UNIVERSITY OF CALIFORNIA

Santa Barbara

Mixed Signal Neurocomputing Based on Floating-gate Memories

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical and Computer Engineering

by

Xinjie Guo

Committee in charge:

Professor Dmitri Strukov, Chair

Professor Kwang-Ting (Tim) Cheng

Professor Margaret Marek-Sadowska

Professor Li-C Wang

March 2017

The dissertation of Xinjie Guo is approved.

_____

Kwang-Ting (Tim) Cheng

_____

Margaret Marek-Sadowska

_____

Li-C Wang

_____

Dmitri Strukov, Committee Chair

March 2017

Mixed Signal Neurocomputing Based on Floating-gate Memories

by

Xinjie Guo

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge my advisor, Dr. Dmitri Strukov, for providing an opportunity of my stay at UCSB, supervising my research and reviewing this thesis. At the meantime, I want to show my deepest appreciation to Dr. Konstantin K. Likharev, Dr. Farnood Merrikh-Bayat, Dr. Mirko Prezioso, Mikhael Klachko and Dr. Ligang Gao for all the discussions we had and your help provided. Many thanks to all my other lab mates specially Dr. Gina Adam, Mohammad Bavandpour, Dr. Bhaswar Chakrabarti, Dr. Brian Hoskins, Dr. Advait Madhavan, M. Reza Mahmoodi and Dr. Elham Zamanidoost for the enjoyable working environment. I wish to thank my committee for the fruitful review of my thesis as well.

I also wish to thank those most important people in my life: my parents and my husband-- Shaodi Wang. Without their endless love, support and paying out, I could never march this far and become who I am today.

VITA OF XINJIE GUO
March 2017

EDUCATION

Bachelor of Science in Microelectronics, Peking University, Berkeley, July 2011
Master of Science in Electrical Computer Engineering, University of California, Santa Barbara, June 2013
Doctor of Philosophy in Electrical Computer Engineering, University of California, Santa Barbara, March 2017 (expected)

PROFESSIONAL EMPLOYMENT

2011-2017: Researching Assistant, Department of Electrical Computer Engineering, University of California, Santa Barbara

PUBLICATIONS

• X. Guo, "Mixed Signal Neurocomputing Based on Floating-gate Memories", unpublished thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical Computer Engineering, University of California, Santa Barbara, 2017.

• F. Merrikh Bayat*, X. Guo*, M. Klachko, M. Prezioso, K.K. Likharev, and D.B. Strukov, "High-Performance Analog Neurocomputing with Nanoscale Floating-Gate Memory Cell Arrays", submitted to Science Advance. *these authors have equal contribution.

• X. Guo, F. Merrikh Bayat, N. Do, M. Prezioso, K.K. Likharev, and D.B. Strukov, "Temperature-Insensitive Analog Vector-by-Matrix Multiplier Based on 55 nm NOR Flash Memory Cells", CICC'17, Austin, TX, May 2017.

• X. Guo*, F. Merrikh-Bayat*, L. Gao, B. D. Hoskins, F. Alibart, B. Linares-Barranco, L. Theogarajan, C. Teuscher, and D.B. Strukov, "Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits", Frontiers in Neuroscience , art. 488, Dec. 2015. *these authors have equal contribution.

• F. Merrikh Bayat, X. Guo, and D. B. Strukov, "Exponential-weight multilayer perceptron", accepted to IJCNN'17, Anchorage, Alaska, May, 2017.

• F. Merrikh Bayat, X. Guo, M. Klachko, N. Do, K. Likharev, and D. Strukov, "Model-based high-precision tuning of NOR flash memory cells for analog computing applications", in: Proc. DRC'16, Newark, DE, June 2016, pp. 1-2.

• F. Merrikh Bayat, M. Prezioso, X. Guo, B. Hoskins, D.B. Strukov, and K.K. Likharev, "Memory technologies for neural networks", in: Proc. IMW'15, Monterey, CA, May 2015, pp. 1-4.

• F. Merrikh Bayat, X. Guo, H.A. Om'mani, N. Do, K.K. Likharev, and D.B. Strukov, "Redesigning commercial floating-gate memory for analog computing applications", in: Proc. ISCAS'15, Lisbon, Portugal, May 2015, pp. 1921-1924.

• L. Gao*, F. Merrikh-Bayat*, F. Alibart, X. Guo, B.D. Hoskins, K.-T. Cheng, and D.B. Strukov, "Digital-to-analog and analog-to-digital conversion with metal oxide memristors

for ultra-low power computing", in: Proc. NanoArch'13, New York, NY, July 2013 (best short paper award).

• Xinjie Guo, Shaodi Wang, Jin He, “A Novel Approach to Simulate Fin-width Line Edge Roughness Effect of FinFET Performance”, IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC'10).

• Shaodi Wang, Xinjie Guo, Jin He, “Analytical subthreshold channel potential model of asymmetric gate under-lap gate-all-around MOSFET”, IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC'10).

• Zhang Chenfei, Ma Chenyue, Guo Xinjie, “Forward Gated-Diode Method for Parameter Extraction of MOSFETs”, Journal of Semiconductors.

PATENTS

• F. Merrikh Bayat, X. Guo, D.B. Strukov, H. Tran, N. Do, V. Tiwari, M, Reiten, “Deep Learning Neural Network Classifier Using Non-volatile Memory Array”, U.S. application serial no. 62/337,760

• X. Guo, F. Merrikh Bayat, D.B. Strukov, H. Tran, N. Do, V. Tiwari, “Flash Memory Array with Individual Memory Cell Read, Program and Erase”, U.S. application serial no. 62/337,751

AWARDS

• Dean's Scholarship, Peking University, 2010
• Industry Scholarship, Peking University, 2009
• 1st Prize in National Mathematics Olympiad in Province, 2006

ABSTRACT

Mixed Signal Neurocomputing Based on Floating-gate Memories

by

Xinjie Guo

Nervous systems inspired neurocomputing has shown its great advantage in object detection, speech recognition and a lot of other machine-learning technology-driven applications from speed and power efficiency. Among handful neurocomputing implementation approaches, analog nanoelectronic circuits are very appealing because they may far overcome digital circuits of the same functionality in circuit density, speed and energy efficiency. Device density is one of the most essential metrics for designing large-scale neural networks, allowing for high connectivity between neurons. Thanks to the high-density nature of traditional memory applications, building artificial neural networks with hybrid complementary metal oxide semiconductor (CMOS)/memory devices would enable the high parallelism as well as achieve the performance advantages.

Synapses, the most numerous elements of neural networks, are efficiently implemented by memory devices. This application, however, imposes a number of requirements, such as the continuous change of the memory resistance state, creating the need for novel engineering approaches. Here we report such engineering approaches for advanced

commercial 180-nm ESF1 and 55-nm ESF3 NOR flash memory, facilitating fabrication and successful test of high performance analog vector-by-matrix multiplication which is the key operation performed at signal propagation through any neuromorphic network. Furthermore, we discuss the recent progress toward neuromorphic computing implementations based on nonvolatile floating-gate devices, in particular the experimental results for a prototype 28×28-binary-input, 10-output, 3-layer neuromorphic network based on arrays of highly optimized embedded nonvolatile floating-gate cells. The fabricated neuromorphic network's active components, including 101,780 floating-gate cells, have a total area below 1 mm^2. The network has shown a 94.7% classification fidelity on the common MNIST benchmark, close to the 96.2% obtained in simulation. The classification of one pattern takes sub-1 μs time and sub-20 nJ energy – both numbers much better than for the best reported digital implementations of the same task. Estimates show that a straightforward optimization of the hardware, and its transfer to the already available 55-nm technology may increase this advantage to more than 100X in speed and 10000X in energy efficiency.

As pure analog circuits cannot address the noise accumulation problem, a practical solution would also require inclusion of analog-to-digital and digital-to-analog stages for signal restoration. High energy-efficient and compact data converters are therefore expected to play an important role in future computing platforms. We perform an experimental demonstration of 6-bit digital-to-analog (DAC) and 4-bit analog-to-digital conversion (ADC) operations implemented with a hybrid circuit consisting of Pt/TiO2-x /Pt resistive switching devices (also known as ReRAMs or memristors) and a CMOS operational amplifier (opamp). In particular, ADC is implemented with a Hopfield neural network circuit.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. Introduction

In the past few years, neuromorphic computing has advanced greatly in solving cognitive problems such as pattern recognition, speech translation, topic classification and so on. Among diverse neuromorphic networks, deep learning approach is a important subfield for its success in both commerce and academia [1- 3]. In general, deep learning neuromorphic networks consist of a large number of processing layers which contain millions of weights (called synapses) connected simple processors called neurons [4]. Despite of various deep learning architectures such as multilayer perceptron, convolutional neural network and recurrent neural networks, all architectures are stack of processing layers which perform linear or nonlinear transformations of previous layer's outputs. The concise and uniform structure of deep learning neuromorphic networks along with its outstanding performance in tremendous domains will allow a revolutionary technological leap toward outperforming conventional complimentary metal-oxide semiconductor (CMOS) technology [5].

Technology breakthroughs and designs are expected in implementing neuromorphic systems, due to the huge difference from traditional Von Neumann computational architecture. There is a booming of approaches to implement deep neural networks models with distinct analog, digital, mixed-mode analog/digital VLSI, and software systems [6 - 14]. For example, the rapid computational capability evolving in graphics processing units (GPUs) facilitates the scaling up of neuromorphic networks to an extent that they can accomplish certain applications with configurable features. However, GPUs are relatively expensive, area costly and prohibitively power hungry [15]. Neuromorphic application-specific integrated circuits (ASICs) have shown their speed and power advantages over

GPUs with the same technology node [14, 16], the advances ASICs achieved is still not adequate. To resolve previous mentioned contemporary issues, dedicated hardware approaches are needed to utilize the similarities between Silicon and neurobiology, e.g., memory, to maximize the potential of neuromorphic architectures [17].

The concept of using nonvolatile memories for signal processing at relatively low precision requirements, for example analog and mixed-signal neuromorphic networks, far superior to digital circuits of the same functionality in speed and energy efficiency, is at least 30 years old [18, 19]. Limited by the technology, precision handicap was one of the major challenges preventing analog computation's prevailing back then. Fortunately, with the technology developments, neuromorphic networks are scaled up, obtaining high tolerance of their operation to synaptic weight variations, and hence the precision requirement in neuromorphic computing is proven to be as low as several bits or even binary [20 - 27]. Without the precision wall, analog computing is the key to break the power, area and cost bottleneck [17]. Benefitting from the inherent similarities between analog circuits and biological systems, many theoretical and experimental works further confirm the advantages of analog approach [28 – 31]. Moreover, since there is storage requirement for a tremendous number of synapses inside any practical neuromorphic networks, utilizing computation in memory would be both efficient and straight forward [32, 33]. Recent works [27, 34 - 35] have shown that such circuits, utilizing nanoscale devices, improves the neuromorphic network performance dramatically, leaving far behind both their digital counterparts and biological prototypes, and approaching the energy efficiency of the human brain.

The key component of such mixed-signal neuromorphic networks is a device with adjustable (tunable) conductance - essentially an analog nonvolatile memory cell,

mimicking the biological synapse. There have been significant recent advances in the development of nanoscale nonvolatile memories, such as ReRAMs, MRAM, PCRAM, FeRAM, NOR Flash memories, NAND Flash memories and 3D ReRAMs – for a review, see, e.g., Refs, 36 - 42. In particular, those emerging memories have already been used to demonstrate small neuromorphic networks [43 - 55]. The background of further advantages among different analog approaches is the fact that in memory based neuromorphic circuits, the vector-by-matrix multiplication, i.e. the key operation performed at signal propagation through any neuromorphic network, is implemented on the physical level, in a resistive crossbar circuit, using the fundamental Ohm and Kirchhoff laws.

However, for most of the emerging memories, their fabrication technology is still in much need for improvement and not ready yet for the large-scale integration, which is necessary for practically valuable neuromorphic networks. Up until recently, such devices were implemented mostly as floating-gate "synaptic transistors" [15, 56], which may be fabricated using the standard CMOS technology. Flash memory device has inherently adjustable conductance and the advancing to three-dimensional integration with higher density makes it more appealing for neuromorphic applications [57]. Recently some rather sophisticated neuromorphic systems were demonstrated [15, 58] using this approach. However, synaptic transistors have relatively large areas ($\sim 10^3 F^2$, where F is the minimum feature size), leading to larger time delays and energy consumption [17].

In this thesis, we mainly use the highly optimized, nanoscale, nonvolatile floating-gate memory cells which are used in the recently developed embedded NOR flash memories [59]. These cells are quite suitable to serve as adjustable synapses in neuromorphic networks, provided that the memory arrays are redesigned to allow for individual, precise adjustment of the memory state of each device. Recently, such modification was performed

[60, 61] using the 180-nm ESF1 embedded commercial NOR flash memory technology of SST Inc. [59], and, more recently, the 55-nm ESF3 technology of the same company [62], with good prospects for its scaling down to at least F = 28 nm. (The last number is just slightly worse than the expected size of the emerging nonvolatile memories with transistor-based selectors.) Though such modification nearly triples the cell area, it is still at least an order of magnitude smaller, in terms of F^2, than that of synaptic transistors [17].

As a key step towards area and power efficient neuromorphic network, small vector-by-matrix multipliers are separately fabricated and tested based on redesigned 180-nm ESF1 and 55-nm ESF3 arrays [60, 62]. The demonstrated vector-by-vector multipliers are operating in low-power subthreshold mode, with gate coupling of array cells to the input (peripheral) cells. In order to reduce the temperature drift, pertinent to the subthreshold operation of the cells, differential versions of multipliers are implemented and characterized to minimize the output signal drift.

The main result reported in this thesis is the first successful use of previous mentioned approach for the experimental implementation of a mixed-signal neuromorphic network performing high-fidelity classification of patterns of the standard MNIST benchmark, with record-breaking speed and energy efficiency [63]. Hardware constrains aware designs are applied throughout the implementation especially when training the network in software before importing into the fabricated chip.

Another prototype reported in this thesis explores the implementation of synapses with the emerging, very promising memristor devices to build a recurrent artificial neural network called Hopfield analog-to-digital (ADC) network [64].

II. Memory Elements

The key component of the most advanced analog computing implementations is a nanodevice with adjustable conductance – essentially an analog nonvolatile memory cell, which could mimic synaptic transmission function by multiplying signal from the input neuron (e.g. encoded as voltage applied to the memory device) by its analog weight (device conductance) and passing the product (the resulting current) to the output. Such functionality enables very dense, fast, and low power implementation of dot-product computation, the most common operation in many artificial neural networks. However, there is a particular challenge for utilizing analog nonvolatile memory cell. The challenge is that the synapse is an analog memory element, and, e.g., at a lot of occasions, 5-bit precision is required for convolutional networks [27, 65]. The most promising analog memory devices are memristive device and floating gate device. The former one is an ideal candidate with super density while the later one is a more practical, mature technology for implementing large scale neural networks for now.

A. 180-nm ESF1 Floating-Gate

In this section, we have modified a commercial NOR flash memory array to enable high-precision tuning of individual floating-gate cells for analog computing applications. The modified array area per cell in a 180-nm process is about 1.5 μm^2. While this area is approximately twice the original cell size, it is still at least an order of magnitude smaller than in the state-of-the-art analog circuit implementations. The new memory cell arrays have been successfully tested, in particular confirming that each cell may be automatically tuned, with ~1% precision, to any desired subthreshold readout current value within an almost three-orders-of-magnitude dynamic range, even using an unoptimized tuning algorithm.

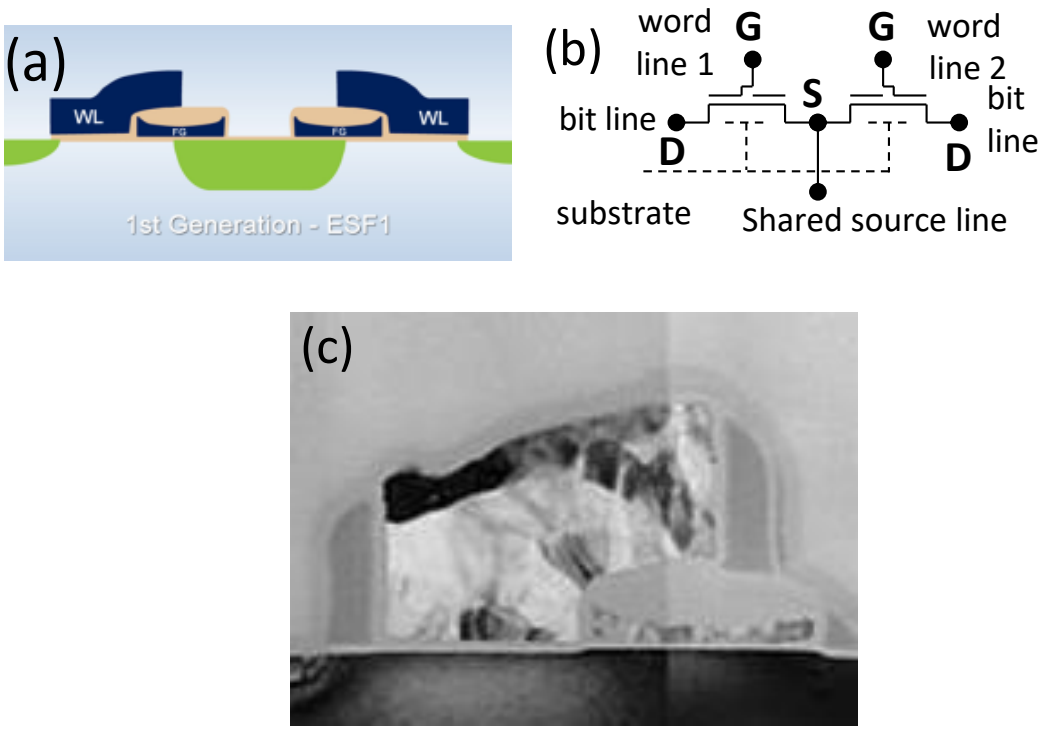


Fig. 1. SST's ESF-1 technology [59]: (a) schematic cross-section of a supercell, (b) its equivalent circuit, and (c) TEM cross-section image of one half of the supercell implemented in a 180-nm process.

1. Device Characterization

The 180-nm NOR memory array consists of "supercells" (Fig. 1). Each supercell is a common-source assembly of two floating-gate memory cells with a highly asymmetric structure: the control gate (usually connected to a "word" line) overlaps the drain region of cell's MOSFET transistor, while being separated from its source region by the floating gate. Because of that, the direct effect of the gate voltage on the process of electron emission by the source is very small. This is evident from the readout characteristics of the cell, shown in Fig. 2: at $V_{DS} > 0$, when the source-to-drain current is due to the electron emission from the source, a large gate voltage is necessary to open the transistor of a fully programmed cell (with negatively charged floating gate). On the other hand, at $V_{DS} < 0$, when electrons are

emitted by transistor's drain, the effect of control gate voltage on the current is much stronger, while that of the floating gate charge is much weaker.

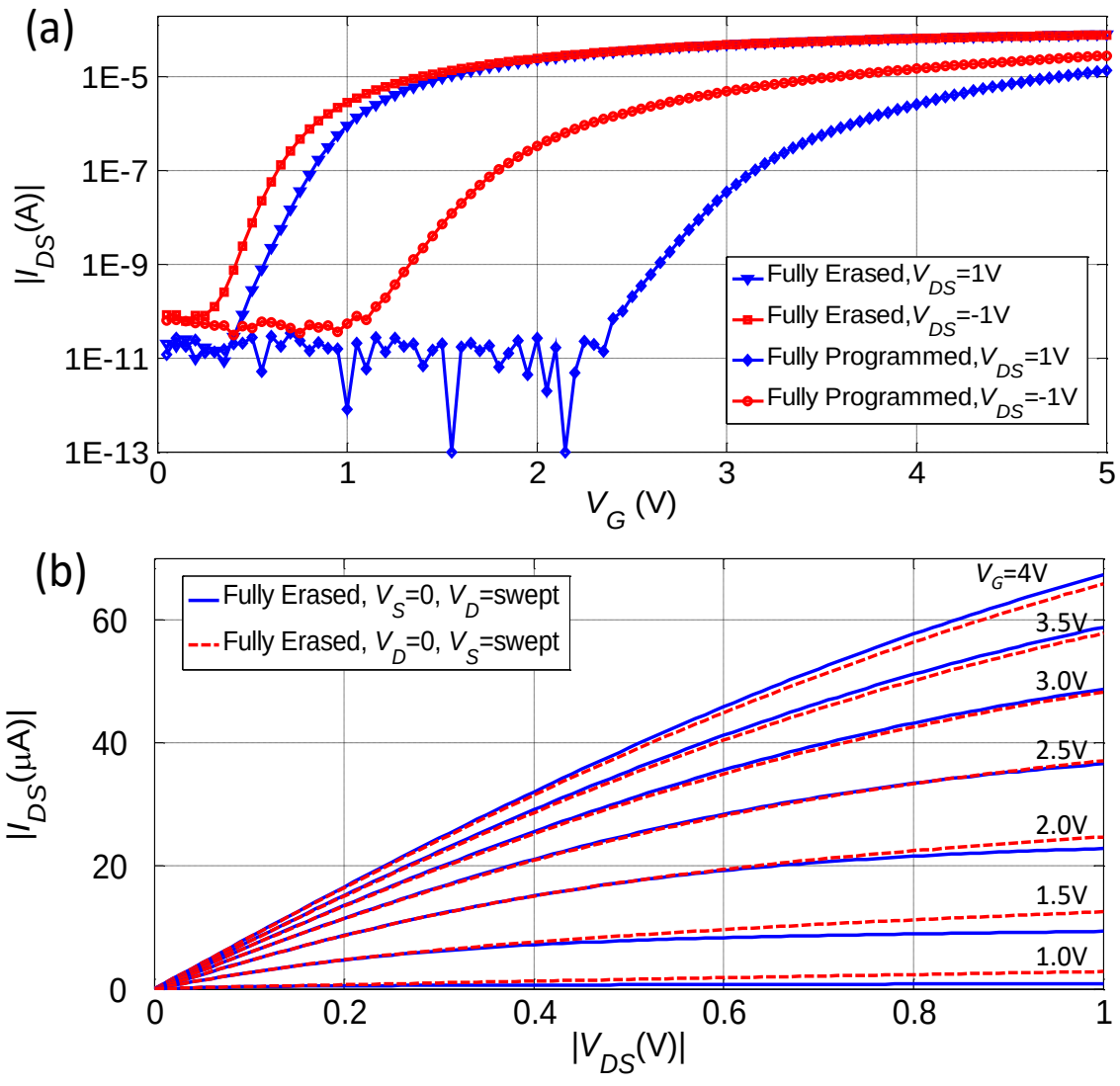
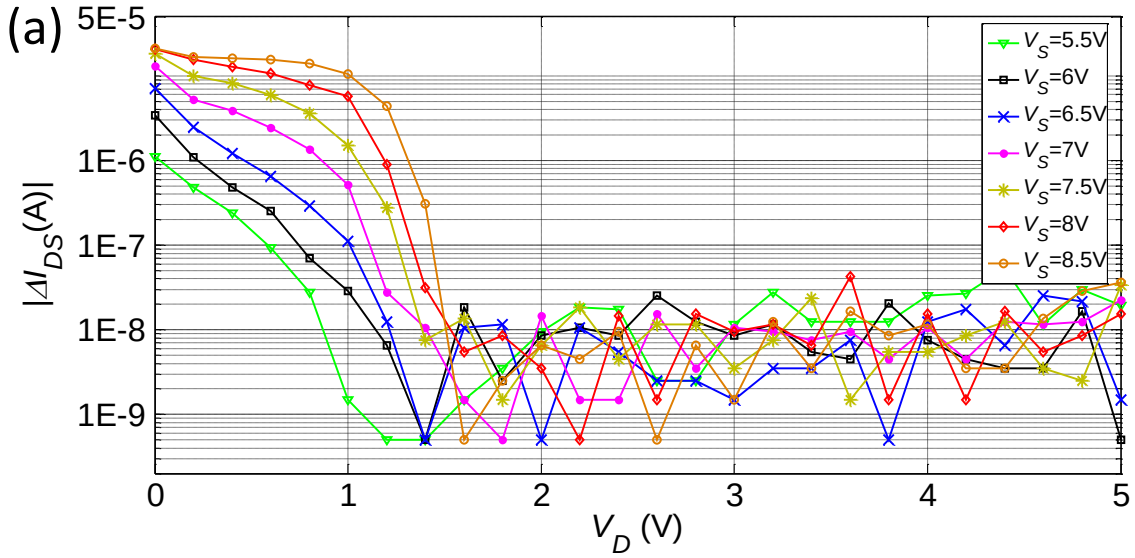
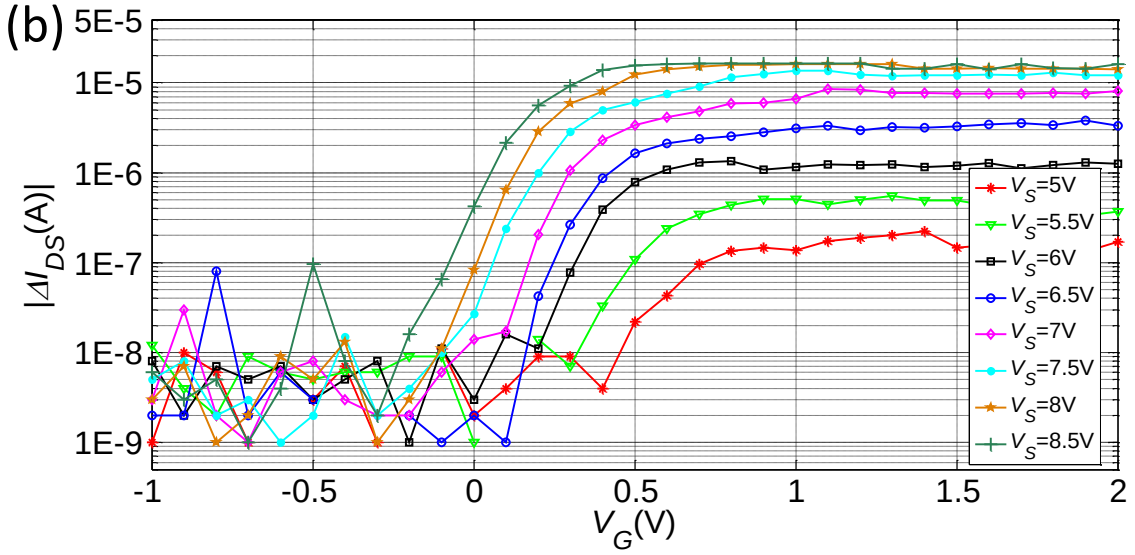
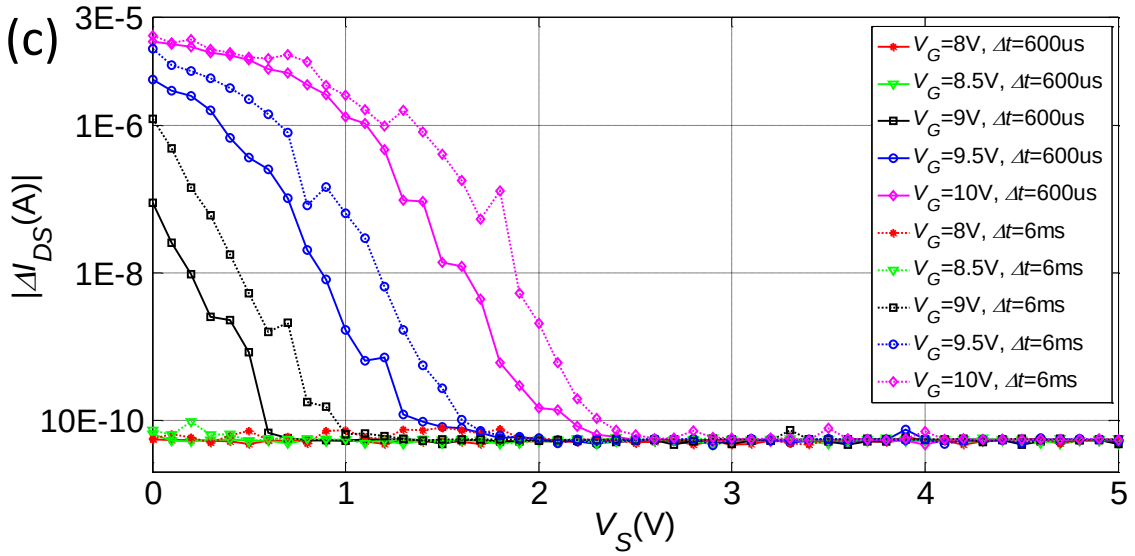


Fig. 2. Readout characteristics of 180-nm ESF-1 memory cells: Drain-source current as a function of (a) gate and (b) drain-source voltage.


