen, P. W. M. Blom, and D. M. de Leeuw, "Retention of intermediate polarization states in ferroelectric materials enabling memories for multi-bit data storage," *Applied Physics Letters*, vol. 108, no. 23, p. 232907, 2016. [Online]. Available: http://dx.doi.org/10.1063/1.4953199

46. S. Boyn, J. Grollier, G. Lecerf, B. Xu, N. Locatelli, S. Fusil, S. Girod, C. Carrétéro, K. Garcia, S. Xavier, J. Tomas, L. Bellaiche, M. Bibes, A. Barthélémy, S. Saïghi, and V. Garcia, "Learning through ferroelectric domain dynamics in solid-state synapses," *Nature Communications*, vol. 8, pp. 14 736 EP –, 04 2017. [Online]. Available: http://dx.doi.org/10.1038/ncomms14736

47. T. S. Böscke, J. Müller, D. Bräuhaus, U. Schröder, and U. Böttger, "Ferroelectricity in hafnium oxide: CMOS compatible ferroelectric field effect transistors," in *2011 International Electron Devices Meeting*, Dec 2011, pp. 24.5.1–24.5.4.

48. S. Mueller, S. R. Summerfelt, J. Müller, U. Schroeder, and T. Mikolajick, "Ten-nanometer ferroelectric Si:HfO_2 films for next-generation FRAM capacitors," *IEEE Electron Device Letters*, vol. 33, no. 9, pp. 1300–1302, Sept 2012.

49. J. Muller, T. S. Boscke, U. Schroder, R. Hoffmann, T. Mikolajick, and L. Frey, "Nanosecond polarization switching and long retention in a novel MFIS-FET based on ferroelectric HfO_2," *IEEE Electron Device Letters*, vol. 33, no. 2, pp. 185–187, Feb 2012.

50. H. Mulaosmanovic, J. Ocker, S. Müller, U. Schroeder, J. Müller, P. Polakowski, S. Flachowsky, R. van Bentum, T. Mikolajick, and S. Slesazeck, "Switching kinetics in nanoscale hafnium oxide based ferroelectric field-effect transistors," *ACS Applied Materials & Interfaces*, vol. 9, no. 4, pp. 3792–3798, 02 2017.

51. S. George, K. Ma, A. Aziz, X. Li, A. Khan, S. Salahuddin, M.-F. Chang, S. Datta, J. Sampson, S. Gupta, and V. Narayanan, "Nonvolatile memory design based on ferroelectric fets," in *Proceedings of the 53rd Annual Design Automation Conference*, ser. DAC '16. New York, NY, USA: ACM, 2016, pp. 118:1–118:6. [Online]. Available: http://doi.acm.org/10.1145/2897937.2898050

52. S. Oh, T. Kim, M. Kwak, J. Song, J. Woo, S. Jeon, I. K. Yoo, and H. Hwang, "HfZrOx-based ferroelectric synapse device with 32 levels of conductance states for neuromorphic applications," *IEEE Electron Device Letters*, vol. 38, no. 6, pp. 732–735, June 2017.

53. D. Kwon, K. Chatterjee, A. J. Tan, A. K. Yadav, H. Zhou, A. B. Sachid, R. D. Reis, C. Hu, and S. Salahuddin, "Improved subthreshold swing and short channel effect in fdsoi n-channel negative capacitance field effect transistors," *IEEE Electron Device Letters*, vol. 39, no. 2, pp. 300–303, Feb 2018.

54. M. Jerry, P. Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, and S. Datta, "Ferroelectric fet analog synapse for acceleration of deep neural network training," in *2017 IEEE International Electron Devices Meeting (IEDM)*, Dec 2017, pp. 6.2.1–6.2.4.

55. X. Li, S. George, K. Ma, W. Y. Tsai, A. Aziz, J. Sampson, S. K. Gupta, M. F. Chang, Y. Liu, S. Datta, and V. Narayanan, "Advancing nonvolatile computing with nonvolatile NCFET latches and flip-flops," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 11, pp. 2907–2919, Nov 2017.

56. A. Aziz, E. T. Breyer, A. Chen, X. Chen, S. Datta, S. K. Gupta, M. Hoffmann, X. S. Hu, A. Ionescu, M. Jerry, T. Mikolajick, H. Mulaosmanovic, K. Ni, M. Niemier, I. O'Connor, A. Saha, S. Slesazeck, S. K. Thirumala, and X. Yin, "Computing with ferroelectric FETs: Devices, models, systems, and applications," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1289–1298.

57. P.-Y. Chen, X. Peng, and S. Yu, "Neurosim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *Electron Devices Meeting (IEDM), 2017 IEEE International*. IEEE, 2017, pp. 6–1.

58. P. Narayanan, A. Fumarola, L. L. Sanches, K. Hosokawa, S. C. Lewis, R. M. Shelby, and G. W. Burr, "Toward on-chip acceleration of the backpropagation algorithm using nonvolatile memory," *IBM Journal of Research and Development*, vol. 61, no. 4, pp. 11:1–11:11, July 2017.

59. C. C. Chang, P. C. Chen, T. Chou, I. T. Wang, B. Hudec, C. C. Chang, C. M. Tsai, T. S. Chang, and T. H. Hou, "Mitigating asymmetric nonlinear weight update effects in hardware neural network based on analog resistive synapse," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

60. M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electronics*, vol. 1, no. 4, pp. 246–253, 2018. [Online]. Available: https://doi.org/10.1038/s41928-018-0054-8

61. S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, 2018. [Online]. Available: https://doi.org/10.1038/s41586-018-0180-5

62. G. Cristiano, M. Giordano, S. Ambrogio, L. P. Romero, C. Cheng, P. Narayanan, H. Tsai, R. M. Shelby, and G. W. Burr, "Perspective on training fully connected networks with resistive memories: Device requirements for multiple conductances of varying significance," *Journal of Applied Physics*, vol. 124, no. 15, p. 151901, 2018. [Online]. Available: https://doi.org/10.1063/1.5042462

63. X. Sun, P. Wang, K. Ni, S. Datta, and S. Yu, "Exploiting hybrid precision for training and inference: A 2T-1FeFET based analog synaptic weight cell," in *2018 IEEE International Electron Devices Meeting (IEDM)*, Dec 2018, pp. 3.1.1–3.1.4.

64. S. V. Kalinin, A. N. Morozovska, L. Q. Chen, and B. J. Rodriguez, "Local polarization dynamics in ferroelectric materials," *Reports on Progress in Physics*, vol. 73, no. 5, p. 056502, 2010. [Online]. Available: http://stacks.iop.org/0034-4885/73/i=5/a=056502

65. R. Materlik, C. Künneth, and A. Kersch, "The origin of ferroelectricity in $Hf_{1-x}Zr_xO_2$: A computational investigation and a surface energy model," *Journal of Applied Physics*, vol. 117, no. 13, p. 134109, 2015. [Online]. Available: https://doi.org/10.1063/1.4916707

66. Undoped 20/80 PZT with platinum electrodes and $400\mu m^2$ area. Radiant Technologies, Part Number RTAB401.

67. G. Karbasian, A. Tan, A. Yadav, E. M. H. Sorensen, C. R. Serrao, A. I. Khan, K. Chatterjee, S. Kim, C. Hu, and S. Salahuddin, "Ferroelectricity in HfO_2 thin films as a function of Zr doping," in *2017 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, April 2017, pp. 1–2.

68. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

69. T. Shiraishi, K. Katayama, T. Yokouchi, T. Shimizu, T. Oikawa, O. Sakata, H. Uchida, Y. Imai, T. Kiguchi, T. J. Konno, and H. Funakubo, "Effect of the film thickness on the crystal structure and ferroelectric properties of $Hf_{0.5}Zr_{0.5}O_2$ thin films deposited on various substrates," *Materials Science in Semiconductor Processing*, vol. 70, pp. 239 – 245, 2017, control of Semiconductor Interfaces and SiGe Technology/Devices. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369800116305935

70. S. J. Kim, J. Mohan, J. Lee, J. S. Lee, A. T. Lucero, C. D. Young, L. Colombo, S. R. Summerfelt, T. San, and J. Kim, "Effect of film thickness on the ferroelectric and dielectric properties of low-temperature (400 °C) $Hf_{0.5}Zr_{0.5}O_2$ films," *Applied Physics Letters*, vol. 112, no. 17, p. 172902, 2018. [Online]. Available: https://doi.org/10.1063/1.5026715

71. F. Ambriz-Vargas, G. Kolhatkar, M. Broyer, A. Hadj-Youssef, R. Nouar, A. Sarkissian, R. Thomas, C. Gomez-Yáñez, M. A. Gauthier, and A. Ruediger, "A complementary metal oxide semiconductor process-compatible ferroelectric tunnel junction," *ACS Applied Materials & Interfaces*, vol. 9, no. 15, pp. 13 262–13 268, 04 2017. [Online]. Available: https://doi.org/10.1021/acsami.6b16173

72. S. Mueller, J. Müller, R. Hoffmann, E. Yurchuk, T. Schlösser, R. Boschke, J. Paul, M. Goldbach, T. Herrmann, A. Zaka, U. Schröder, and T. Mikolajick, "From MFM capacitors toward ferroelectric transistors: Endurance and disturb characteristics of HfO_2-based FeFET devices," *IEEE Transactions on Electron Devices*, vol. 60, no. 12, pp. 4199–4205, Dec 2013.

73. E. W. Kinder, C. Alessandri, P. Pandey, G. Karbasian, S. Salahuddin, and A. Seabaugh, "Partial switching of ferroelectrics for synaptic weight storage," in *2017 Device Research Conference (DRC)*, June 2017, pp. 1–2.

74. H. K. Yoo, J. S. Kim, Z. Zhu, Y. S. Choi, A. Yoon, M. R. MacDonald, X. Lei, T. Y. Lee, D. Lee, S. C. Chae, J. Park, D. Hemker, J. G. Langan, Y. Nishi, and S. J. Hong, "Engineering of ferroelectric switching speed in Si doped HfO_2 for high-speed 1T-FERAM application," in *2017 IEEE International Electron Devices Meeting (IEDM)*, Dec 2017, pp. 19.6.1–19.6.4.

75. W. Chung, M. Si, P. R. Shrestha, J. P. Campbell, K. P. Cheung, and P. D. Ye, "First direct experimental studies of $Hf_{0.5}Zr_{0.5}O_2$ ferroelectric polarization

switching down to 100-picosecond in sub-60mv/dec germanium ferroelectric nanowire FETs," in *2018 IEEE Symposium on VLSI Technology*, June 2018, pp. 89–90.

76. H. Mulaosmanovic, T. Mikolajick, and S. Slesazeck, "Accumulative polarization reversal in nanoscale ferroelectric transistors," *ACS Applied Materials & Interfaces*, vol. 10, no. 28, pp. 23 997–24 002, 07 2018.

77. N. Gong, X. Sun, H. Jiang, K. S. Chang-Liao, Q. Xia, and T. P. Ma, "Nucleation limited switching (NLS) model for HfO_2-based metal-ferroelectric-metal (MFM) capacitors: Switching kinetics and retention characteristics," *Applied Physics Letters*, vol. 112, no. 26, p. 262903, 2018.

78. H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein, "Training quantized nets: A deeper understanding," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5811–5821. [Online]. Available: http://papers.nips.cc/paper/7163-training-quantized-nets-a-deeper-understanding.pdf

79. A. Sharma and K. Roy, "1T Non-Volatile Memory Design Using Sub-10nm Ferroelectric FETs," *IEEE Electron Device Letters*, vol. 39, no. 3, pp. 359–362, March 2018.

80. M. H. Park, H. J. Kim, Y. J. Kim, Y. H. Lee, T. Moon, K. D. Kim, S. D. Hyun, F. Fengler, U. Schroeder, and C. S. Hwang, "Effect of Zr content on the wake-up effect in $Hf_{1-x}Zr_xO_2$ films," *ACS Applied Materials & Interfaces*, vol. 8, no. 24, pp. 15 466–15 475, 2016, pMID: 27237137. [Online]. Available: http://dx.doi.org/10.1021/acsami.6b03586

81. G. Karbasian, R. dos Reis, A. K. Yadav, A. J. Tan, C. Hu, and S. Salahuddin, "Stabilization of ferroelectric phase in tungsten capped $Hf_{0.8}Zr_{0.2}O_2$," *Applied Physics Letters*, vol. 111, no. 2, p. 022907, 2017. [Online]. Available: https://doi.org/10.1063/1.4993739

82. Y.-C. Lin, F. McGuire, and A. D. Franklin, "Realizing ferroelectric $Hf_{0.5}Zr_{0.5}O_2$ with elemental capping layers," *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, vol. 36, no. 1, p. 011204, 2018. [Online]. Available: https://doi.org/10.1116/1.5002558

83. S. J. Kim, D. Narayan, J.-G. Lee, J. Mohan, J. S. Lee, J. Lee, H. S. Kim, Y.-C. Byun, A. T. Lucero, C. D. Young, S. R. Summerfelt, T. San, L. Colombo, and J. Kim, "Large ferroelectric polarization of $TiN/Hf_{0.5}Zr_{0.5}O_2/TiN$ capacitors due to stress-induced crystallization at low thermal budget," *Applied Physics Letters*, vol. 111, no. 24, p. 242901, 2017. [Online]. Available: https://doi.org/10.1063/1.4995619

84. M. H. Park, H. J. Kim, Y. J. Kim, W. Lee, T. Moon, and C. S. Hwang, "Evolution of phases and ferroelectric properties of thin $Hf_{0.5}Zr_{0.5}O_2$ films according to the thickness and annealing temperature," *Applied Physics Letters*, vol. 102, no. 24, p. 242905, 2013. [Online]. Available: http://dx.doi.org/10.1063/1.4811483

85. K. Chatterjee, A. J. Rosner, and S. Salahuddin, "Intrinsic speed limit of negative capacitance transistors," *IEEE Electron Device Letters*, vol. 38, no. 9, pp. 1328–1330, Sept 2017.

86. J. A. Kittl, B. Obradovic, D. Reddy, T. Rakshit, R. M. Hatcher, and M. S. Rodder, "On the validity and applicability of models of negative capacitance and implications for MOS applications," *Applied Physics Letters*, vol. 113, no. 4, p. 042904, 2018. [Online]. Available: https://doi.org/10.1063/1.5036984

87. Y. Ishibashi and Y. Takagi, "Ferroelectric domain switching," *Journal of the Physical Society of Japan*, vol. 31, no. 2, p. 506, 1971.

88. J. F. Scott, L. Kammerdiner, M. Parris, S. Traynor, V. Ottenbacher, A. Shawabkeh, and W. F. Oliver, "Switching kinetics of lead zirconate titanate submicron thinfilm memories," *Journal of Applied Physics*, vol. 64, no. 2, pp. 787–792, 1988.

89. J. Woo, S. Hong, N. Setter, H. Shin, J.-U. Jeon, Y. E. Pak, and K. No, "Quantitative analysis of the bit size dependence on the pulse width and pulse voltage in ferroelectric memory devices using atomic force microscopy," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, vol. 19, no. 3, pp. 818–824, 2001. [Online]. Available: http://avs.scitation.org/doi/abs/10.1116/1.1364697

90. O. Lohse, M. Grossmann, U. Boettger, D. Bolten, and R. Waser, "Relaxation mechanism of ferroelectric switching in $Pb(Zr,Ti)O_3$ thin films," *Journal of Applied Physics*, vol. 89, no. 4, pp. 2332–2336, 2001. [Online]. Available: https://doi.org/10.1063/1.1331341

91. A. K. Tagantsev, I. Stolichnov, N. Setter, J. S. Cross, and M. Tsukada, "Non-Kolmogorov-Avrami switching kinetics in ferroelectric thin films," *Phys. Rev. B*, vol. 66, p. 214109, Dec 2002.

92. J. Y. Jo, H. S. Han, J.-G. Yoon, T. K. Song, S.-H. Kim, and T. W. Noh, "Domain switching kinetics in disordered ferroelectric thin films," *Phys. Rev. Lett.*, vol. 99, p. 267602, Dec 2007.

93. S. Zhukov, Y. A. Genenko, O. Hirsch, J. Glaum, T. Granzow, and H. von Seggern, "Dynamics of polarization reversal in virgin and fatigued ferroelectric ceramics by inhomogeneous field mechanism," *Phys. Rev. B*, vol. 82, p. 014109, Jul 2010.

94. J. Müller, T. S. Böscke, S. Müller, E. Yurchuk, P. Polakowski, J. Paul, D. Martin, T. Schenk, K. Khullar, A. Kersch, W. Weinreich, S. Riedel, K. Seidel, A. Kumar, T. M. Arruda, S. V. Kalinin, T. Schlösser, R. Boschke, R. van Bentum, U. Schröder, and T. Mikolajick, "Ferroelectric hafnium oxide: A CMOS-compatible and highly scalable approach to future ferroelectric memories," in *2013 IEEE International Electron Devices Meeting*, Dec 2013, pp. 10.8.1–10.8.4.

95. C. Alessandri, P. Pandey, A. Abusleme, and A. Seabaugh, "Switching dynamics of ferroelectric Zr-doped HfO_2," *IEEE Electron Device Letters*, vol. 39, no. 11, pp. 1780–1783, Nov 2018.

96. K. Ni, M. Jerry, J. A. Smith, and S. Datta, "A circuit compatible accurate compact model for ferroelectric-FETs," in *2018 IEEE Symposium on VLSI Technology*, June 2018, pp. 131–132.

97. B. Obradovic, T. Rakshit, R. Hatcher, J. A. Kittl, and M. S. Rodder, "Ferroelectric switching delay as cause of negative capacitance and the implications to NCFETs," in *2018 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, 2018, pp. 51–52.

98. A. K. Saha, S. Datta, and S. K. Gupta, ""negative capacitance" in resistor-ferroelectric and ferroelectric-dielectric networks: Apparent or intrinsic?" *Journal of Applied Physics*, vol. 123, no. 10, p. 105102, 2018.

99. N. Gong and T. P. Ma, "A study of endurance issues in HfO_2-based ferroelectric field effect transistors: Charge trapping and trap generation," *IEEE Electron Device Letters*, vol. 39, no. 1, pp. 15–18, Jan 2018.

100. Y. A. Genenko, S. Zhukov, S. V. Yampolskii, J. Schütrumpf, R. Dittmer, W. Jo, H. Kungl, M. J. Hoffmann, and H. von Seggern, "Universal polarization switching behavior of disordered ferroelectrics," *Advanced Functional Materials*, vol. 22, no. 10, pp. 2058–2066, 2012.

101. R. W. Balluffi, S. M. Allen, and W. C. Carter, *Kinetics of Materials*, 1st ed. John Wiley & Sons, 2005.

102. W. Lee, Y. Kim, Y. Song, K. Cho, D. Yoo, H. Ahn, K. Kang, and T. Lee, "Investigation of time–dependent resistive switching behaviors of unipolar nonvolatile organic memory devices," *Advanced Functional Materials*, vol. 28, no. 35, p. 1801162, 2018.

103. Y. Arayashiki, T. Nakajima, Y. Takahashi, and T. Furukawa, "Accelerated and decelerated polarization reversal in thin vinylidene fluoride/trifluoroethylene copolymer films," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 17, no. 4, pp. 1066–1073, August 2010.

104. C. Alessandri, P. Pandey, and A. Seabaugh, "Experimentally validated, predictive monte carlo modeling of ferroelectric dynamics and variability," in *2018 International Electron Devices Meeting*, Dec 2018, pp. 12.2.1–12.2.4.

105. A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018, pp. 124–126.

106. K. Yoshioka, Y. Toyama, K. Ban, D. Yashima, S. Maya, A. Sai, and K. Onizuka, "Phasemac: A 14 tops/w 8bit gro based phase domain mac circuit for in-sensor-computed deep learning accelerators," in *2018 IEEE Symposium on VLSI Circuits*, June 2018, pp. 263–264.

107. B. Sedighi, I. Palit, X. S. Hu, J. Nahas, and M. Niemier, "A CNN-inspired mixed signal processor based on tunnel transistors," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2015, pp. 1150–1155.

108. A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2s, pp. 92:1–92:19, May 2013. [Online]. Available: http://doi.acm.org/10.1145/2465787.2465794

109. S. Liu and J. Han, "Toward energy-efficient stochastic circuits using parallel Sobol sequences," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–14, 2018.

110. Y. Xie, S. Liao, B. Yuan, Y. Wang, and Z. Wang, "Fully-parallel area-efficient deep neural network design using stochastic computing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 12, pp. 1382–1386, Dec 2017.

111. N. Weste, *CMOS VLSI Design : A Circuits and Systems Perspective*. Boston: Addison Wesley, 2011.

112. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

113. C. Parisien, C. H. Anderson, and C. Eliasmith, "Solving the problem of negative synaptic weights in cortical models," *Neural Computation*, vol. 20, pp. 1473–1494, 2008.

114. B. Li, P. Gu, Y. Shan, Y. Wang, Y. Chen, and H. Yang, "RRAM-based analog approximate computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 12, pp. 1905–1917, 2015.

115. B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, "Merging the interface: Power, area and accuracy co-optimization for RRAM crossbar-based mixed-signal computing system," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 13.

116. M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: programming 1t1m crossbar to accelerate matrix-vector multiplication," in *Proceedings of the 53rd annual design automation conference*. ACM, 2016, p. 19.

117. F. Chollet *et al.*, "Keras," https://keras.io, 2015.

118. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

119. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

120. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

121. P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, Aug 2003, pp. 958–963.

122. W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 108–122, Jan 2019.

123. S. Kim, T. Gokmen, H. M. Lee, and W. E. Haensch, "Analog cmos-based resistive processing unit for deep neural network training," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug 2017, pp. 422–425.

124. B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, "Minimum Energy Quantized Neural Networks," *ArXiv e-prints*, Nov. 2017.

125. A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "VLSI implementation of deep neural network using integral stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2688–2699, Oct 2017.