(a)
5E-5
1E-6
1E-7
1E-8
1E-9
$|\Delta I_{DS}(A)|$
0
1
2
3
4
5
V_D (V)
V_S=5.5V
V_S=6V
V_S=6.5V
V_S=7V
V_S=7.5V
V_S=8V
V_S=8.5V


(b)
5E-5
1E-5
1E-6
1E-7
1E-8
1E-9
$|\Delta I_{DS}(A)|$
-1
-0.5
0
0.5
1
1.5
2
V_G(V)
V_S=5V
V_S=5.5V
V_S=6V
V_S=6.5V
V_S=7V
V_S=7.5V
V_S=8V
V_S=8.5V


(c)
3E-5
1E-6
1E-8
10E-10
$|\Delta I_{DS}(A)|$
0
1
2
3
4
5
V_S(V)
V_G=8V, Δt=600us
V_G=8.5V, Δt=600us
V_G=9V, Δt=600us
V_G=9.5V, Δt=600us
V_G=10V, Δt=600us
V_G=8V, Δt=6ms
V_G=8.5V, Δt=6ms
V_G=9V, Δt=6ms
V_G=9.5V, Δt=6ms
V_G=10V, Δt=6ms

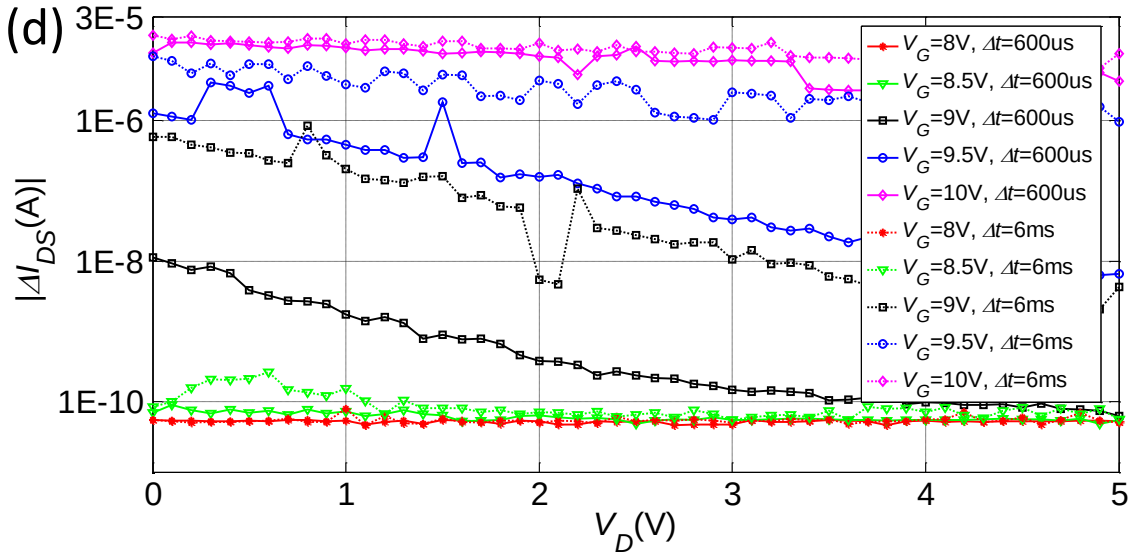


Fig. 3. Analog tuning of 180-nm ESF-1 memory cells, characterized by the change in source-to-drain current I_{DS} (as measured at V_G = 2.5V, V_D = 1V, and V_S = 0V) under effect of applied voltage pulses: (a, b) gradual programming of an (initially erased) device with 5-μs source voltage pulses of various amplitudes V_S; (c, d) gradual erasure of an (initially programmed) device with gate voltage pulses of various amplitudes V_G and durations Δt.

The same structure asymmetry affects the switching dynamics of the cell (Fig. 3). During the "programming" process, the negative charge of the floating gate may be increased very fast using very effective hot-electron injection from the source area of transistor's channel, while the simplest way to decrease it (and hence "erase" the cell) is via the Fowler-Nordheim tunneling of electrons from the floating gate to the control gate, by applying a rather high voltage (~ 11 V) to the latter electrode.

Our network design uses energy-saving gate coupling [17, 66, 67] of the peripheral and array cells, which works well in the subthreshold mode, with a nearly exponential dependence of the drain current I_{DS} of the memory cell on the gate voltage V_{GS} (Fig. 4):

$$I_{\mathrm{DS}} \approx I_0 \exp\left\{\beta \frac{V_{\mathrm{GS}} - V_t}{V_{\mathrm{T}}}\right\}, \qquad (1)$$

where V_T is a threshold voltage depending on the memory state of the cell (physically, the electric charge of its floating gate), $V_T \equiv k_B T/e$ is the voltage scale of the thermal excitations, equal to ~26 mV at room temperature, while $\beta < 1$ is the dimensionless subthreshold slope $d(\ln I_{DS}/dV_{GS})$, measured in the units of V_T, and characterizing the efficiency of the gate-to-channel coupling. As the inset in Fig. 4d shows, in the ESF1 cells this slope stays relatively constant in a broad region of memory states – a feature enabling the gate-coupled circuit operation. (For lower V_t, the slope becomes higher, apparently due to the cell's split-gate design – see Fig. 1a.)

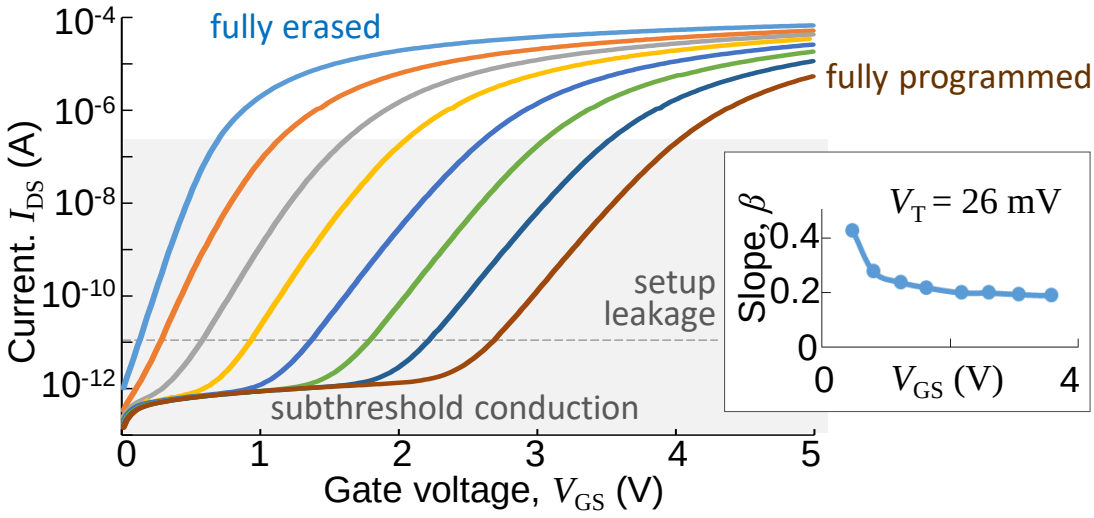


Fig. 4 The grey-shaded region shows the subthreshold conduction region; the currents below I_{DS} = 10 pA (the level shown with the dashed line) are significantly contributed by leakages in the experimental setup used for the measurements. The inset shows the extracted slope of this semi-log plot, measured at I_{DS} = 10 nA, as a function of the memory state (characterized by the corresponding gate voltage).

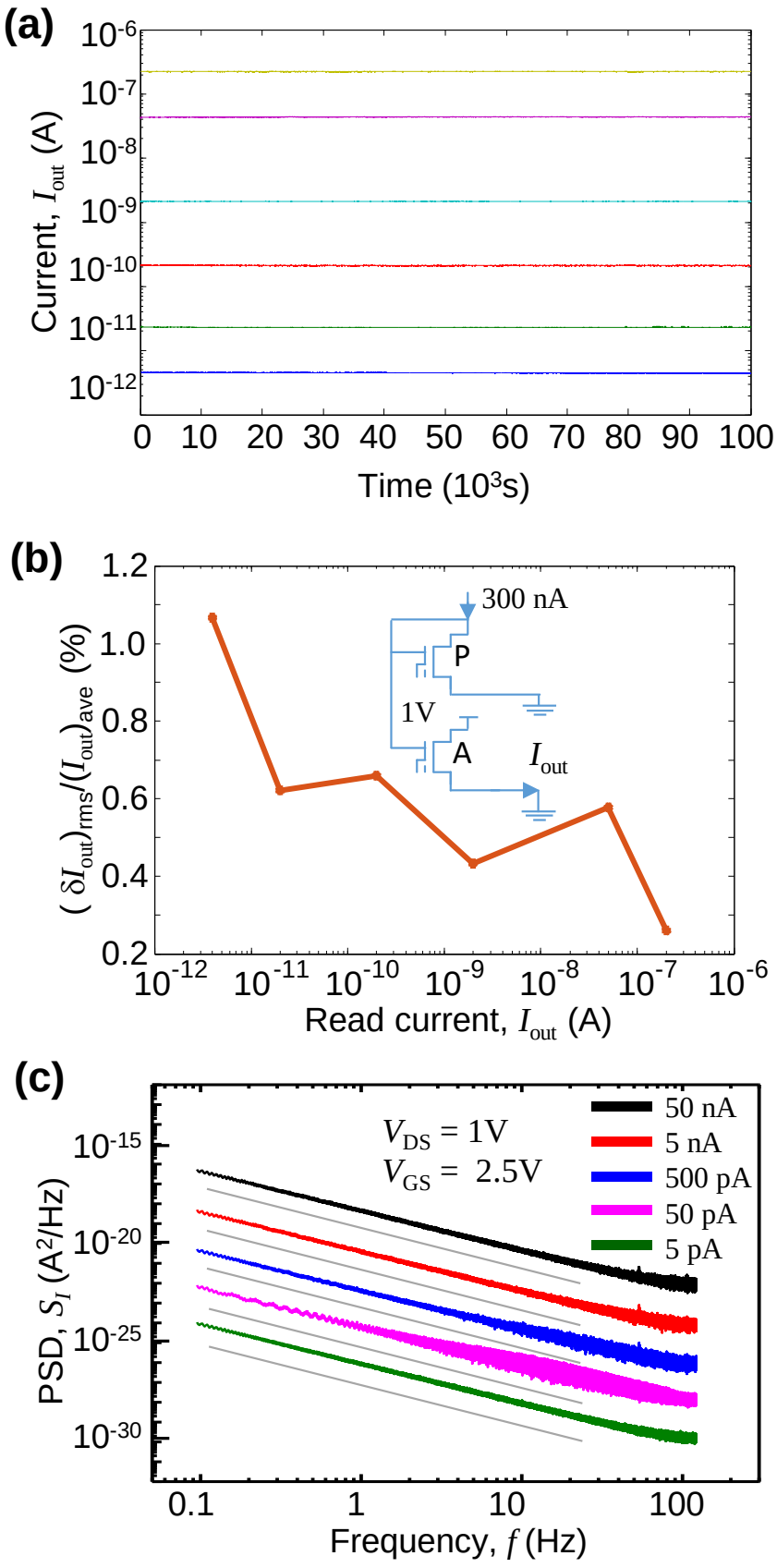


Fig. 5. (a) Results of analog retention measurements for several memory states, performed in the gate-coupled array configuration, and (b) the average relative variation of the currents during the same time interval. The inset shows the equivalent circuit of the used gate coupling. Each point on panel (a) is an average over 65 samples taken within a 130 ms period. (c) Spectral density of cell current's noise measured at room temperature; the gray lines are just guides for the eye, corresponding to $S_I \propto 1/f^{1.6}$.

With the requirement to keep the relative current fluctuations (Fig. 5b) below 1%, the dynamic range of the subthreshold operation is about five orders of magnitude, from ~10 pA to ~300 nA, corresponding to the gate voltage swing of ~1.5 V.

The ESF1 flash technology guarantees a 10-year digital-mode retention at temperatures up to 125˚C [59]. Our experiments have shown that these cells also feature at least a-few-days analog-level retention, with very low fluctuations of the output current – see Fig. 5.

2. Array Design

The top row of Fig. 6 shows the usual structure of the NOR flash memory and its programming/erasure voltage protocols, employing these properties of the SST cells. In this architecture, cells of the same row share transistor source and control gate ("word") lines, while transistor drains of all cells of the same column are connected to the same "bit" line. Fig. 6a shows the set of applied voltages used for programming of the top left cell, while avoiding state disturb in all other cells. In particular, a positive bias $V_D^{P'} > 2V$, applied to all unselected bit lines, inhibits unintentional hot-electron injection in all unselected cells, including type-A half-selected cells (sitting on the selected word line). Also, grounding of unselected word lines guarantees the absence of disturb processes (such as the back Fowler-Nordheim tunneling) in all unselected cells including half-selected cells of type B (sharing the source voltage with the selected cell). As Fig. 3a indicates, the same programming protocol, only with pulsed source voltage and slightly modified voltage values, allows analog programming of the selected cell, also without disturbing the half-selected cells, regardless of their charge state.

Unfortunately, in this memory architecture the opposite process of cell erasure (Fig. 6b) is much less controllable. Namely, the fully selected cell and the type-C half-selected cell

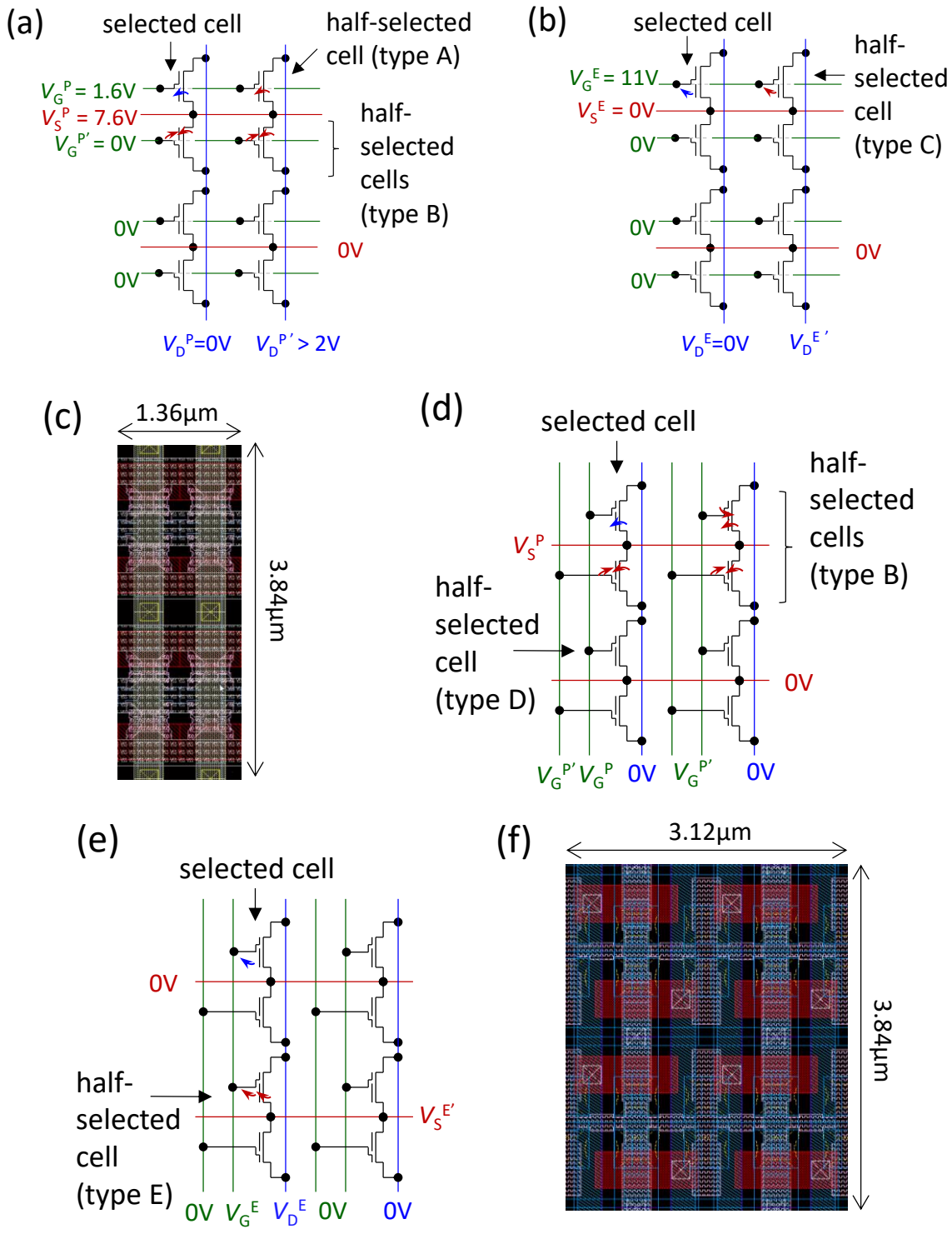


Fig. 6. Floating gate recharging effects: (a, b) – in the original SST array (c), and (d, f) – in the array with modified routing (f), on the example of a 2×2 supercell array fragment. Voltages shown on panels (a, d) correspond to programming of the top left cell, while those on panels (b, e), to its erasure. Blue and red arrows show, respectively, the useful and undesirable recharging processes. Line colors are for clarity only.

share their gate and source voltages, and due to the cell structure (Fig. 1) the process responsible for erasure (the Fowler-Nordheim tunneling of electrons from the floating gate to the control gate) is only weakly affected by the drain voltage V_D – the only voltage which may be different for these two cells. (The possible increase of $V_D^{E'}$ is limited by the onset of large drain-to-source current.) For digital applications, this feature is not a handicap, because in flash memories all cells of the same row are erased simultaneously. However, in analog applications it is highly desirable to perform not only a gradual programming of each cell, but also a gradual erasure of each cell without disturbing its neighbors. Our detailed measurements (see, e.g., Fig. 3c, d) have shown that in the baseline architecture (Fig. 6a-c) the latter operation is impossible for any bias voltage set.

To resolve this problem, we have modified the array structure (without changing the optimized cell fabrication technology) as shown in the bottom row of Fig. 6, i.e. by re-routing the gate lines in the "vertical" direction, i.e. perpendicular to the source lines. A straightforward analysis of the data shown in Figs. 2 and 3 shows that the new design resolves the half-selected cell disturb problem, by using the applied voltage protocol shown in Fig. 6d, e, with $V_G^E \approx 8.5$V, $V_S^{E'} \approx 3$V, and $V_D^E \approx 3$V.

Indeed, for the programming operation, most of the half-selected cells are of the type B, while the disturb in type D cells, with $V_G^{P'} = -1$V, is even less problematic. For the erase operation, the new gate line routing enables taking advantage of the very strong nonlinearity of possible Fowler-Nordheim and hot-electron tunneling currents (as functions of, respectively, the drain and source voltages), to completely inhibit these effects in all unselected cells including half-selected cells of type E.

The SST cell array with the architecture shown in Fig. 6d, e has been designed, fabricated (so far in the 180-nm technology of SST, Inc.) and successfully tested. Fig. 6f

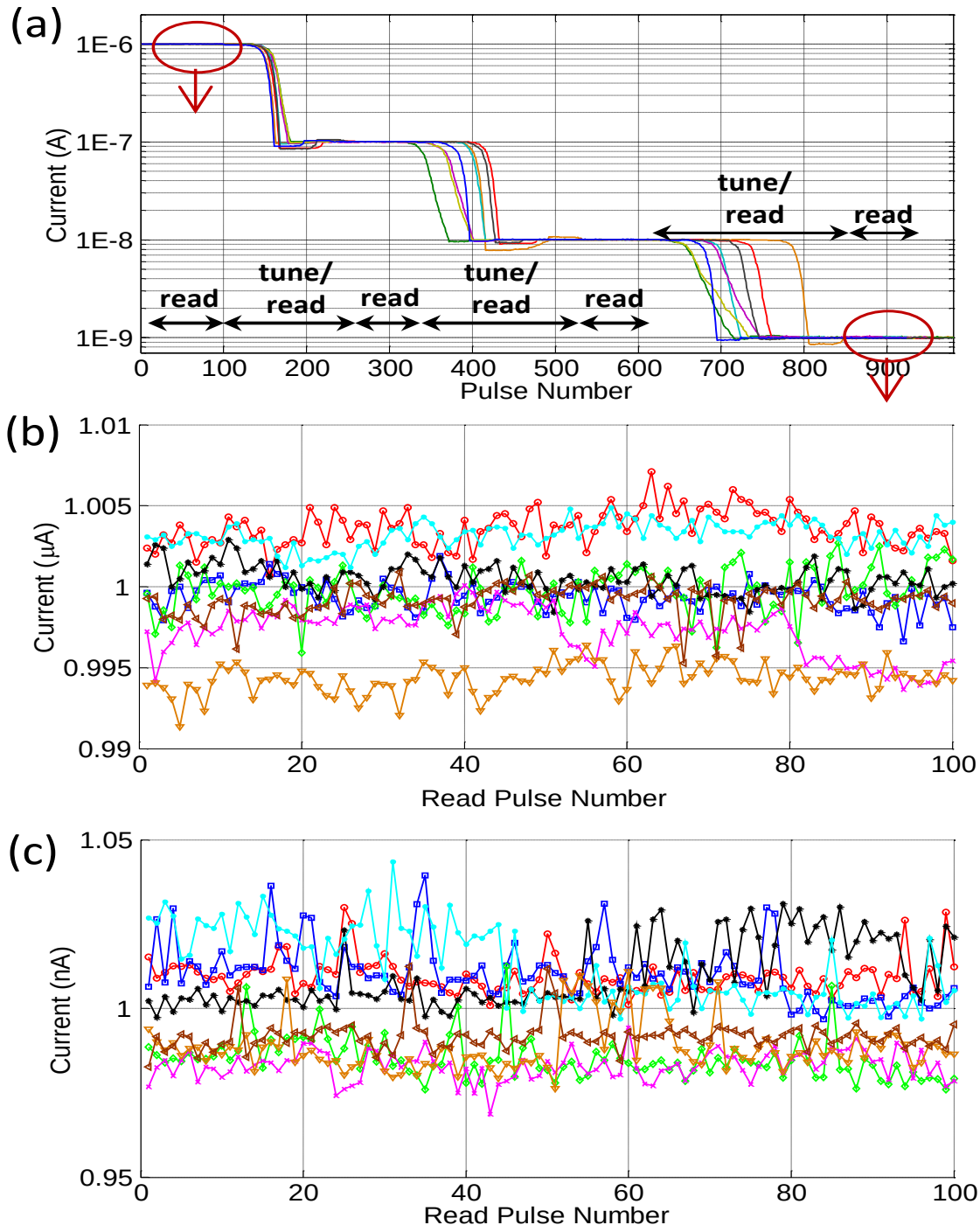


Fig. 7. High-precision tuning of cells of the modified memory: (a) All cells being tuned sequentially to 1 μA, 100 nA, 10 nA, and 1nA readout currents (as measured at V_G = 2.5V, V_D = 1V, V_S = 0V); (b, c) zoom-in on the readout of the first and the last of the tuned states, to highlight the current variations due to intrinsic device noise. On all panels, each point represents the current average over a 10-ms time interval.

shows the layout of the new array. Its area per cell is 2.3 times larger than the original one (Fig. 6c) due to the additional real estate needed to accommodate two gate lines for each cell column.

To verify that the new array architecture enables a full inhibition of half-select disturb effects, we have performed a series of experiments, tuning all 8 cells in a 2×2 supercell array, one by one, to pre-selected goal values with a ~1% precision (Fig. 7), using a simple, fully automated feedback procedure that had been originally developed for tuning memristive devices [46, 68]. Its algorithm consists of alternating "tune" (either program or erase) and "read" pulses applied to the selected device. Every read measurement determines the necessary direction of the next tune operation, i.e. whether program or erase pulses are needed. If a read measurement shows that the desired value has been overshot, the tuning pulse polarity is changed. The tuning procedure stops when the device has reached the desired analog state with the pre-specified precision [46, 68].

In the particular series of experiments shown in Fig. 7, the initial erase was performed with a 10-ms, 10-V gate pulse, keeping $V_D = V_S = 0$, while the initial programming, with a 5-μs, 9-V source pulse, keeping $V_D = 0$V and $V_G = 1.6$ V. The gradual programming was done using 5-μs source voltage pulses with an initial amplitude of 4.5 V, which was then ramped up to 8V in 50-mV steps, while applying dc voltages $V_G^P = 1.6$V, $V_G^{P'} = -1$V and keeping other lines grounded. The gradual erase was performed using 0.6-ms gate pulses with an initial amplitude of 5V, which was then increased to the maximum value of 8.5V, also in 50-mV steps, while applying dc voltages $V_D^E = 2.7$V, $V_S^E = 0$V, $V_S^{E'} = 2.7$V, and keeping other lines grounded. (This choice of voltages is likely suboptimal and may be improved to increase tuning speed.)

B. 55-nm ESF3 Floating-Gate

The advanced commercial 55-nm ESF3 technology of the same company [59], with good prospects for its scaling down to at least F = 28 nm. The modified arrays enable high-precision individual analog tuning of each cell, with sub-1% accuracy, while keeping the highly-optimized cells, with their long-term state retention, intact. The array has an area of 0.33 μm^2 per cell, and is at least one order of magnitude denser than the reported prior implementations of nonvolatile analog memories.

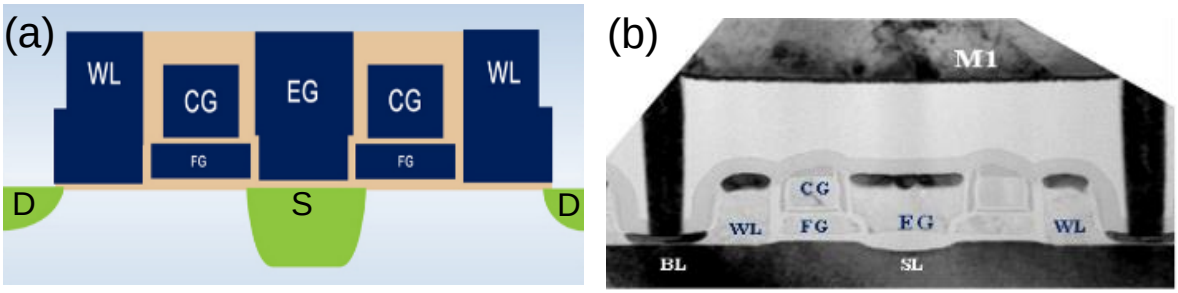


Fig. 8. SST's 55-nm ESF3 NOR flash memory cells: (a) schematic view, and (b) TEM image of the cross-section of a "supercell" incorporating two floating-gate transistors with a common source (S) and erase gate (EG) [59].

1. Device Characterization

The ESF3 NOR flash memory is based on "supercells" with two floating-gate transistors sharing the source (S) and the erase gate (EG), but are controlled by different word-line (WL) and coupling (CG) gates - see Fig. 8. In the original ESF3 memory arrays, the cells are connected as Fig. 9a shows, with six row lines per supercell, connecting transistor sources, erase gates, coupling gates, and word-line gates, while each column has only one ("bit") line connecting transistor drains (D). In this array topology, each cell may be programmed individually, by hot-electron injection into its floating gate. For that, the voltage on the source line (SL in Fig. 9) of the cell's row is increased to 4.5 V (while those in other rows are kept at 0.5 V), with the proper column selected by lowering the bit line (BL) voltage to

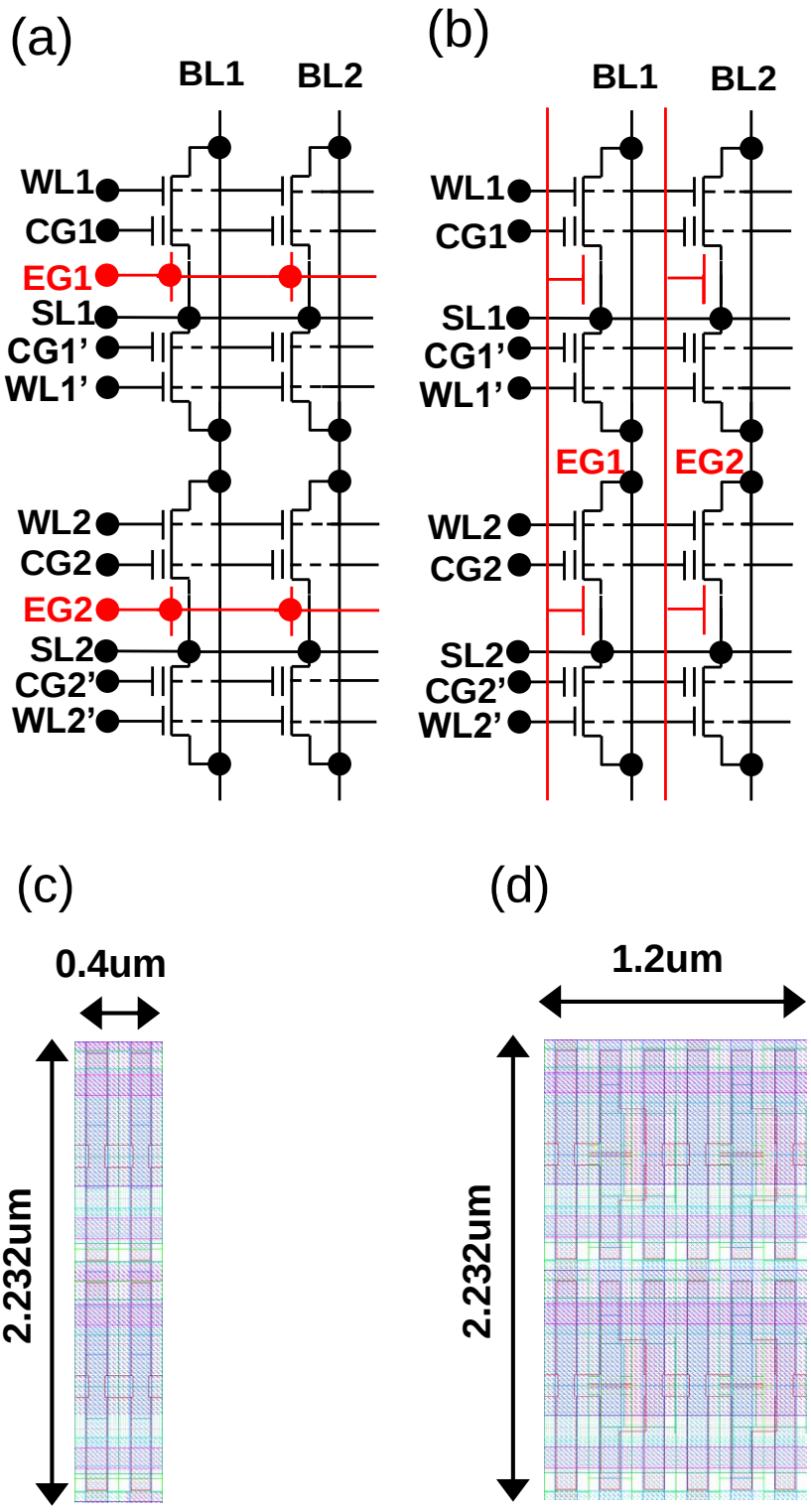


Fig. 9. (a, c) Original and (b, d) modified circuitry and layout for 4-supercell ESF3 55-nm memory array.

0.5 V (while keeping all other bit line voltages above 2.25 V). This process works well for providing proper digital state, with 1- or even 2-bit accuracy. However, it is insufficient for cell tuning with analog (say, 1%) precision. Unfortunately, the reverse process ("erasure"), using the Fowler-Nordheim tunneling of electron from the floating gates to the erase gates, may be performed, in the original arrays, only in the whole row, selected by applying a high voltage of ~ 11.5 V to the corresponding erase gate line (with all other EG voltages kept at 0 V). So, these arrays do not allow for a precise analog cell tuning, which unavoidably requires an iterative, back-and-forth (program-read-erase-read-program...) process, with the run-time result control. The required modification details will be discussed in the next section.

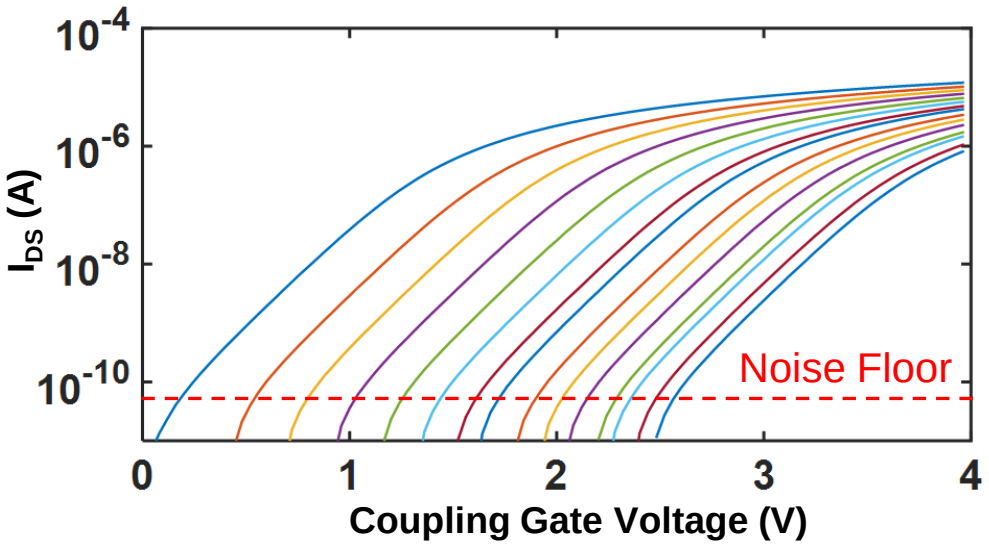


Fig. 10. Drain-source current of a modified ESF3 cell as a function of the coupling gate voltage, for several different memory states.

The ability to tune the floating gate cells of the modified arrays continuously is illustrated in Fig. 10 which shows the readout current as a function of the coupling gate voltage for a selected equidistant series of cell states. These semi-log plots have wide quasi-

linear segments, corresponding to the nearly-exponential behavior of the current in the subthreshold region. In the current range from 100 pA and 30 nA, the subthreshold slope factor n, defined by the well-known relation $I \propto \exp\{qV_{CG}/nk_BT\}$, varies only from 5 to 5.1 for all the 15 states shown in Fig. 10. This low variability of n enables the implementation of highly linear signal transfer in gate-coupled current mirrors using these cells [17].

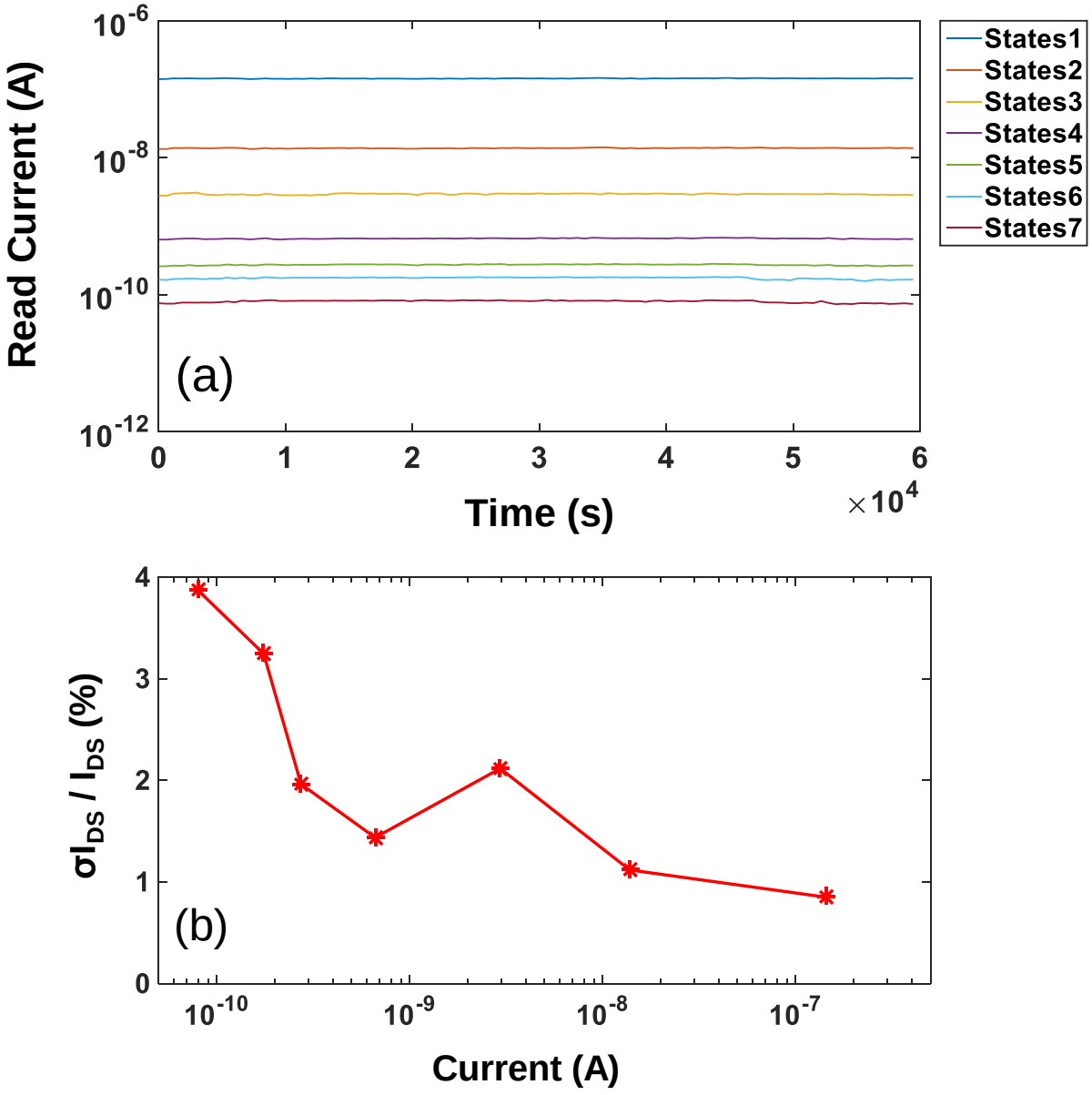


Fig. 11. (a) Retention measurements for several cells in different memory states at 85°C, and (b) the average relative variation of the readout currents during this time interval.

The ESF3 flash technology guarantees a 10-year digital-mode retention at temperatures up to 125˚C [59]. To explore its analog mode retention, we have programmed 7 memory cells to 7 different states from around 100 pA to 100 nA covering the whole subthreshold region, and then were continuously monitoring their output current within a day under 85 ˚C as shown in Fig. 11a. Each point on this panel is an average over 128 samples taken during 16 ms periods. Fig. 11b shows the relative r.m.s. variation of the current during the measurement period for the 7 states shown in Fig. 11a. For larger currents the variation is below 1%, increasing to ~4% only at the lower boundary of the range.

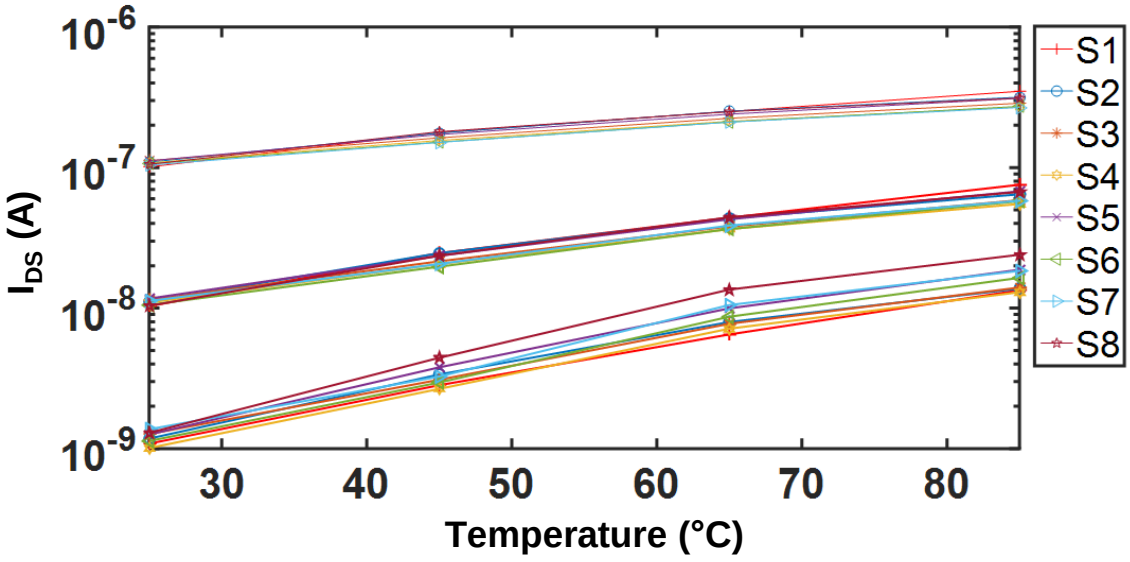


Fig. 12. Temperature dependence of drain-source current for several memory cells with different memory states, with the initial readout currents close to 1 nA, 10 nA and 100 nA.

In order to fairly characterize the temperature dependence of the cell output current in the subthreshold region, we have programmed 8 cells to 8 different states, equally spread over the useful dynamic range. Then, in 3 different experiments, appropriate coupling gate voltages were applied to each cell, to make the readout currents of them all equal to,

sequentially, 1 nA, 10 nA and 100 nA at 25 ˚C. After that, temperature was ramped up from 25˚C all the way to 85˚C, and the readout current of each cell was monitored. Fig. 12 shows the results of these 3 experiments. In accordance with our expectations (and the measured values of *n*), the currents increased significantly – more than by an order of magnitude for the lowest initial current. Though in the gate-coupling scheme (see below) this changes are mostly compensated by similar changes in the input (peripheral) transistors, this fact still shows that the temperature sensitivity of the subthreshold current requires special attention.

2. Array Design

We have modified the ESF3 memory arrays as shown in Fig. 9b, by connecting the erase gates of all cells of one column with an additional line, while eliminating the row lines connecting these gates. (Note that this redesign is different from the one performed by our group earlier [60] with the 180-nm ESF1 technology, because of a different structure of its supercells.) Fig. 13 shows our test 10×10 modified ESF3 cell array. The peripheral cells designed for multiplier purpose are located in the additional columns on the left and the right from the basic array. (Two columns are necessary because with the ESF3 supercell structure we can use only one half of each supercell as the peripheral cell.)

In the modified arrays, the analog hot-electron programming of each cell may be performed by applying 10 μs pulses of a fixed amplitude of 4.5 V to the source line of the corresponding row. In this process, the proper column is selected by applying a positive voltage ~4 V between the erase-gate and bit lines, while keeping this voltage negative for all un-selected columns [69]. Fig. 14a documents the inhibition of the unwanted programming process in a half-selected cell at the increase of the bit-line (i.e. drain) voltage.

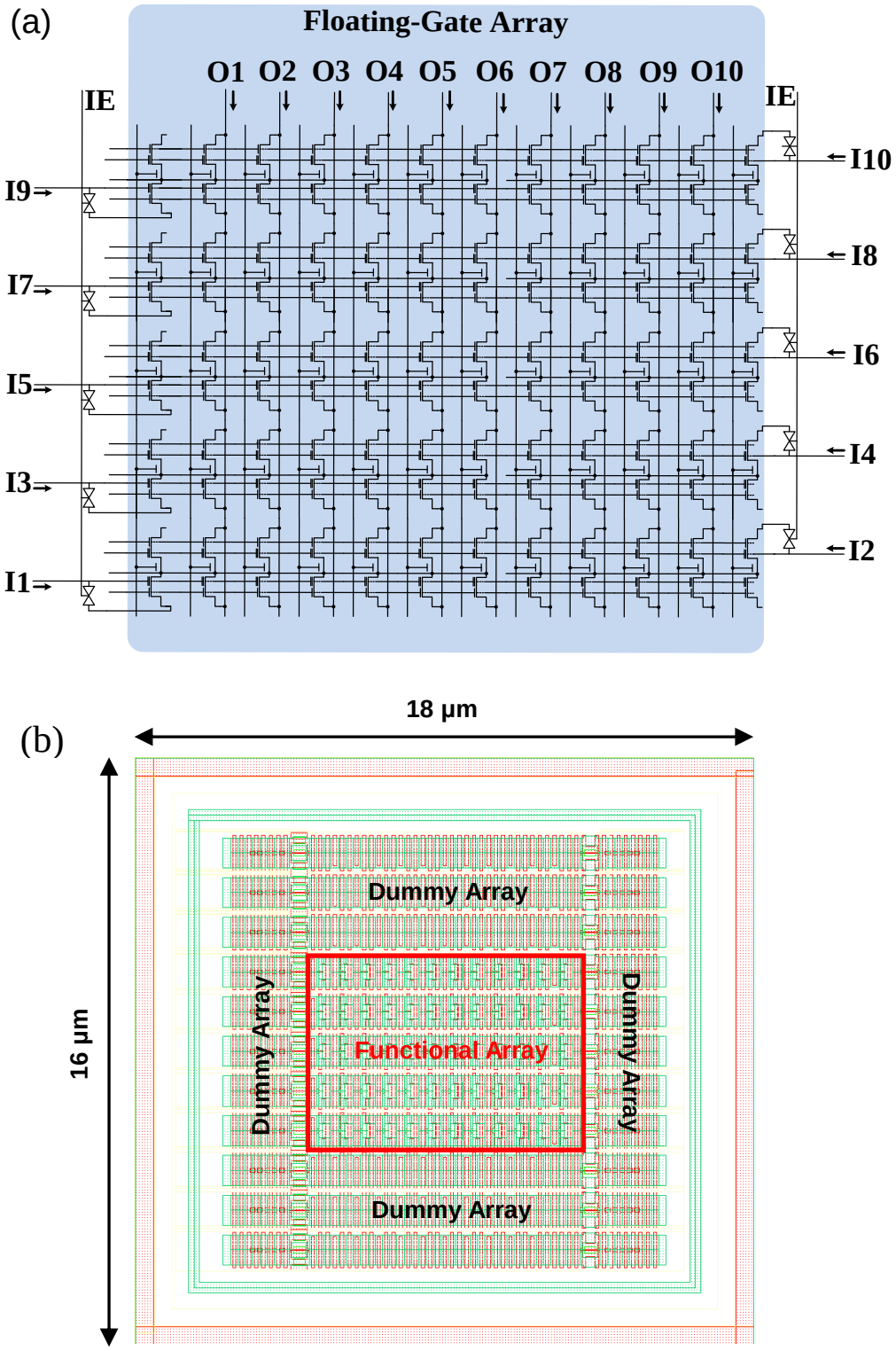


Fig. 13. Gate-coupled vector-by-matrix multiplier based on a 10×(10+2) array of ESF3 floating-gate cells, together with auxiliary pass-gates (which are disabled during tuning with IE signal): (a) schematics; (b) layout for 55-nm fabrication.

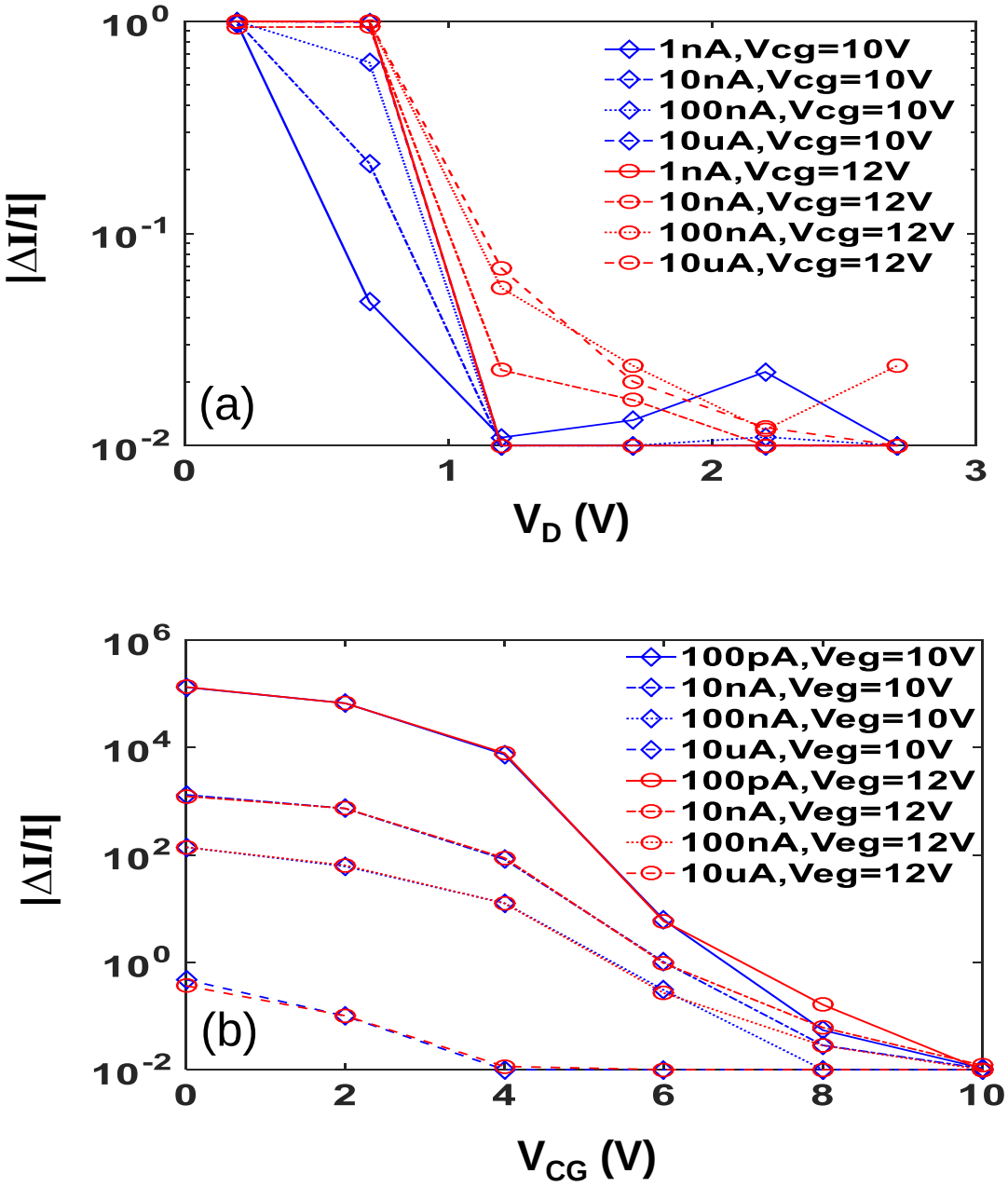


Fig. 14. (a) Programming inhibition and (b) erasure inhibition in the transistors of half-selected cells. Unless specified otherwise, the shown readout (source-to-drain) currents have been measured at V_{WL} = 2.5 V, V_{CG} = 2.5 V, V_D = 1 V, V_S = 0 V, and V_{EG} = 0 V.

The opposite process of individual analog erasure via the Fowler-Nordheim tunneling is now also possible, by using the new column lines to apply high-amplitude (11.5 V), 0.5 ms pulses to the erasure gates of the selected column. The proper row is selected by grounding

the corresponding coupling gate line, while keeping a high voltage (+8 V) on these lines of unselected rows. As Fig. 14b shows, such a positive bias inhibits the Fowler-Nordheim tunneling in half-selected cells, due to a relatively high capacitance between the coupling gate and the floating gate of the same transistor [70].

Due to the line rerouting, the array area per cell has nearly tripled – cf. Figs. 9c and 9d. However, even with this increase the area is still as small as 0.33 μm^2, i.e. ~ 110 F^2, much smaller than in any other design we are aware of.

C. Memristive Device

Another synapse candidate with high density is nonvolatile memristive devices. In their simplest form, memristors are two-terminal passive elements, the conductance of which can be modulated reversibly by applying electrical stress. Due to the simple structure and ionic nature of their memory mechanism, metal-oxide memristors have excellent scaling prospects, often combined with fast, low energy switching and high retention [36]. Many metal oxide based memristors can also be switched continuously, i.e. in analog manner, by applying electrical bias (current or voltage pulses) with gradually increasing amplitude and/or duration. The Pt/TiO_{2-x}/Pt memristive device we fabricated is shown Fig.15.

1. Device Characterization

Fig. 16a shows typical continuous switching *I-V*s for the considered Pt/TiO_{2-x}/Pt devices [71]. The devices were implemented in "bone-structure" geometry with an active area of ~1 μm^2 using the atomic layer deposition technique. An evaporated Ti/Pt bottom electrode (5nm/25nm) was patterned by conventional optical lithography on a Si/SiO_2 substrate (500 μm/200 nm, respectively). A 30 nm TiO_2 switching layer was then realized by atomic layer deposition at 200°C using Titanium Isopropoxide ($C_{12}H_{28}O_4Ti$) and water as precursor and

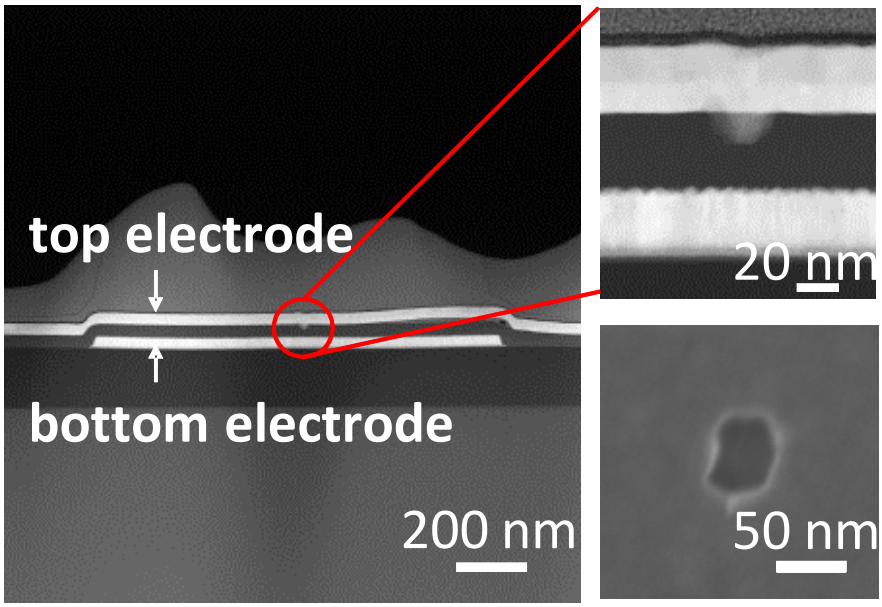


Fig. 15. TEM images of 50-nm-thick titanium dioxide devices with e-beam defined protrusion

reactant, respectively. A Pt/Au electrode (15nm/25nm) was evaporated on top of the TiO_2 blanket layer, and the device was finally rapidly annealed at 500° C in an N_2 and N_2+O_2 atmosphere for 5 minutes to improve the crystallinity of the TiO_2 material. Details of the fabrication and characterization of the considered memristors are given in Ref. [67].

After programming the memristors to the desired resistance, it was important for their state to remain unchanged during operation of the Hopfield network, so to prevent any disturbance the voltage drop across them was always kept within the $|V| \leq 0.2$V "disturb-free" range [71].

2. Modeling

The static *I*-*V* characteristics (i.e., those within disturb-free regime) for several different memory states are shown in Fig. 16b.

To assist SPICE simulation, the experimental *I*-*V* curves at small biases were fitted by the following static equation with a single memory state *G*:

$$I = GV + \beta(\alpha_1 G + \alpha_2 G^2 + \alpha_3 G^3)V^4 . \qquad (2)$$

where β = 1, α_1 = 14.7 V^{-3}, α_2 = -5.9×10^4 ΩV^{-3}, α_3 = 1.5 ×10^8 $\Omega^2 V^{-3}$ for $V > 0$, and α_1 = 34.6 V^{-3}, α_2 = -1.9×10^5 ΩV^{-3}, α_3 = 3.65 ×10^8 $\Omega^2 V^{-3}$ for $V < 0$. As it is obvious from Equation 2, memory state G is simply a conductance (I-V slope) at zero voltage.

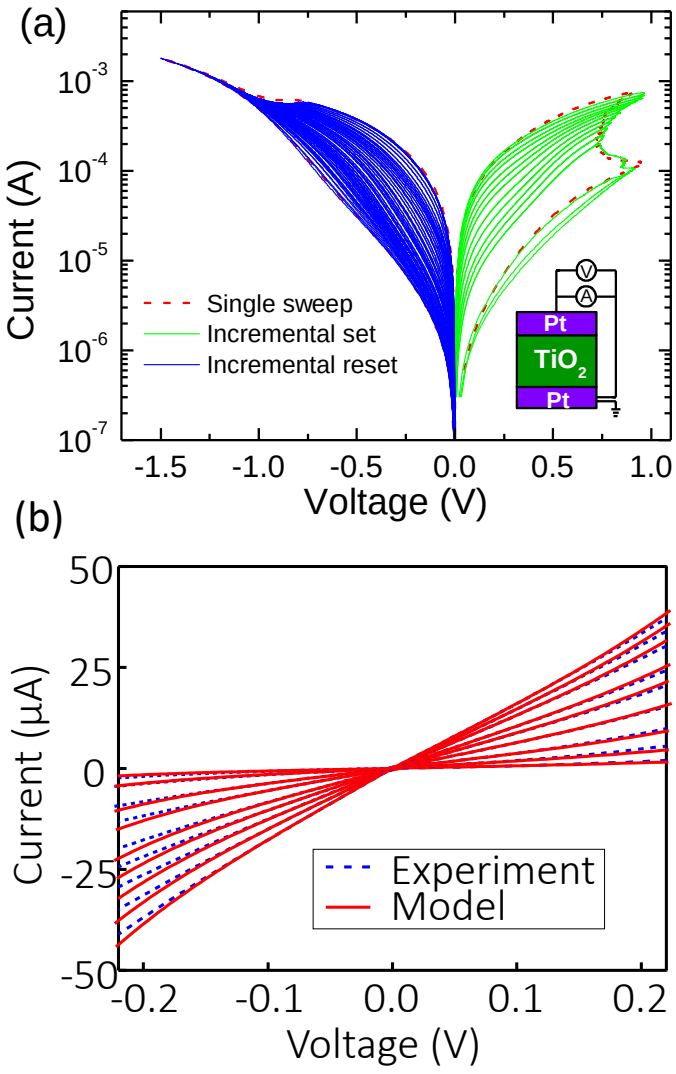


Fig. 16. (a) Typical I-V curves with current-controlled set and voltage-controlled reset switching for the considered Pt/TiO2-x/Pt memristors. (b) Modeling of static I-V curves at small disturb-free voltages for several different states.

III. High-Precision Tuning of Memory Elements

Based on the detailed characterization of multilevel property (Figs. 4 and 10) in commercial NOR memory developed by SST Incorporation, it confirms the potential of flash memories as analog weights. Although for many analog computing applications, weights are typically changed infrequently so that tuning time and energy are of less importance, it is still essential to demonstrate feasibility of high precision tuning in flash memories and potential fast tuning methodology for large scale analog systems.

A. Model-Based Fast Tuning of 180-nm ESF1 Floating-Gate Array

For ESF1 NOR flash memory, high-precision tuning experiments were performed within 10×10 array of modified memory cells with an additional two rows of supercells included to implement gate-coupled vector matrix multiplier, which is the most critical component in neural network classifiers [35] (Fig. 17). The main idea of the algorithm (Fig. 18) is to use switching dynamics of the erasure and programming processes to calculate appropriate write pulse amplitudes. Based on the fitted behavior, a formula for the required voltage amplitude required to change the readout current (i.e. from current to desired state) is derived. Due to significant device-to-device variations in switching behavior, the parameters of the model are adjusted at the initial stage of the algorithm for the specific cell being tuned (Fig. 18). Due to significant cycle-to-cycle variations, the tuning cannot be implemented by applying a single pulse. Instead, the iterative scheme with the feedback is realized in which a sequence of a write (erase or program) and read pulses are applied in each iteration. Due to much steeper erase switching, governed by Fowhler-Nordheim tunneling as opposed to hot-electron injection for programming, overshooting at the programming stage was avoided by using smaller than ideal (i.e. determined by the model amplitude) program pulses. The

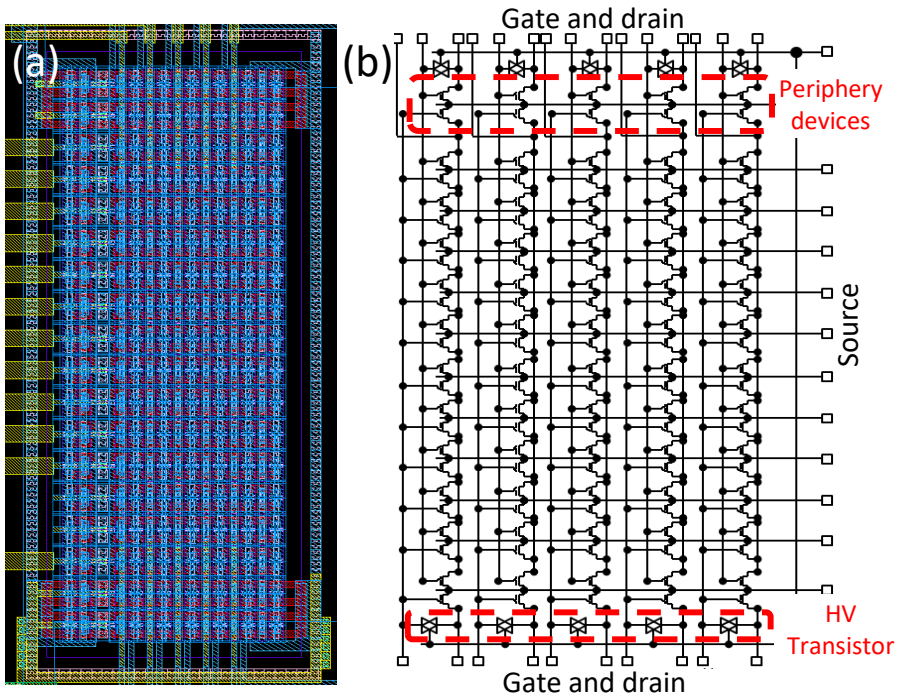


Fig. 17. Fabricated gate-coupled vector-matrix multiplier with (10+4) ×10 memory cells: (a) layout in 180-nm process, and (b) its schematics. The first and the last row of supercells are part of current mirror circuitry [66] that converts input currents into voltages that are applied to the gates of FG transistors in the array.