126. H. Sim and J. Lee, "A new stochastic computing multiplier with application to deep convolutional neural networks," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2017, pp. 1–6.

127. V. T. Lee, A. Alaghi, J. P. Hayes, V. Sathe, and L. Ceze, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 13–18.

APPENDIX A

LIST OF PUBLICATIONS AND PATENTS

Along with the work presented in the body of this thesis, I completed other research projects during the beginning of the dual degree, which are not directly related to hardware acceleration for DNNs. These projects were published in peer-reviewed journals and are included as appendices.

Journal Publications

1. **C. Alessandri**, P. Pandey, A. Abusleme, and A. Seabaugh, *Monte Carlo simulation of switching dynamics in polycrystalline ferroelectrics* IEEE Transactions on Electron Devices, Under Review

2. **C. Alessandri**, P. Pandey, A. Abusleme, and A. Seabaugh, *Switching dynamics of ferroelectric Zr-doped HfO_2,* IEEE Electron Device Letters, vol. 39, no. 11, pp. 17801783, 2018.

3. M. Jara, **C. Alessandri**, A. Abusleme,: *Time-domain 1/f noise analysis of a charge-redistribution track-and-hold circuit*, IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 2, pp. 161-165, 2018

4. **C. Alessandri**, S. Fathipour, H. Li, I. Kwak, A. Kummel, M. Remskar, A. C. Seabaugh: *Reconfigurable electric double layer doping in an MoS_2 nanoribbon transistor*, IEEE Trans. Electron Devices, vol. 64, no. 12, pp. 5217-5222, 2017. (Appendix D)

5. **C. Alessandri**, A. Abusleme, D. Guzman, I. Passalacqua, E. Alvarez-Fontecilla, M. Guarini: *Optimal CCD readout by digital correlated double sampling*, Monthly Notices of the Royal Astronomical Society, vol. 455, pp. 1443-1450, 2015. (Appendix E)

Conference Publications

1. **C. Alessandri**, P. Pandey, and A. Seabaugh, *Experimentally validated, predictive Monte Carlo modeling of ferroelectric dynamics and variability,* IEEE International Electron Devices Meeting (IEDM), pp. 162, IEEE, 2018.

2. E. Kinder, **C. Alessandri**, P. Pandey, G. Karbasian, S. Salahuddin, A. C. Seabaugh: *Partial switching of ferroelectrics for synaptic weight storage*, 2017 75th Annual Device Research Conference (DRC), Notre Dame, IN, 2017

3. J. Zhang, **C. Alessandri**, P. Fay, A. C. Seabaugh, T.Ytterdal, E. Memisevic, L.E. Wernersson: *Projected performance of experimental InAs/GaAsSb/GaSb TFET as millimeter-wave detector*, 2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Burlingame, CA, 2017

Patent Applications

1. **C. Alessandri**, E. Kinder, A. Seabaugh. Partial polarization resistive electronic devices, neural network systems including partial polarization resistive electronic devices and methods for operating the same. United States Patent Application No. 16,180,453. 2018, Nov 5. (Appendix B)

APPENDIX B

US PATENT APPLICATION NUMBER 16180453

Notre Dame Reference No.: 18-002
MARCO Reference: P1742
W&S Reference: 170964-00006

PARTIAL-POLARIZATION RESISTIVE ELECTRONIC DEVICES, NEURAL NETWORK SYSTEMS INCLUDING PARTIAL-POLARIZATION RESISTIVE ELECTRONIC DEVICES AND METHODS OF OPERATING THE SAME

STATEMENT OF GOVERNMENT SUPPORT

[001] This invention was made with government funds under Agreement No HR0011-13-3-0002 awarded by the Defense Advanced Research Projects Agency (DARPA) and Award Grant No. ECCS1631717 awarded by the National Science Foundation. The U.S. Government has certain rights in the invention.

FIELD

[002] The present invention relates to the field of electronics in general, and more particularly, to resistive-based memory devices.

BACKGROUND

[003] Deep Neural Networks (DNN) can perform cognitive tasks such as speech recognition, drug discovery and object detection with high accuracy and efficiency. Training a DNN, however, can be an energy- and time-consuming task. Hardware-based accelerators have the potential to out-perform software implementations. A synaptic element of the DNN can be important in this type of approach.

[004] It has been proposed to utilize a nonvolatile memory with multilevel conductance that can perform a weight update operation by stochastic multiplication as the synaptic element to provide a Resistive Processing Unit (RPU). Such an RPU, may require 1000 levels of conductance with an increase/decrease systematic mismatch below 5%, which may be difficult to accomplish with phase-change memory (PCM) and resistive random access memory (RRAM) due to their fundamental asymmetry.

SUMMARY

[005] Embodiments according to the invention can provide partial-polarization resistive electronic devices, neural network systems including partial-polarization resistive electronic

devices and methods of operating the same. Pursuant to these embodiments, an electronic device can include a semiconductor material including a channel region configured to conduct a current, a source contact electrically coupled to the channel region at a first location, a drain contact electrically coupled to the channel region at a second location spaced apart from the first location, a partial-polarization material on the semiconductor material between the source contact and the drain contact opposite the channel region and a gate contact on the partial-polarization material opposite the channel region and ohmically coupled to the drain contact or ohmically coupled to the source contact.

[006] In some embodiments, a method of programming an electronic device can be provided by applying a programming voltage pulse or sequence of pulses between a drain contact and a source contact to set a partial-polarization state of a partial-polarization material opposite a channel region of the electronic device and applying a read voltage pulse to generate a current in the channel region responsive to the partial-polarization state.

In some embodiments, a neural network circuit can include a plurality of serially connected layers of the neural network circuit, wherein each layer includes a respective plurality of neurons, a plurality of resistive processing cross-bar circuits, wherein each of the resistive processing cross-bar circuits is connected between directly adjacent ones of the serially connected layers to provide a respective weighting to data provided from the respective plurality of neurons included in an upstream layer of the neural network circuit that is summed by a down-stream layer of the neural network circuit and each of the resistive processing cross-bar circuits comprises a respective array of partial-polarization electronic devices each being arranged in a diode-connected configuration.

BRIEF DESCRIPTION OF THE FIGURES

[007] FIG. 1 is a schematic drawing of a Deep Neural Network (DNN) having 4 layers of neurons (L^i), using diode connected partial-polarization electronic devices to implement a Resistive Processing Unit (RPU) wherein voltage signals are applied in the upstream layer (L1) and a resulting current is integrated in the downstream layer (L2) in some embodiments according to the invention.

[008] FIG. 2 is a schematic drawing of an RPU using diode connected partial-polarization electronic devices in some embodiments according to the invention.

[009] FIG. 3 illustrates a weight update operation implemented using stochastic multiplication in some embodiments according to the invention.

[0010] FIG. 4 is an illustration of a partial polarization operation of a 10 nm thick TiN/HZO/TiN capacitor in some embodiments according to the invention.

[0011] FIG. 5 is a graph illustrating measured ferroelectric polarization (circles) and fitted model ferroelectric polarization (lines) of the capacitor of FIG. 4 in some embodiments according to the invention.

[0012] FIGs. 6-8 are cross-sectional views of an electronic device including a planar semiconductor opposite a partial-polarization material in some embodiments according to the invention.

[0013] FIGs. 9-10B are perspective views of an electronic device including a vertically protruding semiconductor structure opposite a partial-polarization material in some embodiments according to the invention.

[0014] FIG. 11A-C are comparative graphs that illustrate saturation behavior of weighting.

[0015] FIG. 12 is a graph that shows the training error achieved using the ideal update rule of Equation (1) in some embodiments according to the invention.

[0016] FIG. 13A-C are graphs illustrating the distribution of values for weights W1, W2, and W3 obtain after training with the base model in some embodiments according to the invention.

[0017] FIG. 14A-C are graphs illustrating test and training error obtained after 30 epochs with the weight update rule in eq (3) for a) Δw_o=0.001, b) Δw_o=0.005 and c) Δw_o=0.01 in some embodiments according to the invention.

[0018] FIG. 15 is a graph that shows non-systematic asymmetry as a function of the weight saturation value w_{max}, computed from the update rule in equation (3) for a weight $w_{ij}^{L} = 1$ in some embodiments according to the invention.

[0019] FIG. 16 is a table listing ideal prior art RPU specifications used to achieve a 2.3% test accuracy, compared with the polarization-based RPU in some embodiments according to the invention.

Notre Dame Reference No.: 18-002
MARCO Reference: P1742
W&S Reference: 170964-00006

DETAILED DESCRIPTION OF EMBODIMENTS

[0020] Exemplary embodiments of the present disclosure are described in detail with reference to the accompanying drawings. The disclosure may, however, be exemplified in many different forms and should not be construed as being limited to the specific exemplary embodiments set forth herein. Rather, these exemplary embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the disclosure to those skilled in the art.

[0021] FIG. 1 shows a Deep Neural Network (DNN) having 4 layers of neurons L1-L4 that are connected to subsequent layers by a weight matrix (W^i) with two layers of N2 and N3 neurons (sometimes referred to as hidden layers) and an output layer with 10 neurons 0-9. Each layer has a bias unit $x_0^L = 1$. Layer numbers are indicated by superscript, whereas neuron rows are indicated by subscript. Each neuron receives a weighted sum of the neurons in the previous layer and can compute a sigmoid activation function *f(z)*, bounded between 0 and 1. The DNN shown in FIG. 1 can use diode connected partial-polarization electronic devices to implement an RPU in some embodiments according to the invention.

[0022] FIG. 2 is a schematic drawing that illustrates an RPU implementation of a DNN, where each layer of neurons is connected to the next layer in a crossbar array of RPUs. It will be understood that the RPUs can be diode connected partial-polarization electronic devices as illustrated in, for example FIGs. 6-10B, in some embodiments according to the invention. In particular, the depicted example shows that voltage inputs are generated from L1 neurons and the resulting current is integrated by L2 neurons to perform the equivalent of a weighted sum. The backwards cycle can then be performed by applying the inputs from L2 and integrating in L1.

[0023] FIG. 3 illustrates a weight update operation implemented using stochastic multiplication in some embodiments according to the invention. According to FIG. 3, the analog values x_i and δ_j can be converted into stochastic bit streams of length BL. The RPU weight can increase by Δw_o for each coincidence of pulses. The weight can be decreased by inverting the polarity of the pulses.

[0024] In the forward operation, an input vector is applied at the first layer (L^1). Each of the neurons in the next layer measures a weighted sum of the input and produces a nonlinear

activation function *f(z)* that will be the input of the following layer. The signal is propagated until the last layer (L^{OUT}), where the neuron with the highest output indicates the classification given by the network. A trained network can perform the forward operation to carry out classification tasks.

[0025] The training of the DNN can be performed by the backpropagation algorithm using three operations: 1) the forward operation; 2) the backwards cycle, in which the error measured at the output is given as an input to the last, output layer L^{out}, and allowed to propagate back to the first layer L^1 in the same manner as in the forward cycle; and 3) the weight update, performed by the rule:

$$w_{ij}^{L} = w_{ij}^{L} - \eta x_i^{L} \delta_j^{L+1}, \qquad (1)$$

where w_{ij}^{L} is the weight that connects the neuron i in layer L (pre-neuron) to the neuron j in layer $L+1$ (post-neuron), x_i is the pre-neuron output, δ_j^{L+1} is the back-propagated error fed by the post-neuron, and η is the learning rate.

[0026] FIG. 4 shows measurements of the ferroelectric polarization of an exemplary 10-nm thick TiN/HfZrO2/TiN capacitor, which can be partially polarized with pulses ranging from 2 to 3 volts, whereas pulses below 1 V can be used for readout in some embodiments. As shown in FIG. 5, measurements of the partial polarization of the exemplary capacitor show an exponential settling, similar to that observed with ions. Some approaches may consider an ideal saturation and only nonlinearities that keep the up/down symmetry, as shown in FIGs. 11A and 11B, respectively. In particular, as shown in FIG. 11A, the weights saturate at values ±wmax without affecting the shape of the curve before saturation, whereas in FIG. 11B nonlinearities provide a symmetric up/down behavior. In FIG 11C shows a typical nonlinear behavior observed in ferroelectrics and ionic memories, where the states evolve exponentially towards the saturation level. The up/down behavior is symmetric at zero polarization, and becomes asymmetric as the states approach the maximum polarization.

[0027] FIGs. 6-8 are cross-sectional views of an electronic device including a planar semiconductor opposite a partial-polarization material in some embodiments according to the

invention. As shown in the configurations of FIGs. 6-8 the device can be configured to appear similar to a MOSFET with an internally connected gate and drain in some embodiments. The polarization of the partial-polarization material can be altered by applying voltage pulses between the contacts at biases great enough to partially-polarize the ferroelectric or displace ions. The conductance state of the semiconductor channel can be determined (or "read out") using a low voltage that does not disturb the partial-polarization state so that the level of polarization remains substantially unchanged.

[0028] Figures 6-8 are cross sectional views of partial polarization electronic devices having planar channel regions in some embodiments according to the invention. According to Figure 6, the partial-polarization electronic device 100 includes a semiconductor material 105 that is generally planar and includes a channel region that extends between a drain contact 125 and a source contact 130 located on either end of the semiconductor material. It will be understood that the semiconductor material can be any semiconductor material that is suitable for use in resistive type electronic devices as described herein, such as silicon.

[0029] It will be further understood that each of the drain contact 125 and the source contact 130 can be ohmically connected to a respective portion of the semiconductor material 105. When the channel region is formed in the semiconductor material 105, a current can flow in the channel region from the drain 125 to the contact 130. It will be understood that the current in the channel region is responsive to a voltage that is applied between the source and drain contacts. The resistance provided by the channel region is responsive to the level of polarization of a partial-polarization material 110 that is on the semiconductor material 105 opposite the channel region. In particular, the partial-polarization material 110 is located on the semiconductor material 105 opposite the channel region and responds to a program voltage applied between the source contact 130 and the drain contact 125 to change the polarization of the partial-polarization material 110 as described above in reference to Figures 4 and 5. For example, in operation, a voltage can be applied to the drain contact 125 and the source contact 130 to change the level of polarization exhibited by the partial polarization material 110 as shown for example in Figure 4 and 5 above. Changing the level of polarization of the partial-polarization material 110 can set the resistance provided by the channel region in the semiconductor material 105.

[0030] As further shown in FIG.6, the gate contact 115 is located on the partial-polarization material 110 and is ohmically coupled to the drain contact 125 to place the electronic device 100 in what is sometimes referred to as a diode-connected configuration. Accordingly, when the voltage is developed across the source contact 130 and the drain contact 125, the voltage is also developed between the source contact 130 and the gate contact 115 which (when the voltage is great enough) can change the polarization level of the partial-polarization material 110. When the voltage is developed across the source contact 130 and the drain contact 125, the same voltage is provided to the gate contact 115. When a programming voltage level is provided between the source contact 130 and the drain contact 125/gate contact 115, the polarization level of the partial-polarization material 110 can modify the resistance of the channel region. When a read voltage is applied to the device 100, the current that flows in the channel region is responsive to the resistance, which indicates the polarization level of the partial-polarization material 110.

[0031] FIG.7 and 8 are cross sectional views of partial-polarization electronic devices 100 having first and second gate contacts which are provided in various ohmic configurations relative to the source and drain contacts in some embodiments according to the invention. According to FIG.7, the gate contact is separated into a first gate contact 115a and a second gate contact 115b, both of which are located on the partial-polarization material 110 and are electrically isolated from one another by a void 135. It will be understood that the void may include any material that electrically isolates the first gate contact 115a and the second gate contact 115b from one another.

[0032] As further shown in FIG. 7, the first gate contact 115a is ohmically connected to the drain contact 125 and the second gate contact 115a is ohmically connected to the source contact 130. As shown in FIG. 8, the first gate contact 115a is ohmically coupled to the source contact 130 whereas the second gate contact 115b is ohmically coupled to the drain contact 125.

[0033] FIG.9, 10A, and 10B are perspective views of partial-polarization electronic devices 100 including vertically protruding structures protruding from the semiconductor material which provide the channel region therein in some embodiments according to the invention. The partial-polarization material is provided in two separate layers located on opposite sides of the vertically protruding structure 150. In particular, a first partial polarization material 110a is

located on a first vertical side wall of the vertically protruding structure 150 and a second partial-polarization material 110b is located on an opposing vertical side wall of the vertically protruding structure 150. The first and second partial-polarization materials 110a-b extend upward from the base of the vertically protruding structure 150 to just short of an upper most portion of the vertically protruding structure 150.

[0034] As further shown in FIG. 9, a first drain contact 125a is located on the semiconductor material 105 and is ohmically coupled to a base of the first partial-polarization material 110a whereas a second drain contact 125b is ohmically coupled to a base of the second partial-polarization material 110b. The uppermost portion of the vertically protruding structure is covered by a source contact 130a which extends downward along the vertical side walls of the vertically protruding structure 150 to meet the respective upper most portions of the first and second partial-polarization materials 110a-b.

[0035] As further shown in Figure 9, first and second gate contacts 115a-b are located on the first and second the partial-polarization materials 110a and 110b, respectively and overlap the source contact 130a. Accordingly, the first and second gate contacts 115a-b are ohmically coupled to the source contact 130a and are electrically isolated from the first and second drain contacts 125a-b.

[0036] FIG.10A illustrates that, in some embodiments, the first and second drain contacts 125a-b can extend onto the vertical side walls of the first and second partial polarization materials 110a-b to also provide the first and second gate contacts 115a-b. As further shown in FIG.10A, the source contact 130a is electrically isolated from the first and second gate contacts 115a-b and the first and second drain contacts 125a-b.

[0037] FIG.10B is a perspective view of yet another embodiment of a vertically protruding structure configuration for a partial-polarization electronic device 100 in some embodiments according to the invention. According to FIG.10B, the first and second drain contacts 125-b are both ohmically connected to vertical side walls of the vertically protruding structure 150 and to the first and second partial-polarization materials 110a-b, which can extend over the first and second drain contacts 125a-b respectively to contact the semiconductor material 105. Still further, the upper portions of the first and second partial polarization materials 110a-b both extend onto the vertical side walls of the vertically protruding structure 150 above the first and

second drain contacts 125a-b. The source contact 130c extends over an uppermost portion of the vertically protruding structure 150 onto the outer surfaces of the first and second partial-polarization materials 110a-b.

EXPERIMENTAL DATA

[0038] The DNN in FIG. 1 was implemented for classification of a MNIST dataset of handwritten digits. The dataset included 60,000 images for training and 10,000 test images that were used to evaluate the network performance. The input layer size was 784 (28 × 28 pixels, normalized between 0 and 1), followed by two hidden layers with 256 and 128 neurons, and an output layer with 10 neurons for labels from 0 to 9. Sigmoid activation functions where used in hidden layers, softmax activations at the output and log-likelihood cost function. The base model was implemented with the ideal weight update in (1) and achieved 1.96% accuracy on the test set of FIG. 12.

[0039] For the stochastic weight update, see for example FIGs. 2 and 3, the following rule was used:

$$w_{ij}^{L} = w_{ij}^{L} \mp \Delta w_0 N, \qquad (2)$$

where N is the number of pulse coincidences, and Δw_0 is the nominal weight update. The exponential behavior shown in FIG. 11C was modeled by modifying the update rule as:

$$w_{ij}^{L} = w_{ij}^{L} \mp \Delta w_0 N \left(1 \pm \frac{w_{ij}^{L}}{w_{max}}\right), \qquad (3)$$

where w_{max} is the device saturation point. The number of levels used for the RPU can be determined by the largest Δw_0 and smallest w_{max} that can be simultaneously tolerated.

[0040] FIG. 13A shows the distribution of values in the weight matrices W^1, W^2 and W^3 obtained after training with the base model. Given that the input of a neuron was computed as the weighted sum of the output from the previous layer, the weights tend to scale inversely as the number of elements on each layer.

[0041] In some embodiments, the weighted sums performed by the neurons can be scaled in the forward and backward cycles, which can be done at the hardware level by changing the gain

during integration. The weight update can be scaled back by changing the scale factors in the random bit generators. Both scaling steps are common to all neurons in a layer and do not affect the parallel operation. Two layer-normalization rules were tested, as depicted in FIGs. 13B and 13C. The normalization is transparent to the algorithm, so all cases achieve the same performance when the ideal update rule in (1) is applied, and only become relevant when nonidealities are introduced.

[0042] FIGs. 14A-C shows the performance of the neural network after 30 epochs (*i.e.* iterations of the 60,000 training images), measured as the percentage of mislabeled images. The three scale rules were evaluated using the update rule in equation (3) with $\Delta w_o = 0.001$, 0.005 and 0.01, and saturation values w_{max} from 1 to 100. The scaling effectively reduces the impact of saturation for small w_{max}, but is also more sensible to the nominal weight update Δw_o: for $\Delta w_o = 0.001$, the scale rule 2 achieves a lower error for the full range of w_{max}, whereas when $\Delta w_o = 0.01$ the unscaled weights perform better for $w_{max} > 10$. Scale rule 1 achieves a good overall performance due to the balance between these two factors: a test error of 2.14% is obtained with 4000 levels, and 2.62% with 1000 levels.

[0043] In all cases, the error on the training set is monotonically reduced as the saturation w_{max} increases, but in some embodiments, the test error can increase for large values of w_{max}. This is caused by the non-systematic asymmetry that favors updates towards zero than to the extreme values $\pm w_{max}$, and therefore disfavors large weights. When the nominal weight update is small enough not to affect the performance ($\Delta w_o = 0.001$), an improvement over the base model was observed due to this regularization effect.

[0044] A measure of the non-systematic asymmetry obtained with the update rule in equation (3) is shown in FIG. 15, computed for a weight $w_{ij}^{L} = 1$, which is within the range of weight values obtained after training. A 20% up/down asymmetry is obtained for $w_{max} = 10$ and up to 50% for $w_{max} = 5$, with errors of 2.14 and 2.62% respectively.