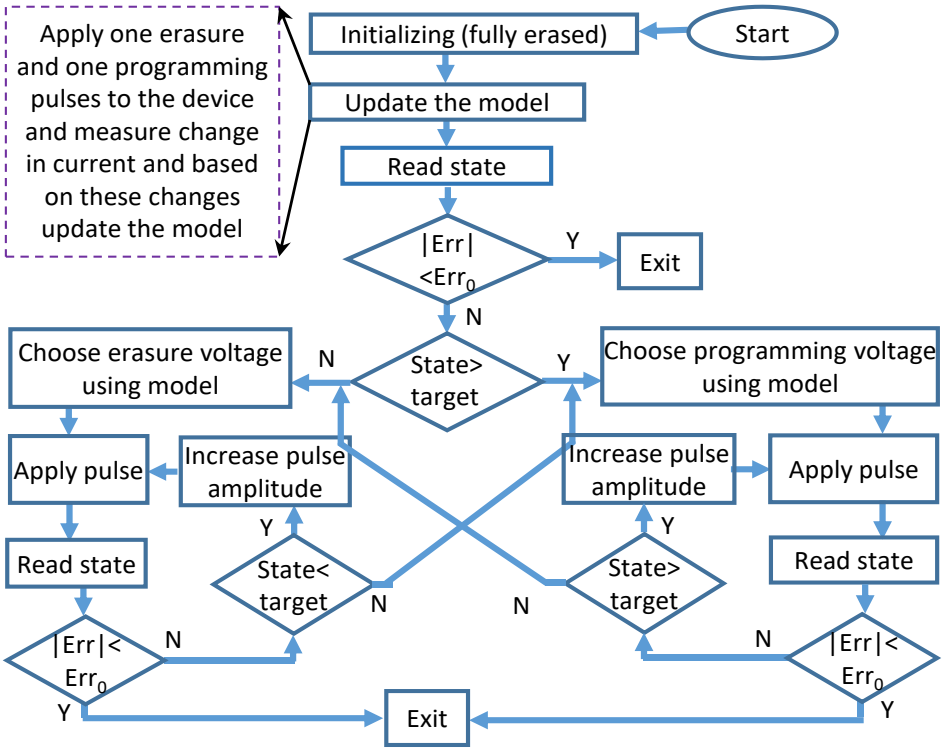


Fig. 18. Flowchart of the proposed tuning algorithm.

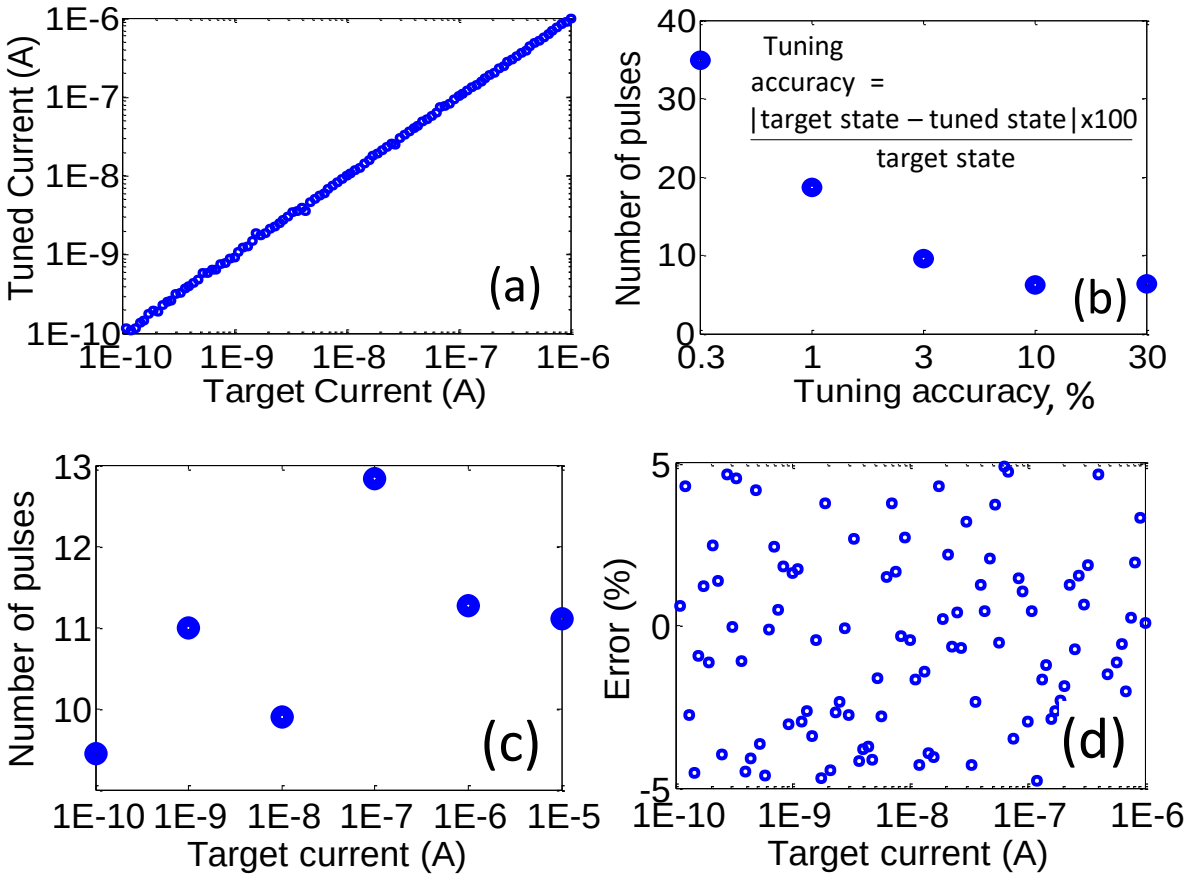


Fig. 19. (a) Measured versus target final states for 100 devices when taking into account half-select problem, and (b) tuning error as a function of the required number of tuning (erase & program) pulses. Panel a confirms that the disturbance of half-selected devices during tuning is negligible, while panel b shows that the number of tuning pulses grows exponentially with the tuning accuracy. (c) Average number of pulses and (d) tuning error for tuning 100 devices to different logarithmically-spaced target states.

optimal algorithm parameters leading to the smallest number of tuning pulses (i.e. faster tuning time) were found via exhaustive search. The tuning algorithm functionality was successfully verified in variety of conditions (Fig. 19). Naturally, the tuning was faster when tuning precision was low with roughly exponential increase in the number of tuning pulses required to get higher precision (Fig. 19b), which is similar to the tuning of phase change memory [72, 73].

B. Tuning of 55-nm ESF3 Floating-Gate Array

In order to gain continuous analog levels in memory cells and performing high precision and fast tuning for analog computing applications using 55-nm ESF3 flash memory, different amplitude and duration programming and erasing pulses will be applied. Considering the similar tuning strategy as fully automated feedback procedure in [61] for 180-nm reconfigured Flash memory, we explore the erasure dynamics by applying continuously pulses on EG with pulse amplitude (PA) of 7 V, 7.3 V, 7.6 V, 7.9 V, 8.2 V and pulse width (PW) of 10 us, 25 us, 50 us at a fully programmed device. As illustrated in Fig. 20 (a), we observe an exponential dependent on PA during erasure under subthreshold region. Moreover, the curves in different color are sited in good consistency on each other demonstrating total erasing time other than individual PW is essential for erasing. If we take advantage of that property, we can apply one long pulse instead of many short pulses when we have a model based on short pulse measurement [61]. In that way, we would save huge time on module communication for sending pulses. We observe similar programming dynamics when applying continuous programming pulses on source with PA of 3.2 V, 3.4 V, 3.6 V, 3.8 V, 4 V and PW of 2 us, 4 us at a fully erased device as shown in Fig. 20 (b). Exponential dependent on programming PA is observed. Similar to erasure process, matched curves on different color demonstrate a total programming time dependent property. That property could also be used to facilitate fast programming and save control module communication time.

Since our proposed memory array will be used to implement large scale analog computing systems, it is crucial to understand device to device variations in switching dynamics for whole memory array tuning. Fig. 21 (a, b) illustrate the threshold voltage spread for programming and erasing in 100 memory cells respectively. The programming

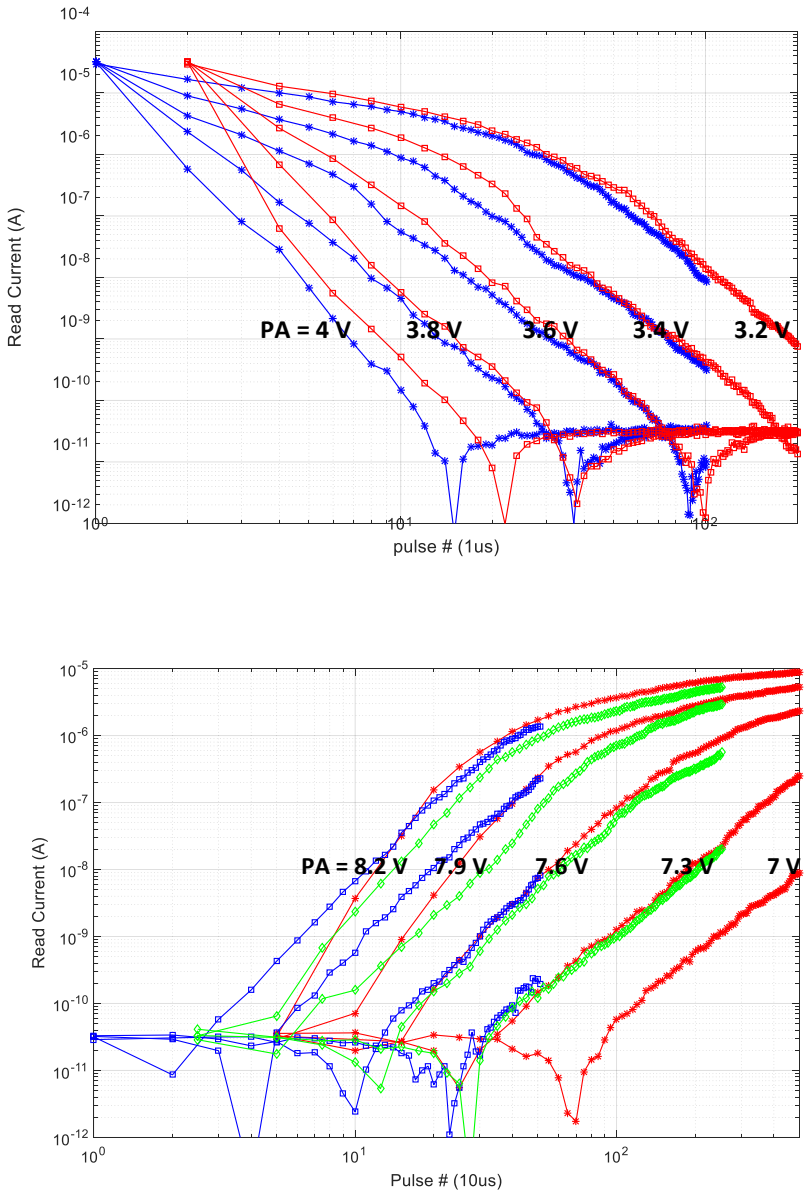


Fig. 20. Switching dynamics of a single FG transistor with different pulse amplitude (PA) as well as different pulse width: (a) programming from fully erased state for each curve, (b) erasure from fully programmed state for each curve.

threshold voltage is defined as a source voltage that will change the device from fully erased state by 30% and the erasing threshold voltage is defined as an EG voltage that will change the device from fully programmed state by 30%. A very tight spread in programming threshold voltage with few outliers and relatively larger spread in erasing voltage are observed. That is mainly because hot electron injection process is less affected by the gate

oxide thickness compared with Fowler-Nordheim tunneling, and gate oxide thickness is severely affected when technology scaling down. As a result, a programming pulse is preferred when we are approaching the target during tuning when utilizing a model based tuning strategy [61].

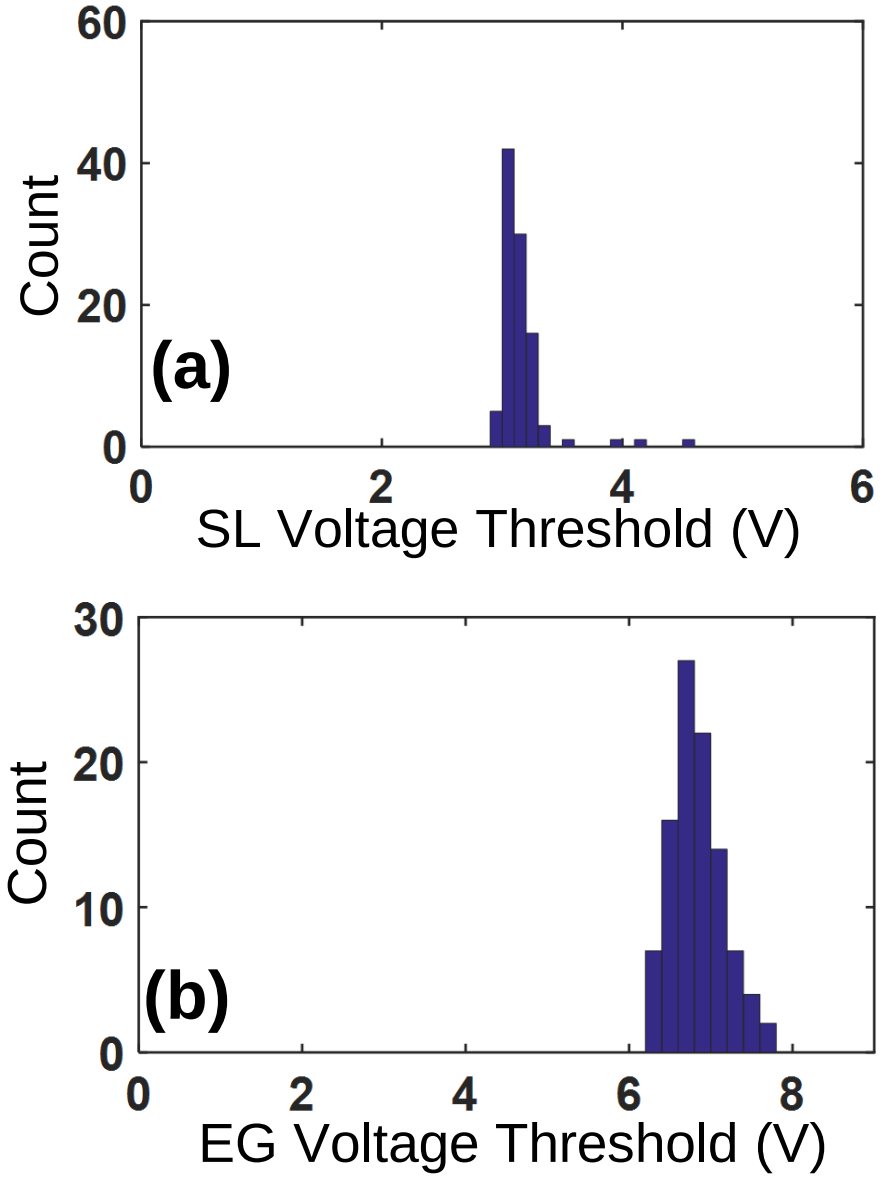


Fig. 21. (a, b) Device-to-device variations for programming and erasing voltage thresholds for 100 devices.

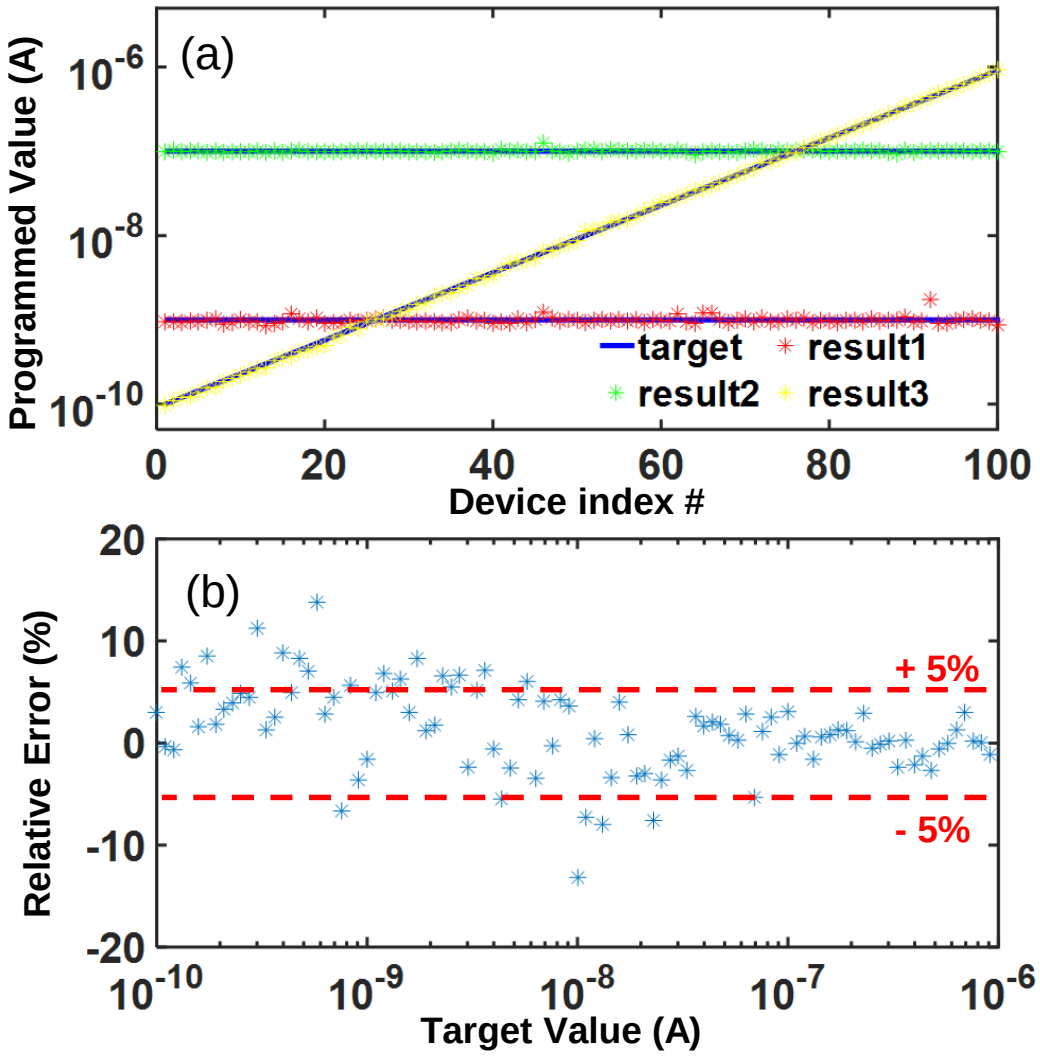


Fig. 22. (a) Measured versus target weights for 100 devices, and (b) Measured tuning error for 100 devices at a tuning precision target of 5%.

Fig. 22 illustrates the analog tuning capability of the array. All 10×10 array cells have been tuned one-by-one by an automatic feedback controlled application of alternating programming pulses to their source electrodes and erasing pulses to their erase gates. After each tuning pulse, the external control circuitry read out the cell output current at standard bias conditions, and made a decision about the next pulse's destination and amplitude, until the read-out current has reached the target value with the 5% precision [68]. Fig. 22a shows the results of 3 separate experiments of tuning all 100 cells of the array to different target

values of the output current: 1 nA (red line), 100 nA (green line), and an exponential function of the cell number, within the rage from 100 pA to 1 μA (yellow line). Fig. 22b shows the relative errors achieved in last experiment. The data mean that larger tuning errors (of the order of 10%) take place for smaller target currents, because of the relative large intrinsic noise of the devices.

IV. Vector Matrix Multiplication

The essence of the advantages using nonvolatile memory for neuromorphic network is the fact that in analog circuits, the vector-by-matrix multiplication, i.e. the key operation performed at signal propagation through any neuromorphic network, is implemented on the physical level, in a resistive crossbar circuit, using the fundamental Ohm and Kirchhoff laws (Fig. 23). On the other hand, the basic handicap of analog circuits, their finite precision, is not crucial in neuromorphic networks, due to the inherently high tolerance of their operation to synaptic weight variations [27].

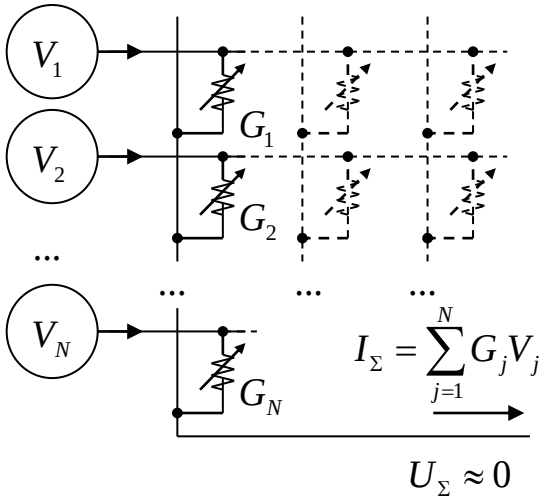


Fig. 23. Analog vector-by-matrix multiplication in a crossbar with adjustable crosspoint devices. For clarity, the output signal is shown for just one column of the array.

A. Based on 180-nm ESF1 Floating-Gate Array

We redesigned and optimized the ESF1 Flash memory from Silicon Storage Technology, Inc. (SST) (See Fig. 6) to build an analog vector-by-matrix multiplier in 180-nm technology for deep learning friendly hardware implementations. Because the original Flash memory from SST was optimized for digital memory applications, we reconfigured the original

memory array to enable precise tuning of individual cell inside the array for analog computing purposes [61].

As a first step, we have used the high-precision tuning in the modified array for a preliminary demonstration of a small-scale four-quadrant gate-coupled vector-by-matrix multiplication [66], in which peripheral floating-gate transistors had been implemented with the same SST memory technology and integrated on the same chip shown in Fig. 17.

To implement the vector-by-matrix multiplication, we have used the gate coupling of the tunable floating gate cells of each column of the array with a similar "peripheral" cell, with the virtual-bias condition imposed (by external circuitry) on the output (row) wires [18] (Fig. 24a). Since all the cells sharing the same gate have the same gate voltage, in the subthreshold operation mode the component w_i of the output I_{out} is proportional to the input I_{in}:

$$I_{in} = I_0 \exp\left\{ q \frac{V_g - V_{\text{th}}^{(p)}}{n k_{\text{B}} T} \right\}, \qquad I_{out} = I_0 \exp\left\{ q \frac{V_g - V_{\text{th}}^{(i)}}{n k_{\text{B}} T} \right\} \equiv w_i I_{in} \tag{3}$$

with current-independent proportionality coefficients w_i, which are determined by the differences of threshold voltages V_{th} of the array cells and the peripheral transistors:

$$w_i = \exp\left\{ q \frac{V_{\text{th}}^{(p)} - V_{\text{th}}^{(i)}}{n k_{\text{B}} T} \right\} \tag{4}$$

In turn, each threshold voltage is determined by the analog state (physically, the floating gate charge) of the cell, so that each w_i may be adjusted to the desirable value.

The results (Fig. 24) show an excellent linearity (derivate variation below 1%) of circuit's transfer characteristics over a wide range of input currents. In the meanwhile, we achieved an area of ~50F^2 for multiply-and-accumulate (MAC) unit.

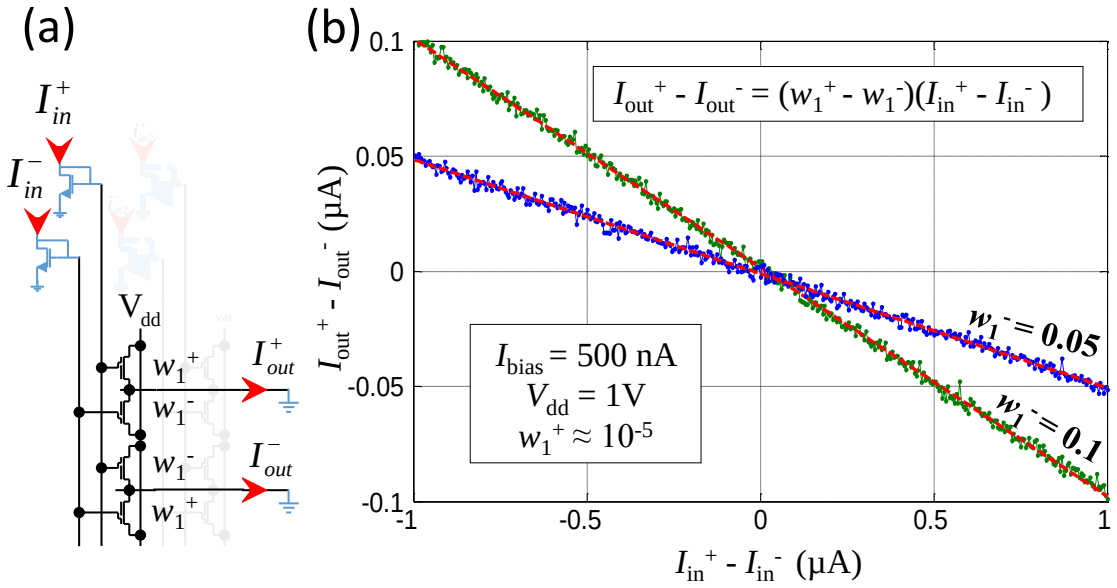


Fig. 24. Preliminary experimental results for a gate-coupled vector-by-matrix multiplier: (a) circuit schematics and (b) measured transfer characteristics for two sets of "weights" (matrix elements) w_1^-. Dotted lines show another column of the array, disengaged in these experiments.

As a simple illustration of multiplier's operation, we also fabricated and tested a 4 × 4 analog vector-by-matrix multiplier (See Fig. 25) to better evaluate the performance of nonvolatile memory approach. Fig. 26 shows the results of multiplication of 4 input signals by 16 different weights: {w_{11}, w_{12},⋯, w_{44}} = {0.1875, 0.5, 0.125, 1; 0.125, 0.9375, 0.0625, 0.4375, 0.875, 0.125, 0.375, 0.125, 0.0625, 0.6875, 0.8125, 0.25}, performed by 4 columns and 4 rows of the array, tuned with a 1% precision. We also investigated the sensitivity of multiplier precision on a selected range of array and peripheral weights, current range, and find optimal operating conditions with presence of mismatch, variations, weight-dependent subthreshold conductance slope, capacitive cross-talk, noise, retention and tuning precision.

With all factors mentioned above, we evaluate our fabricated 4 × 4 analog vector-by-matrix multiplier to achieve a total precision of ~5%.

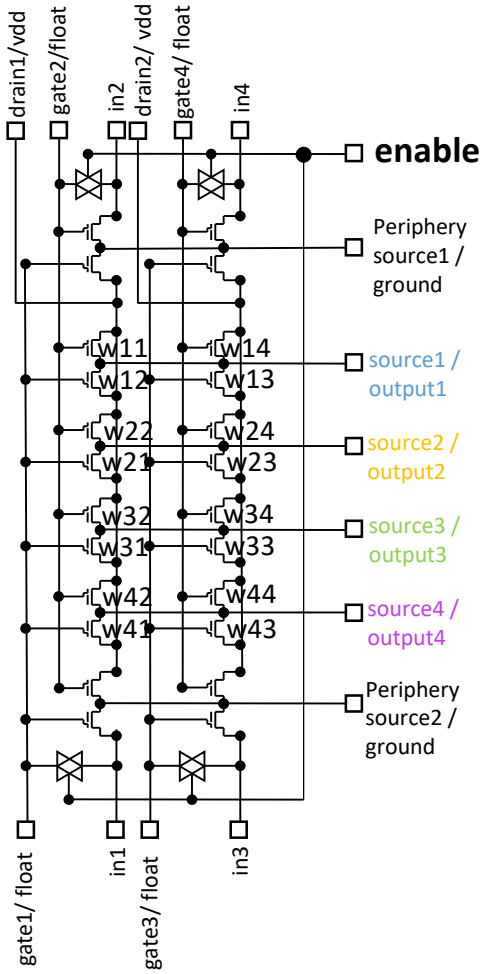


Fig. 25. Fabricated 4 × 4 analog vector-by-matrix multiplier based on 180-nm ESF1 NOR flash memory and high voltage pass-gate integrated on chip.

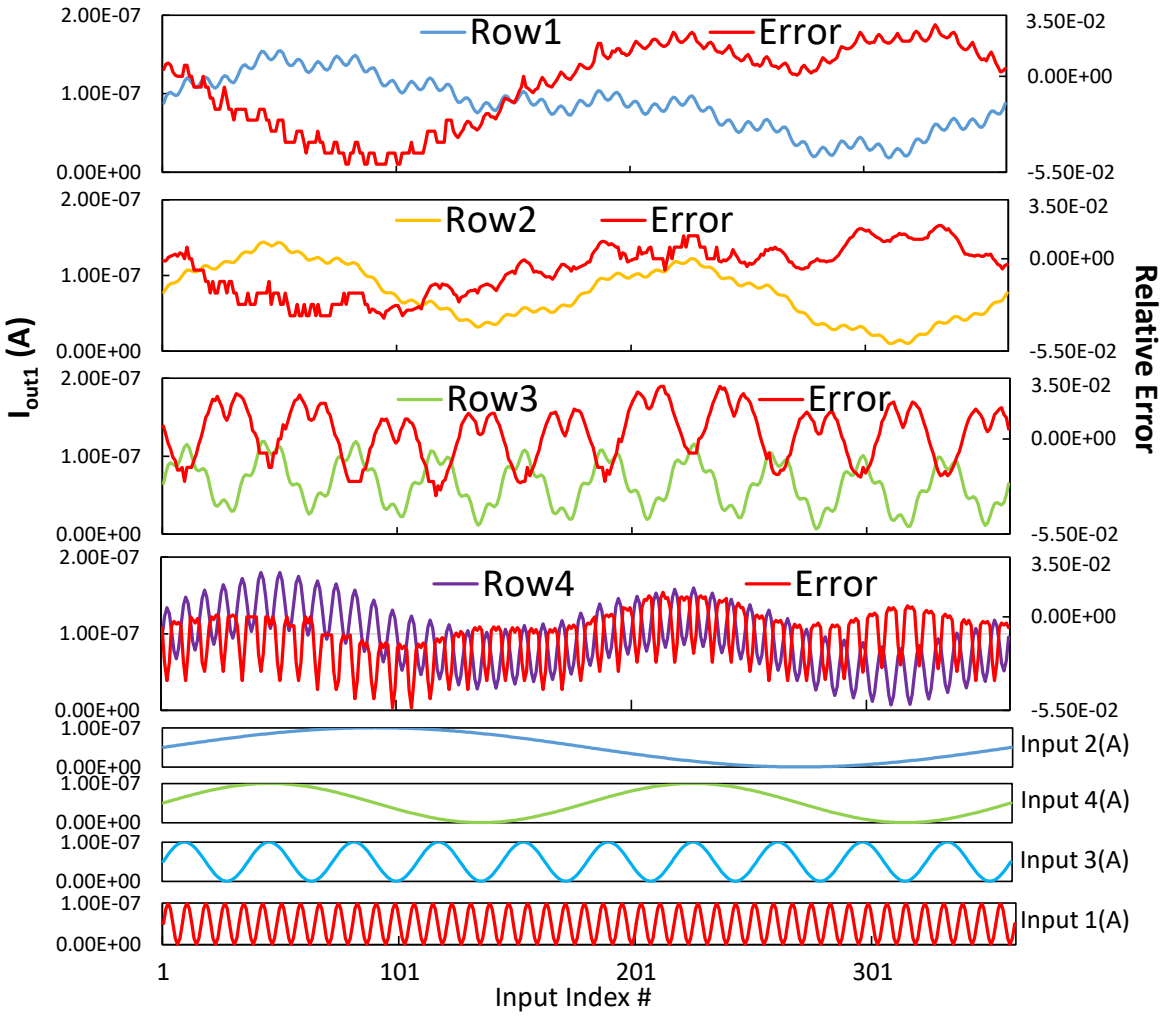


Fig. 26. Real outputs at a 4-input vector-by-vector multiplication, and their difference (red lines). The four inputs are quasi-DC currents sampled from sine function 50 nA × [1+ Sin(2π × Input Index# × f)], with f = 1/8, 1/36, 1/180, and 1/360.

B. Based on 55-nm ESF3 Floating-Gate Array

Similar to 180-nm ESF1 technology, to implement the vector-by-matrix multiplication, we have used the gate coupling of the tunable floating gate cells of each row of the array with a similar "peripheral" cell, with the virtual-bias condition imposed (by external circuitry) on the output (column) wires (Fig. 27).

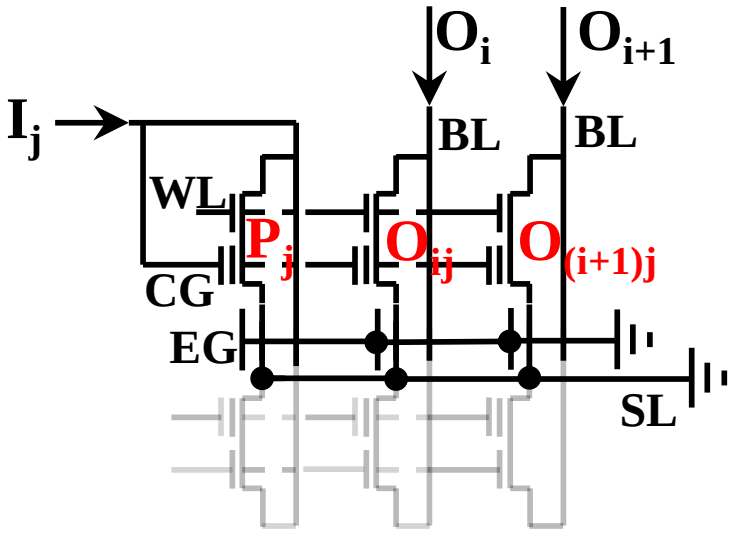


Fig. 27. The vector-by-matrix multiplication scheme based on gate coupling of the floating-gate cells. (For clarity, only one peripheral (P) and two array (O) cells of the same (j^{th}) row are shown.

Since all cells of the same row share the same coupling gate voltage V_j, in the subthreshold operation mode the j-th component O_{ij} of the output current O_i in the i-th column is proportional to the input current I_j in the j-th-row:

$$I_j = I_0 \exp\left\{q\frac{V_j - V_{\text{th}}^{(j)}}{nk_{\text{B}}T}\right\},$$
$$O_i = \sum_j O_{ij} = \sum_j I_0 \exp\left\{q\frac{V_j - V_{\text{th}}^{(i,j)}}{nk_{\text{B}}T}\right\} \equiv \sum_j w_{ij} I_j, \quad (5)$$

with current-independent proportionality coefficients w_{ij}, which are determined by the differences of threshold voltages V_{th} of the array cells and the peripheral transistors:

$$w_{ij} = \exp\left\{q\frac{V_{\text{th}}^{(j)} - V_{\text{th}}^{(i,j)}}{nk_{\text{B}}T}\right\} \quad (6)$$

In turn, each threshold voltage is determined by the analog state (physically, the floating gate charge) of the cell, so that each w_{ij} may be adjusted to the desirable value (typically, below 1).

As a simple illustration of multiplier's operation, Fig. 28 shows the results of multiplication of 4 input signals by 4 different weights: $w_1 = 0.25$, $w_2 = 1$, $w_3 = 0.5$, and $w_4 = 0.125$, performed by 4 cells of one column of the array, tuned with a 1% precision. This experiment demonstrates that the relative error, incorporating contributions from all sources (device noise, state retention, impedance mismatch, parameter variation, tuning precision, and capacitive crosstalk) does not exceed 2%.

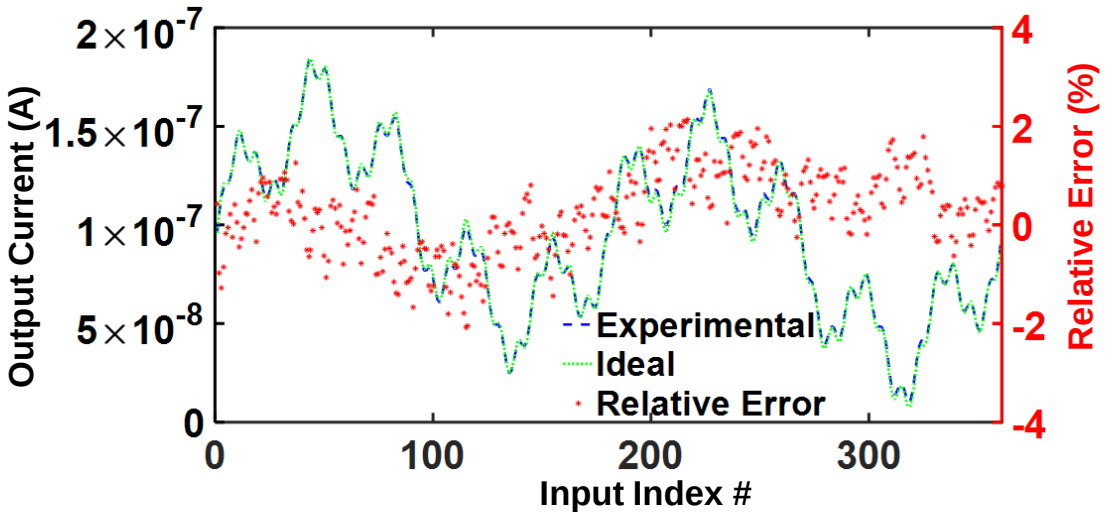


Fig. 28. Ideal (green line) and real (blue dashes) outputs at a 4-input vector-by-vector multiplication, and their difference (red dots). The four inputs are quasi-DC currents sampled from sine function 50 nA × [1+ Sin(2π × Input Index# × *f*)], with *f* = 1/8, 1/36, 1/180, and 1/360.

According to Eq. (6), in the coupled-gate operation mode, much of the thermal dependence of the subthreshold current is compensated, but besides the special case $w_{ij} = 1$, the compensation is incomplete. Indeed, our measurements have confirmed that in

agreement with this relation, that as temperature is raised from 25°C to 85°C, weight w_{ij}, initially equal to 0.9, increases by ~10%.

However, there is a straightforward way to decrease the temperature sensitivity, at the cost of a two-fold increase of hardware. For that, one can subtract output currents of two cells (say, those shown in Fig. 27), with their individual weights tuned to, respectively, (w_b + w/2) and (w_b - w/2). Here w is the desired net weight, and w_b is the "bias weight", which may be optimized to suppress the temperature dependence of the new output current. A straightforward analysis of this scheme, using Eq. (6), shows that after such optimization, the temperature drift of the output may be reduced to less than 1% at the [25°C, 85°C] interval, for any weight $0 < w_{ij} < 1$. Fig. 29 shows the results of our preliminary experiments with this mode, showing the drifts not exceeding 2.7% in that temperature interval.

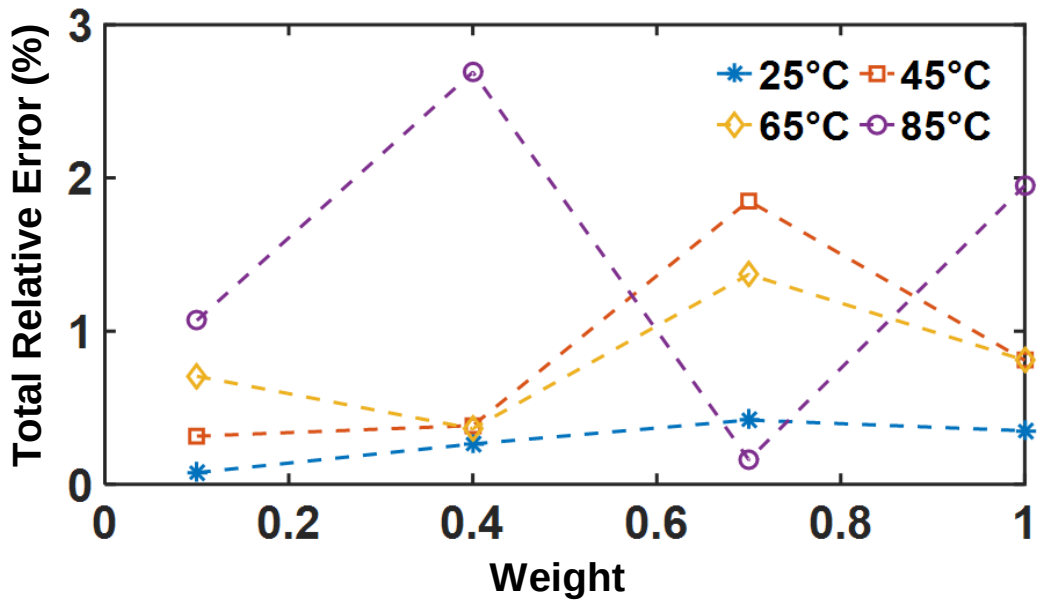


Fig. 29. The total relative error of the reproduction of a 100 nA input signal for several values of *w*, at various temperatures.

V. Mixed-Signal Neurocomputing Systems

A. Fabricated Pattern Classifier based on NOR Flash Array

Here we report a prototype 28×28-binary-input, 10-output, 3-layer neuromorphic network based on arrays of highly optimized embedded nonvolatile floating-gate cells, redesigned from a commercial 180-nm NOR flash memory. The implemented network could perform a high-fidelity classification of patterns of the standard MNIST benchmark with record-breaking speed and energy efficiency.

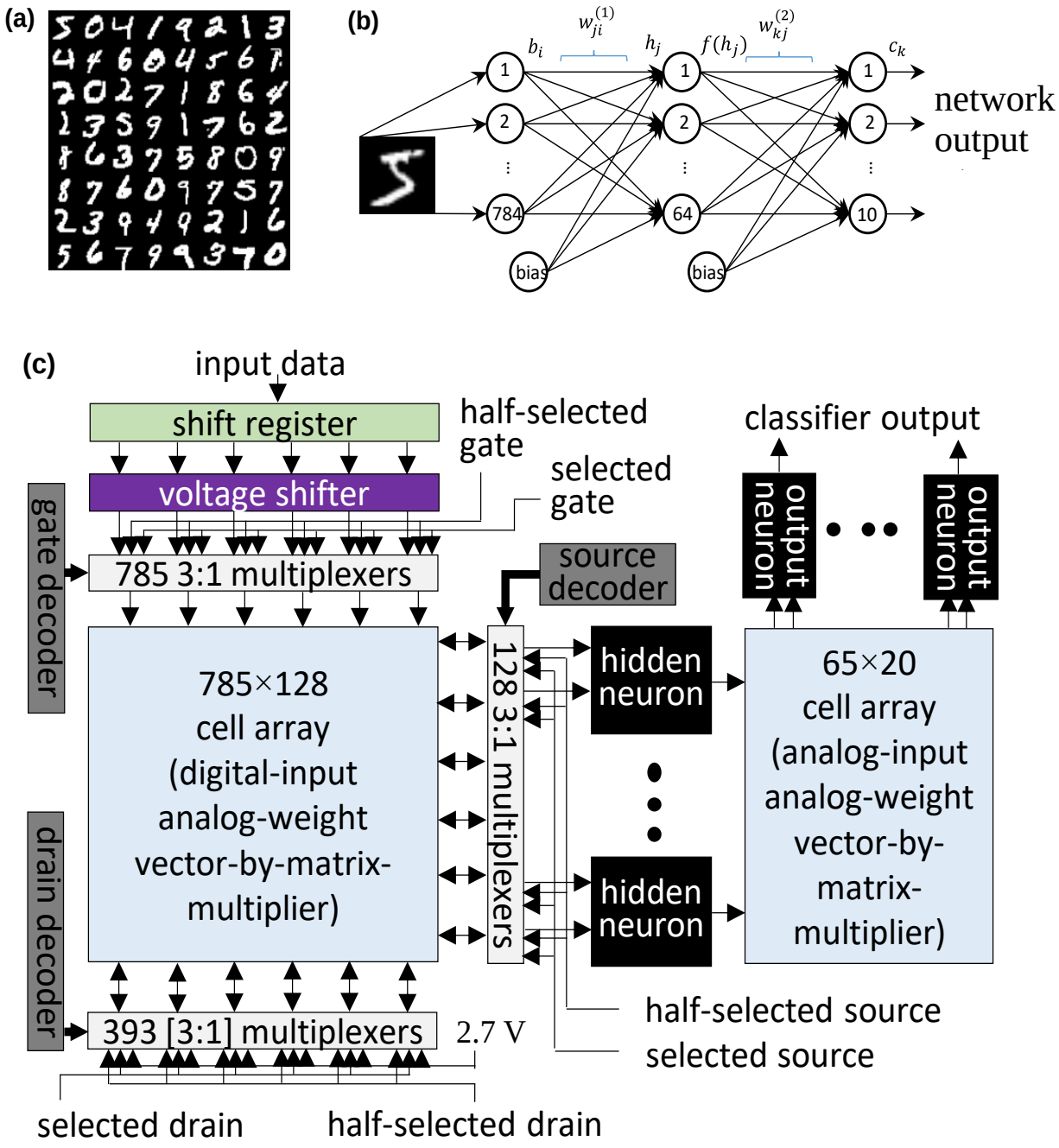
1. Network Deisgn

The implemented neuromorphic network (Fig. 30) is a 3-layer (one-hidden-layer) perceptron with 784 binary inputs b_i, which may represent, for example, 28×28 black-and-white pixels of an input image (such as the MNIST dataset images shown in Fig. 30a), 64 hidden layer neurons with the rectified-tanh activation function, and 10 output neurons (Fig. 30b). The goal of the network is to perform the pattern inference by the following sequential transformation of the input signals:

$$h_j = \sum_{i=1}^{784} w_{ji}^{(1)} b_i + w_{j,785}^{(1)}, \quad c_k = \sum_{j=1}^{64} w_{kj}^{(2)} f\left(h_j\right) + w_{k,65}^{(2)} f_{\max}, \tag{7}$$

$$f\left(h\right) \equiv f_{\max} \times \begin{cases} \tanh\left(h\right), & \text{for } h \geq 0, \\ 0, & \text{for } h < 0. \end{cases}$$

Here h_j and f_j (with j = 1, 2, ⋯, 64) are, respectively, the input and output signals of the hidden-layer neurons, c_k (with k = 1, 2, ⋯, 10) are the output signals, providing the class of the input pattern, while $w^{(1)}$ and $w^{(2)}$ are two matrices of tunable synaptic weights,


(a)
(b)
b_i
$w_{ji}^{(1)}$
h_j
$f(h_j)$
$w_{kj}^{(2)}$
c_k
1
2
784
bias
64
10
network output
(c)
input data
shift register
half-selected gate
selected gate
voltage shifter
gate decoder
785 3:1 multiplexers
source decoder
785×128 cell array (digital-input analog-weight vector-by-matrix-multiplier)
128 3:1 multiplexers
hidden neuron
65×20 cell array (analog-input analog-weight vector-by-matrix-multiplier)
classifier output
output neuron
drain decoder
393 [3:1] multiplexers
2.7 V
half-selected source
selected source
selected drain
half-selected drain

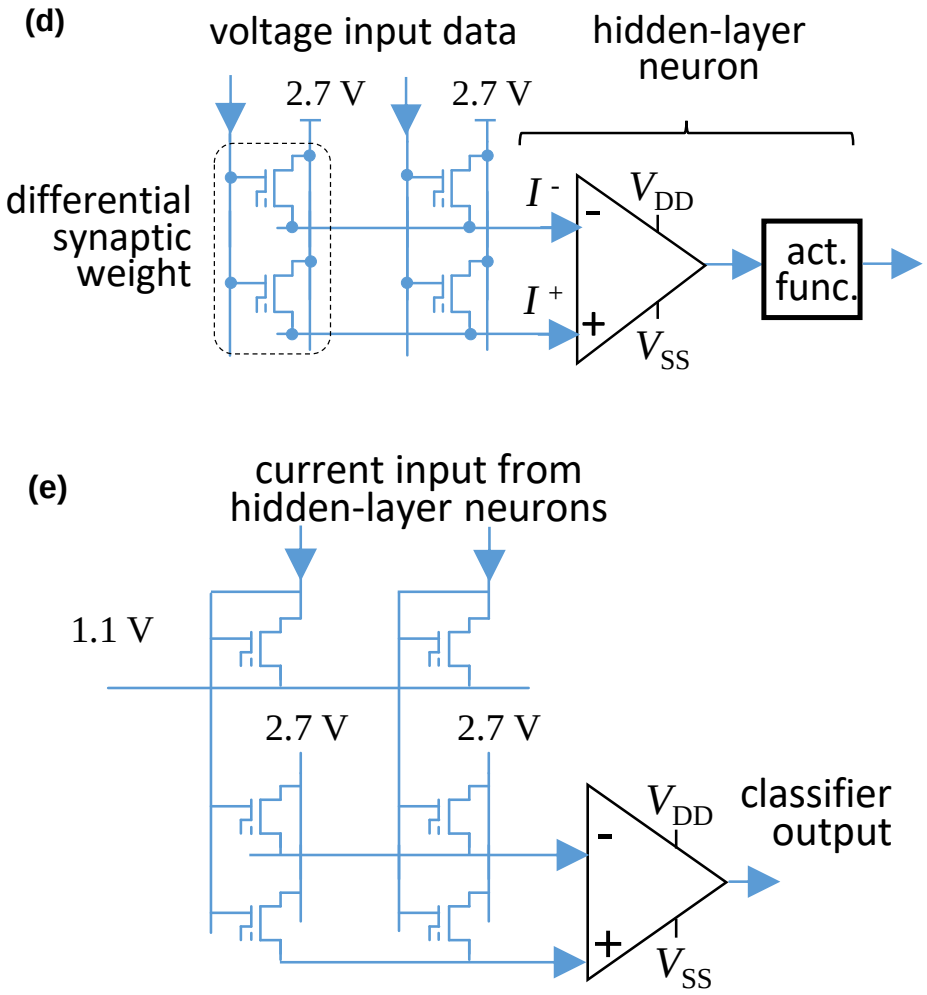


Fig. 30. Network architecture: (a) Typical examples of B/W hand-written digits of the MNIST benchmark set. (b) Graph representation of our 3-layer perceptron network. Each synapse is implemented using a differential pair of floating-gate memory cells. (c) High-level architecture, with the weight tuning circuitry for the second array (like that of the first one) not shown for clarity. (d) A 2×2-cell fragment of the first crossbar array shown together with a hidden-layer neuron, consisting of a differential summing operational amplifier pair and an activation-function circuit. (e) A 2×2-cell fragment of the second crossbar array with an output-layer neuron; these neurons do not implement an activation function. The voltage shifter, shown on panel (c), enables using voltage inputs of both polarities over a 1.65V bias, and is also used to initiate the classification process by increasing the input background from 1.8 V to 4.2 V.

characterizing the coupling of the adjacent network layers. In our network, these weights are provided by floating-gate cells of two crossbar arrays of the floating-gate memory cells providing tunable weights (Fig. 30c). Each neuron also gets an additional input from a bias

node, with a tunable weight based on a similar cell (Fig. 30b). With the differential-pair implementation of each synapse (see below), the total number of utilized floating-gate memory cells is $2\times[(28\times28+1)\times64 + (64+1)\times10] = 101{,}780$.

The mixed-signal vector-by-matrix multiplication in the first crossbar array is implemented by applying input voltages (4.2 V for black pixels or 0 V for white ones) directly to the gates of the array cell transistors, with fixed voltages on their sources (1.65 V) and drains (2.7 V) – see Fig. 30d. As a result, the transistor source-to-drain current of the cell located at the crosspoint of the i^{th} column and the j^{th} row of the array does not depend on the state of any other cells, and is equal to the product of the binary input voltage b_i by the analog weight $w_{ji}^{(1)}$ pre-recorded in the memory cell. The sources of the transistors of each row are connected to a single wire (with an externally-fixed voltage on it), so that the j^{th} output current of the array is just the sum of products $w_{ji}^{(1)}b_i$ over all columns i, thus implementing the vector-by-matrix multiplication described by the first of Eqs. (7).

Actually, in order to reduce the random drifts, and also to work with zero-centered signals h_j, we are using a differential scheme, in which each synaptic weight is recorded in two adjacent cells of each column, and the output currents (in Fig. 30d, I_j^+ and I_j^-) of two adjacent cell rows are subtracted in an operational amplifier, with its output, $h_j \propto I_j^+ - I_j^-$, passed to the activation function circuit performing the function $f(h)$. The accepted sharing of the weight $w_{ji}^{(1)}$ between the two cells of the differential pair is very simple: one of the cells (depending of the sign of the desirable weight) is completely turned off, giving virtually no contribution to the output current. This arrangement keeps half of the cells virtually idle, but simplifies the design and speeds up the weight tuning process.

The analog vector-by-matrix calculation in the second array is performed using the gate-coupled approach (Fig. 30e). In this approach [66], the synaptic gate array is complemented

by the additional row of "peripheral" cells, which are physically similar to the array cells, and hence sharing the same subthreshold slope β. The gate electrode of the peripheral cell of each column is connected to those of all cells of this column, so that their voltages V_{GS} are also equal. Applying Eq. (1) to the current of the cell located at the crosspoint of the k^{th} row and the j^{th} column of the array (I_{kj}), and that of the peripheral cell of this column (I_j), and dividing the results, we get

$$w_{kj}^{(2)} \equiv \frac{I_{kj}}{I_j} = \exp\left\{\beta \frac{(V_t)_j - (V_t)_{kj}}{V_T}\right\} \tag{8}$$

The resulting currents I_{kj} are summed up exactly as those in the first array (actually, with the similar differential scheme for drift reduction), so that if the array is fed by the output currents of the activation function circuits, $I_j \propto f(h_j)$, it performs the second vector-by-matrix multiplication described by Eq. (7), with the synaptic weights given by Eq. (8), which depend on the preset memory states of the corresponding cells, but are independent of the input currents. To minimize the error due to the dependence of β on the memory state (see the inset in Fig. 4), in the second array we used a higher gate voltage range (1.1 V to 2.7 V), with the upper bound due to the technology restrictions.

Fig. 31a shows the circuit used to subtract the currents I^+ and I^- of the differential-scheme rows, based on two operational amplifiers (Fig. 31c). Assuming that the resistances R_F are equal, that the outputs of both opamps do not saturate (which is ensured by the following relation between of the chosen value R_F = 16 KΩ in the first layer and R_F = 128 KΩ in the second one, and the maximum value of currents $I^{\pm}$: $I_{max}R_F$ < 1 V) the output voltage of the scheme is

$$V = R_{\mathrm{F}}\left(I^{+} - I^{-}\right) + const \tag{9}$$

Fig. 31b shows the rectified-tanh activation function $f(h)$ used in the hidden-layer neurons (see Eq. (5) of the main text), with h [V] = $10R_{\mathrm{F}}$ [Ω] (I^{+} - I^{-}) [A] and $f_{\max}$ = 300 nA, while Fig. 31d shows the circuit used for the implementation of this function.

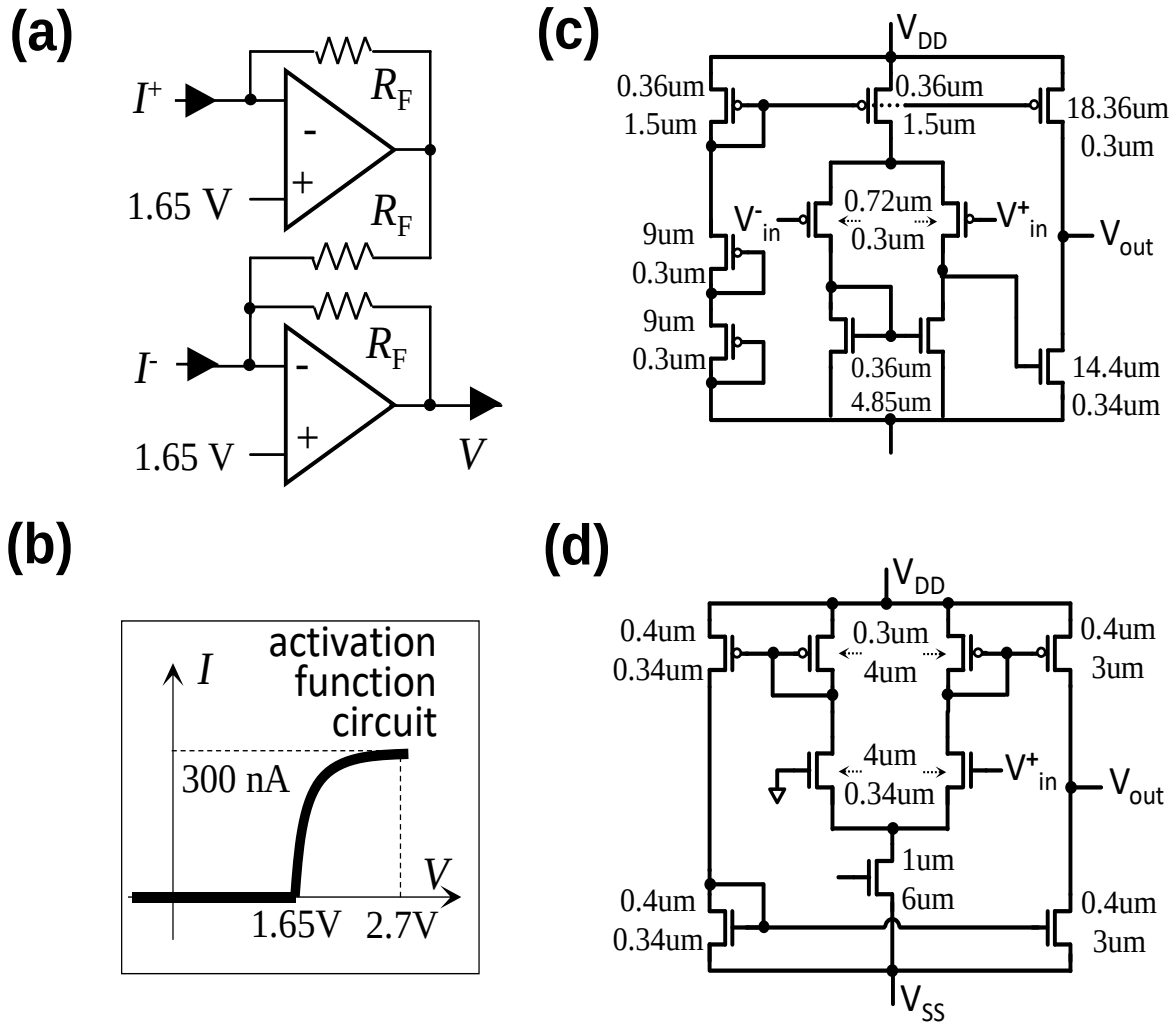


Fig. 31. (a) Circuit-level diagram of a differential summing amplifier used in the hidden-layer and output-layer neurons; RF_1 = 16 KΩ for hidden neurons, and RF_2 = 128 KΩ for output neurons. (b) Implemented activation function. (c, d) Transistor-level schematics of: (c) the operational amplifier and (d) the activation function; V_{SS} = 0 V, V_{DD} = 2.7 V.

The desirable synaptic weights, calculated in an external computer implementing a similar "precursor" network, using the standard error backpropagation algorithm, were imported into the network by analog tuning of the memory state of each floating-gate cell using peripheral analog demultiplexer circuitry (Fig. 30c). In order to simplify this first, prototype design, the weights were tuned one-by-one, by applying proper bias voltage sequences to selected and half-selected lines [60]. (In principle, this process may be significantly parallelized.) The large voltages required for the weight import are decoupled from the basic, low-voltage circuitry, using high-voltage pass transistors. The input pattern bits are shifted serially into a 785-bit register before each classification; to start it, the bits are read out into the network in parallel.