[0045] FIG. 16 is a table listing prior art RPU specifications used to achieve a 2.3% test accuracy, compared with the polarization-based RPU in some embodiments according to the invention. According to FIG. 16, simulations of a symmetric PCM show a 3% test error, but real devices show a large systematic asymmetry that limits practical use. The CMOS based RPU

achieves a good performance, but it is limited by the capacitor leakage, which requires a large area. The proposed polarization-based RPU can achieve a low error and tolerate a large non-systematic asymmetry while keeping realistic area constrains. The device area for the polarization-based RPU is estimated assuming a planar ferroelectric with a grain dimension of 10 nm. A smaller area could be achieved with ions or in a vertical fin structure.

[0046] In the drawings, the shapes and dimensions of elements may be exaggerated for clarity, and the same reference numerals will be used throughout to designate the same or like elements. It will be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and, similarly, a second element could be termed a first element, without departing from the scope of the present invention. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0047] It will be understood that when an element such as a layer, region or substrate is referred to as being "on" or extending "onto" another element, it can be directly on or extend directly onto the other element or intervening elements may also be present. In contrast, when an element is referred to as being "directly on" or extending "directly onto" another element, there are no intervening elements present. It will also be understood that when an element is referred to as being "connected" to another element, it can be directly connected to the other element or intervening elements may be present. In contrast, when an element is referred to as being "directly connected" to another element, there are no intervening elements present. Other words used to describe the relationship between elements should be interpreted in a like fashion (i.e., "between" versus "directly between", "adjacent" versus "directly adjacent", etc.).

[0048] Relative terms such as "below" or "above" or "upper" or "lower" or "horizontal" or "vertical" may be used herein to describe a relationship of one element, layer or region to another element, layer or region as illustrated in the figures. It will be understood that these terms are intended to encompass different orientations of the device in addition to the orientation depicted in the figures.

[0049] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an"

and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" "comprising," "includes" and/or "including" when used herein, specify the presence of stated elements but do not preclude the presence or addition of one or more other elements.

[0050] While exemplary embodiments have been shown and described above, it will be apparent to those skilled in the art that modifications and variations could be made without departing from the spirit and scope of the present disclosure as defined by the appended claims.

Notre Dame Reference No.: 18-002
MARCO Reference: P1742
W&S Reference: 170964-00006

WHAT IS CLAIMED:

1. An electronic device comprising:

a semiconductor material including a channel region configured to conduct a current;

a source contact electrically coupled to the channel region at a first location;

a drain contact electrically coupled to the channel region at a second location spaced apart from the first location;

a partial-polarization material on the semiconductor material between the source contact and the drain contact opposite the channel region; and

a gate contact on the partial-polarization material opposite the channel region and ohmically coupled to the drain contact or ohmically coupled to the source contact.

2. The electronic device of Claim 1 wherein the gate contact comprises a first gate contact extending on the partial-polarization material to about half a length of the channel region, the electronic device further comprises:

a second gate contact extending on the partial-polarization material opposite the channel region and ohmically coupled to the source contact; and

a void between the second gate contact and the first gate contact on the partial-polarization material at about a mid-point of the length of the channel region, the void electrically isolating the second gate contact from the first gate contact.

3. The electronic device of Claim 2 wherein the source contact is ohmically coupled to the first gate contact and the drain contact is ohmically coupled to the second gate contact.

4. The electronic device of Claim 1 wherein the partial-polarization material comprises a ferroelectric material.

5. The electronic device of Claim 1 wherein the partial-polarization material comprises doped hafnium oxide.

6. The electronic device of Claim 1 wherein the partial-polarization material comprises an ion doped polymer.

7. The electronic device of Claim 1 wherein the semiconductor material includes a vertically protruding structure providing the channel region and having opposing first and second vertical side walls, and the source contact comprises a first source contact the electronic device further comprising:

wherein the partial-polarization material is located on side walls the vertically protruding structure to provide first and second vertical side walls of the partial-polarization material

a second source contact is on the semiconductor material and ohmically coupled to a base of the second vertical side wall opposite the first source contact on the semiconductor material and ohmically coupled to a base of the first vertical side wall; and

wherein the drain contact is ohmically coupled to an uppermost surface of the vertically protruding structure and extends on the first and second vertical side walls of the partial-polarization material to provide the gate contact.

8. The electronic device of Claim 1 wherein the semiconductor material includes a vertically protruding structure providing the channel region and having opposing first and second vertical side walls, and the source contact comprises a first source contact the electronic device further comprising:

wherein the partial-polarization material is located on side walls the vertically protruding structure to provide first and second vertical side walls of the partial-polarization material

a second source contact is on the semiconductor material and is ohmically coupled to a base of the second vertical side wall opposite the first source contact and extends on the second vertical side wall toward the drain contact to provide a second gate contact;

wherein the first source contact is on the semiconductor material and is ohmically coupled to a base of the first vertical side wall and extends on the first vertical side wall toward the drain contact to provide a first gate contact; and

wherein the drain contact is ohmically coupled to an uppermost surface of the vertically protruding structure.

9. The electronic device of Claim 1 wherein the channel region is planar.

10. The electronic device of Claim 1 wherein the source contact is ohmically coupled to the channel region at the first location and the drain contact is ohmically coupled to the channel region at the second location.

11. A method of programming an electronic device, the method comprising:

applying a programming voltage pulse or sequence of pulses between a drain contact and a source contact to set a partial-polarization state of a partial-polarization material opposite a channel region of the electronic device; and

applying a read voltage pulse to generate a current in the channel region responsive to the partial-polarization state.

12. The method of Claim 11 wherein the electronic device comprises:

a semiconductor material including the channel region;

the source contact electrically coupled to the channel region at a first location;

the drain contact electrically coupled to the channel region at a second location spaced apart from the first location;

the partial-polarization material on the semiconductor material between the source contact and the drain contact opposite the channel region; and

a gate contact on the partial-polarization material opposite the channel region and ohmically coupled to the drain contact or ohmically coupled to the source contact.

13. The method of Claim 12 wherein the gate contact comprises a first gate contact extending on the partial-polarization material to about half a length of the channel region, the electronic device further comprises:

a second gate contact extending on the partial-polarization material opposite the channel region and ohmically coupled to the source contact; and

a void between the second gate contact and the first gate contact on the partial-polarization material at about a mid-point of the length of the channel region, the void electrically isolating the second gate contact from the first gate contact.

14. The method of Claim 13 wherein the source contact is ohmically coupled to the first gate contact and the drain contact is ohmically coupled to the second gate contact.

15. The method of Claim 12 wherein the partial-polarization material comprises a ferroelectric material.

16. The method of Claim 12 wherein the partial-polarization material comprises doped hafnium oxide.

17. A neural network circuit comprising:

a plurality of serially connected layers of the neural network circuit, wherein each layer includes a respective plurality of neurons;

a plurality of resistive processing cross-bar circuits, wherein each of the resistive processing cross-bar circuits is connected between directly adjacent ones of the serially connected layers to provide a respective weighting to data provided from the respective plurality of neurons included in an upstream layer of the neural network circuit that is summed by a down-stream layer of the neural network circuit; and

each of the resistive processing cross-bar circuits comprises a respective array of partial-polarization electronic devices each being arranged in a diode-connected configuration.

18. The neural network circuit of Claim 17 wherein each of the partial-polarization electronic devices in the array comprises:

a semiconductor material including a channel region configured to conduct a current;

a source contact electrically coupled to the channel region at a first location, the source contact comprising a first terminal of the diode-connected configuration;

a drain contact electrically coupled to the channel region at a second location spaced apart from the first location, the drain contact comprising a second terminal of the diode-connected configuration;

a partial-polarization material on the semiconductor material between the source contact and the drain contact opposite the channel region; and

a gate contact on the partial-polarization material opposite the channel region and ohmically coupled to the drain contact.

19. The neural network circuit of Claim 18 wherein the gate contact comprises a first gate contact extending on the partial-polarization material to about half a length of the channel region, the electronic device further comprises:

a second gate contact extending on the partial-polarization material opposite the channel region and ohmically coupled to the source contact; and

a void between the second gate contact and the first gate contact on the partial-polarization material at about a mid-point of the length of the channel region, the void electrically isolating the second gate contact from the first gate contact.

20. The neural network circuit of Claim 19 wherein the source contact is ohmically coupled to the first gate contact and the drain contact is ohmically coupled to the second gate contact.

21. The neural network circuit of Claim 18 wherein the partial-polarization material comprises a ferroelectric material.

22. The neural network circuit of Claim 18 wherein the partial-polarization material comprises doped hafnium oxide.

Notre Dame Reference No.: 18-002
MARCO Reference: P1742
W&S Reference: 170964-00006

ABSTRACT

An electronic device can include a semiconductor material including a channel region configured to conduct a current, a source contact electrically coupled to the channel region at a first location, a drain contact electrically coupled to the channel region at a second location spaced apart from the first location, a partial-polarization material on the semiconductor material between the source contact and the drain contact opposite the channel region and a gate contact on the partial-polarization material opposite the channel region and ohmically coupled to the drain contact or ohmically coupled to the source contact.

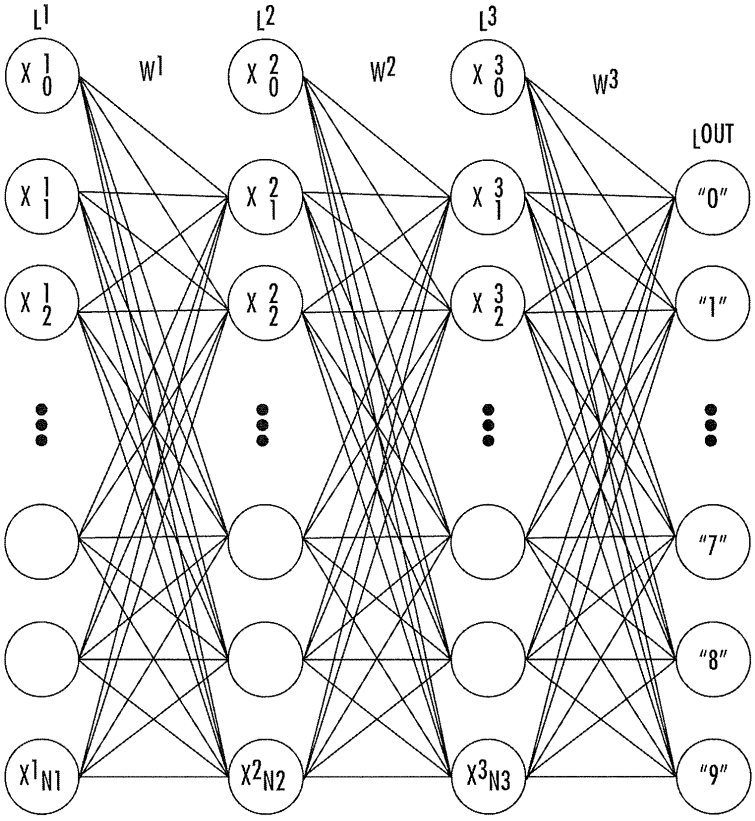
L1
L2
L3
LOUT
W1
W2
W3
X 1 0
X 1 1
X 1 2
X1N1
X 2 0
X 2 1
X 2 2
X2N2
X 3 0
X 3 1
X 3 2
X3N3
"0"
"1"
"7"
"8"
"9"

FIG. 1

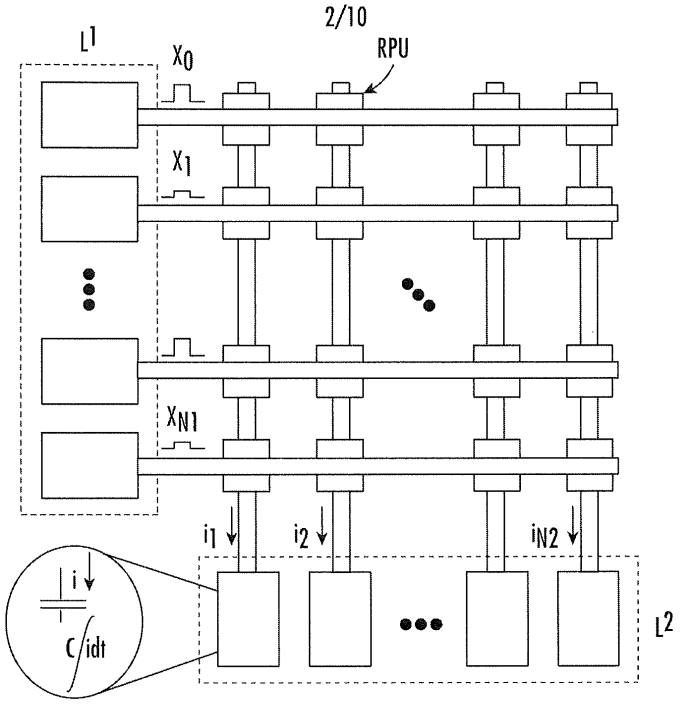
2/10
L1
X0
RPU
X1
XN1
i1
i2
iN2
i
C/idt
L2

FIG. 2

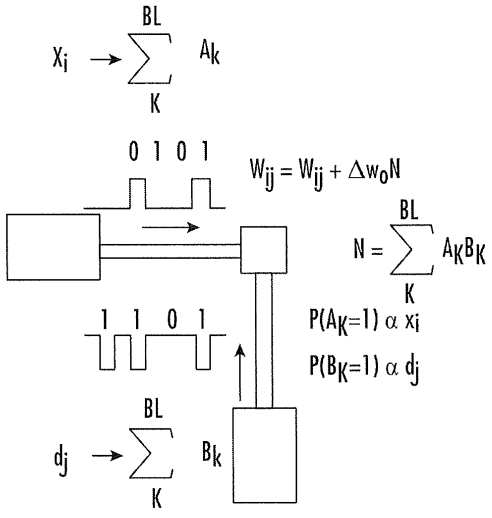
BL
Xi
Ak
K
0 1 0 1
Wij = Wij + Δw0N
BL
N = AKBK
K
1 1 0 1
P(AK=1) α xi
P(BK=1) α dj
BL
dj
Bk
K

FIG. 3

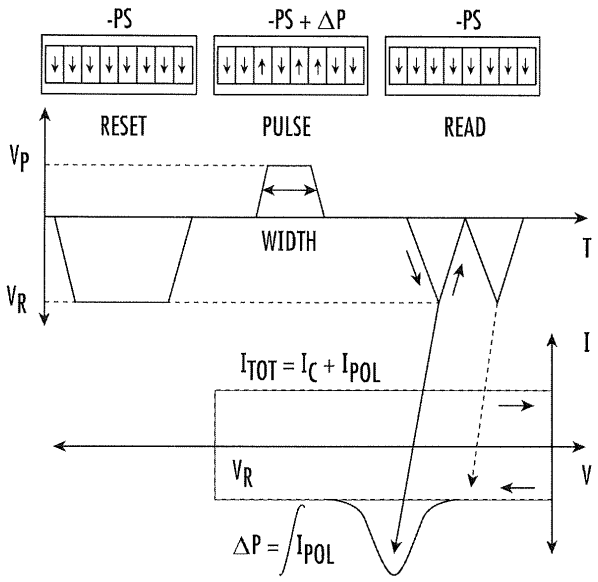
-PS
-PS + ΔP
-PS
RESET
PULSE
READ
VP
WIDTH
T
VR
I
ITOT = IC + IPOL
VR
V
ΔP = ∫IPOL

FIG. 4

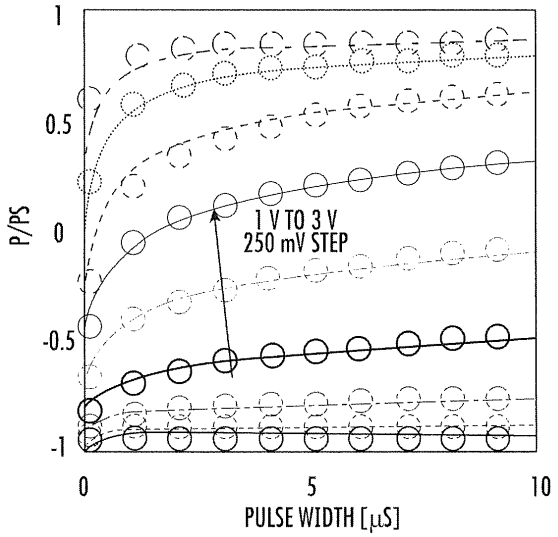
1
0.5
0
-0.5
-1
P/PS
1 V TO 3 V
250 mV STEP
0
5
10
PULSE WIDTH [μS]