The digital encoders and shift register circuits and their layouts were synthesized from Verilog in a standard 1.8 V digital CMOS process. All other circuits were designed manually for the embedded 180-nm process of SST Inc.. (Such a design was practicable due to the modular, repetitive design of the circuit.) All active components of the circuit have a total area of 0.78 mm^2 (Fig. 32), with the two synaptic arrays occupying less than a quarter of this area, while the total chip area, including very sparse routing (which was not yet optimized for this design), is about 5×5 mm^2.

2. Network Testing

Because of the digital (fixed-voltage) input of the first synaptic array, the subthreshold conduction was not enforced there, so that the output currents of some cells exceeded 300 nA (Figs. 33a, b). To reduce the computation error due to the potential slope mismatch between peripheral and array cells, all peripheral floating gate transistors in the second array were tuned to provide output currents of 300 nA at V_G = 2.7 V, i.e. at the largest voltage that

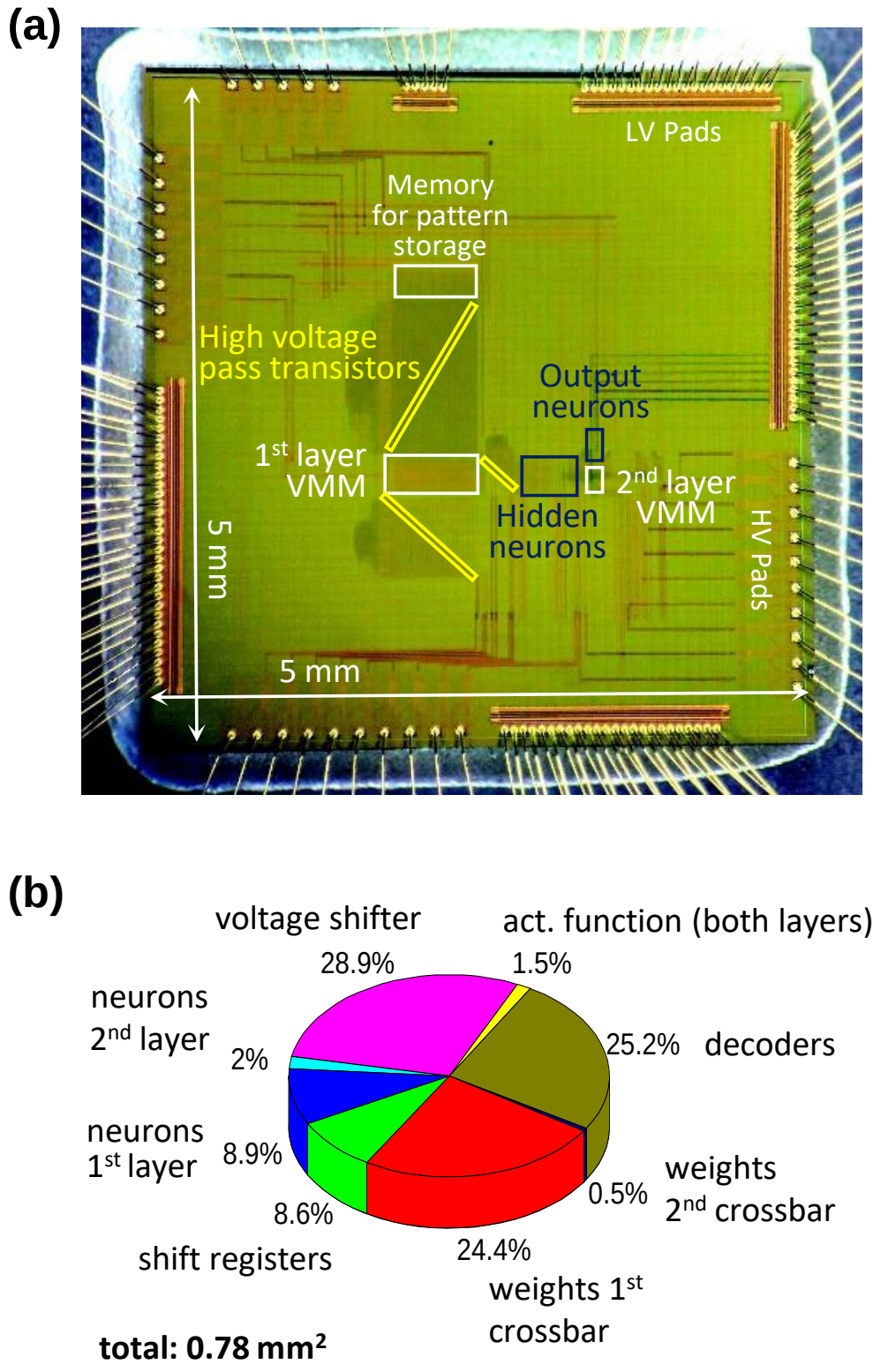


Fig. 32. (a) A micrograph of the chip, and (b) an area breakdown of its active components (excluding wiring between the blocks, which was not optimized at this stage).

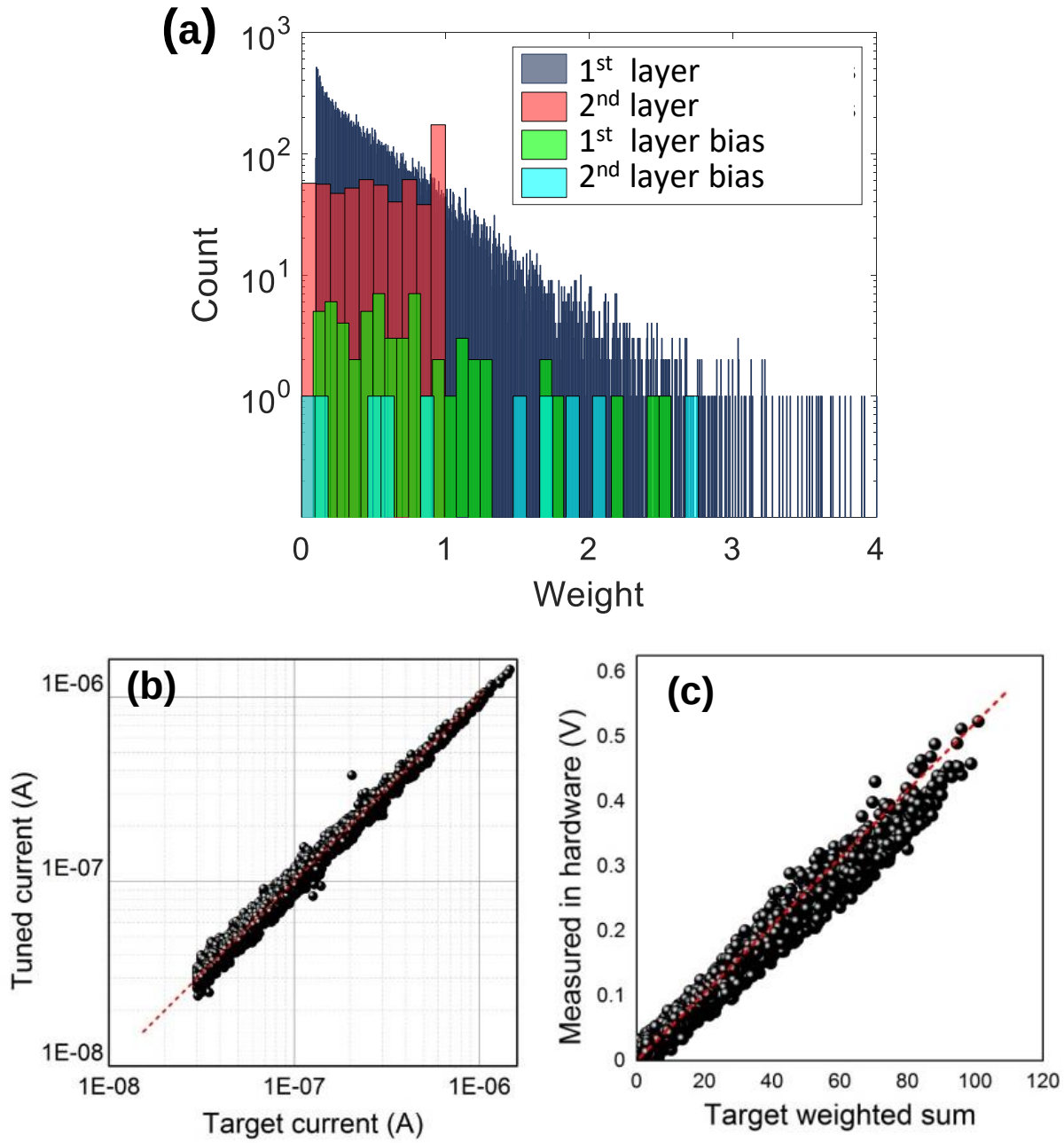
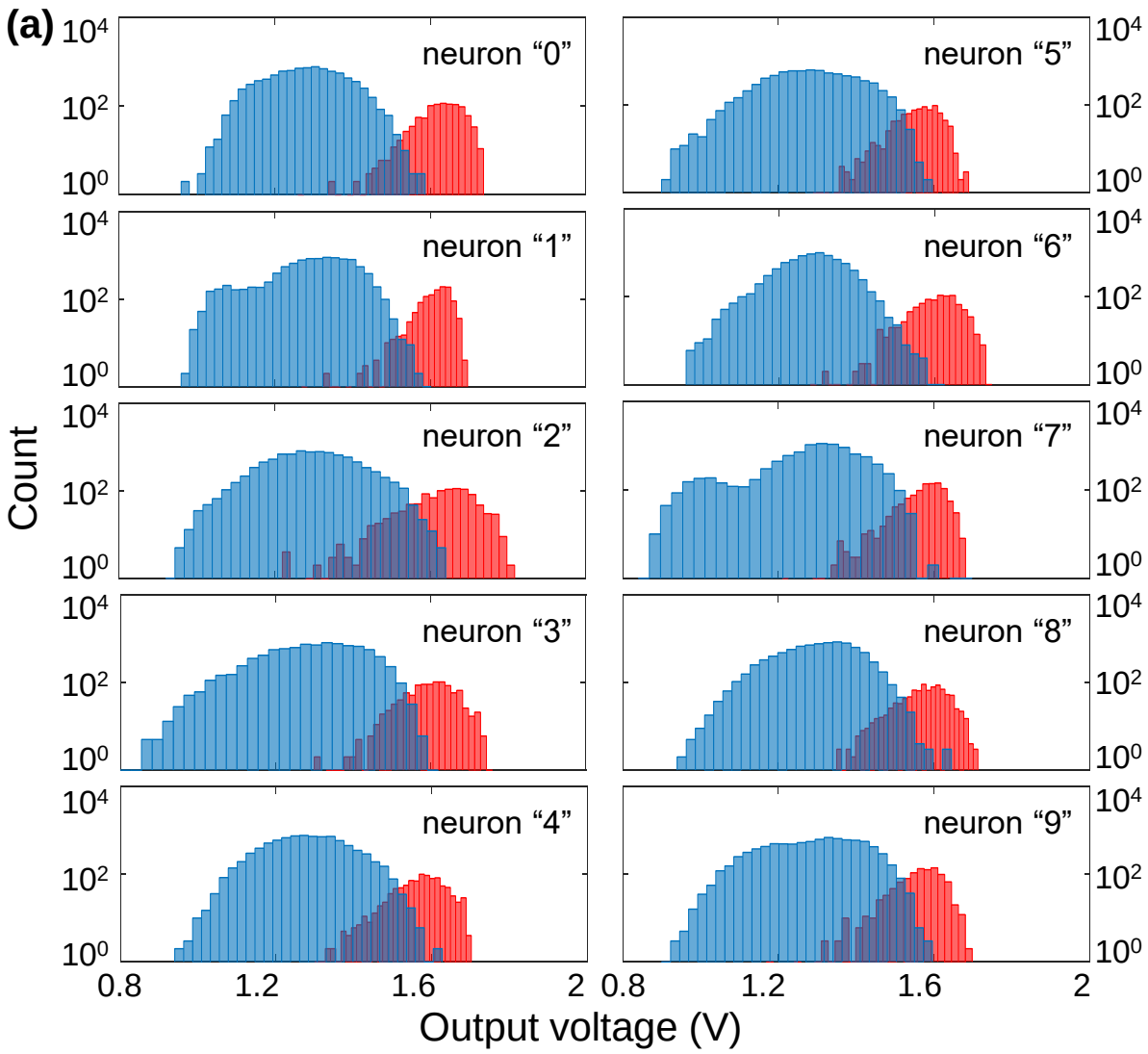


Fig. 33. Weight export statistics: (a) A histogram showing the imported cell current values (weights), measured at V_D = 2.7 V, and V_S = 1.65 V and V_G = 4.2 V in the first synaptic array, and V_S = 1.1 V and V_G = 2.7 V for the second one, which were used in the experiment. (b) Comparison between the target synaptic cell currents (computed at the external network training) and the actual cell currents measured after their import, i.e. cell tuning. (c) The similar comparison for the positive fraction of hidden neuron output, computed for all test patterns. (The negative outputs are not shown, because they are discarded by the used activation function.) Red dashed lines are guides for the eye, corresponding to the perfect weight import.

can be supplied by the hidden layer neuron in our design. With such scheme, the error is conveniently smallest for the largest weight $w_{ki} = 1$, corresponding to the array cell tuned to run a current 300 nA at $V_{GS} = 1.6$ V. The target current values for all cells in the second array (excluding bias ones) were ensured to be between 0 and 300 nA by clipping the weights during training of the precursor network.

To decrease the weight import time, only one cell of each pair, corresponding to a particular sign of the weight value, was tuned, while its counterpart was kept at a very small, virtually zero, and initial conductance. Additionally, all non-bias cells in the first array, for which the target conductances were below 30 nA, were also not tuned, because of their negligible impact on the classification fidelity, confirmed by modeling. As a result, only about ~30% of the cells were actually fine-tuned. Because of the sequential character of the tuning process, it took several hours to complete it, with the chosen accuracy, for the whole chip. (In future, the tuning may be greatly sped up by adjusting multiple weights at a time via integrated on-chip tuning circuitry [58], and using better tuning algorithms we have developed [61].)

Moreover, also to speed up the import process, the weigh import accuracy for a single cell tuning was set to relatively high value of 5%. As Fig. 33b indicates, some of the already tuned cells were disturbed beyond the target accuracy during the subsequent weight import. In this first experiment, these cells were not re-tuned, in part because even for such rather crude weight import the experimentally tested classification fidelity (94.65%) on MNIST benchmark test patterns (Fig. 34) is already remarkably close to the simulated value (96.2%) for the same network (Fig. 35). Both these numbers are also not too far from the maximum fidelity (97.7%) of the similar perceptron of this size, optimized without hardware constrains.


(a)
neuron “0”
neuron “1”
neuron “2”
neuron “3”
neuron “4”
neuron “5”
neuron “6”
neuron “7”
neuron “8”
neuron “9”
Count
Output voltage (V)
0.8
1.2
1.6
2

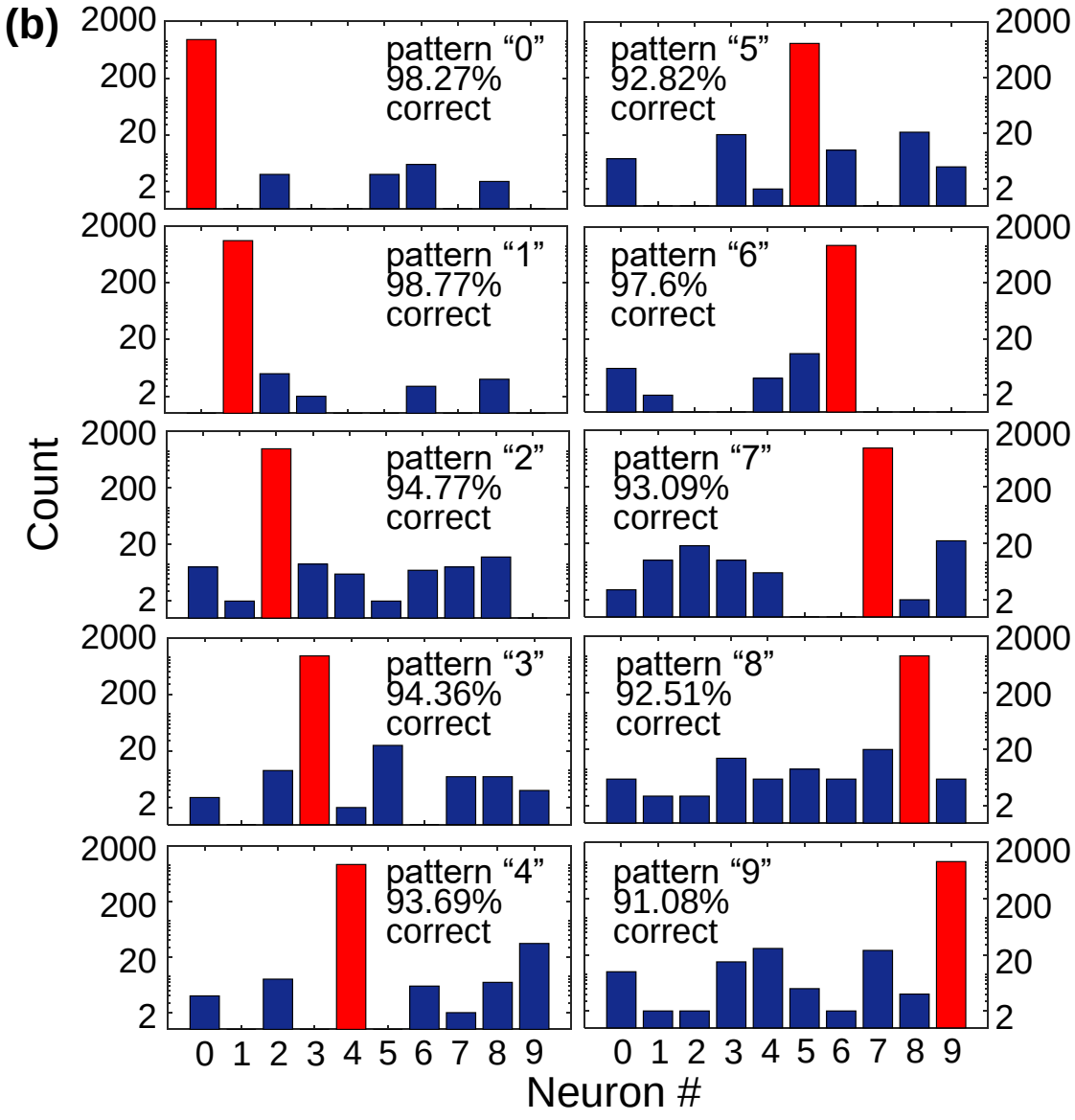


Fig. 34. Experimental results for the classification of all 10,000 MNIST test set patterns: (a) Histograms of voltages measured on each output neuron. Red bars correspond to the patterns whose class belongs to this particular output, while the blue ones are for all remaining patterns. (b) Histograms of the largest output voltages (among all output neurons) for all test patterns of each class, showing that the correct outputs (red bars) always dominate. Note the logarithmic vertical scales.

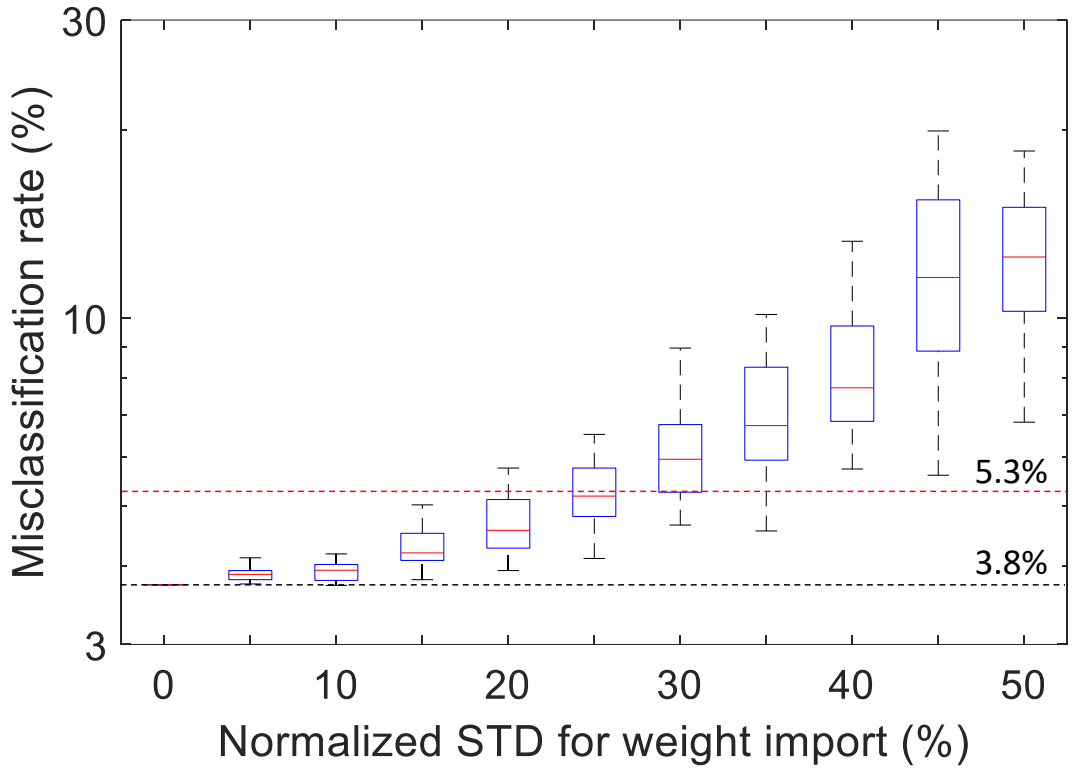


Fig. 35. The simulated classification fidelity as a function of weight precision import for the implemented network, with the particular set of weights used in the experiment. The weight error was modeled by adding, to its optimized value, a normally distributed noise with the shown standard deviation. The red, blue (rectangles), and black (segment) markers denote, respectively, the median, the 25%-75% percentile, and the minimum and maximum values for 30 simulation runs. The black and red horizontal dashed lines show, respectively, the calculated misclassification rate for perfect (no noise) weights, and the rate obtained in the experiment.

Excitingly, such classification fidelity in network, with large optimization reserves (see below), is achieved at an ultralow (sub-20-nJ) energy consumption per average classified pattern (Fig. 36a), and the average classification time below 1 μs (Fig. 36b). The upper bound of the energy is calculated as a product of the measured average power, 5.6 mA × 2.7 V + 2.9 mA × 1.05 V ≈ 20 mW, consumed by the network, by the upper bound, 1 μs, of the average signal propagation delay. A more accurate measurement of the time delay, and hence the energy, requires a redesign of the signal input circuitry, currently rather slow – see Fig. 36b.

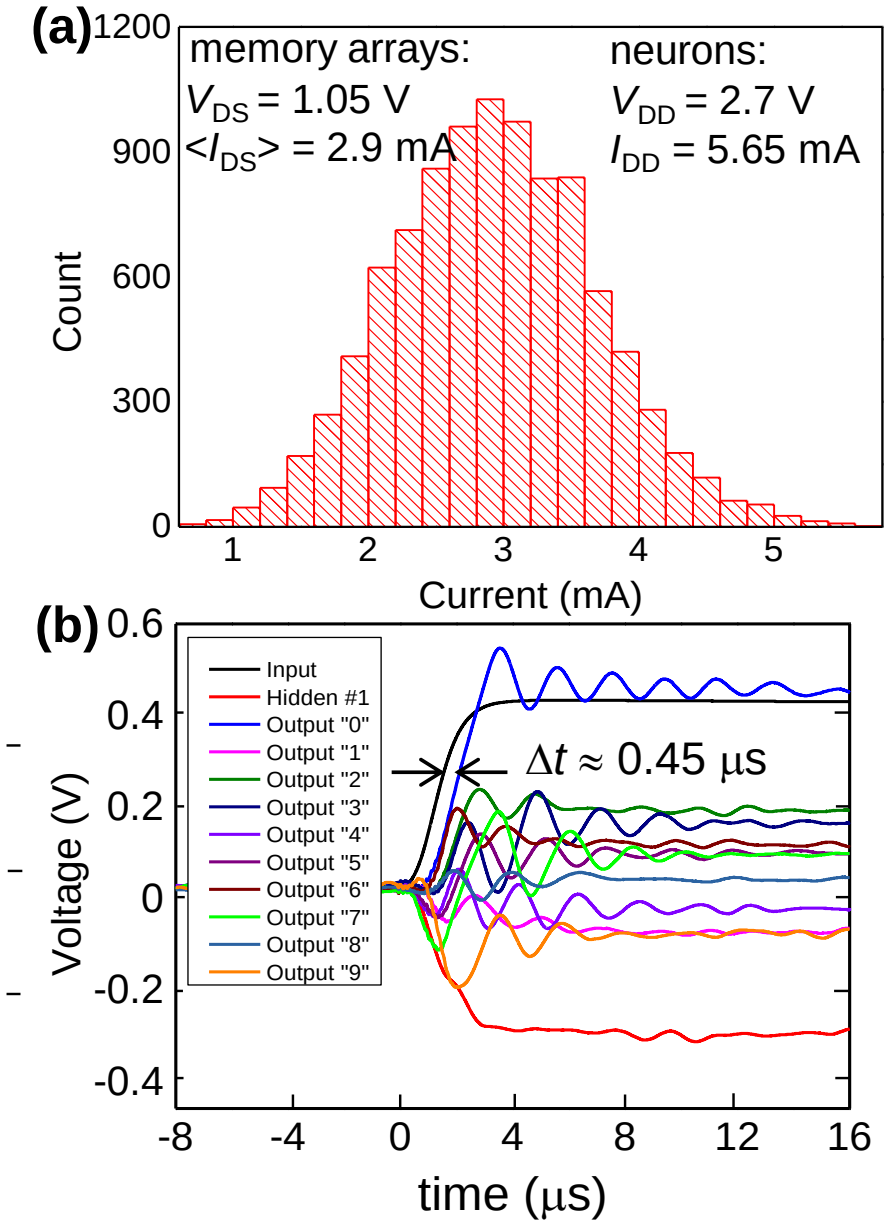


Fig. 36. Physical performance: (a) Histogram of the total currents flowing into the circuit, characterizing the static power consumption of the both memory cell arrays, for all patterns of the MNIST test set. The inset lists the pattern-independent static current of the neurons. (b) The typical output signal dynamics after an abrupt turn-on of the voltage shifter power supply, measured simultaneously at the network input, at the output of a sample hidden-layer neuron, and at all network's outputs. (The actual input voltage is 10× larger.) The oscillatory behavior of the outputs is a result of a suboptimal phase stability design of the operational amplifiers. Before it has been improved, and the input circuit is sped up, we can only claim a sub-1-μs average time delay of the network, though it is probably closer to 0.5 μs.

3. Network Evaluation

The achieved speed and energy efficiency are much better than those demonstrated, for the same task, at any digital network we are aware of. For example, the best results for the same MNIST benchmark classification were reported for IBM's TrueNorth chip [67]. For the comparable 95% fidelity, that chip can classify 1,000 images per second while consuming 4 μJ energy per image [74], i.e. is at least three orders of magnitude slower and less energy efficient than our, still unoptimized analog circuit. This difference is rather impressive, taking into account the advanced 28-nm CMOS process used for the TrueNorth chip implementation.

In a less direct comparison, in terms of energy per a MAC operation, our network also outperforms the best reported digital systems. Indeed, the measured upper bound of the energy efficiency of our circuit is 0.2 pJ per MAC. This is a factor of 60 smaller than the 12 pJ per MAC reported for 65-nm Eyeriss chip [14], which is highly optimized for machine learning applications. (It performs 16-bit operations and, like the TrueNorth chip, was implemented using an advanced fabrication technology.) Note that both the TrueNorth and Eyeriss chip, in turn, far outperform the modern graphics processing units (GPUs) for neuromorphic-network applications. Our result is also much better than the ~1 pJ per analog operation, recently reported for a small 130-nm mixed-signal neural networks based on synaptic transistors [15] and is comparable to the best results obtained with the switched-capacitor approach [45], for example the recent ~0.1 pJ per operation achieved in a much smaller circuit, with only 8×8×3 discrete (3-bit) synaptic weights, using a 40-nm process [1]. (Note that this approach does not allow fine-tuning of the synapses, and its extension to larger circuits may be problematic because of the relatively large capacitor size.)

It should be also noted that the energy-per-MAC metric is generally less objective, because it does not account for the operation precision and the complexity and functionality of the implemented system (e.g., general-purpose systems like a typical GPU vs application-specific ones like the Eyeriss chip).

B. Exponential-Weight Multilayer Perceptron with NOR Flash Array

The name "perceptron" today typically implies a well-known algorithm for supervised training of linear classifiers. Interestingly, a more general version of perceptron algorithm was proposed in early 1960s [75]. In such algorithm, all sensory (input), association (hidden) and response (output) neurons are of the form $f(\alpha_i)$, where α_i is the algebraic sum of all input signals while synaptic/transmission function is a function of input and current synaptic state of the synapse, i.e. $g(x_i, w_i)$, so that the output of the neuron is written as:

$$y = f\left[\sum_i g(x_i, w_i)\right] \tag{10}$$

For linear synaptic transmission function, i.e. the most commonly used case today, this equation simplifies to a typical weighted-sum (dot-product) operation:

$$y = f\left[\sum_i x_i w_i\right] \tag{11}$$

with the corresponding well-known single-layer perceptron network.

In the early years of artificial neural networks, the motivation for more general algorithm was rather obvious considering the inherent limitations of linear perceptron, which can perform classification for only linearly-separable patterns [75]. It seems though that this idea has not received much attention, which is likely due to subsequent invention of more powerful multilayer networks in which the needed nonlinearity is provided by neurons.

Additionally, simple product operation is certainly easier, as compared to nonlinear transmission function, to compute using digital circuits, which has been traditionally used to implement artificial neural network implementations.