FIG. 5

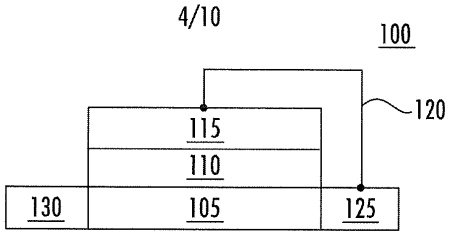

FIG. 6

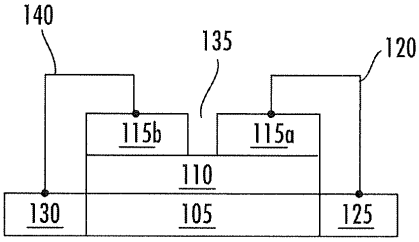

FIG. 7

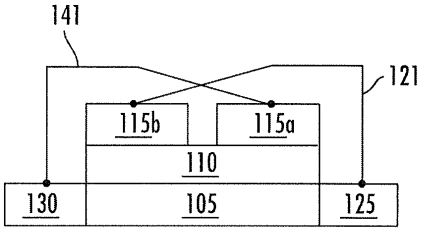

FIG. 8

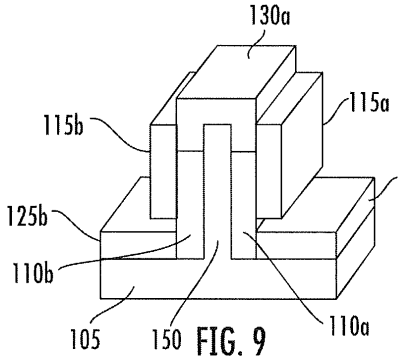

FIG. 9

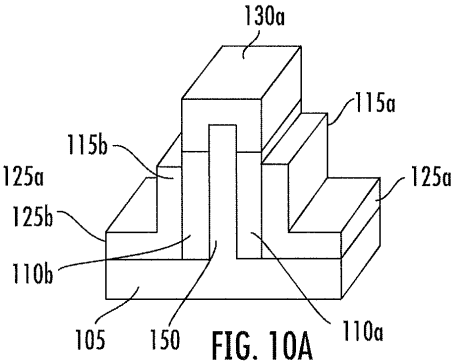

FIG. 10A

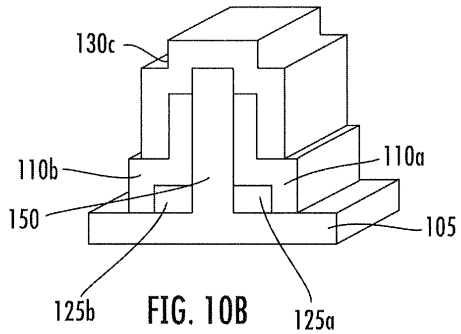

FIG. 10B

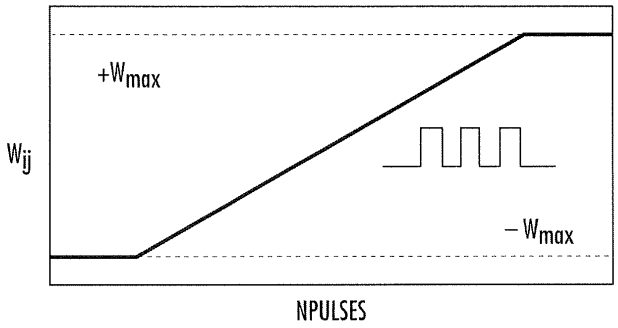

FIG. 11A

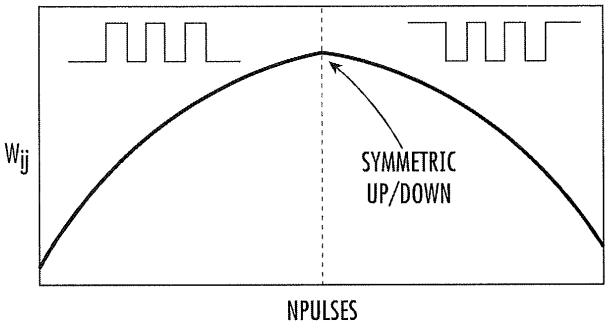

FIG. 11B

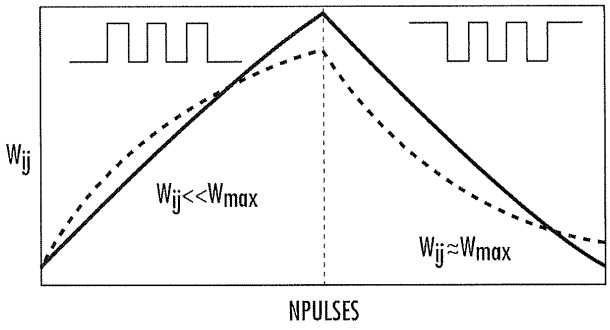

FIG. 11C

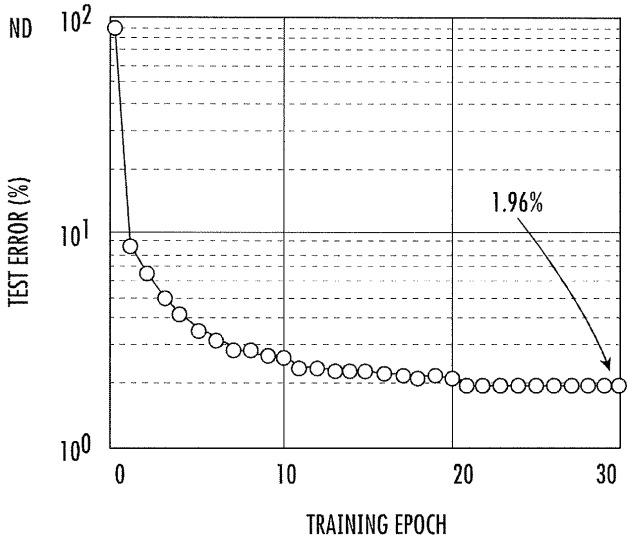

FIG. 12

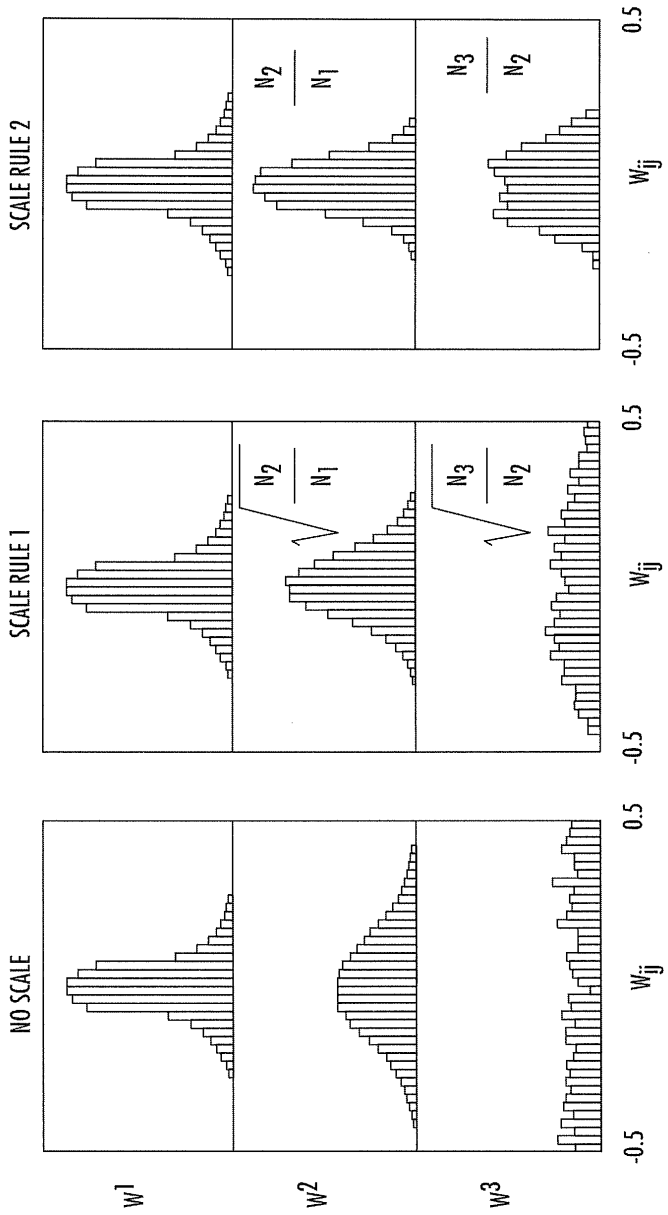

FIG. 13A

FIG. 13B

FIG. 13C

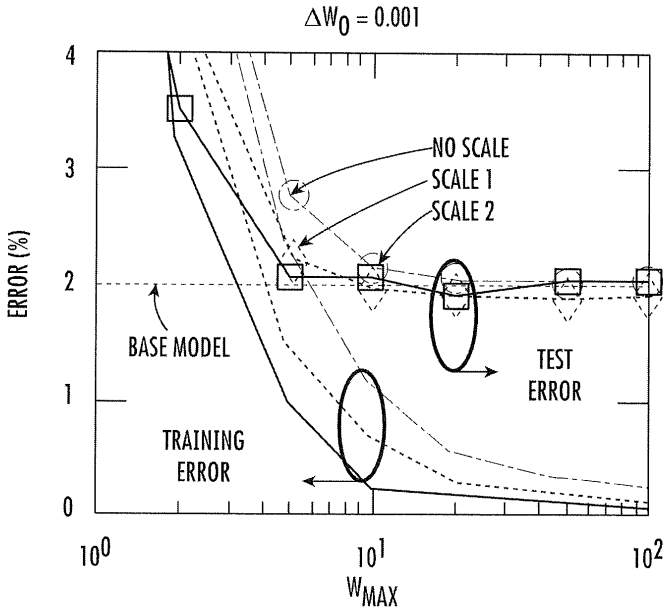

FIG. 14A

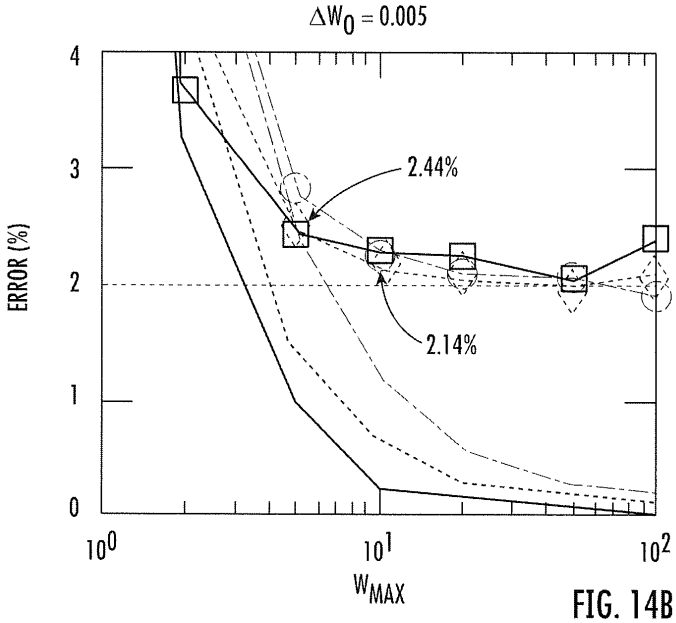

FIG. 14B

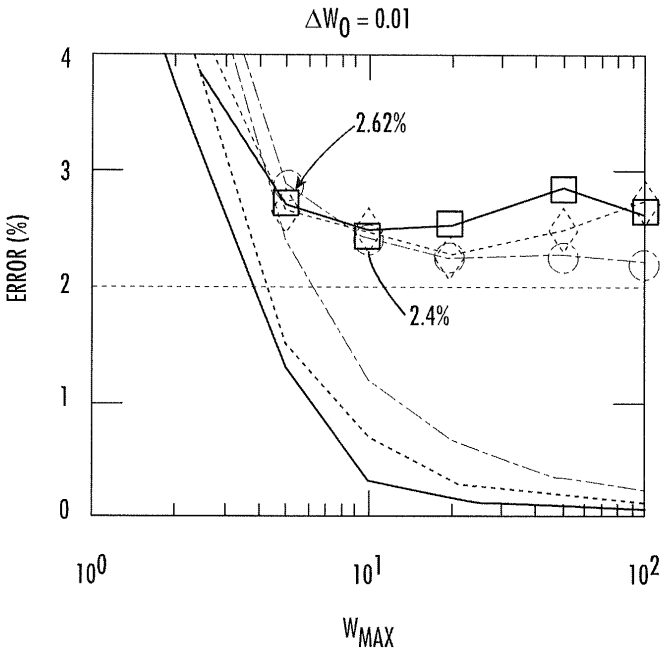

FIG. 14C

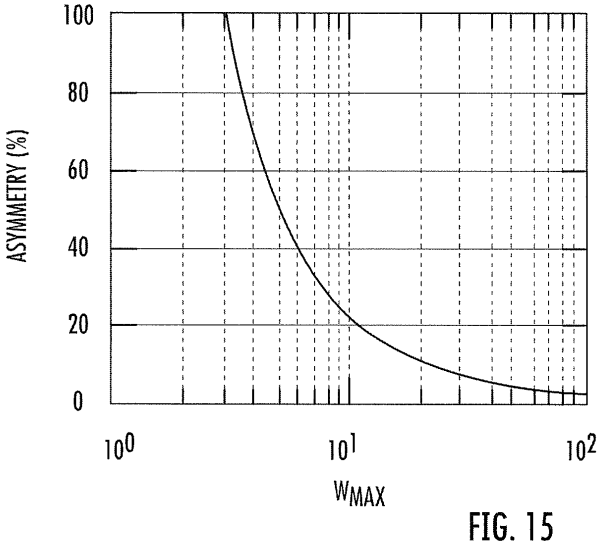

FIG. 15

	IDEAL RPU [3]	PRPOSED RPU		PCM [10]		
TEST ERROR [%]	2.3	2.14	2.62	3	20	CMOS RPU [9]
LEVELS	>1000	4000	1000	~1000		2-3
ASYMMETRY	BELOW 5%	20 TO 50%		NONE	LARGE	>1000
AREA [μM^2]	0.04	0.36	0.09	NOT SPECIFIED		LARGER

FIG. 16

APPENDIX C

COUNTER-BASED NEURAL NETWORK ARCHITECTURE WITH RATE-WIDTH MULTIPLICATION

Deep learning models are challenging to implement in embedded devices due to their high computational complexity, which is dominated by memory access and multiply accumulate operations (MAC). Several hardware accelerators have been proposed to reduce the computational cost of DNNs for embedded devices [3, 10]. These approaches exploit data reuse by storing data in local buffers, and parallel hardware for efficient MAC operations. Furthermore, the resolution of MAC operations can be reduced to minimize energy consumption, although this requires configurable resolution to meet the requirements of different models [13, 124]. In this appendix, an architecture to perform reduced-precision vector multiply accumulate operations is developed by encoding the factors as pulse width and frequency signals. With this approach, the multiplication itself is performed by an AND gate and a counter. The effort of the signal encoding is mitigated by performing vector operations, where each encoded signal is used for many multiplications. A statistical analysis of the bias and variance of the proposed method is presented and compared against integer multiplication and multiplication based on stochastic bit streams. Experimental results with the MNIST dataset show that the proposed architecture achieves equivalent or better performance over integer multiplication with the same bit resolution, while potentially reducing the area and energy consumption. Furthermore, precision and throughput can be dynamically traded to meet different deep learning model requirements.

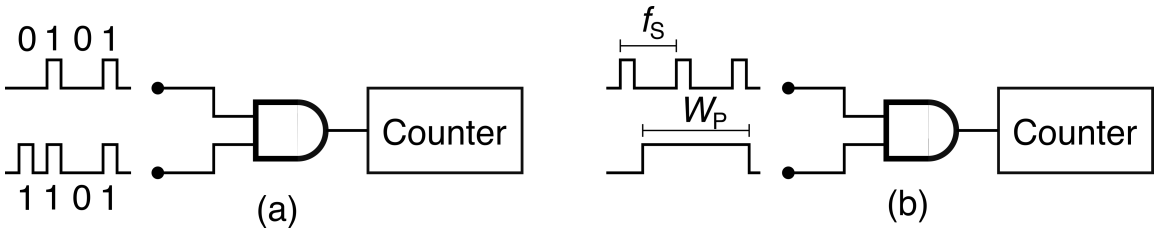

Figure C.1. Basic structure of (a) multiplication by stochastic pulses and (b) width-rate multiplication.

Variable-precision MAC operations can be implemented by stochastic multiplication [26, 110, 125–127]. The basic operation of this method is depicted in Fig. C.1a. The two factors are encoded as a stream of stochastic pulses with the same length, where the probability of each pulse to be asserted is proportional to the factor value. The encoded signals are fed to an AND gate and a counter, which produces an output proportional to the desired multiplication [21, 26]. Although this method produces an unbiased multiplication, it suffers from a large variance.

The proposed architecture is depicted in Fig. C.1b for two scalar factors. One of the factors is encoded as a single pulse with variable width W_P and the other factor is encoded as a sequence of short pulses with variable frequency f_S. The number of counts is proportional to $W_P \times f_S$, and the output of a sequence of inputs can be accumulated in the counter.

The basic structure can be replicated as shown in Fig. C.2 to perform vector-scalar multiplication. Matrix-vector multiplication is performed with the same structure by sequentially feeding the elements of the vector and the columns of the matrix. Replicating this structure vertically, as depicted in Fig. C.3, enables both vector outer product and matrix multiplication.

In Section C.1 the multiplication mechanisms depicted in Fig. C.1a and C.1b are analyzed to evaluate the multiplication bias (mean) and noise (variance). In Section C.2, the proposed architecture is extended to perform signed multiplication

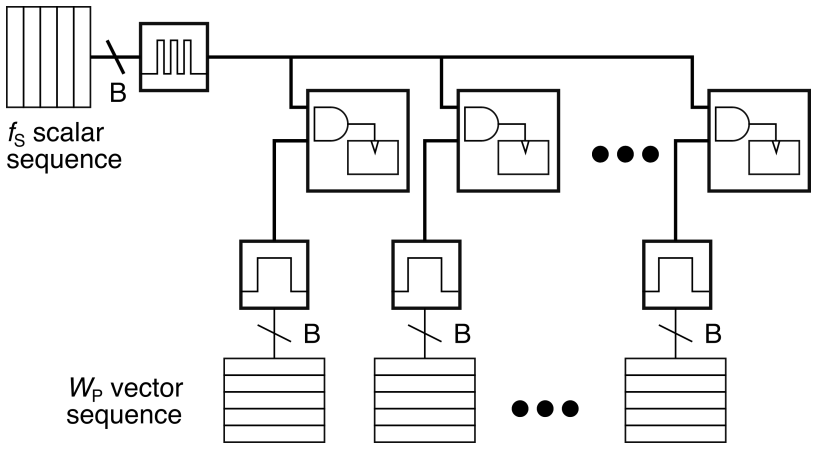

Figure C.2. Structure for vector-scalar multiplication and matrix-vector MAC.

with dynamic fixed point precision. In Section C.3 simulated results are provided to evaluate the performance of DNN training with integer multiplication, stochastic multiplication and the proposed architecture for different MAC resolutions.

C.1 Statistical analysis

Consider two factors α and β to be unsigned integers with length B bits. The product $\gamma = \alpha \times \beta$ is an unsigned integer, with length up to $2B$. Consider also a minimum pulse width T_{min} and maximum clock rate f_c, which are related by $T_{min} = 1/f_c$.

C.1.1 Stochastic pulse multiplication

The two factors are encoded as bit streams of length N_L, where each bit has a probability of being asserted given by

$$\alpha \rightarrow P(A_k = 1) = C_1\alpha \tag{C.1}$$

$$\beta \rightarrow P(B_k = 1) = C_2\beta, \tag{C.2}$$

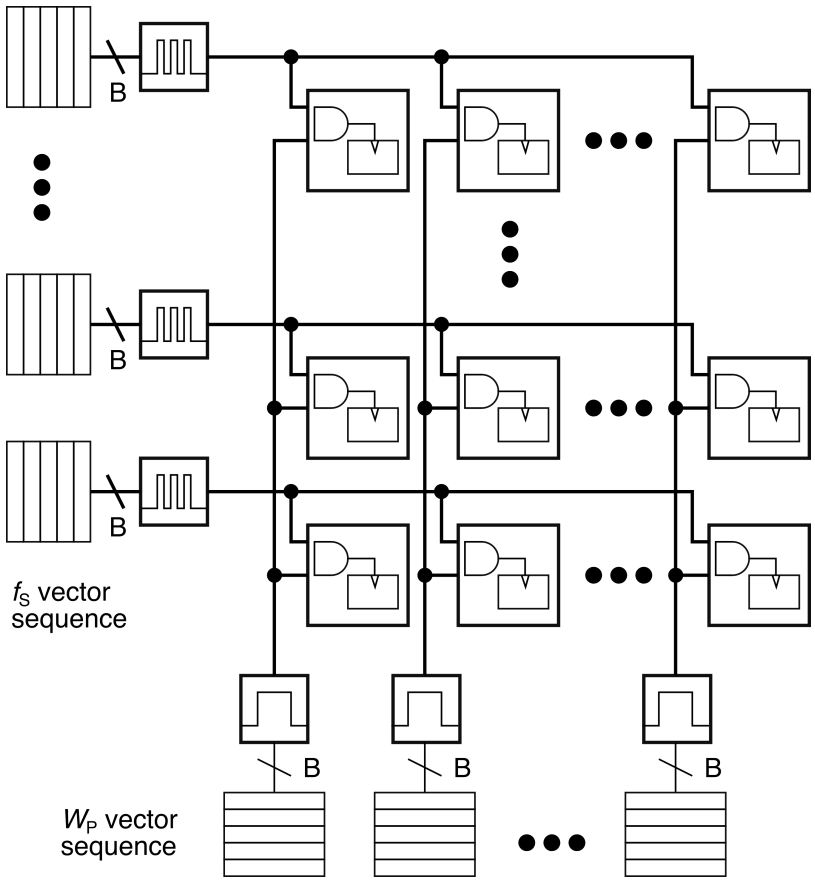

Figure C.3. Structure for vector outer product and matrix multiplication.

where C_1 and C_2 are proportionality constants. The number of pulse coincidences N is computed as

$$N = \sum_{k=1}^{N_L} A_k \wedge B_k, \tag{C.3}$$

which is a sequence of N_L Bernoulli trials with probability $P(A_k \wedge B_k) = \gamma C_1 C_2$, so its mean and variance are given by

$$\mu_N = \gamma N_L C_1 C_2 \tag{C.4}$$

$$\sigma_N^2 = \mu_N \left(1 - \mu_N / N_L\right). \tag{C.5}$$

The time required to perform the stochastic multiplication is $N_L T_{min}$.

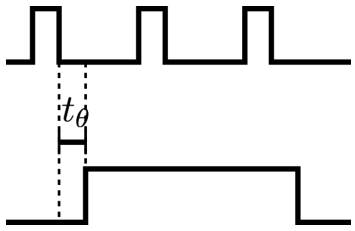

Figure C.4. Phase difference between the frequency- and width-modulated signals.

C.1.2 Rate-width multiplication

Considering the maximum frequency f_c, the frequency corresponding to the least significant bit (LSB) of the frequency modulated signal is $f_c 2^{-B}$. The two factors are encoded as

$$\alpha \rightarrow fs = \alpha f_c 2^{-B} \tag{C.6}$$

$$\beta \rightarrow T_p = \beta T_{min} = \beta / f_c \tag{C.7}$$

Consider the timing diagram depicted in Fig. C.4, where the time difference between the frequency- and width-modulated signals is defined as $t_\theta \in (0, 1/f_s)$. The number of pulses is given by

$$\begin{aligned} N &= \lfloor (\beta/f_c + t_\theta)\, \alpha f_c 2^{-B} \rfloor \\ &= \lfloor \gamma 2^{-B} + \theta \rfloor \end{aligned} \tag{C.8}$$

where $\lfloor x \rfloor$ denotes the integer part of x and $\theta \in (0, 1)$. If the signals have non-synchronized clock sources, they will have a random phase, so $\theta \sim U(0, 1)$. This

results in stochastic rounding [78], so its mean and variance are given by

$$\mu_N = \gamma 2^{-B} \tag{C.9}$$

$$\sigma_N^2 = \Delta(1-\Delta), \tag{C.10}$$

where $\Delta = \gamma 2^{-B} - \lfloor \gamma 2^{-B} \rfloor$. For example, $\gamma 2^{-B} = 3.5$ will be rounded up with probability 0.5 and variance $\sigma^2 = 0.25$, whereas $\gamma 2^{-B} = 3.2$ will be rounded up with probability 0.2 and variance $\sigma^2 = 0.16$. The time required to perform the multiplication is $2^B T_{min}$. To make a fair comparison with stochastic multiplication, we set $N_L = 2^B$ so both multiplication methods use the same number of clock cycles. With $C_1 = C_2 = 2^{-B}$, the mean value for both methods is the same, but the variance is significantly larger for stochastic multiplication. According to Eq. (C.9), scaling N by 2^B gives an unbiased product $\hat{\gamma}$, where the B least significant bits will be zero due to quantization.

C.2 Signed multiplication with dynamic fixed point precision

The proposed approach can be extended to perform signed multiplication with dynamic fixed point precision. Consider first the case for integer multiplication as depicted in Fig. C.5a. The data is stored with resolution B_S plus the sign bit and the multiplications are performed with resolution B. The point position is set as a configuration parameter for each factor independently (p_1 and p_2), but kept constant throughout the MAC operations. For each multiplication, the multiplier receives the B most significant bits of the data and performs an integer multiplication. The sign of the multiplication is determined by the exclusive OR of the sign bits, and the accumulator is updated accordingly. Once all the MAC operations have been executed, the output is read as an integer with resolution B_S and point position p_o (also a configuration parameter). This is done by shifting right the accumulated

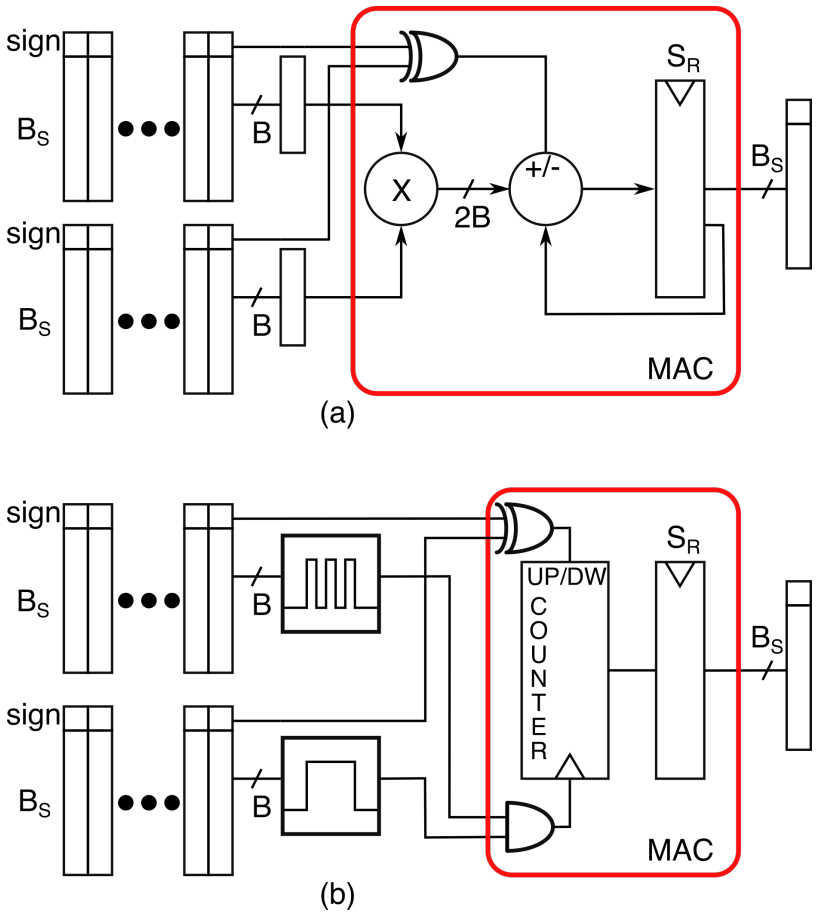

Figure C.5. Block diagram for signed multiplication with dynamic fixed point precision with (a) integer MAC unit and (b) rate-width multiplication. The number of shift right bits S_R for readout depends on the configuration parameters B_S, B, the inputs point position (p_1 and p_2) and the output point position p_o.

output by $S_R = 2\,(B_S - B) - p_1 - p_2 + p_o$ and reading the B_S least significant bits. If any bit higher than B_S is 1, the result overflows and is set to its maximum value.