However, the recent advances in emerging nanoelectronic memory devices [76] and the opportunity to use such devices in low-energy analog circuit implementations of artificial neural networks make the idea of nonlinear synaptic transmission function very appealing again. In our architecture, we implement the exponential synaptic weights through floating-gate transistors which are biased in sub-threshold voltage and show that the whole network maps very efficiently to the modified NOR flash memory originally designed for digital memory applications. The proposed algorithm and architecture are experimentally verified by implementing and training a small-scale classifier on a 10 × 10 flash memory array fabricated in 180-nm process.

According to the exponential relationship of floating gate devices in subthreshold region—Eq. (1), we have $g(\cdot,\cdot)$ as $g(x,w)=e^{\alpha(x-w)}$. In order to have bipolar weights, we have to implement each weight as a difference of two non-negative weights. In that way, we have:

$$y=f\left[\sum_i g(x_i,w_i^+)-\sum_i g(x_i,w_i^-)\right],\qquad \forall w_i^+,w_i^-\geq 0 \tag{12}$$

By defining f (A - B) = $ln(A)$ - $ln(B)$, we have our network with exponential weight as shown in Fig. 37.

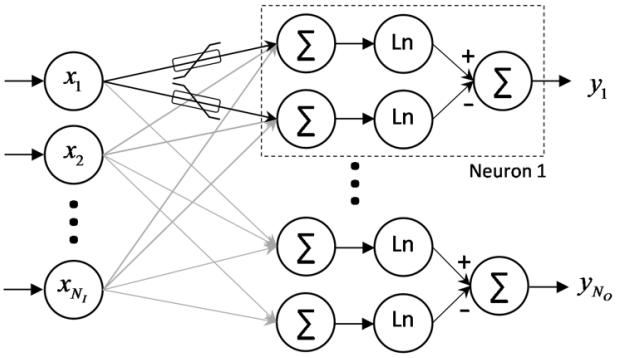


Fig. 37. The proposed perceptron implementation with nonlinear (exponential) synaptic weights. In this architecture, the natural logarithmic function represents activation function.

Practicality and functionality of the proposed neural network was tested by fabricating a small scale (10 x10) flash memory array of Figs. 38(a and b) in 180-nm process using SST's SuperFlash technology [60]. The fabricated array was then used to implement a single layer neural network with the top-level (functional) scheme shown in Fig. 38(a) while neuron circuits were emulated in software. The network had ten inputs and three outputs, fully connected with 10 x 3 x 2 = 60 differential weights each implemented with a single floating-gate transistor. Such a network is sufficient for performing, for example, the classification of 3 x 3-pixel images into three classes with nine network inputs corresponding to the pixel values. We tested the network on a set of 30 patterns, including three stylized letters ('z', 'v' and 'n') and three sets of nine noisy versions of each letter, formed by flipping one of the pixels of the original image (see the inset of Fig. 39). Physically, each input signal was represented by a voltage equal to either 0.5 V or 0 V, corresponding, respectively, to the black or white pixel, while bias input was equal to 0.5 V.

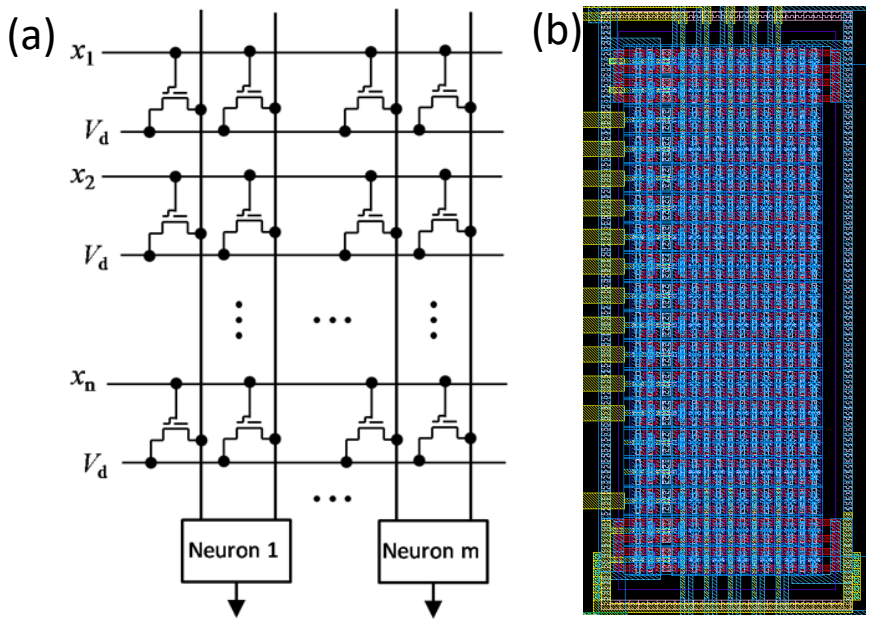


Fig. 38. (a) Implementing Eq. (10) using an array of floating-gate transistors. (b) Layout of the fabricated 10 x 10 memory array (obtained by modifying the commercially available NOR flash memory).

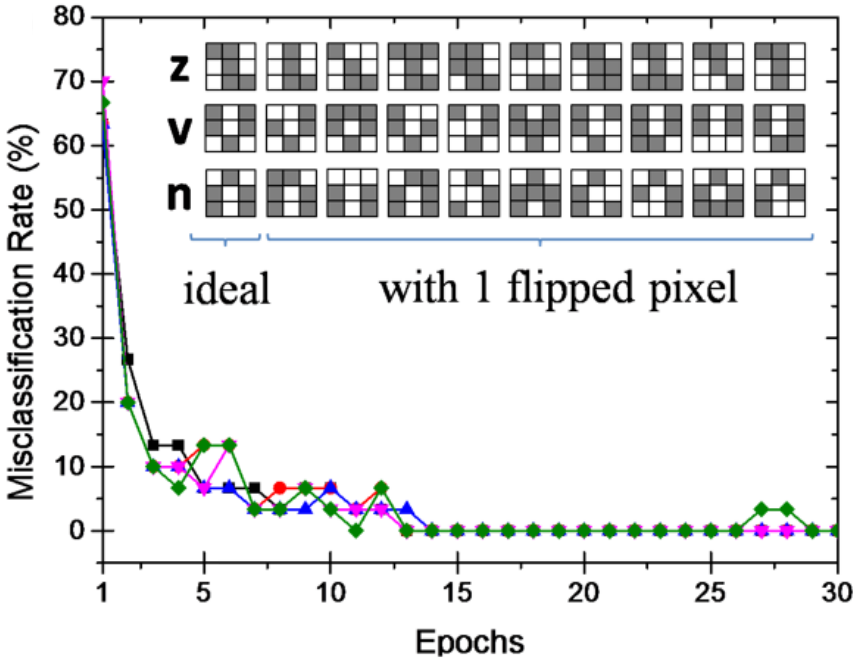


Fig. 39. Results of experimentally testing the proposed architecture with exponential weights.

The network was trained in situ, that is, without using its external computer model, using the Manhattan update rule, which is essentially a coarse-grain, hardware-friendly, batch-

mode variation of the usual delta rule of supervised training. After initializing all flash transistors to near fully-erased state, at each iteration ('epoch') of this procedure, patterns from the training set were applied, one by one, to the network's inputs while V_d = 0.6 V and V_s = 0V, and then its outputs were measured according to Eq. (12) (note that $\sum_i e^{(\alpha(x_i - w_i))}$ corresponds to the flowing current on each source line and has been measured experimentally by virtually grounding source lines). Once all N patterns of the training set had been applied and all gradients are calculated, the synaptic weights were updated in parallel based the Manhattan update rule (sign of the gradients) using fixed-amplitude erasure and programming pulses. For updating flash transistors in parallel, in each epoch we used the modified version of the algorithm originally proposed for parallel updating of memristive devices in crossbar structure [46]. The weight stored in each flash transistor was decreased by applying a single programming pulse (6V, 5 us) to the source line while synapse potentiation is done by applying an erasure pulse (8 V, 2.5 ms) to the control gate (see Ref. [60] for more details). For this particular classification task, the perfect classification for five runs was reached, on average, after 13 training epochs (see Fig. 39). For the fully trained network, the averaged read current of each floating-gate transistor was 0.34 nA at specific read condition of V_d = 0.6 V, V_g = 0:5 V and V_s = 0 V which corresponds to the power consumption of 200 pW/synaptic weight (when the input is at its maximum). Note that in this architecture power consumption of the whole network can be decreased even further by putting flash transistors in deeper subthreshold (by lowering Vg) with the cost of having smaller signal-to-noise ratio.

C. Hopfield Network with Hybrid CMOS/Memristor Circuits

Recurrent artificial neural networks are an important computational paradigm capable of solving a number of optimization problems [77, 78]. One classic example of such networks is a Hopfield analog-to-digital converter [78 - 80]. Although such a circuit may be of little practical use, and inferior, for example, to similar-style feed forward-type ADC implementations [81], it belongs to a broader constrained optimization class of networks which minimize certain pre-programmed energy functions and have several applications in control and signal processing [78]. The Hopfield network ADC circuit also represents an important bridge between computational neuroscience and circuit design, and an understanding of the potential shortcomings of such a relatively simple circuit is therefore important for implementing more complex recurrent neural networks.

An example of a 4-bit Hopfield network ADC is shown in Fig. 40 [78]. The originally proposed network consists of an array of linear resistors (also called *weights* or *synapses*) and four peripheral inverting amplifiers (*neurons*). Each neuron receives currents from the input and reference lines and from all other neurons via corresponding synapses. The analog input voltage V_S is converted to the digital code $V_3V_2V_1V_0$, i.e.

$$V_s = \sum_{i=0}^{3} 2^i V_i \tag{13}$$

by first forcing all neuron outputs to zero [79] and then letting the system evolve to the appropriate stationary state.

To understand how the Hopfield network performs the ADC operation, let us first describe its electrical behavior. Assuming leakage-free neurons with infinite input and zero output impedances, the dynamic equation governing the system evolution of the input

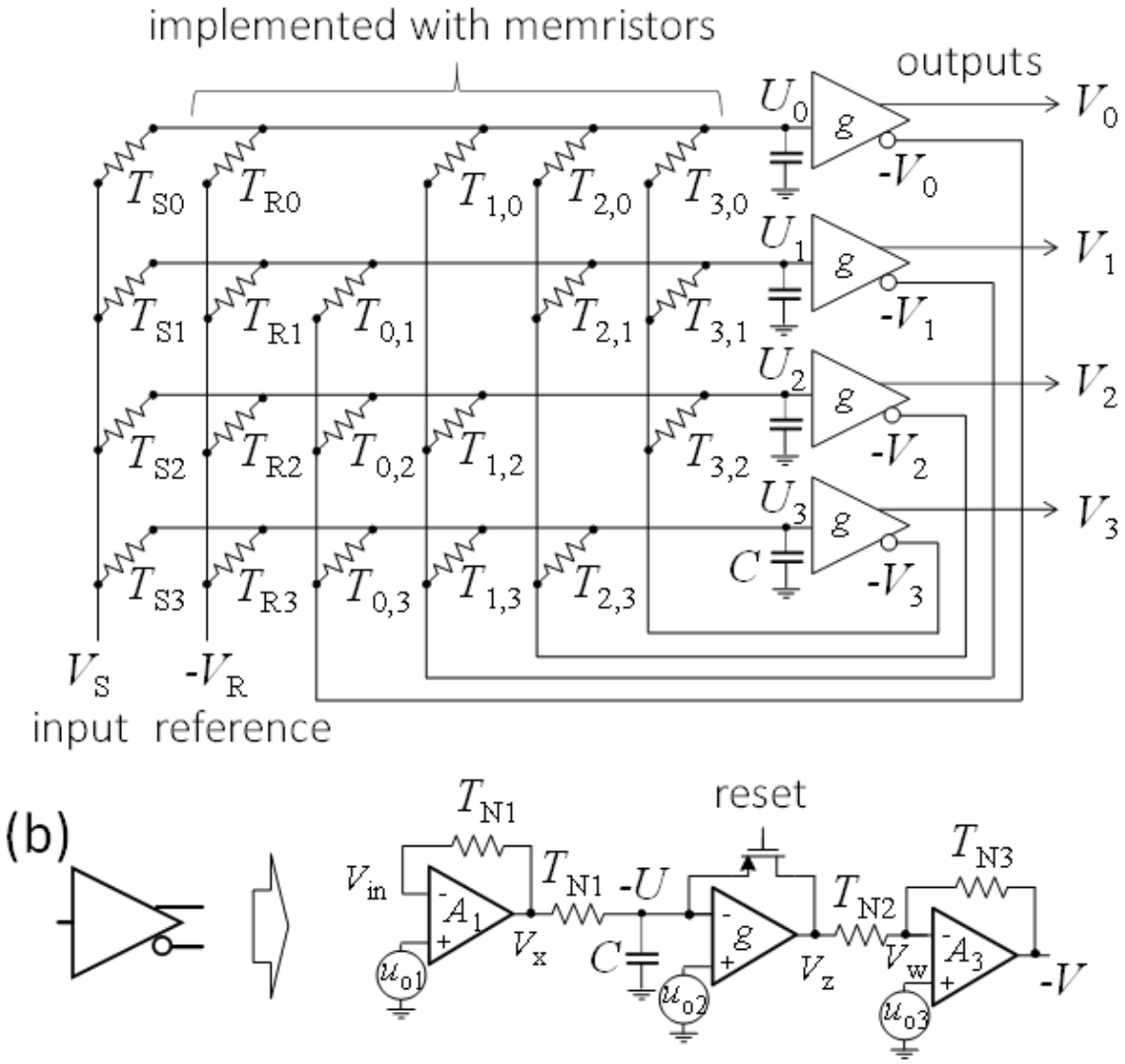


Fig. 40. (a) Conventional Hopfield network implementation of a 4-bit ADC and (b) specific implementation of a neuron as considered in this thesis.

voltage U_j of the j-th neuron is described as

$$C\dot{U}_j = -\sum_i T_{ij} V_i - T_j U_j + I_j \tag{14a}$$

$$V_i = g(U_i) \tag{14b}$$

where $g(\cdot)$ is a neuron activation function, C is the neuron's input capacitance, T_{ij} is a conductance of the synapse connecting the output of the i-th neuron with the input of the j-th neuron, while

$$I_j = T_{Sj} V_S - T_{Rj} V_R \tag{15}$$

$$T_j \equiv T_{Sj} + T_{Rj} + \sum_j T_{ij} \tag{16}$$

are the corresponding effective offset input current and effective input conductance for the j-th neuron. Here V_R is a reference voltage, while T_R and T_S are conductances of reference and input weights, respectively (Fig. 40a). Note that neuron input U_i can be either positive or negative, but the output of the neuron is either zero or positive. The inverted outputs of the neurons, which are fed back to the network, are therefore either negative or zero. One activation function suitable for such mapping is the sigmoid function $1/(1+\exp[-U])$. Neuron output needs to be inverted to keep the feedback weights positive and thus to allow physical implementation with passive devices, such as resistors.[1]

Alternatively, the Hopfield network operation can be described by an energy function. The evolution of the dynamic system described by Eq. (14) is equivalent to a minimization of the energy function:

$$E = \frac{1}{2} \sum_{ij} T_{ij} V_i V_j - \sum_j V_j I_j - \sum_j T_j \int_0^{V_j} g^{-1}(V) dV \tag{17}$$

[1]The sign of the first term on the left in Eq. 14a, and of all right hand terms in Eq. 17, is different from that of the original paper [73]. In this work we assume that all weights are strictly positive, making it necessary explicitly to flip the neuron feedback signal sign.

where the last term can be neglected for very steep transfer functions [77]. In Ref. 78, Tank and Hopfield showed that a 4-bit ADC task (Eq. 13) can be described by the following energy function:

$$E = \frac{1}{2}(V_S - \sum_{i=0}^{3} 2^i V_i)^2 - \frac{1}{2}\sum_{i=0}^{3} 2^{2i} V_i (V_i - 1) \tag{18}$$

Here the first term tends to satisfy Eq. (13), while the second tends to force each digital output V_i to be either "0 or "1". After rearranging the terms in Eq. (18) and comparing the result with Eq. (17), the appropriate weights for performing the ADC task are

$$T_{ij} = 2^{(i+j)}, \qquad T_{Sj} = 2^j, \qquad T_{Rj} = 2^{(2j-1)}. \tag{19}$$

In the Hopfield ADC network, the number of synapses grows quadratically with the number of neurons. Compact implementation of the synapses is therefore required if such circuits are to be practical. This is certainly challenging to achieve with conventional CMOS technology, because, according to Eq. 19, it requires analog weights with a relatively large dynamic range, i.e., in the order of 2^{2N}, where N is the bit precision. Weights can be stored digitally, but this approach comes with a large overhead [82]. On the other hand, analog CMOS implementations of the synapses have to cope with the mismatch issues often encountered in CMOS circuits [34]. Consequently, several attempts have been made to implement synapses with alternative, nonconventional technologies. In some of the early implementations of Hopfield networks, weights were realized as corresponding thin film [83] or metal line [84, 85] conductance values, patterned using e-beam lithography and reactive-ion-etching. The main limitation of these approaches was that the weights were essentially one-time programmable, with rather crude accuracy. A much more attractive solution was very recently demonstrated in Ref. 86, which describes a Hopfield network implementation

with synapses based on phase change memory paired with conventional field-effect transistors. That work, together with other recent advances in device technologies [87, 88] revived interest in the theoretical modeling of recurrent neural networks based on hybrid circuits [36, 89 - 92].

This thesis explores the implementation of synapses with an emerging, very promising type of memory devices, namely metal-oxide resistive switching devices ("memristor") [87, 88]. In the next section we discuss the general implementation details of the Hopfield network ADC, including the memristor devices which were utilized in the experimental setup. This is followed by a theoretical analysis of the considered hybrid circuits' sensitivity to certain representative sources of non-ideal behavior and discussion of a possible solution to such problems. The theoretical results were validated with Spice simulations and experimental work. It should be noted that preliminary experimental results, without any theoretical analysis, were reported earlier in Ref. 93, where we first presented a Hopfield network implementation with metal-oxide memristors. The only other relevant experimental work on memristor-based Hopfield networks that we are aware of was published recently in Ref. 94. However, the network demonstrated in Ref. 94 was based on 9 memristors whereas the circuit presented in this work involves 16.

Following on from our earlier works [46, 95 - 96], we here consider the implementation of a hybrid CMOS/memristive circuit (Fig. 40). In this circuit, density-critical synapses are implemented with Pt/TiO_{2-x}/Pt memristive devices, while neurons are implemented by CMOS circuits.

In their simplest form, memristors are two-terminal passive elements, the conductance of which can be modulated reversibly by applying electrical stress. Due to the simple structure and ionic nature of their memory mechanism, metal-oxide memristors have excellent scaling

prospects, often combined with fast, low energy switching and high retention [36]. Many metal oxide based memristors can also be switched continuously, i.e. in analog manner, by applying electrical bias (current or voltage pulses) with gradually increasing amplitude and/or duration.

Fig. 16a shows typical continuous switching *I*-*V*s for the considered Pt/TiO_{2-x}/Pt devices [71]. After programming the memristors to the desired resistance, the voltage drop across them was always kept within the $|V| \leq 0.2$V "disturb-free" range [71].

The need to keep the voltage drop across memristive devices small also affects neuron design. A simple leaky operational amplifier (op-amp) integrator could be sufficient to implement neuron functionality, but ensuring disturb-free operation with such a design is not easy. This issue was resolved by implementing neurons with three op-amps connected in series (Fig. 40b). The first op-amp was an inverting amplifier which held virtual ground even if the neuron's output was saturated. The second op-amp was an open loop amplifier implementing a sign-like activation function. The field effect transistor in the negative feedback of this op-amp was initially turned on to force the neuron's outputs to zero (i.e. to set into initial state before computing output) and then turned off during network convergence. The last op-amp inverted the signal and ensured that the neuron output was within the $-0.2\text{V} \leq V \leq 0$ voltage range. Note that since the neuron bandwidth was mainly determined by the input capacitance of the second amplifier, and the other sources of parasitic capacitance could be neglected for simplicity, the capacitive load of the second amplifier (Fig. 40b) was effectively a neuron input capacitance (Fig. 40a).

Assuming negligible op-amp input currents and output impedances, the Hopfield network is described by the following equations, which also account for limited gain and voltage offsets:

$$V_{xj} = A_{1j}(u_{o1j} - V_{inj}) \tag{20}$$

$$V_{zj} = g(u_{o2j} + U_j) \tag{21}$$

$$-V_j = A_{3j}(u_{o3j} - V_{Wj}) \tag{22}$$

$$\begin{aligned} T_{N1}(V_{inj} - V_{xj}) = T_{Rj}(-V_R - V_{inj}) \\ + T_{Sj}(V_S - V_{inj}) \\ + \sum_i T_{ij}(-V_i - V_{inj}) \end{aligned} \tag{23}$$

$$-C\dot{U}_j = T_{N1}(V_{xj} + U_j) \tag{24}$$

$$T_{N2}(V_{zj} - V_{Wj}) = T_{N3}(V_{Wj} + V_j) \tag{25}$$

Solving these equations results in the following dynamic equation

$$a_j C\dot{U}'_j = -\sum_i T_{ij} V'_i - a_j T_{N1} U'_j + I_j + I_{oj} \tag{26a}$$

$$b_j V'_j = g(U'_j) \tag{26b}$$

where $g(\cdot)$ is a transfer function of the saturating amplifier implemented with the second op-amp, and

$$U'_j = u_{o2j} + U \tag{27}$$

$$V'_i = u_{o3j}(1 + \frac{T_{N3j}}{T_{N2j}}) / b_j + V_i \tag{28}$$

$$a_j = 1 + (1 + \frac{T_j}{T_{N1j}}) / A_{1j} \tag{29}$$

$$b_j = \frac{T_{N3j}}{T_{N2j}} + (1 + \frac{T_{N3j}}{T_{N2j}}) / A_{3j} \tag{30}$$

$$\begin{aligned} I_{oj} = &-(T_{N1j} + T_j) u_{o1j} \\ &+ a_j T_{N1j} u_{o2j} \\ &+ \frac{1 + \frac{T_{N3j}}{T_{N2j}}}{b_j} \sum_i T_{ij} u_{o3j} \end{aligned} \tag{31}$$

Based on previous derivations, assuming ideal op-amps and no possibility of saturation by the first and last amplifiers, the dynamic equation for this neuron design can be written as

$$C\dot{U}_j = -\sum_i T_{ij} V_i - T_{N1} U_j + I_j \tag{32a}$$

$$V_j = -\frac{T_{N2}}{T_{N3}} g(U_j) \tag{32b}$$

where $g(\cdot)$ is a transfer function of the second op-amp .

For a very steep transfer function, the second term in the right hand part of Eq. 32a can be neglected [77]. The network is then described by the original energy function (Eq. 17) and the weights are proportional to those defined in Eq. (19), i.e.

$$T_{ij}' = 5T_{ij}, \qquad T_{Sj}' = T_{Sj}, \qquad T_{Rj}' = 5T_{Rj}. \tag{33}$$

where the additional coefficient 5 is due to the reduced, i.e. 0.2 V, output voltage corresponding to digital "1" in the considered circuit (as opposed to output voltage 1 V assumed in the original ADC energy function in Eq. (18) for ADC and the weights in Eq. (19) derived from that energy function).

The physical implementation of this Hopfield network ADC posed a number of additional challenges. However, it should first be mentioned that variations in neuron delay and input capacitances, which may result in oscillatory behavior and the settling in of false energy minima [79, 80], were not a problem in our case thanks to the slow operating speed, which was enforced to reduce capacitive coupling. The specific problems regarding the considered implementation were offsets in virtual ground, resulting from the voltage offsets (u_o) and limited gain (A) of the op-amps (Fig. 40b). Another, somewhat less severe, problem was the nonlinear conductance of the memristive devices (defined via parameter β–, see Eq. 2). In the Eq. (26), it is shown how limited gain and non-zero offset result in an additional constant term I_0 in dynamical equation, which can be factored into the reference weights as follows

$$T_{Rj}'' = T_{Rj}' + \frac{I_{0j}}{V_R} \tag{34}$$

The Hopfield network with practical, non-ideal neurons can still therefore be approximated by the original energy equation and it should be possible to circumvent the effects of limited gain and voltage offset by fine-tuning the reference weights. This idea was verified via SPICE modeling and experimental work, as described in the next section.

Using Eq. (2) for the memristors and SPICE models for the IC components, in the next series of simulations we studied how particular non-ideal behavior affects differential (DNL)

and integral (INL) nonlinearities in ADC circuits [97]. Fig. 41a shows INL and DNL as a function of the open loop DC gain, which was varied simultaneously for all three op-amps, assuming ideal memristors with $\beta = 0$ and no voltage offset. Note that in this simulation, the gain-bandwidth product (*GBP*) was increased proportionally to the open loop DC gain, and was equal to 3MHz at $A_{DC} = 2\times10^5$. Because the circuit operated at about 1.5 KHz, the effective gain $A \approx A_{DC}/100$ for all simulations (and also for the experimental work discussed below). Fig. 41b shows the impact of the voltage offset on DNL and INL (simulated as an offset on the ground nodes), which was varied simultaneously for all three op-amps. Finally, Fig. 41c shows the effect of *I*-*V* nonlinearity, which was varied by changing constant β in Eq. (2), assuming all other parameters of the network to be close to ideal, i.e. that the voltage offset $u_o = 0$ and the open loop DC gain $A_{DC} = 10^6$. Note that for $\beta > 0$, the memristor weights were chosen in such a way that the conductance of the device at -0.2V matched the corresponding values prescribed by Eq. (33).

The results shown in Fig. 41 confirm the significant individual contribution of the considered sources of non-ideal behavior on the ADC's performance. Fig. 42a shows the simulation results considering all these factors together for the specific values $u_o = 3$mV, $\beta = 1$, $A_{DC} = 2\times10^5$, and *GBP* = 3 MHz, which are representative of the experimental setup. The gain and voltage offset values were taken from the specifications of the discrete IC op-amps used in the experiment. Clearly, the ADC output is distorted and contains numerous errors, with the largest contribution to INL being due to finite gain (Fig. 41). Fig. 42b and 43 show the simulation results with new values for the reference weights calculated according to Eq. (34) for the 4-bit and 8-bit ADCs, respectively. The results shown in these figures confirm that non-ideal behavior op-amps, such as limited gain and voltage offsets, can be efficiently compensated by fine-tuning memristors.

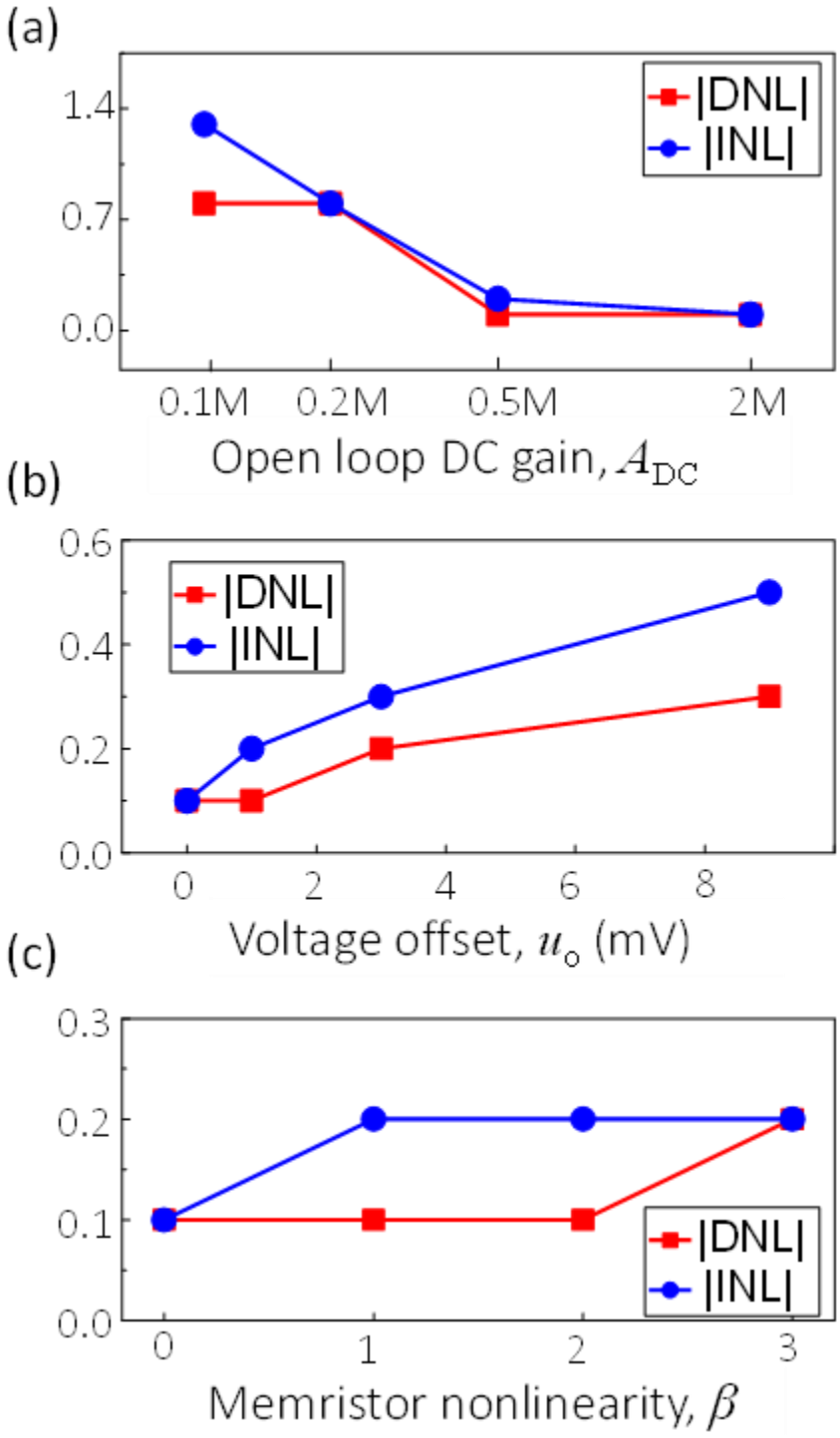


Fig. 41. Theoretical analysis of the performance sensitivity of a 4-bit Hopfield network ADC with respect to (a) open-loop DC gain, (b) voltage offsets in the operational amplifiers, and (c) the nonlinearity of memristive devices.