Consider now the case for rate-width multiplication, depicted in Fig.C.5b. The B most significant bits of the factors are encoded and fed to the AND gate as before. An UP/DOWN counter is used for the accumulator, and its direction is controlled by the exclusive OR of the sign bits. According to Eq. (C.8), the result of the multiplication is already scaled by 2^{-B}, so the output is now shifted by $S_R = 2B_S - B - p_1 - p_2 + p_o$ for readout. Note that the rate-width multiplication requires a smaller accumulator than integer multiplication because the B least significant bits are stochastically rounded at each multiplication. Also note that the sign bit doubles the dynamic range, but does not require a longer N_L. The implementation of variable point multiplication for stochastic pulses is equivalent to that of rate-width multiplication. Finally, for practical neural network implementations it is sometimes convenient to scale the accumulated sums. This can also be done by adding an additional output scaling parameter S_o, which is added to the shift right S_R.

C.3 Experimental evaluation

To evaluate the inference and training performance of the proposed architecture, we use the MNIST dataset of handwritten digits [68], comprised of gray-scale images of handwritten digits, size-normalized and centered in a fixed-size 28×28 image. This dataset has a training set of 60,000 examples, and a test set of 10,000 examples. Although it is considered a low-complexity task, it is well benchmarked and the models used for this dataset are manageable for simulations. The neural network in Fig. 1.5 was implemented with and input layer size of 784 (determined by the input), hidden layers of 256 and 128 elements with ReLu activations, and an output layer with 10 categories. The output activations are sigmoid functions with log-likelihood loss function.

The weights, biases and activations are stored as integer values with resolution B_S plus the sign bit. Each layer has 3 parameters to define the point position for its activations, weights and biases. The multiplications were implemented with integer, stochastic and rate-with multiplication with resolution $B \leq B_S$. The same multiplication precision is implemented for the forward propagation, back propagation and weight update. The learning rate is defined as

$$\eta = 2^{-\eta_1 - \eta_2}, \tag{C.11}$$

where η_1 is a user-defined parameter and $\eta_2 = \lfloor N_{batch} \rfloor$. With this definition, the learning rate can be applied directly by shift operations with the parameter S_o, and accounts for the size of the minibatch N_{batch}.

The training set of 60,000 images was divided into 10 stratified sets of 6,000 images for a 10-fold cross validation [6]. The training was performed with gradient descent for 40 epochs in minibatches of $N_{batch} = 36$ images, with $\eta_1 = 2$ for the first 20 epochs and $\eta_1 = 3$ for epochs 21 to 40. The weights were initialized to be in the range

$$W^L \in \left(-\sqrt{\frac{6}{N^{L-1} + N^L}}, \sqrt{\frac{6}{N^{L-1} + N^L}} \right). \tag{C.12}$$

The equivalent integer range was determined according to Eq. (C.12) and the weight point position W_p^L, and integer values were drawn from a uniform distribution. The same random seed was used for all training initializations. The biases for ReLu layers 1 and 2 were initialized to 2^{-3}, whereas the biases for layer 3 were initialized to 0.

A baseline model was tested with $B_S = 24$ and integer multiplication with the same resolution. The point position was set to 16 for all the weights, activations and biases. Two reduced precision models were implemented: RP1 with $B_S = 12$ and RP2 with $B_S = 8$. For these models, the three multiplication methods were tested with resolution varying from 3 to 8 bits. The point positions were tuned for

TABLE C.1

PARAMETERS FOR BASELINE DNN MODEL AND TWO REDUCED MODELS (RP1 AND RP2).

Parameter	Baseline	RP1	RP2
B_S	24	12	8
B	24	3-8	3-8
A_p^L	16,16,16,16	12,9,8,7	8,5,4,3
W_p^L	16, 16, 16	13,12,11	13,12,11
b_p^L	16, 16, 16	12,12,12	12,12,12
D_p^L	16, 16, 16	12,12,12	12,12,12

each layer to minimize overflow and underflow, and are shown in Table C.1. The parameters are storage resolution B_S, multiplication resolution B, activations point position A_p^L (layers 0 to 3), weights point position W_p^L (layers 1 to 3), bias point position b_p^L (layers 1 to 3) and backpropagated error point position D_p^L (layers 1 to 3). Note that the storage resolution is only relevant for training. For inference, the storage resolution is reduced to the multiplication resolution.

The test error obtained with 10-fold cross validation is shown in Fig. C.6. The reduced precision networks with $B_S = 12$ and $B_S = 8$ are compared against the baseline model. The mean error is plotted with 95% confidence intervals for the different multiplication methods. There is no statistically significant difference between rate-width multiplication for $B_S = 12$ and B ranging from 4 to 8 bits, whereas rate-with is statistically significantly better for $B = 3$. With a storage resolution $B_S = 8$, the rate-width multiplication is better for multiplication resolutions ranging from 3 to 5 bits. When compared with stochastic multiplication, the difference is statistically

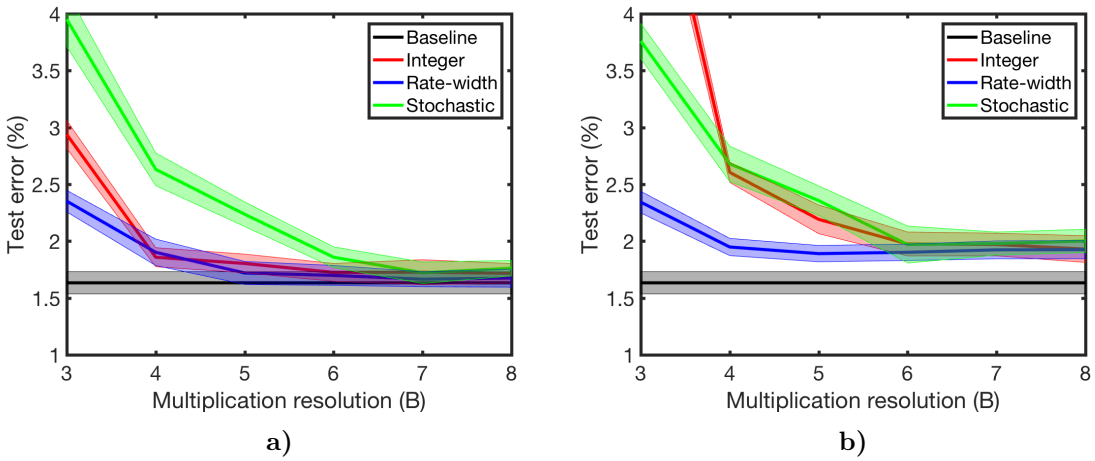

Figure C.6. Test error for 10-fold cross validation and different multiplication methods. Reduced precision networks with (a) $B_S = 12$ and (b) $B_S = 8$ compared with baseline model. The mean error is plotted with 95% confidence intervals.

significant for multiplication resolutions ranging from 3 to 6 bits. Overall, the rate-width multiplication method achieves the same or better accuracy than the integer and stochastic multiplication with the same bit resolution.

C.4 Conclusion

An architecture to implement reduced precision vector MAC operations was proposed and evaluated. The MAC hardware is reduced to AND and XOR logic gates and a counter in an array structure, which enables the implementation of signed matrix multiplication with dynamic fixed point precision. Experimental results show that the proposed architecture achieves equivalent or better accuracy than integer multiplication with the same bit resolution, while potentially reducing the area and energy consumption. Furthermore, precision and throughput can be dynamically traded to meet different deep learning model requirements. An analysis of the transistor-level implementation is required to evaluate the area and energy efficiency

of the proposed architecture as a function of the array size and the multiplication resolution, which is proposed as future work.

APPENDIX D

RECONFIGURABLE ELECTRIC DOUBLE LAYER DOPING IN AN MoS_2 NANORIBBON TRANSISTOR

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Reconfigurable Electric Double Layer Doping in an MoS_2 Nanoribbon Transistor

Cristobal Alessandri, Sara Fathipour, Huamin Li, Iljo Kwak, Andrew Kummel, Maja Remškar, and Alan C. Seabaugh

***Abstract*— A back-gated multilayer nanoribbon molybdenum disulfide (MoS_2) transistor grown by chemical vapor transport and doped using polyethylene oxide cesium perchlorate is fabricated and characterized. Ions in the polymer dielectric are directed by side gates to the source and drain access regions where they form electric double layers (EDLs) that control the carrier densities. This allows the junctions of the same transistor channel to be reconfigured as an n-MOSFET, p-MOSFET, and as a tunnel field-effect transistors. The EDLs are formed at room temperature and then locked into place by cooling the polymer below the glass transition temperature (~240 K). Transport measurements are presented and explained using simulated band diagrams. Both n and p-conduction in MoS_2 is demonstrated using solid polymer ion doping, enabling characterization of a semiconductor in which the doping of the same channel has been reconfigured to form three different transistor configurations.**

***Index Terms*— Electric double layer, ion doping, molybdenum disulfide, multilayer nanoribbon molybdenum disulfide (MoS_2), tunnel field-effect transistor (FET) (TFET), TFET.**

I. Introduction

2-D SEMICONDUCTORS are being widely explored for beyond-CMOS electronics [1]. Electric double layers (EDLs) formed using solid polymers, such as polyethylene oxide (PEO) containing cesium perchlorate ($CsClO_4$), and 2-D crystals can induce degenerate sheet electron and hole densities exceeding 1×10^{14} cm^{-2} [2], [3], beyond the limits of substitutional doping in bulk semiconductors. Using PEO:$CsClO_4$, n-contact resistance as low as 200 Ω μm has been achieved in multilayer multilayer nanoribbon molybdenum disulfide (MoS_2), with a record current of 300 $\mu A/\mu m$ at 1.6 V for a channel length of 0.8 μm [4]. As a point of reference, an n-MOSFET with a 0.5-μm gate length and biased at $V_{DS} = V_{GS} = 1.6$ V has a current of roughly 170 $\mu A/\mu m$ [5].

While the use of electrolytes to gate transition metal dichalcogenide field-effect transistors (FETs) has been previously discussed [6]–[10] this is not the approach taken here. We use the PEO:$CsClO_4$ to dope the access regions of the transistor. Once the doping is established, the temperature is lowered to lock the ions in place. With this transistor structure, where only the access regions are exposed to the ions, the device can then be operated with a metal/Al_2O_3 back gate. This doping and locking approach using PEO:$CsClO_4$ has also been successfully applied to the formation of p-n junctions in $MoTe_2$ [11] and in WSe_2 [12]. Other approaches for n-doping [13]–[16] or p-doping [16]–[18] of MoS_2 have been reported, but here the focus is on ion doping which enables the reconfigurability.

II. Device Fabrication and Doping

The MoS_2 was grown by chemical vapor transport (CVT) from MoS_2 powder, using a two-zone furnace and an iodine transport agent [19]. This method enables the vapor-phase growth of nanotubes and nanoribbons [4], [19]. The CVT growth method is being explored to avoid the unpassivated dangling bonds that are obtained at the edges of exfoliated materials [20]. While thicknesses at the few nanometer level are desired, the nanoribbons and nanotubes produced by the CVT growth method, as currently applied, are in the range 10 ± 5 nm. The device, with cross section shown in Fig. 1, has a 13-nm body thickness and a 700-nm width. This will be referred to as a nanoribbon. The fabrication started with electron-beam (e-beam) evaporation of Ti/Au (5/100 nm) on the back of a p^+ Si wafer. The nanoribbons were tape transferred from the CVT source onto a 27-nm Al_2O_3 oxide formed by atomic layer deposition (ALD) on the wafer top surface. E-beam lithography and lift off were used to form

Manuscript received July 17, 2017; revised October 1, 2017; accepted October 23, 2017. The review of this paper was arranged by Editor F. Schwierz. This work was supported in part by the CONICYT-PCHA/Doctorado Nacional/2014-2114059 and in part by Taiwan Semiconductor Manufacturing Company. *(Cristobal Alessandri and Sara Fathipour contributed equally to this work.) (Corresponding author: Alan C. Seabaugh.)*

C. Alessandri is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA and also with the Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile (e-mail: calessan@nd.edu).

S. Fathipour and A. C. Seabaugh are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: seabaugh.1@nd.edu).

H.-M. Li is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA and also with the Department of Electrical Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260 USA.

I. Kwak and A. Kummel are with the Materials Science and Engineering Program, Departments of Chemistry and Biochemistry, and Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093 USA (e-mail: akummel@ucsd.edu).

M. Remškar is with the Solid State Physics Department, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia (e-mail: maja.remskar@ijs.si).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TED.2017.2767501

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

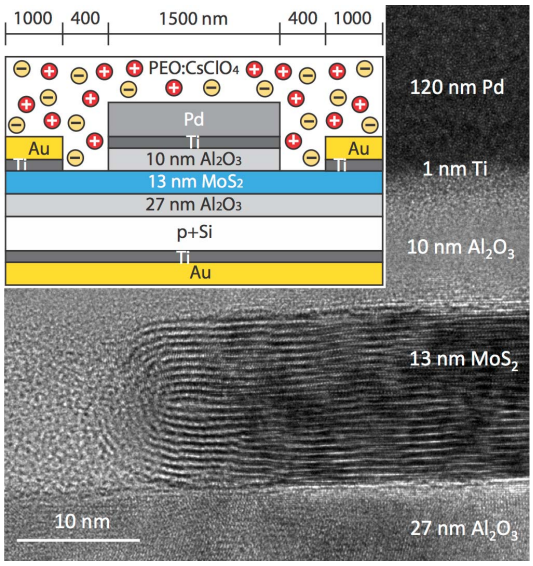

Fig. 1. Nanoribbon MoS_2 transistor cross-sectional schematic and transmission electron micrograph (TEM) along the 700-nm width. This TEM was made on completion of the measurements discussed in this paper.

Ti/Au (5/100 nm) source/drain contacts using e-beam evaporation. Side gates, not shown in Fig. 1, were also formed in the source/drain metallization step and are located 30 μm from the channel.

The ALD of Al_2O_3 on the MoS_2 nanoribbon utilized a two-step process consisting of a low-temperature physisorption step at 50 °C followed by ALD at 120 °C. In this way, conformal deposition of Al_2O_3 is achieved, wrapping around and under the nanoribbon edge without pinholes, as shown in Fig. 1. A top gate was formed by e-beam evaporation of Ti/Pd (1/120 nm), and the Al_2O_3 was etched in buffered HF using the gate as a mask to form access regions for the ion doping. The drain current was not notably reduced after etching indicating that the MoS_2 etch was insignificant. The top-gate pad away from the channel was mechanically damaged which prevented use of the gate terminal, but left the transistor fully functional when operated using the back gate.

The PEO and $CsClO_4$ were dissolved in acetonitrile and drop-cast to cover the entire surface of the wafer, followed by a 3 min anneal at 90 °C in an Ar-filled glove box. At room temperature, Cs^+ and ClO_4^- ions move on the polymer chains in the PEO in response to potentials applied between the side gates and the channel. With the source and drain grounded, a negative side gate bias V_{SG} pushes ClO_4^- ions into the channel access regions inducing hole conduction for the *p*-MOSFET. A positive side gate bias pushes Cs^+ into the access regions and induces electron conduction to set up the *n*-MOSFET. After positioning the ions with the side gates, the transistor is cooled below the glass transition temperature of the electrolyte (~240 K) to lock the ions in place and fix the doping. Measurements were carried out using an Agilent B1500 semiconductor parameter analyzer in a Cascade PLC50 vacuum probe station at 1.2×10^{-6} Torr.

Dozens of transistors have been fabricated based on this approach. The transistor reported here, however, was tested extensively over several months and represents the most thoroughly characterized of the CVT MoS_2 transistors we have tested to date. The doping results are reproducible and after locking the ions, I–V curves are reproducible with insignificant hysteresis. When the devices were reset and doped, the results were repeatable and reproducible. This device was initially tested with double sweeps and no noticeable hysteresis was observed after the ions are locked (below 220 K). The measurements reported in this paper are for single sweeps. It was verified that the polymer does not contribute any significant current below the glass transition temperature by measuring 2-μm long gaps without an MoS_2 channel and filled with PEO:$CsClO_4$. These measurements showed less than 1 pA/μm currents in the PEO:$CsClO_4$ for biases up to 4.5 V.

To provide a simple description of the connections and a consistent analysis, the left and right contacts in Fig. 1 will be referred to as the source and drain, respectively. To isolate the effect of the EDL doping on the channel, the same biasing conditions are measured for all configurations, even though for some transistor configurations this will not always be the usual transistor reporting convention.

III. MoS_2 N- and P-Doping Configurations

When polymer ion doping is used, it is essential to establish repeatable and reproducible starting conditions. Before setting the ion configuration, all the device terminals were grounded for 5 min at room temperature to reset any previous ion configuration. The EDL was then formed at room temperature by applying a potential to the side gate and grounding the drain–source, and back-gate contacts. A hold time of 0 or 3600 s (1 h) was applied before cooling at selected biases. The potentials on the terminals were maintained during cooling, which takes approximately 20 min.

Fig. 2(a) shows the transfer characteristics measured for different EDL forming conditions and for a negative drain–source bias, $V_{DS} = -0.4$ V. Following each measurement, the transistor was warmed back to 300 K, reset for 5 min and then cooled at a different side gate bias condition to set the access region doping. For $V_{SG} = 0$ V, an *n*-type FET characteristic is established, which suggests that the Fermi level is close to the MoS_2 conduction band, as is commonly observed in unintentionally doped MoS_2 [17]. Without any hold time before cooling, applying $V_{SG} = 2$ V produces a small negative shift in the threshold voltage, whereas applying $V_{SG} = -4$ V produces a slight positive shift. This is consistent with weak *n*- and *p*-type doping, respectively.

With a hold time of 1 h, a high *n*-doping is achieved with $V_{SG} = 2$ V, and back-gate modulation becomes negligible. In contrast, for $V_{SG} = -4$ V, both *n*- and *p*-branches can be observed with strong back-gate modulation. Increasing the side gate bias to −4.5 V further reduces the *n*-branch maxima, but has no significant effect on the *p*-branch maxima or the ability to back-gate the transistor.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

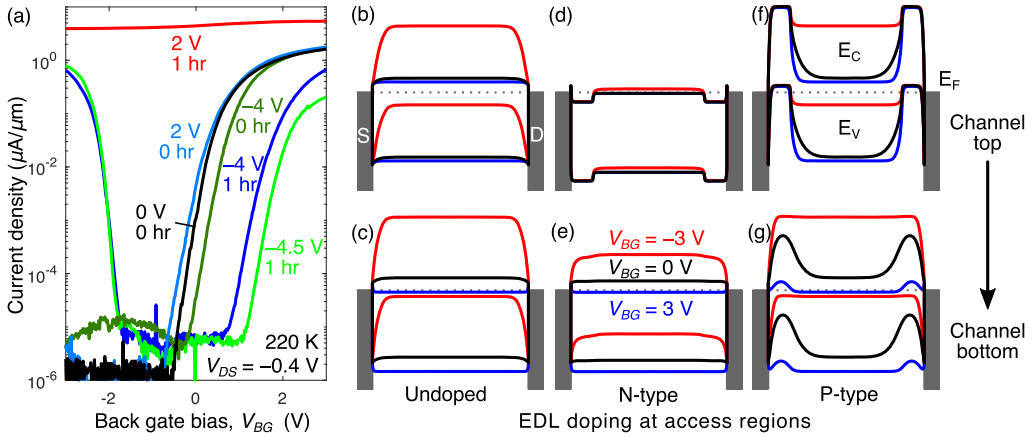

Fig. 2. (a) Measured back-gate transfer characteristics versus EDL doping condition in the transistor access regions. The EDL positions are specified by the side gate bias and hold time before cooling to freeze the ions into place. Simulated electrostatic band diagrams at the top of the channel (top row of band diagrams) and at the bottom of the channel (bottom row). The columns of band diagrams from left to right are for (b) and (c) no EDL doping, (d) and (e) n-type doping, and (f) and (g) p-type doping.

The induced carrier density from the EDL, similar to a gate, is peaked in concentration at the surface and decays with distance away from the surface. Similarly, induced carriers from the back-gate bias are maximized at the back of the MoS_2 channel and decay toward the surface. For this reason, the transport results can be expected to depend on thickness. For a thick device, independent conduction channels are induced at the surface and back-gate faces of the MoS_2. For layer thicknesses less than a Debye length in the vertical direction, the capacitances of the surface and back channels become coupled and a single conduction channel can be expected. To understand this behavior, a 2-D COMSOL multiphysics model was implemented across the channel length (source to drain) and thickness (top to bottom). The ionic charge at the access regions was modeled as a uniform fixed charge density at the PEO/MoS_2 interface of 2 and -2 μC/cm^2 for the n-doping and p-doping, respectively. These charge densities are conservative values considering measured EDL capacitances of 4 μF/cm^2 [6] have been obtained with PEO:$CsClO_4$. Electrostatic Poisson simulations were performed under different doping configurations and back-gate biases to explain the transport. Horizontal cuts of the band diagrams at the top and bottom of the MoS_2 channel are plotted in Fig. 2. Given that the channel is 13-nm thick, the effect of the ion doping is strong at the channel surface, but the back gate dominates at the bottom of the channel.

Consider the case where the ions are homogeneously distributed during cooling to arrest the ion motion, i.e., all terminal voltages are set to zero during cooling to 220 K as indicated by the black curve in Fig. 2(a). The band diagrams at the top and bottom of the channel are shown in Fig. 2(b) and (c), respectively, simulated with no surface charge in the access regions. The carrier densities are n-type both in the access regions and channel; negative back-gate bias raises the channel barrier and the measurements are consistent with the simulated band diagrams showing that the transistor turns off with over six orders of magnitude current ratio. For large, positive back-gate bias, the ON-current is likely limited by the contact barriers, and the electron concentration is peaked at the bottom of the channel.

Fig. 2(d) and (e) depicts the simulated band diagrams at the top and bottom of the channel for n-doping with $V_{SG} = 2$ V and 1 h hold time. Because of the degenerate doping induced in the access regions, the Schottky barrier is much thinner than the case shown in Fig. 2(c) and a 4× higher ON-current is observed at $V_{BG} = 3$ V. Given that the device channel is long (1.5 μm), the energy band in the middle of the channel is not determined by the doping in the access regions. Therefore, the back-gate bias should still allow modulation of the channel (far from the access regions) in the same way as in Fig. 2(b). However, the measured current shows a 20% back-gate modulation and the device cannot be turned off. The observed behavior is similar to what we observe on ion-doped and locked, back-gated transistor channels on MoS_2 [4] and WSe_2 [21] when no top gate is present. In this case the EDL controls the channel and the back-gate modulation is weak. This suggests that Cs+ ions have penetrated under the gate or in the region where the gate metal goes over the nanoribbon edge. The band diagrams computed in Fig. 2(d) and (e) follow this assumption. A recent report by Piatti *et al.* [22] using polymer ion gating indicates that Li and Na can intercalate in MoS_2 and affect channel conductance.

When a 1-h hold time is applied at room temperature for the -4 and -4.5 V side gate biases, the condition of Fig. 2(f) and (g) is achieved. A degenerate p-type doping is induced in the access regions at the top of the channel, which decays toward the bottom. Schottky tunneling of holes at the source/drain contacts should be enabled by the thin tunneling barriers. Unlike the Cs^+ doping case, the results suggest that the ClO_4^- does not intercalate or diffuse underneath the gate, which is likely because of its larger size.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
MOBILITY AND CONTACT RESISTANCE EXTRACTED FROM BACK-GATE TRANSFER CHARACTERISTICS, COMPARED WITH RESULTS REPORTED IN THE LITERATURE. VALUES ARE REPORTED AT ROOM TEMPERATURE UNLESS OTHERWISE NOTED

	Doping type	Mobility ($cm^2/V{\cdot}s$)	Contact resistance ($k\Omega{\cdot}\mu m$)
This work	**n**	**23 at 220 K**	**<36**
Fang 2013 [13]	n	25	Not reported
Kiriya 2014 [14]	n	24.7	1
Rai 2015 [15]	n	102	0.18
Giannazzo 2017 [16]	n	11.15	Not reported
This work	**p**	**20 at 200 K**	**28**
Chuang 2014 [17]	p	Not reported	2000
Nipane 2016 [18]	p	8.4 to 137.7	Not reported
Giannazzo 2017 [16]	p	7.2	Not reported

The p-branch observed in Fig. 2(a) for negative back-gate bias is produced by lowering the hole barrier in the channel. A positive back-gate bias turns off the p-channel as expected. However, an n-branch is still observed due to conduction at the bottom of the channel, where the ion doping is weak. When increasing the side gate bias from -4 to -4.5 V during the ion configuration, the n-branch is further reduced and shifted to the right, as shown in Fig. 2(a). The p-branch has no significant change because it is mainly controlled by the channel barrier, which does not change with the doping in the access region.

Table I shows the mobility and contact resistance extracted from the transfer characteristic in linear region ($V_{DS} < V_{GS} - V_t$) using the equation $I_D = (\mu C_{OX} W/L)(V_{GS} - V_{TH})V_{DS}$, where V_{DS} and V_{GS} are corrected for the series resistance and $C_{OX} = 0.26$ μF/cm^2 for 27-nm Al_2O_3 with 8.1 dielectric constant. The parameters were extracted for the p-MOSFET doped with $V_{SG} = -4$ V and 1 h hold time, and the n-MOSFET doped with $V_{SG} = 2$ V and no hold time. The n-MOSFET doped with $V_{SG} = 2$ V and 1 h hold time could not be used to extract mobility, but an upper bound for the contact resistance of 36 k$\Omega \cdot \mu$m was estimated from the saturation current. The p-MOSFET, on the other hand, compares well with previous reports in terms of mobility and contact resistance.

IV. MoS_2 TFET CONFIGURATION

To form doping with opposite carrier types at the source and drain contacts, opposite biases are applied to the drain and source, respectively [12]. The EDL was formed at room temperature by applying +2 V to the drain and −2 V to the source, while the back gate was grounded. In this way, Cs^+ ions are drawn to the negative source contact and ClO_4^- ions are drawn to the positive drain contact. A hold time of 1 h was applied and the device was then cooled to 220 K while keeping the above biases. The cooling process was again approximately 20 min. After reaching 220 K, the biases were released and measurements were then taken for six temperatures between 80 and 220 K.

Under this bias condition, associated with biasing the transistor as a TFET, there is no evidence that ions of either type intercalate or diffuse under the gate. This is consistent with

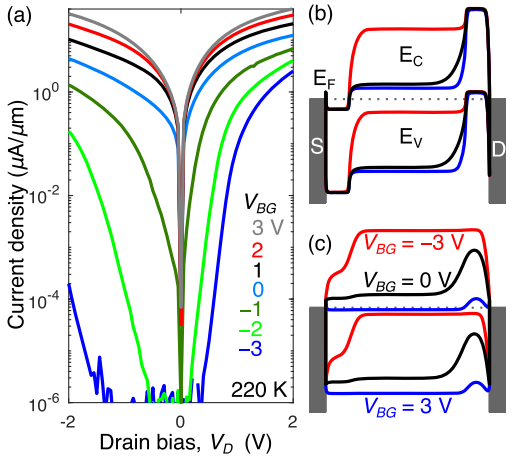

Fig. 3. (a) I_D versus V_D characteristics measured at different back-gate bias. Simulated band diagram for TFET configuration at the (b) top and (c) bottom of the channel.

simulations discussed in [12] that show that the ions accumulate adjacent to the contacts. The simulated electrostatic band diagram for the TFET configuration at the top of the channel is depicted in Fig. 3(b) as an n^+np^+ profile along the channel. The back gate modulates the channel and has no significant effect in the access regions. However, at the bottom of the channel the ion doping is weak and the back-gate modulation dominates, as shown in Fig. 3(c). Fig. 3(a) shows the $I_D - V_D$ measurements for different back-gate biases, which are readily explained by the simulated band diagrams. For large positive back-gate biases, the channel induced at the bottom dominates, so there is conduction for both negative and positive V_{DS} with almost no rectification. For zero back-gate bias, a small rectification is observed, due to the weak junction at the bottom of the channel.

The back-gate transfer characteristics measured with $V_{DS} = -0.4$ V are shown in Fig. 4(a) for different temperatures, and the subthreshold slope (SS) is shown in Fig. 4(b). At current densities below approximately 10^{-3} μA/μm, a linear positive temperature dependence in the SS is observed, as expected in the subthreshold region, i.e., $(kT/mq)\ln(10)$, see the inset, where m is a factor related to gate efficiency. However, at current densities above 3×10^{-3} μA/μm the swing decreases with temperature which suggests that the resistance is increasing with temperature, perhaps related to mobility degradation. No clear evidence for tunneling was observed when biased as a TFET. While high doping can be induced at the channel surface, tunneling could not be measured by back-gating the 13-nm-thick MoS_2 channel. This is because an abrupt tunnel junction could not be induced in such a thick channel.

The electrostatics of the device can be improved by reducing the channel thickness to a few monolayers to obtain a homogeneous heavy doping of the access regions and a better gate control in the channel. Although our simulations provide a qualitative understanding of the coupling between

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

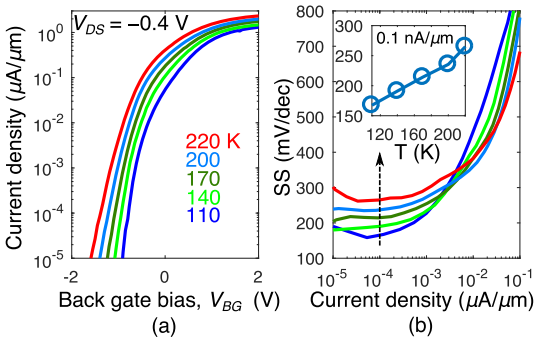

Fig. 4. (a) Temperature dependence of the back-gate transfer characteristic and (b) subthreshold swing for $V_{DS} = -0.4$ V. The inset shows the SS temperature dependence at a 10 nA/μm drain current density.

the EDL and back-gate capacitances, a more complete model is needed for a quantitative understanding, including quantum confinement effects and interlayer conduction. The use of a top gate (not operational in this device) would further improve the electrostatics for two reasons: first the top gate would modulate the doping at the top of the nanoribbon on the same plane as the EDL doping. Second, the top gate would modulate only the channel region and would not compete with the doping at the access regions as the back gate does.

V. Conclusion

Experimental measurements of EDL doping in a nanoribbon MoS_2 back-gated transistor have been presented showing *n*-MOS, *p*-MOS, and TFET configurations characterized in the same FET channel. Simulated band diagrams taking into account the front and back ends of the channel are used to explain the behavior showing that the characteristics of the 13-nm-thick channel can be readily explained. While band-to-band tunneling has been sought in these transistors for operation as TFETs, we show that the temperature dependence indicates the subthreshold transport is predominantly thermionic. To enable band-to-band tunneling, the device electrostatics must be improved by reducing the nanoribbon thickness to a few monolayers.

Acknowledgment

The authors would like to thank L. Yeh, W. Tsai, Y. Lin, and E. Chen, TSMC, for their insights and support.

References

[1] G. Fiori *et al.*, "Electronics based on two-dimensional materials," *Nature Nanotechnol.*, vol. 9, no. 10, pp. 768–779, 2014, doi: 10.1038/nnano.2014.207.

[2] D. K. Efetov and P. Kim, "Controlling electron-phonon interactions in graphene at ultrahigh carrier densities," *Phys. Rev. Lett.*, vol. 105, no. 25, p. 256805, 2010, doi: 10.1103/PhysRevLett.105.256805.

[3] K. Ueno, H. Shimotani, H. Yuan, J. Ye, M. Kawasaki, and Y. Iwasa, "Field-induced superconductivity in electric double layer transistors," *J. Phys. Soc. Jpn.*, vol. 83, no. 3, pp. 032001-1–032001-16, Mar. 2014, doi: 10.7566/JPSJ.83.032001.

[4] S. Fathipour *et al.*, "Record high current density and low contact resistance in MoS_2 FETs by ion doping," in *Proc. Int. Symp. VLSI Technol. Syst. Appl. (VLSI-TSA)*, Apr. 2016, pp. 1–2, doi: 10.1109/VLSI-TSA.2016.7480511.

[5] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2001.

[6] Y. Zhang, J. Ye, Y. Matsuhashi, and Y. Iwasa, "Ambipolar MoS_2 thin flake transistors," *Nano Lett.*, vol. 12, no. 3, pp. 1136–1140, Mar. 2012, doi: 10.1021/nl2021575.

[7] M. M. Perera *et al.*, "Improved carrier mobility in few-layer MoS_2 field-effect transistors with ionic-liquid gating," *ACS Nano*, vol. 7, no. 5, pp. 4449–4458, May 2013, doi: 10.1021/nn401053g.

[8] J. Pu, Y. Yomogida, K.-K. Liu, L.-J. Li, Y. Iwasa, and T. Takenobu, "Highly flexible MoS_2 thin-film transistors with ion gel dielectrics," *Nano Lett.*, vol. 12, no. 8, pp. 4013–4017, 2012, doi: 10.1021/nl301335q.

[9] M.-W. Lin *et al.*, "Mobility enhancement and highly efficient gating of monolayer MoS_2 transistors with polymer electrolyte," *J. Phys. D, Appl. Phys.*, vol. 45, no. 34, p. 345102, 2012, doi: 10.1088/0022-3727/45/34/345102.

[10] J. Pu *et al.*, "Fabrication of stretchable MoS_2 thin-film transistors using elastic ion-gel gate dielectrics," *Appl. Phys. Lett.*, vol. 103, no. 2, p. 023505, 2013, doi: 10.1063/1.4813311.

[11] H. Xu, S. Fathipour, E. W. Kinder, A. C. Seabaugh, and S. K. Fullerton-Shirey, "Reconfigurable ion gating of 2H-$MoTe_2$ field-effect transistors using poly(ethylene oxide)-$CsClO_4$ solid polymer electrolyte," *ACS Nano*, vol. 9, no. 5, pp. 4900–4910, May 2015, doi: 10.1021/nn506521p.

[12] S. Fathipour, P. Paletti, S. K. Fullerton-Shirey, and A. C. Seabaugh, "Demonstration of electric double layer p-i-n junction in WSe_2," in *Proc. Device Res. Conf. (DRC)*, 2016, pp. 213–214, doi: 10.1109/DRC.2016.7548485.

[13] H. Fang *et al.*, "Degenerate n-doping of few-layer transition metal dichalcogenides by potassium," *Nano Lett.*, vol. 13, no. 5, pp. 1991–1995, 2013, doi: 10.1021/nl400044m.

[14] D. Kiriya, M. Tosun, P. Zhao, J. S. Kang, and A. Javey, "Air-stable surface charge transfer doping of MoS_2 by benzyl viologen," *J. Amer. Chem. Soc.*, vol. 136, no. 22, pp. 7853–7856, 2014, doi: 10.1021/ja5033327.

[15] A. Rai *et al.*, "Air stable doping and intrinsic mobility enhancement in monolayer molybdenum disulfide by amorphous titanium suboxide encapsulation," *Nano Lett.*, vol. 15, no. 7, pp. 4329–4336, 2015, doi: 10.1021/acs.nanolett.5b00314.

[16] F. Giannazzo *et al.*, "Ambipolar MoS_2 transistors by nanoscale tailoring of Schottky barrier using oxygen plasma functionalization," *ACS Appl. Mater. Interfaces*, vol. 9, no. 27, pp. 23164–23174, 2017, doi: 10.1021/acsami.7b04919.

[17] S. Chuang *et al.*, "MoS_2 p-type transistors and diodes enabled by high work function MoO_x contacts," *Nano Lett.*, vol. 14, no. 3, pp. 1337–1342, 2014, doi: 10.1021/nl4043505.

[18] A. Nipane, D. Karmakar, N. Kaushik, S. Karande, and S. Lodha, "Few-layer MoS_2 p-type devices enabled by selective doping using low energy phosphorus implantation," *ACS Nano*, vol. 10, no. 2, pp. 2128–2137, 2016, doi: 10.1021/acsnano.5b06529.

[19] M. Remškar, Z. Skraba, F. Cleton, R. Sanjines, and F. Levy, "MoS_2 microtubes: An electron microscopy study," *Surf. Rev. Lett.*, vol. 5, no. 1, pp. 423–426, Feb. 1998, doi: 10.1142/S0218625X98000785.

[20] S. Fathipour *et al.*, "Synthesized multiwall MoS_2 nanotube and nanoribbon field-effect transistors," *Appl. Phys. Lett.*, vol. 106, no. 2, p. 022114, 2015, doi: 10.1063/1.4906066.

[21] S. Fathipour, P. Pandey, S. Fullerton-Shirey, and A. Seabaugh, "Electric-double-layer doping of WSe_2 field-effect transistors using polyethylene-oxide cesium perchlorate," *J. Appl. Phys.*, vol. 120, no. 23, p. 234902, Dec. 2016, doi: 10.1063/1.4971958.

[22] E. Piatti, Q. Chen, and J. Ye, "Strong dopant dependence of electric transport in ion-gated MoS_2," *Appl. Phys. Lett.*, vol. 111, no. 1, p. 013106, 2017, doi: 10.1063/1.4992477.

Cristobal Alessandri received the B.Sc. degree in electrical engineering from the Pontificia Universidad Católica de Chile, Santiago, Chile, in 2013. He is currently pursuing the dual Ph.D. degree with the Pontificia Universidad Católica de Chile and the University of Notre Dame, Notre Dame, IN, USA.

His current research interests include nanoelectronic devices and circuits.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Sara Fathipour is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA.

Her current research interests include memory and steep-slope devices.

Huamin Li received the Ph.D. degree in nano science and technology from Sungkyunkwan University, Seoul, South Korea, in 2013.

He is currently an Assistant Professor with the Department of Electrical Engineering, the University at Buffalo, Buffalo, NY, USA. His current research interests include low-dimensional materials and their advanced applications in electronics and optoelectronics.

Iljo Kwak is currently pursuing the Ph.D. degree in material science and engineering with the University of California at San Diego, La Jolla, CA, USA.

He has developed a deposition process of quality gate oxides on inert 2-D semiconductor surfaces using low temperature ALD. He is trying to extend this technique to very advance materials such as carbon nanotube and 3-D MoS2 nanoribbon and tube. His current research interests include novel semiconductors for future electronic devices.

Andrew Kummel received the bachelor's degree in chemical engineering from Yale University, New Haven, CT, USA, and the Ph.D. degree in chemistry from Stanford University, Stanford, CA, USA.

He is currently a Distinguished Professor with the Department of Chemistry and BioChemistry, University of California at San Diego, La Jolla, CA, USA, where he is also the Assistant Director of the Moores Cancer Center for Engineering and Physical Sciences.

Maja Remškar received the Degree in physics from University of Ljubljana, Ljubljana, Slovenia.

She is currently with the Solid State Physics Department, Jozef Stefan Institute, where she leads a laboratory for synthesis of inorganic nanotubes. Her current research interests include low-dimensional nanomaterials, nanophysics, and nanosafety.

Alan C. Seabaugh received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in 1985.

He was with Raytheon, Dallas, TX, USA, from 1997 to 1999. He joined the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA, in 1999, where he is currently the Frank Freimann Professor of Electrical Engineering.

APPENDIX E

OPTIMAL CCD READOUT BY DIGITAL CORRELATED DOUBLE SAMPLING

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY
MNRAS **455,** 1443–1450 (2015) doi:10.1093/mnras/stv2410

Optimal CCD readout by digital correlated double sampling

C. Alessandri,[1,2⋆] A. Abusleme,[1⋆] D. Guzman,[1] I. Passalacqua,[1] E. Alvarez-Fontecilla[1,3] and M. Guarini[1⋆]

[1] *Pontificia Universidad Catolica de Chile, Department of Electrical Engineering, Santiago, 7820436 Chile*
[2] *University of Notre Dame, Department of Electrical Engineering, Notre Dame, IN 46556*
[3] *University of California San Diego, Department of Electrical and Computer Engineering, La Jolla, CA 92093*

Accepted 2015 October 15. Received 2015 October 14; in original form 2015 September 9

ABSTRACT

Digital correlated double sampling (DCDS), a readout technique for charge-coupled devices (CCD), is gaining popularity in astronomical applications. By using an oversampling ADC and a digital filter, a DCDS system can achieve a better performance than traditional analogue readout techniques at the expense of a more complex system analysis. Several attempts to analyse and optimize a DCDS system have been reported, but most of the work presented in the literature has been experimental. Some approximate analytical tools have been presented for independent parameters of the system, but the overall performance and trade-offs have not been yet modelled. Furthermore, there is disagreement among experimental results that cannot be explained by the analytical tools available. In this work, a theoretical analysis of a generic DCDS readout system is presented, including key aspects such as the signal conditioning stage, the ADC resolution, the sampling frequency and the digital filter implementation. By using a time-domain noise model, the effect of the digital filter is properly modelled as a discrete-time process, thus avoiding the imprecision of continuous-time approximations that have been used so far. As a result, an accurate, closed-form expression for the signal-to-noise ratio at the output of the readout system is reached. This expression can be easily optimized in order to meet a set of specifications for a given CCD, thus providing a systematic design methodology for an optimal readout system. Simulated results are presented to validate the theory, obtained with both time- and frequency-domain noise generation models for completeness.

Key words: instrumentation: detectors – methods: analytical – techniques: imaging spectroscopy – telescopes.

1 INTRODUCTION

Charge-coupled devices (CCDs) are widely used for scientific imaging because of their high quantum efficiency, linearity and photon dynamic range. However, the dynamic range of astronomical CCDs is usually limited by the readout noise produced by the on-chip amplifier and the reset noise at the sensing capacitor (White et al. 1974; Barbe 1975; Janesick 2001). A correlated double sampling (CDS) scheme removes the reset noise and attenuates low-frequency noise components (White et al. 1974; Barbe 1975). White noise components can also be reduced by using a limited-bandwidth preamplifier. However, lowering the bandwidth requires a longer separation between samples due to the signal settling, which increases the pixel time and the low-frequency noise contribution (Kansy 1980; Hopkinson & Lumb 1982).

In the search for a better noise reduction, a differential-averaging scheme was proposed, which was proven to be optimal for white noise components (Hegyi & Burrows 1980). The usual implementation, known as dual slope integration, comprises analogue switches and an integrator (Janesick 2001). By using this technique on a standard CCD, the noise can be lowered at the expense of a reduced frame rate by using longer pixel integration times. However, the readout noise cannot be reduced without bound due to the contribution of low-frequency noise, which imposes a noise floor that limits the performance of CCDs for low-light applications. A comprehensive analysis of analogue readout schemes can be found in Hopkinson & Lumb (1982), which provides analytical expressions useful for design.

The development of low-noise readout techniques was inactive for over two decades, until Gach et al. (2003) proposed the digital correlated double sampling (DCDS) scheme. In this scheme, most of the analogue circuitry is replaced by an oversampling ADC and a digital filter. Due to the development of high-speed, high-resolution ADCs, the digital implementation of the differential-averaging has

⋆ E-mail: calessa2@uc.cl (CA); angel@uc.cl (AA); mguarini@ing.puc.cl (MG)

outperformed the traditional dual slope integration. Furthermore, the DCDS scheme allows us to implement any arbitrarily shaped filter instead of a simple averaging filter, thus increasing the design complexity compared to that of the well-studied analogue techniques.

Based on a qualitative understanding of noise correlation properties, Gach et al. (2003) experimentally found that, for a particular CCD, a weighted filter performs better than an averaging filter. However, this result was only optimal for a specific setup and was not supported by an analytical framework. Using a different experimental setup, Clapp (2012) tested similar weighted profiles, but reported a better performance for the averaging filter. Clapp also presented an approximated expression to compute the noise of the DCDS system, although it was derived only for an averaging filter. Therefore, the theory failed to explain the disagreement with Gach et al. (2003). Afterwards, Tulloch (2013) simulated the performance of several weighted filters and reported a marginal noise reduction over the averaging filter at low pixel rates. A first approach to compute optimal weights analytically was presented by Alessandri et al. (2013), who analysed the design of the digital filter for noise reduction under ideal settling conditions of the video signal. Other design variables such as the ADC sampling frequency and resolution, and the amplifier bandwidth have been studied independently (Smith 2013; Tulloch 2013; Stefanov & Murray 2014). However, there has been no analysis for the overall performance of a DCDS readout system with arbitrary weighted filters.

In this work, an in-depth theoretical analysis of a generic DCDS readout system is presented as follows: Section 2 provides a mathematical description of the DCDS system. In Section 3, the output statistics of the system are computed with the proper continuous- and discrete-time treatment of the noise processes involved. The signal-to-noise ratio (SNR) optimization model is depicted in Section 4, and a simulation model for a DCDS readout system is depicted in Section 5. Theoretical and simulated results are presented in Section 6. In Section 7, conclusions are drawn.

2 READOUT SYSTEM

Fig. 1 depicts a generic setup of a DCDS readout system along with the characteristic waveforms of a CCD. The measurement of each pixel is performed as follows: the sensing capacitor C_s is reset to V_{ref} by the analogue switch M_1. Due to thermal noise, charge injection and clock feedthrough, a voltage drop ΔV produces an uncertain initial voltage, which will be referred to as the reset voltage. At $t = t_d$, the pixel charge is transferred to the sensing capacitor, discharging the capacitor by a voltage V_p, which is related to the pixel charge n_e by the output sensitivity S_v, thus

$$V_p = S_v n_e. \tag{1}$$

Therefore, the voltage at the sensing capacitor can be expressed as

$$v_a(t) = V_r - v_p(t), \tag{2}$$

where $V_r = V_{ref} - \Delta V$ is the reset voltage, $v_p(t) = V_p u(t - t_d)$ is the pixel signal and $u(t)$ is the Heaviside function. The reset pulse is left out of equation (2) for simplicity, and it is assumed that the reset voltage is fully settled.

The voltage at the sensing capacitor is buffered by an on-chip amplifier, which adds noise to the measurement. This amplifier can be modelled as a noiseless amplifier (block Amp in Fig. 1) preceded by an equivalent series noise voltage source with two-sided Power Spectral Density (PSD) $S(i\omega)$ (Gray 2009). Hence, the voltage at the input of the noiseless amplifier is given by

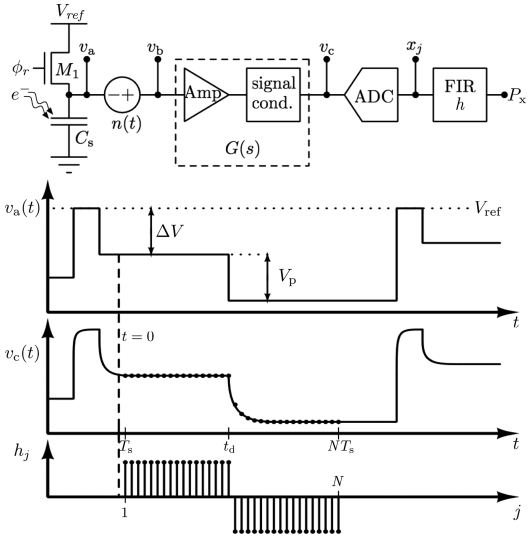

Figure 1. Generic setup of a DCDS readout system (top), and typical waveforms of a CCD (bottom), where v_a is the voltage at the CCD sensing capacitor, and v_c is the voltage after the on-chip amplifier and the signal conditioning stage, both described by $G(s)$. The signal is sampled starting at $t = 0$, and the digital filter depicted by h_j is applied to compute the pixel value. The amplifier noise, modelled by $n(t)$, is not considered in the plots for simplicity.

$$v_b(t) = V_r - v_p(t) + n(t), \tag{3}$$

where $n(t)$ is the amplifier input-referred series noise voltage.

The CCD output is processed by a signal conditioning stage as depicted in Fig. 1. For analysis purposes, the noiseless amplifier and the signal conditioning circuit can be described by a single generic transfer function $G(s)$ with impulse response $g(t)$.

The signal at the ADC input can be computed as a linear convolution between $v_b(t)$ and $g(t)$, hence

$$v_c(t) = \{V_r * g\}(t) - \{v_p * g\}(t) + \{n * g\}(t), \tag{4}$$

where $*$ is the convolution operator. Then, the signal is sampled with a period T_s, where $t = 0$ is arbitrarily defined before the first sample, as shown in Fig. 1. A column vector of N samples $x = [x_1, \ldots, x_j, \ldots, x_N]^t$ is taken at $t = jT_s$, with $j = 1, \ldots, N$, thus

$$x_j = v_j + r_j + n_j + q_j, \tag{5}$$

where $v_j = -\{v_p * g\}(jT_s)$, $r_j = \{V_r * g\}(jT_s)$, $n_j = \{n * g\}(jT_s)$ and q_j is the quantization and electronic noise introduced by the ADC. Finally, the digital filter described by $h = [h_1, \ldots, h_j, \ldots, h_N]^t$ is applied to compute the pixel value as

$$\begin{aligned} P_x &= h^t x \\ &= \sum_{j=1}^{N} h_j x_j, \end{aligned} \tag{6}$$

which is the output of the DCDS system.

3 OUTPUT STATISTICS

Knowing the noise characteristics of the system at the ADC input, the expression for the SNR after the digital filter is derived as follows. The mean value of the pixel measurement can be computed as

$$\begin{aligned}\mu_{\rm x} &= E\{P_{\rm x}\}\\ &= \sum_{j=1}^{N} h_j(E\{v_j\} + E\{r_j\} + E\{n_j\} + E\{q_j\})\\ &= \sum_{j=1}^{N} h_j(v_j + r_j),\end{aligned} \tag{7}$$

since n_j and q_j are zero-mean random variables (see Section 3.2), and both v_j and r_j are deterministic functions of $V_{\rm r}$ and $V_{\rm p}$, which are constant within a pixel. The variance of the pixel measurement is given by

$$\begin{aligned}\sigma_{\rm x}^2 &= E\{\left(h^t x - \mu_{\rm x}\right)^2\}\\ &= E\left\{\left(\sum_{j=1}^{N} h_j n_j + h_j q_j\right)^2\right\}.\end{aligned} \tag{8}$$

Considering that n_j and q_j are independent variables (see Section 3.2), the expected value of their product is zero, thus

$$\begin{aligned}\sigma_{\rm x}^2 &= \sum_{j=1}^{N}\sum_{k=1}^{N} h_j h_k E\{n_j n_k\} + \sum_{j=1}^{N}\sum_{k=1}^{N} h_j h_k E\{q_j q_k\}\\ &= \sum_{j=1}^{N}\sum_{k=1}^{N} h_j h_k R_{\rm n}[j,k] + \sum_{j=1}^{N}\sum_{k=1}^{N} h_j h_k R_{\rm q}[j,k]\\ &= \sigma_{\rm amp}^2 + \sigma_{\rm ADC}^2,\end{aligned} \tag{9}$$

where $R_{\rm n}[j,k]$ and $R_{\rm q}[j,k]$ are the terms of the discrete autocorrelation matrices of the amplifier and ADC noise, respectively. The noise models for these processes and the procedures to compute $\sigma_{\rm amp}^2$ and $\sigma_{\rm ADC}^2$ are presented separately in the following subsections.

3.1 Output amplifier noise

The noise of the CCD output amplifier usually comprises white noise and one or more low-frequency noise components (Hopkinson & Lumb 1982; Janesick 2001). For mathematical purposes, the two-sided PSD of the amplifier input-referred series noise voltage is described as a superposition of power-law noise sources given by

$$\begin{aligned}S(\mathrm{i}\omega) &= \sum_{\rm m} A_{\rm m}\,|\omega|^{\alpha_{\rm m}}\,[V^2/\mathrm{Hz}]\\ &= \sum_{\rm m} S_{\rm m}(\mathrm{i}\omega)[V^2/\mathrm{Hz}],\end{aligned} \tag{10}$$

which describes white noise ($\alpha_{\rm m} = 0$) and low-frequency noise, where $\alpha_{\rm m}$ is usually between -1 and -2. Accordingly, at the ADC input, the noise spectrum is given by

$$S_{\rm c}(\mathrm{i}\omega) = \sum_{\rm m} S_{\rm m}\,(\mathrm{i}\omega)\,|G(\mathrm{i}\omega)|^2\,[V^2/\mathrm{Hz}]. \tag{11}$$

Given the composition of equation (11), the output-referred voltage noise will be derived for a single power-law noise source $S_{\rm m}(\mathrm{i}\omega)$, and the total noise can be computed as the superposition in quadrature of the contribution of each power-law noise source.

Although the autocorrelation matrix from equation (9) could be computed by the inverse Fourier transform of $S_{\rm c}(\mathrm{i}\omega)$, it usually does not yield a closed-form expression and requires N infinite-length numerical integrations. Therefore, the resulting expression for $\sigma_{\rm x}^2$ provides little insight for design. An alternative approach, widely used in instrumentation for detectors in particle physics experiments, employs a time-domain noise model to design optimal filters. The noise is modelled as a sequence of pulses with a certain shape $\tilde{y}(t)$, arriving poissonianly at times $t_{\rm a}$ with an average rate ν and random sign (Goulding 1972; Radeka 1988; Pullia & Gatti 2001; Pullia & Riboldi 2004; Avila, Alvarez & Abusleme 2013). The pulse shape that models a noise source $S_{\rm m}(\mathrm{i}\omega)$ referred to the ADC input is expressed as (see Appendix A)

$$\tilde{y}_{\rm m}(t) = \sqrt{\frac{A_{\rm m}}{\nu}}\,\frac{\mathrm{d}^{\alpha_{\rm m}/2}}{\mathrm{d}t^{\alpha_{\rm m}/2}}\,g(t). \tag{12}$$

The total integrated noise $\sigma_{\rm m}^2$ measured at the ADC input is computed in the time domain using Campbell theorem (Papoulis & Pillai 2002).

$$\begin{aligned}\sigma_{\rm m}^2 &= \nu\int_{-\infty}^{t} \tilde{y}_{\rm m}^2(t - t_{\rm a})\mathrm{d}t_{\rm a}\\ &= \int_{-\infty}^{\infty} y_{\rm m}^2(t_{\rm a})\mathrm{d}t_{\rm a},\end{aligned} \tag{13}$$

which is equivalent to the amplifier noise autocorrelation function evaluated at $t = 0$ (see Appendix B). When the noise converges to a finite value, and according to Parseval theorem, $\sigma_{\rm m}^2$ can also be computed in the frequency domain (Radeka 1988), thus

$$\sigma_{\rm m}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{\rm m}(\mathrm{i}\omega)\,|G(\mathrm{i}\omega)|^2\,\mathrm{d}\omega. \tag{14}$$

The total integrated noise can be decomposed into two uncorrelated noise sources: the noise contribution of pulses that arrive before sampling (i.e. $t_{\rm a} < 0$) and the noise generated within the sampling window (i.e. $0 < t_{\rm a} < NT_{\rm s}$), hence

$$\begin{aligned}\sigma_{\rm m}^2 &= \int_{-\infty}^{0} y_{\rm m}^2(t - t_{\rm a})\mathrm{d}t_{\rm a} + \int_{0}^{t} y_{\rm m}^2(t - t_{\rm a})\mathrm{d}t_{\rm a}\\ &= \sigma_{\rm m,0}^2(t) + \sigma_{{\rm m},t}^2(t).\end{aligned} \tag{15}$$

Since $\sigma_{\rm m,0}^2(t)$ is the contribution of pulses generated before the first sample, its autocorrelation matrix is given by

$$R_{\rm m,0}[j,k] = \sigma_{\rm m,0}(jT_{\rm s})\sigma_{\rm m,0}(kT_{\rm s}), \tag{16}$$

and its contribution after the filter is directly computed as

$$\hat{\sigma}_{\rm m,0}^2 = \left(\sum_{j=1}^{N} h_j\sigma_{\rm m,0}(jT_{\rm s})\right)^2. \tag{17}$$

Using equation (15), this can be written as

$$\hat{\sigma}_{\rm m,0}^2 = \left(\sum_{j=1}^{N} h_j\sqrt{\sigma_{\rm m}^2 - \sigma_{{\rm m},t}^2(jT_{\rm s})}\right)^2. \tag{18}$$

The contribution of the noise generated within the sampling window is computed by the same principle, which is developed in detail by Avila et al. (2013). Thus,

$$\hat{\sigma}_{{\rm m},t}^2 = \sum_{j=1}^{N}\left(\sum_{k=0}^{N-j} h_{j+k}\sqrt{\sigma_{{\rm m},t}^2((k+1)T_{\rm s}) - \sigma_{{\rm m},t}^2(kT_{\rm s})}\right)^2. \tag{19}$$

Finally, the output-referred contribution of $S_{\rm m}(\mathrm{i}\omega)$ is

$$\hat{\sigma}^2_{\rm m} = \hat{\sigma}^2_{{\rm m},0} + \hat{\sigma}^2_{{\rm m},t} \tag{20}$$

and the total amplifier noise contribution is added in quadrature, hence

$$\sigma^2_{\rm amp} = \sum_{\rm m} \hat{\sigma}^2_{\rm m}. \tag{21}$$

3.2 ADC noise autocorrelation

Consider an ADC with a resolution of B bits and a full-scale voltage range $V_{\rm FSR}$, so $\Delta = V_{\rm FSR}/2^B$ is the least-significant bit (LSB). If the ADC is not overloaded, and if the input signal is large and active enough to span over several quantization levels, the quantization noise is modelled as an uncorrelated, zero-mean white noise with variance $\sigma^2_q = \Delta^2/12$ (Widrow 1956). In the case of a DCDS system, a slow varying but noisy signal is sampled, and the aforementioned conditions are met if Δ is comparable to the standard deviation of the independent noise between two samples. This noise is composed by the CCD noise contribution generated within two samples and the ADC electronic noise $\sigma^2_{\rm e}$, also called transition noise. Therefore, the LSB is upper-limited by

$$\Delta < \sqrt{\sigma^2_{\rm e} + \sum_{\rm m} \sigma^2_{{\rm m},t}(T_{\rm s})}. \tag{22}$$

Under this assumption, the autocorrelation matrix of the ADC noise is given by

$$R_{\rm q}[j,k] = \delta[j,k]\left(\sigma^2_{\rm q} + \sigma^2_{\rm e}\right), \tag{23}$$

where $\delta[j,k]$ is the Kronecker delta. The ADC noise contribution at the filter output is directly computed as

$$\sigma^2_{\rm ADC} = \left(\sigma^2_{\rm q} + \sigma^2_{\rm e}\right)\sum_{j=1}^{N} h_j^2. \tag{24}$$

For larger values of Δ, the quantization noise may be partially correlated and the noise contribution will be higher than that predicted in equation (24). Therefore, in order to benefit from the quantization noise reduction of the digital filter, the ADC resolution is lower-limited by

$$B > \log_2\left(\frac{V_{\rm FSR}}{\sqrt{\sigma^2_e + \sum_{\rm m}\sigma^2_{{\rm m},t}(T_{\rm s})}}\right). \tag{25}$$

Nevertheless, a higher resolution still provides a benefit in the optimal setup due to a lower quantization noise, and equation (25) is rarely an active restriction in low-noise applications. Furthermore, typical high-resolution ADCs have a transition noise of several LSB, so this equation is met regardless of the CCD noise. If the ADC resolution is fixed, the full-scale range referred to the sensing capacitor can be adjusted by the gain at the signal conditioning stage, thus trading the electrons range for a lower quantization noise. Although there are more thorough models for the quantization noise autocorrelation matrix (Gray 1990; Gray & Neuhoff 1998), the model presented here is accurate for the conditions of operation of a DCDS system and was chosen for its simplicity.

4 SNR OPTIMIZATION

In order to optimize the SNR, an analytical expression for the impulse response of the signal conditioning stage should be given, since it determines both the mean value and the variance of the pixel measurement. A typical signal conditioning stage for a DCDS system has a transfer function of the form

$$G(s) = G_0\frac{\tau_2 s}{(1+\tau_2 s)(1+\tau_1 s)}, \tag{26}$$

which comprises a single-pole high-pass filter defined by τ_2, static gain G_0 and a single-pole low-pass filter with time constant τ_1. However, it is straightforward to extend the analysis presented here for higher order systems.

Even though $G(s)$ comprises the effect of the AC coupling capacitor, in a well-designed system the coupling capacitor will be large enough so as to keep the signal integrity within a pixel (Hegyi & Burrows 1980). Hence $G(s) \approx G_0/(1+\tau_1 s)$. By setting $t_{\rm d} = \frac{N}{2}T_{\rm s}$, and according to equation (7), the pixel mean value is

$$\mu_{\rm x} = G_0\left(V_{\rm p}\sum_{j=\frac{N}{2}+1}^{N} h_j\left(1-\mathrm{e}^{-\left(j-\frac{N}{2}\right)T_{\rm s}/\tau_1}\right) + V_{\rm r}\sum_{j=1}^{N} h_j\right). \tag{27}$$

Since the reset voltage remains constant within a pixel, it can be completely removed if the filter coefficients add up to zero, which is the basis of the differential sampling scheme. Replacing the signal conditioning impulse response into equation (12), and computing the fractional derivative, the pulse shape $y_{\rm m}(t)$ can be expressed as

$$y_{\rm m}(t) = \sqrt{A_{\rm m}}G_0 u(t)\left(\frac{\tau_2/\tau_1}{\tau_2-\tau_1}t^{-\alpha_{\rm m}/2}E_{1,1-\alpha_{\rm m}/2}(-t/\tau_1)\right.$$
$$\left. -\frac{1}{\tau_2-\tau_1}t^{-\alpha_{\rm m}/2}E_{1,1-\alpha_{\rm m}/2}(-t/\tau_2)\right), \tag{28}$$

where $E_{\rm a,\,b}(t)$ is the Mittag–Leffler function (Mathai & Haubold 2008). Finally the SNR is expressed as

$$\mathrm{SNR} = \frac{\left(G_0 V_{\rm p}\sum_{j=\frac{N}{2}+1}^{N} h_j\left(1-\mathrm{e}^{-\left(j-\frac{N}{2}\right)T_{\rm s}/\tau_1}\right)\right)^2}{\sigma^2_{\rm amp}+\sigma^2_{\rm ADC}}, \tag{29}$$

which is an analytic function of the CCD noise parameters, the filter coefficients and a set of design variables $\boldsymbol{\gamma} = \{G_0, \tau_1, \tau_2, T_{\rm s}, N, B, V_{\rm FSR}\}$. The signal power, the reset noise and the amplifier noise are proportional to G_0^2, therefore changing the gain only affects the overall SNR due to the quantization noise.

The optimization is performed as follows. Given a fixed set of design variables $\tilde{\boldsymbol{\gamma}} = \{\tilde{G}_0, \tilde{\tau}_1, \tilde{T}_{\rm s}, \tilde{N}, \tilde{B}, \tilde{V}_{\rm FSR}\}$, the noise coefficients $\sigma^2_{\rm m}$ and $\sigma^2_{{\rm m},t}(jT_{\rm s})$ can be pre-computed with a single, finite-length numerical integration, and the SNR can be expressed solely as a function of the filter coefficients. Since the SNR is a highly nonlinear function, the filter optimization is carried out by fixing the pixel gain and minimizing the noise. Hence, the optimization problem is formulated as

$$\underset{h}{\text{minimize}} \quad \sigma^2_{\rm read}(h,\tilde{\boldsymbol{\gamma}}) = \sigma^2_{\rm amp} + \sigma^2_{\rm ADC}$$

subjectto

$$\sum_{j=\frac{N}{2}+1}^{N} h_j\left(1-\mathrm{e}^{-\left(j-\frac{N}{2}\right)\tilde{T}_{\rm s}/\tilde{\tau}_1}\right) = 1$$

$$\sum_{j=1}^{N} h_j = 0. \tag{30}$$

This problem can be solved with standard optimization software tools (Fourer, Gay & Kernighan 2003; Byrd, Nocedal & Waltz 2006). The overall optimization is performed as a semi-exhaustive

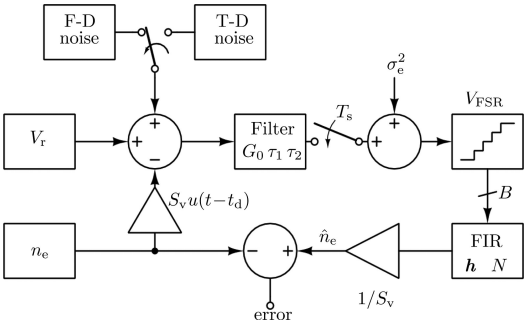

Figure 2. Simulation diagram: for each pixel, the reset voltage, pixel charge and amplifier noise are randomly generated and added as voltages. Time- and frequency-domain models can be selected for noise generation. An analogue filter is emulated with the simulation time-step, and then the signal is downsampled to T_s. The ADC electronic noise is added and the signal is quantized. Finally, the digital filter is applied to compute the error.

search in the design space $\boldsymbol{\gamma}$, which is usually bounded by the application requirements, available hardware and other design-related trade-offs.

5 DCDS READOUT SYSTEM SIMULATION SETUP

Based on the mathematical description of the DCDS readout system presented in Section 2, a set of simulations were programmed in MATLAB. As depicted in Fig. 2, a random reset voltage V_r is generated for each pixel. The pixel charge is computed as a random, integer number of electrons n_e, which is converted into voltage with the output sensitivity and added to the reset voltage at $t = t_d$. The amplifier PSD is defined by white noise and a single low-frequency noise component, hence

$$S(\mathrm{i}\omega) = A_s + A_f|\omega|^b, \tag{31}$$

with $-2 \leq b \leq -1$. It is usual to describe the low-frequency noise amplitude by the corner frequency f_c, defined as the frequency at which the low-frequency noise power is equal to the white noise power. In this case,

$$S(\mathrm{i}\omega) = A_s\left(1 + \left|\frac{\omega}{2\pi f_c}\right|^b\right) \tag{32}$$

and $A_f = A_s(2\pi f_c)^{-b}$.

For completeness, the noise can be generated by two methods.

(i) Time-domain (T-D) generation of noise pulses, based on the method proposed by Pullia & Riboldi (2004).

(ii) Frequency-domain (F-D) generation of noise, implemented by the method proposed by Kasdin (1995).

The noise is added to the signal, and the analogue filter, described by G_0, τ_1 and τ_2, is emulated to obtain the signal at the ADC input. The time-step of the simulation is defined by an oversampling rate over T_s for accuracy in the noise generation and filtering, so the signal is downsampled to T_s at the ADC to generate N samples. The ADC electronic noise is added to these samples, which are quantized with resolution B over a voltage range V_{FSR} and digitally filtered by the FIR described by $\boldsymbol{h}$. The pixel value is converted to electrons and compared with n_e to compute the error. The simulation is entirely determined by the design variables $\boldsymbol{\gamma} = \{G_0, \tau_1, \tau_2, T_s, N, B, V_{FSR}\}$, the filter coefficients and the system parameters $\boldsymbol{\zeta} = \{A_s, A_f, b, S_v, \sigma_e\}$.

6 THEORETICAL AND SIMULATED RESULTS

A set of theoretical and simulated results are presented to validate the theory and illustrate the potential of the proposed method. The results were generated for the two sets of parameters shown in Table 1, which are characterized by the noise PSD depicted in Fig. 3. The CCD1 parameters were estimated from Cancelo et al. (2012), whereas the parameters for CCD2 were taken from Tulloch (2013), which depicts a typical E2V CCD. The LSB is set to 1 electron, so a full-well of up to 262.144 electrons could be read for an 18-bit ADC, and the ADC electronic RMS noise σ_e was set at 3Δ. The high-pass filter time constant is fixed at 10 Hz to keep the signal integrity.

Figs 4 and 5 show a set of optimal filter coefficients for different scenarios. Since CCD2 has a higher corner frequency than CCD1, the optimal coefficients for CCD2 are always steeper near the charge dump, which is consistent with the principle introduced by Gach et al. (2003). Figs 4(a) and (b) show that the coefficients are not symmetrical for low bandwidths, whereas for a higher bandwidth as in Figs 4(c) and (d), the optimal filter approaches those already reported in the literature for ideal signal setting (Alessandri et al. 2013). These results can be understood by considering that a lower bandwidth enlarges the noise temporal correlation, thus producing a better noise cancellation by the subtraction near the charge dump. Therefore, the optimal solution assigns more weights to the middle coefficients. However, some samples after the charge dump are attenuated because the charge is not fully settled, thus there is an optimal bandwidth for this trade-off. In this case, for both CCDs the noise performance was better at 1 MHz. This approach defies the accepted convention to use a high bandwidth and discard samples until the signal is settled after the charge dump. Imposing these conditions, the optimal coefficients tend to be flat but produce a sub-optimal result due to the additional restrictions. This explains

Table 1. CCD1 and CCD2 noise parameters and sensitivity. The noise PSD is described by equation (32).

Parameter	CCD1	CCD2
A_s	$\left(0.5\,\frac{nV}{\sqrt{\mathrm{Hz}}}\right)^2$	$\left(1.5\,\frac{nV}{\sqrt{\mathrm{Hz}}}\right)^2$
f_c	20 kHz	150 kHz
b	-1.2	-1
S_v	$2.5\mu V/e^-$	$8\mu V/e^-$

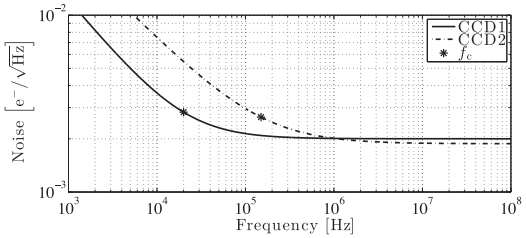

Figure 3. Noise PSD of CCD1 and CCD2. The noise amplitude is referred to the sensing capacitor by the sensitivity S_v and shown in units of $e^-/\sqrt{\mathrm{Hz}}$ for a fair comparison. The low-frequency noise corner frequency f_c is marked for each CCD.

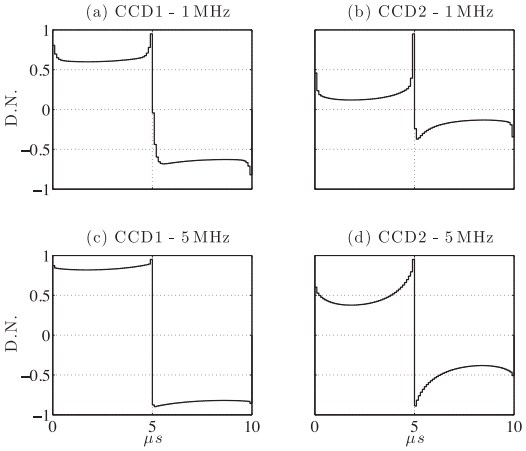

Figure 4. Normalized filter coefficients for a 10 μs sampling window and 100 samples. The optimal coefficients were computed for both CCDs with 1 and 5 MHz bandwidths.

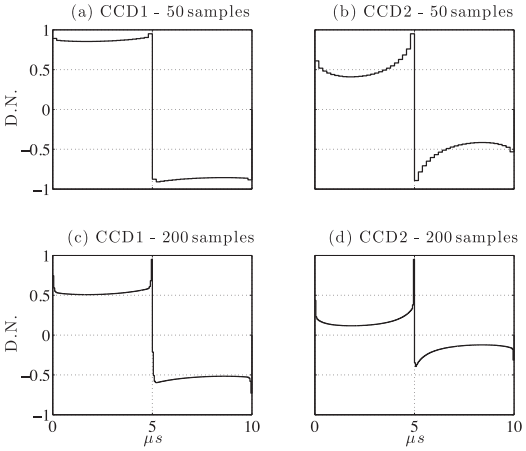

Figure 5. Normalized filter coefficients for a 10 μs sampling window and 2.5 MHz bandwidth. The optimal coefficients were computed for both CCDs with 50 and 200 samples.

the disagreement between Gach et al. (2003) and Clapp (2012), and supports the results reported by Tulloch (2013).

Fig. 6 shows the contribution of all noise sources and the total RMS noise over the pixel rate, taken with a 40 MSPS ADC and a fixed bandwidth for every pixel rate. The theoretical predictions are plotted with solid lines, whereas the error bars were generated with simulations. The simulated results were obtained with the frequency-domain method for noise generation, although the time-domain method produces equivalent results. Each simulation point was computed for 100 pixels and repeated 20 times to compute the mean value and the error bars. The pixel rate is computed as the inverse of the sampling window, so it only depends on the sampling rate and number of samples. The time required for the reset pulse and charge transfer is not considered because it can vary for different CCDs and does not depend on the presented method, so the actual

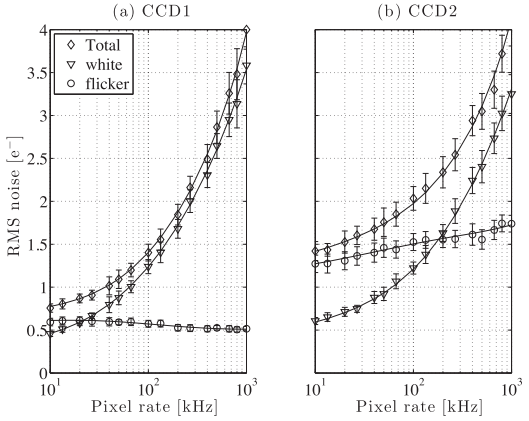

Figure 6. RMS noise along with white and flicker noise contributions versus pixel rate. The results were generated with a 40 MSPS ADC and 500 kHz bandwidth. The theoretical predictions are plotted with solid lines and the simulation results are shown by the error bars.

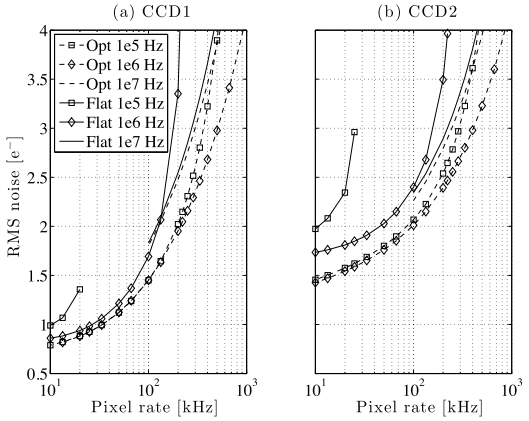

Figure 7. RMS noise versus pixel rate for both CCDs. The standard averaging filter (flat) is compared with the optimal filter computed by the proposed method (opt). The results were generated with a 20 MSPS ADC and different bandwidths at the signal conditioning stage.

pixel rate is slightly lower. Due to the corner frequency location, white noise is dominant in CCD1 over most of the plotted range, whereas its contribution in CCD2 is dominant below 200 kHz.

The optimal setup was compared with the standard setup for a DCDS system with flat weights. In the latter, half of the samples are taken at the reset level. After the charge dump, some samples are discarded until the signal is settled to $\Delta/2$ and the remaining samples are used to compute a simple differential average. Fig. 7 depicts the RMS noise over the pixel rate for both configurations and different bandwidths at the signal conditioning stage. Since the optimal filter is computed as a function of the bandwidth for every pixel rate, the proposed method performs adequately for a typical range of pixel frequencies, even if the bandwidth is fixed. This is an appealing feature, since it does not require to modify electronic components. Furthermore, the proposed method performs better than the averaging filter for any given bandwidth.

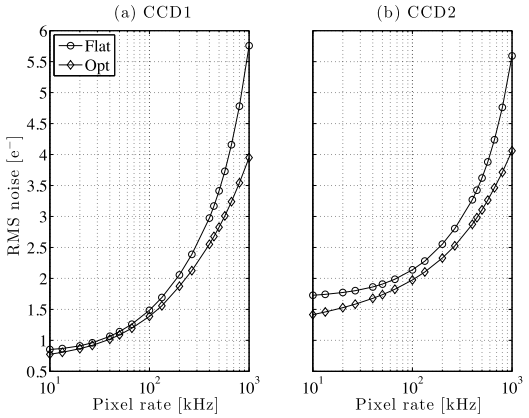

Figure 8. RMS noise versus pixel rate for both CCDs. The standard averaging filter (flat) is compared with the optimal filter computed by the proposed method (opt). The results were generated with a 40 MSPS ADC. For each setup and pixel rate, the bandwidth that produced the lowest noise was selected in order to make a fair comparison of the achievable performance of both methods.

The overall optimal setup is reached by selecting the best bandwidth at each pixel rate, which is a result of the semi-exhaustive search depicted in Section 4. Fig. 8 shows the RMS noise for both CCDs read out with an averaging filter and with an optimal filter, where the optimal bandwidth was selected independently for both setups in order to make a fair comparison of the achievable performance. The optimal filter noise is always lower, and a significant noise reduction is achieved at high pixel rates due to the use of low bandwidths and the settling period of the CCD. When white and low-frequency noise contributions are commensurable, the optimal coefficients are successful in lowering the noise floor, particularly at low pixel rates.

7 CONCLUSION

A detailed and thorough mathematical model to describe a DCDS system was presented. Based on this model, the noise statistics at the system output were computed as a function of the CCD parameters and the system design variables. An optimization model to maximize the SNR was developed, thus providing a systematic design methodology for an optimal DCDS readout system. Theoretical results were compared with realistic simulations to validate the theory and show the potential of the optimization method. As a result, the trade-offs involved in the design of a DCDS system were analysed and previous experimental disagreements were explained.

ACKNOWLEDGEMENTS

The authors would like to thank Peter Moore and Marco Bonati for their helpful discussions and suggestions throughout the development of this manuscript. The authors would also like to thank the engineering team at Cerro Tololo International Observatory (CTIO) for providing a Torrent CCD controller and constant technical support.

This work is funded by the National Commission of Scientific and Technologic Research (CONICYT, Chile) through the FONDECYT project 1130334 and the National Doctoral grant 21140599.

REFERENCES

Alessandri C. et al., 2013, in Proc. Scientific Detectors Workshop 2013, Theoretical Framework and Simulation Results for Implementing Weighted Multiple Sampling in Scientific CCDs. Florence
Avila D., Alvarez E., Abusleme A., 2013, IEEE Trans. Nucl. Sci., 60, 4634
Barbe D., 1975, Proc. IEEE, 63, 38
Byrd R., Nocedal J., Waltz R., 2006, in Di Pillo G., Roma M., eds, Nonconvex Optimization and Its Applications, Vol. 83, Large-Scale Nonlinear Optimization. Springer, USA, p. 35
Cancelo G., Estrada J. C., Moroni G. F., Treptow K., Zmuda T., Diehl H., 2012, Exp. Astron., 34, 13
Clapp M. J., 2012, in Holland A. D., Beletic J. W., eds, Proc. SPIE Conf. Ser. Vol. 8453, High Energy, Optical, and Infrared Detectors for Astronomy V. SPIE, Bellingham, p. 84531D
Fourer R., Gay D. M., Kernighan B. W., 2003, AMPL: a modeling language for mathematical programming (2nd ed.). Thomson/Brooks/Cole, Pacific Grove, CA
Gach J., Darson D., Guillaume C., Goillandeau M., Cavadore C., Balard P., Boissin O., Boulesteix J., 2003, PASP, 115, 1068
Goulding F., 1972, Nucl. Instrum. Methods, 100, 493
Gray R., 1990, IEEE Trans. Inf. Theory, 36, 1220
Gray P., 2009, Analysis and Design of Analog Integrated Circuits. Wiley, New York
Gray R., Neuhoff D., 1998, IEEE Trans. Inf. Theory, 44, 2325
Hegyi D. J., Burrows A., 1980, AJ, 85, 1421
Hopkinson G. R., Lumb D. H., 1982, J. Phys. E: Sci. Instrum., 15, 1214
Janesick J., 2001, Scientific Charge-Coupled Devices. SPIE, Bellingham
Kansy R., 1980, IEEE J. Solid-State Circuits, 15, 373
Kasdin N., 1995, Proc. IEEE, 83, 802
Mathai A., Haubold H., 2008, Special Functions for Applied Scientists. Springer, New York
Papoulis A., Pillai S., 2002, Probability, Random Variables, and Stochastic Processes. McGraw-Hill series in electrical engineering: Communications and signal processing, Boston, McGraw-Hill, New York
Pullia A., Gatti E., 2001, in IEEE Nucl. Sci. Symp. Conf. Rec., 2, 778
Pullia A., Riboldi S., 2004, IEEE Trans. Nucl. Sci., 51, 1817
Radeka V., 1988, Ann. Rev. Nucl. Part. Sci, 38, 217
Smith R., 2013, in Proc. Scientific Detectors Workshop 2013, Digital Correlated Double Sampling for ZTF. Florence, available at: http://www.oir.caltech.edu:8080/DETECT/publications/2013_sdw-digital-cds/SDW2013_Digital_CDS_Smith_Kaye_2013-11-12.pdf
Stefanov K., Murray N., 2014, Electron. Lett., 50, 1022
Tulloch S., 2013, in Proc. Scientific Detectors Workshop 2013, Theoretical Comparison of CCD Video Processors. Florence, available at: http://www.qucam.com/technicalnotes/TN4.pdf
White M., Lampe D., Blaha F., Mack I., 1974, IEEE J. Solid-State Circuits, 9, 1
Widrow B., 1956, IRE Trans. Circuit Theory, 3, 266

APPENDIX A: PULSE SHAPE DERIVATION

An arbitrary two-sided noise power spectrum given by

$$S_{\rm m}(\mathrm{i}\omega) = A_{\rm m}|\omega|^{\alpha_{\rm m}} \tag{A1}$$

can be expressed as

$$S_{\rm m}(\mathrm{i}\omega) = A_{\rm m}\left((\mathrm{i}\omega)^{\alpha_{\rm m}/2}(-\mathrm{i}\omega)^{\alpha_{\rm m}/2}\right) \tag{A2}$$

$$= \left(A_{\rm m}^{1/2}(\mathrm{i}\omega)^{\alpha_{\rm m}/2}\right)\left(A_{\rm m}^{1/2}(\mathrm{i}\omega)^{\alpha_{\rm m}/2}\right)^{*}. \tag{A3}$$

Following the same procedure shown in Pullia & Riboldi (2004), the frequency core pulse is given by

$$H(\mathrm{i}\omega) = A_{\rm m}^{1/2}(\mathrm{i}\omega)^{\alpha_{\rm m}/2} \tag{A4}$$

and the frequency core pulse after a system $G(\mathrm{i}\omega)$ can be computed as

$$Y_{\mathrm{m}}(\mathrm{i}\omega) = A_{\mathrm{m}}^{1/2}(\mathrm{i}\omega)^{\alpha_{\mathrm{m}}/2}G(\mathrm{i}\omega), \tag{A5}$$

which is a hermitian function. The time-domain core pulse can be computed in terms of the system impulse response $g(t)$ and the Fourier derivative property as

$$y_{\mathrm{m}}(t) = \sqrt{A_{\mathrm{m}}}\frac{\mathrm{d}^{\alpha_{\mathrm{m}}/2}}{\mathrm{d}t^{\alpha_{\mathrm{m}}/2}}g(t), \tag{A6}$$

which is a real function. The core pulse is finally scaled in amplitude to make the noise energy consistent with the arrival rate

$$\tilde{y}_{\mathrm{m}}(t) = \sqrt{\frac{A_{\mathrm{m}}}{\nu}}\frac{\mathrm{d}^{\alpha_{\mathrm{m}}/2}}{\mathrm{d}t^{\alpha_{\mathrm{m}}/2}}g(t). \tag{A7}$$

APPENDIX B: AUTOCORRELATION, PSD AND STATIONARITY

Consider the Fourier transform pair from Appendix A

$$y_{\mathrm{m}}(t) \rightarrow Y_{\mathrm{m}}(\mathrm{i}\omega). \tag{B1}$$

The autocorrelation function of $y_{\mathrm{m}}(t)$, defined as

$$\begin{aligned} R_{\mathrm{y}}(t_1, t_2) &= \int_{-\infty}^{\infty} y_{\mathrm{m}}(\tau - t_1)y_{\mathrm{m}}(\tau - t_2)\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} y_{\mathrm{m}}(\tau')y_{\mathrm{m}}(\tau' - (t_2 - t_1))\mathrm{d}\tau', \end{aligned} \tag{B2}$$

can be expressed only as a function of $t = t_2 - t_1$

$$R_{\mathrm{y}}(t) = \int_{-\infty}^{\infty} y_{\mathrm{m}}(\tau)y_{\mathrm{m}}(\tau - t)\mathrm{d}\tau. \tag{B3}$$

If $R_{\mathrm{y}}(t)$ is absolutely integrable, its Fourier transform can be computed as

$$\begin{aligned} S_{\mathrm{y}}(\mathrm{i}\omega) &= Y_{\mathrm{m}}(\mathrm{i}\omega)Y_{\mathrm{m}}(\mathrm{i}\omega)^* \\ &= \left(A^{1/2}(\mathrm{i}\omega)^{\alpha/2}G(\mathrm{i}\omega)\right)\left(A^{1/2}(\mathrm{i}\omega)^{\alpha/2}G(\mathrm{i}\omega)\right)^* \\ &= A\,|\omega|^{\alpha}\,|G(\mathrm{i}\omega)|^2\,, \end{aligned} \tag{B4}$$

which is the noise spectrum of $S_{\mathrm{m}}(\mathrm{i}\omega)$ referred to the ADC input, whereas the full spectrum $S_{\mathrm{c}}(\mathrm{i}\omega)$ can be computed from superposition. Therefore, $S_{\mathrm{c}}(\mathrm{i}\omega)$ is a wide sense stationary (WSS) process if

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} S_{\mathrm{m}}(\mathrm{i}\omega)\,|G(\mathrm{i}\omega)|^2\,\mathrm{d}\omega < \infty. \tag{B5}$$

for all m. This means that even if $S_{\mathrm{m}}(\mathrm{i}\omega)$ is not WSS, like flicker noise components, the noise at the ADC input can behave as a WSS process if the signal conditioning stage has a high-pass filter. Furthermore, even in the absence of a high-pass filter, the limited-bandwidth approximation of flicker noise produces the same result.

This paper has been typeset from a TEX/LATEX file prepared by the author.