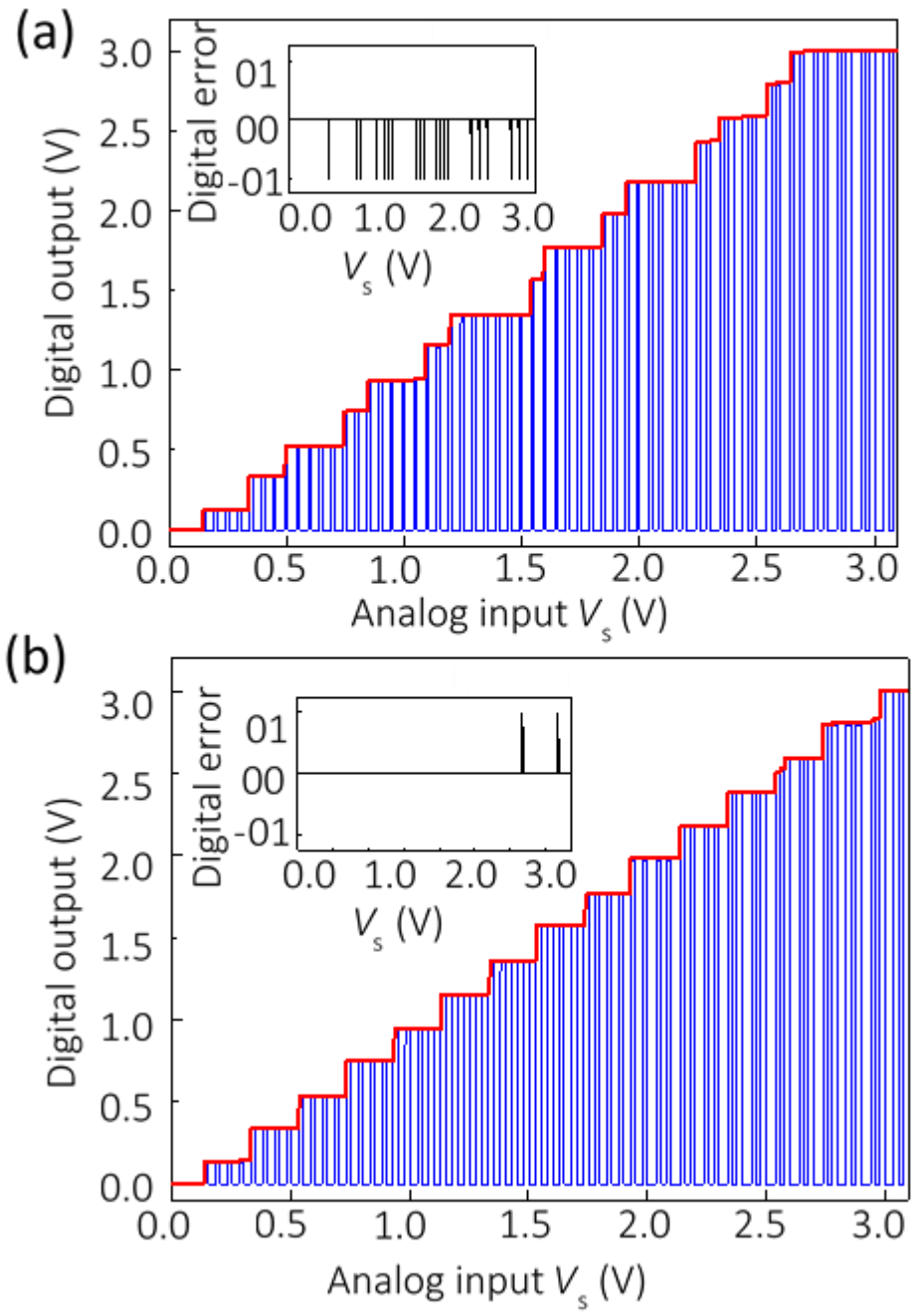


Fig. 42. Simulation results for (a) the original and (b) the optimized 4-bit Hopfield network ADC with β = 1, A_{DC} = 2×10^5, and u_o = 3mV voltage offset, which are representative parameters for the experimental setup. For the optimized network, T_R'' = 0.97 T_{R1}, T_{R2}'' = 0.86 T_{R2}, T_{R3}'' = 0.95 T_{R3}, T_{R4}'' = 0.97 T_{R4}.

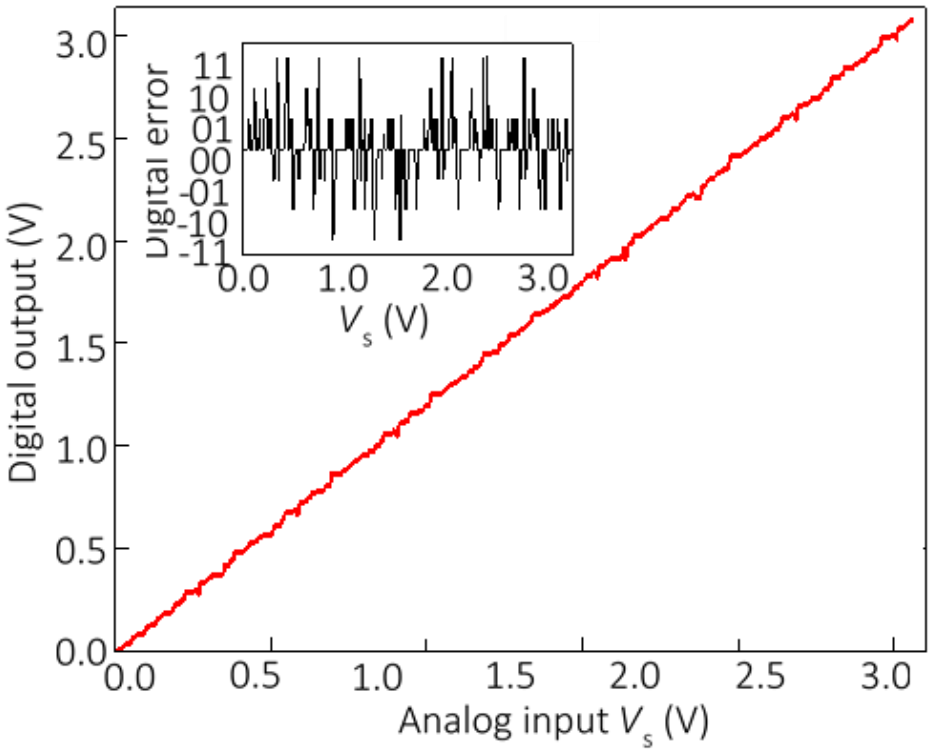


Fig. 43. Simulation results for the optimized 8-bit Hopfield network ADC *with*T_{R1}'' = 0.8 T_{R1}, T_{R2}'' = 0.81 T_{R2}, T_{R3}'' = 0.89 T_{R3}, T_{R4}'' = 0.83 T_{R4}, T_{R5}'' = 0.55 T_{R5}, T_{R6}'' = 0.74 T_{R6}, T_{R7}'' = 0.71 T_{R7}, T_{R8}'' = 0.75 T_{R8}. All other parameters are equal to those used for Fig. 42.

The simulation results were also validated experimentally by implementing a 4-bit Hopfield network ADC in a breadboard setup consisting of Pt/TiO_{2-x}/Pt memristive devices and discrete IC CMOS components (Fig. 44a). The memristor chips were assembled in standard 40-pin DIP packages by wire-bonding 20 standalone memristive devices. Because input voltage range is $0 \leq V_S \leq V_S^{max} = 3.0$ V, the weights T_S were realized with regular resistors.[2] The discrete memristors and other IC components were then connected as shown

[2] In principal, input voltage range could be decreased by increasing input weights correspondingly. However, such rescaling would require larger a dynamic range of conductances to implement Equation 18, and this was not possible with the considered memristive devices.

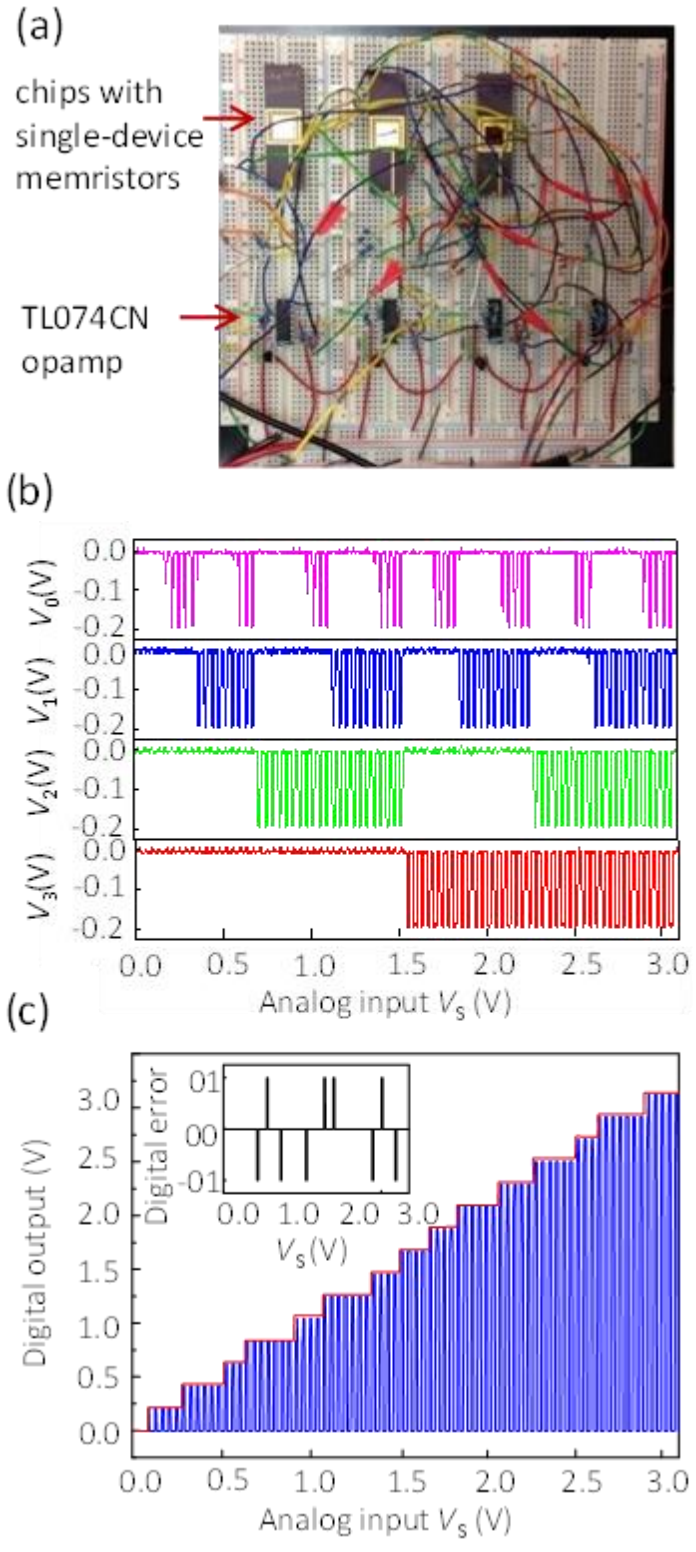


Fig. 44. Experimental results for the optimized 4-bit Hopfield ADC: (a) experimental setup, (b) applied input voltages, and (c) measured transfer characteristics.

in Fig. 40 with external wires.

The memristors implementing feedback and reference weights were first tuned ex-situ using a previously developed algorithm [71] to the values defined by Eq. (33). The ex-situ tuning for each memristor was performed individually before the devices were connected in a circuit. This was done to simplify the experiment and it is worth mentioning that in general, it should be possible to tune memristors after they are connected in the crossbar circuit, as it was experimentally demonstrated by our group for standalone devices connected in crossbar circuits [71, 98] and integrated passive crossbar circuits [47, 99].

As was discussed, limited gain and voltage offsets of operational amplifiers can be compensated by adjusting reference weights according to Eq. (31) and (34). To demonstrate in-field configurability of memristors, the reference weights were fine-tuned in-situ. In particular, reference weights were adjusted to ensure correct outputs at four particular input voltages, when V_S is equal to 1/16, 1/8, 1/4, and 1/2 of its maximum value. The tuning is performed first for $V_S = 1/16\ V_S^{max}$, for which the correct operation of ADC assumes that the least significant output bit V_0 flips from 0 to 1 (corresponding to voltage 0.2 V in our case), which is ensured by fine-tuning reference weight T_{R0}. Similarly, the output bit V_1 should flip from 0 to 1 when $V_S = 1/8\ V_S^{max}$, which is ensured by fine-tuning reference weight T_{R1} and so on. Because we started fine-tuning from the least significant output, it is sufficient to fine-tune only one corresponding reference weight at a time for a particular input voltage, which greatly simplified in-situ tuning procedure. Also, the direction of adjustment was always straightforward to determine due to monotonic dependence of the input voltage at which a particular output bit flips from 0 to 1, on the corresponding reference weight (Eq. 34).

The network parameters for the experimental work are summarized in Table I. Although there were a few A/D conversion errors in the experimental work (Fig. 44), the results are comparable with the simulations of the optimized network, and much better than those obtained for the unoptimized network. The experimental results for the unoptimized network were significantly worse in comparison with the simulation, and are not shown in this thesis.

It is worth mentioning that for the considered memristors drift of conductive state over time was negligible due to highly nonlinear switching kinetics specific to these devices [46, 47, 71]. In principle, for other types of memristors with inferior retention properties it should be possible to occasionally fine-tune memristor state to cope with conductance drift. A related issue might be measurement noise upon reading the state of the memristor, e.g. due to the fluctuations in the device conductance over time, which is sometimes observed as random telegraph noise [98 - 100]. Such noise can be tolerated by performing quasi DC read measurements, however, the downside would be potentially much slower tuning process.

Table I. Parameters for the experimentally demonstrated Hopfield network ADC.

Feed-back	Conductance (S@0.2V)	Refer-ence	Conductance (S@0.2V)
$T_{2,1}$	2e-5	T_{1R}	4.75e-6
$T_{3,1}$	4e-5	T_{2R}	2.19e-5
$T_{4,1}$	7.9e-5	T_{3R}	9.33e-5
$T_{1,2}$	2e-5	T_{4R}	41.85e-5
$T_{3,2}$	7.9e-5	Input	Conductance (S)
$T_{4,2}$	15e-5	T_{1S}	8.33e-6
$T_{1,3}$	4e-5	T_{2S}	1.67e-5
$T_{2,3}$	7.9e-5	T_{3S}	3.33e-5
$T_{4,3}$	30.9e-5	T_{4S}	6.67e-5
$T_{1,4}$	7.9e-5	Neuron	Conductance (S)
$T_{2,4}$	15e-5	T_{N1}	1e-3
$T_{3,4}$	30.9e-5	T_{N2}	1e-5
		T_{N3}	5e-4

VI. Discussion

There are still several unused reserves in our pattern classifier chip design. The first is the most straightforward improvement that uses the current-mirror design for neurons (similar to the gate-coupled circuits shown in Fig. 30e, but implemented with the floating-gate-free transistors, and hence with the signal transfer weight $w = 1$), which currently give dominant contributions to the network latency and energy dissipation (Fig. 36a). The second direct path forward is to use the more advanced 55-nm memory technology ESF3 of the same company [59]. (Our preliminary testing [62] has not found any evident showstoppers on that path.) The time delay and energy dissipation of the network with current-mirror neurons will be dominated by the synaptic arrays, and may be readily estimated for different memory technologies using the experimental values of parameter β for 180-nm ESF1 cells and more advanced 55-nm ESF3 cells. For example, our modeling of a large-scale network deep-learning convolutional networks, suitable for classification of large, complex patterns [74] (i.e. the same network which was implemented by Eyeriss chip [14]). These two improvements are shown at least a ~100X advantage in the operation speed, and an enormous, >10000X advantage in the energy efficiency, over the state-of-the-art purely digital (GPU and custom) circuits – see Table II. Moreover, the energy efficiency of these circuits would closely approach that of the human visual cortex, at much higher speed – see the last two columns of the table.

On the other hand, we also paid attention to a more general case of perceptron architecture, in particular focusing on exponential-weight networks. Although the idea of having nonlinear weights in neural networks is not new, most previous attempts has failed to gain attention either due to the lack of having specific nonlinear element to effectively

TABLE II

SPEED AND ENERGY CONSUMPTION OF THE SIGNAL PROPAGATION THROUGH THE CONVOLUTIONAL (DOMINATING) PART OF A LARGE DEEP NETWORK [1]

AlexNet[30] single pattern classification:	Digital circuits [TNNLS27]		Mixed-signal floating-gate circuits (estimates)		Visual cortex (crude estimates)
	GPU 28 nm	ASIC 65 nm	ESF1 180 nm	ESF3 55 nm	
time (s)	1.5×10^{-2}	2.9×10^{-2}	$\sim1\times10^{-4}$	$\sim6\times10^{-5}$	$\sim3\times10^{-2}$
energy (J)	1.5×10^{-1}	0.8×10^{-2}	$\sim3\times10^{-7}$	$\sim2\times10^{-7}$	$\sim5\times10^{-8}$

The estimates for floating-gate networks take into account the 55×55 = 3,025-step time-division multiplexing (natural for this particular network), and the experimental values of the subthreshold current slope of the cells - see, e.g., the inset in Fig. 2d. The (very crude) estimate of the human visual cortex operation is based on the ~25 W power consumption of $\sim10^{11}$ neurons of the whole brain, and a 30-ms delay of the visual cortex, and assumes the uniform distribution of the power over the neurons, and the same number of neurons participating in a single-pattern classification process.

implement synaptic weights or because of the inconsistency between the algorithm and underlying hardware [101, 102]. In this thesis, not only we investigate the performance of neural networks with nonlinear weights at software level but also we show that the proposed architecture has a very compact, efficient and trainable hardware implementation (from the area and power consumption points of view) that can dramatically speeds up the deep neural networks during operation.

Moreover, we investigated hybrid CMOS/metal-oxide-memristor circuit implementation of a Hopfield recurrent neural network performing analog-to-digital conversion tasks. We showed that straight forward implementation of such networks, with weights prescribed by the original theory, produces many conversion errors, mainly due to the non-ideal behavior of the CMOS components in the integrated circuit. We then proposed a method of adjusting weights in the Hopfield network to overcome the non-ideal behavior of the network components and successfully validated this technique experimentally on a 4-bit ADC circuit. The ability to fine-tune the conductance of memristor in a circuit is essential for

implementing the proposed technique. In our opinion, the work is proved to be an important milestone and its results will be valuable for implementing more practical large-scale recurrent neural networks with CMOS/memristor circuits. From a broader perspective, this thesis demonstrates one of the main advantages of utilizing memristors in analog circuits, namely the feasibility of fine-tuning memristors after fabrication to overcome variations in analog circuits. With the emerging and development of memristive corssbars [50, 104], consistent research into CMOS/nanodevice neural networks is still a very attractive future field.

Recent progress [20, 21, 23, 26, 103] in the development of machine learning algorithms using binary weights imply that our approach may be also extended to novel 3D NAND flash technologies [105] or other 3D memories [106]. Such memories may ensure much higher areal densities of the floating-gate cells, but their redesign to analog weights may be rather problematic. Note, however, the results of the most recent work [23] show a significant drop in classification performance that results from using binary weights in convolutional layers of large-scale neuromorphic networks. The performance of such networks may be improved by increasing the network size; however, its speed and energy efficiency may suffer. So, the tradeoff between the density and weight precision effects in 3D memories is far from certain yet, and requires further study.

Except for previous mentioned potential future direction, spiking neural networks are very promising candidates from energy point of view. Tremendous work [107 - 112] has already well established the foundations for large scale spiking neural networks.

To summarize, we believe that the reported results in this thesis give an important proof-of-concept demonstration of the exciting possibilities opened for neuromorphic networks by mixed-signal circuits based on industrial-grade floating-gate memories.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", in: *Proc. NIPS'12*, Lake Tahoe, CA, Dec. 2012, pp. 1097-1105.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, pp. 436-444, 2015.

[3] L. Cavigelli, M. Magno, and L. Benini, "Accelerating real-time embedded scene labeling with convolutional networks", in: *Proc. DAC'15*, San Francisco, California, Jun. 2015, p. 108.

[4] J. Schmidhuber, "Deep learning in neural networks", Neural Networks, vol. 61, pp. 5-117, 2015.

[5] "Neuromorphic Computing: From Materials to Systems Architecture", Gaithersburg, Mar. 2016, available online at http://science.energy.gov/~/media/ascr/pdf/programdocuments/docs/Neuromorphic-Computing-Report_FNLBLP.pdf

[6] C. Farabet, Y. LeCun, K. Kavukcuoglu, E. Culurciello, B. Martini, P. Akselrod, and S. Talay, "Large-scale FPGA-based convolutional networks", in: *Scaling Up Machine Learning*, ed. by R. Bekkerman, M. Bilenko, and J. Langford, Cambridge University Press, 2011, pp. 399-419.

[7] A. Putnam, A. M. Caulfield, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Y. Xiao, and D. Burger, "A reconfigurable fabric for accelerating large-scale datacenter services", in: *Proc. ISCA'14*, Minneapolis, MN, June 2014, pp. 13-24.

[8] T. Moreau, M. Wyse, J. Nelson, A. Sampson, H. Esmaeilzadeh, L. Ceze, and M. Oskin, "SNNAP: Approximate computing on programmable SoCs via neural acceleration", in: *Proc. HPCA'15*, Burlingame, CA, Feb. 2015, pp. 603-614.

[9] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 G-OPs/s mobile coprocessor for deep neural networks", in: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, Jun. 2014, pp. 682-687.

[10] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H.P. Graf, "A massively parallel coprocessor for convolutional neural networks", in: *Proc. of 20th IEEE International Conference on Application-specific Systems, Architectures and Processors*, Boston, MA, Jul. 2009, pp. 53-60.

[11] S. Cadambi, A. Majumdar, M. Becchi, S. Chakradhar, and H. P. Graf, "A programmable parallel accelerator for learning and classification", in: *Proc. PACT'10*, Vienna, Austria, Sep. 2010, pp. 273-284.

[12] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, “DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning”, in: *Proc. ASPLOS’14,* Salt Lake City, UT, Mar. 2014, pp. 269-284.

[13] P. H. Pham, D. Jelaca, C. Farabet, B. Martini, Y. LeCun, and E. Culurciello, “Neuflow: Dataflow vision processing system-on-a-chip”, in: *Proc. International Midwest Symposium on Circuits and Systems’10,* Boise, ID, Aug. 2010, pp. 1044-1047.

[14] Y. H. Chen, T. Krishna, J. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks”, in: *Proc. ISSCC’16,* San Francisco, CA, Jan. 2016, pp. 262-263.

[15] J. Lu, S. Young, I. Arel, and J. Holleman, “A 1 TOPS/W Analog Deep Machine-Learning Engine with Floating-Gate Storage in 0.13 μm CMOS,” *ISSCC Dig Tech. Papers,* pp. 504-505, Feb. 2014.

[16] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H. Yoo, “A 1.93 TOPS/W Scalable Deep Learning/Inference Processor with Tera-Parallel MIMD Architecture for Big-Data Applications,” *ISSCC Dig. Tech. Papers*, pp. 80-81, 2015.

[17] J. Hasler and H. Marr, “Finding a roadmap to achieve large neuromorphic hardware systems,” *Frontiers Neuroscience*, vol. 7, art. 118, 2013.

[18] C. Mead, *Analog VLSI and Neural Systems,* Addison Wesley, 1989.

[19] C. Mead, “Neuromorphic electronic systems”, *Proc. IEEE*, vol. 78 (10), pp. 1629-1636, 1990.

[20] M. Courbariaux, Y. Bengio, and J. P. David, “Binaryconnect: Training deep neural networks with binary weights during propagations,” *Advances in Neural Information Processing Systems.* pp. 3123-3131, 2015.

[21] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, “YodaNN: An ultra-low power convolutional neural network accelerator based on binary weights”, in: *Proc. ISVLSI’16,* Pittsburgh, PA, July 2016, pp. 236-241.

[22] J. Binas, D. Neil, G. Indiveri, S. C. Liu, and M. Pfeiffer, “Precise deep neural network computation on imprecise low-power analog hardware”, *ArXiv*:1606.07786. 2016.

[23] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, “Quantized neural networks: training neural networks with low precision weights and activations”, *ArXiv*:1609.07061, 2016.

[24] S. Han, H. Mao, and W. J. Dally, “Deep compression: compressing deep neural network with pruning, trained quantization and Huffman coding”, *ArXiv*: 1510.00149, 2015.

[25] R. S. Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmaeilzadeh, A. Hassibi, L. Ceze, and D. Burger, "General-purpose code acceleration with limited-precision analog computation", in: *Proc. ISCA'14*, Minneapolis, MN, June 2014, pp. 505-516.

[26] V. Lee, A. Alaghi, J. Hayes, V. Sathe, and L. Ceze, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing", in: *Proc. DATE'17*, Lausanne, Switzerland, Apr. 2017 (accepted).

[27] E. Säckinger, "Measurement of finite-precision effects in handwriting- and speech-recognition algorithms", *Lect. Notes on Comp. Sci*., vol. 1327, pp. 1223-1228, 1997.

[28] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology", *Neural Computation*, vol. 10, pp. 1601-1638, 1998.

[29] J. Hasler, "Opportunities in physical computing driven by analog realization", in: *Proc. IEEE International Conference on Rebooting Computing*, San Diego, CA, Oct. 2016, pp. 1-8.

[30] E. Säckinger, B. E. Boser, J. M. Bromley, Y. LeCun, and L. D. Jackel, "Application of the ANNA neural network chip to high-speed character recognition", *IEEE Transactions on Neural Networks*, vol. 3, pp. 498-505, May 1992.

[31] S. Wang, S. Pal, T. Li, A. Pan, C. Grezes, P. Amiri, Kang L. Wang, P. Gupta, "Hybrid VC-MTJ/CMOS Non-volatile Stochastic Logic for Efficient Computing", *Design, Automation & Test in Europe Conference (DATE)*, Lausanne, Switzerland, 2017.3.27-3.31.

[32] K. Likharev, "Hybrid CMOS/nanoelectronic circuits", *J. Nanoelectron. & Optoelectron*., vol. 3, pp. 203-230, Dec. 2008.

[33] L. Ceze, J. Hasler, K. Likharev, J.-s. Seo, T. Sherwood, D. Strukov, Y. Xie, and S. Yu, "Nanoelectronic neurocomputing: Status and prospects", in: *Proc. DRC'2016*, Newark, DE, June 2016, pp. 1-2.

[34] G. Indiveri *et al*., "Neuromorphic silicon neuron circuits", *Front. Neurosci*., vol. 5, pp. 1–23, 2011.

[35] K. Likharev, "CrossNets: Neuromorphic hybrid CMOS/nanoelectronic networks", *Sci. Adv. Mat.*, vol. 3, pp. 322 - 331, 2011.

[36] D.B. Strukov and H. Kohlstedt, "Resistive switching phenomena in thin films: Materials, devices, and applications", *MRS Bulletin*, vol. 37, pp. 108-114, 2012.

[37] S. Raoux, D. Ielmini, M. Wuttig, and I. Karpov, "Phase change materials", *MRS Bulletin*, vol. 37 (2), pp. 118-123, 2012

[38] W. Lu, D. Seok Jeong, M. Kozicki, and R. Waser, "Electrochemical metallization cells—blending nanoionics into nanoelectronics?", *MRS Bulletin*, vol. 37 (2), pp. 124-130, 2012.

[39] J. J. Yang, I. H. Inoue, T. Mikolajick, and C. Seong Hwang, "Metal oxide memories based on thermochemical and valence change mechanisms", *MRS Bulletin*, vol. 37 (2), pp. 131-137, 2012.

[40] E.Y. Tsymbal, A. Gruverman, V. Garcia, M. Bibes, A. Barthélémy, "Ferroelectric and multiferroic tunnel junctions", *MRS Bulletin*, vol. 37 (2), pp. 144-149, 2012.

[41] S. Wang, A. Pan, C. Chui, and P. Gupta, "PROCEED: A Pareto optimization-based circuit-level evaluator for emerging devices", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016, 24 (1):: 192-205.

[42] S. Wang, H. Lee, F. Ebrahimi, P. Amiri, K. Wang, and P. Gupta, "Comparative evaluation of spin-transfer-torque and magnetoelectric random access memory", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2016, 6(2): 134-145.

[43] D. Kim, J. Kung, S. Chai, S. Yalamanchili, and S. Mukhopadhyay, "Neurocube: A programmable digital neuromorphic architecture with high-density 3D memory", in: *Proc. ISCA'16*, Seoul, South Korea, Jun. 2016, pp. 380-392.

[44] F. Merrikh Bayat, M. Prezioso, X. Guo, B. Hoskins, D. Strukov, and K. Likharev, "Memory technologies for neural networks", in: *Proc. IMW'15*, Monterey, CA, May 2015, pp. 1-4.

[45] E. H. Lee, and S. S. Wong, "A 2.5 GHz 7.7 TOPs/W switched-capacitor matrix multiplier with co-designed local memory in 40nm", in: *Proc. ISSCC'16*, San Francisco, CA, Feb. 2016, pp. 418-420.

[46] F. Alibart, E. Zamanidoost, and D. Strukov, "Pattern classification by memristive crossbar circuits with ex-situ and in-situ training", *Nature Commun.*, vol. 4, pp. 2072-2074, 2013.

[47] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. Likharev, and D. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors", *Nature*, vol. 521, pp.61-64, 2015.

[48] M. Prezioso, I. Kataeva, F. Merrikh-Bayat, B. Hoskins, G. Adam, T. Sota, K. Likharev, and D. Strukov, "Modeling and simulation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al_2O_3/TiO_{2-x}/Pt memristors", *IEDM'15 Tech. Dig*, Washington, DC, Dec. 2015, pp. 17.4.1-17.4.4.

[49] B. Govoreanu, G. Kar, Y-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. Wouters, J. Kittl, and M. Jurczak, "10×10 nm^2 Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation", *IEDM'11 Tech. Dig*, Washington, DC, Dec. 2011, pp. 31.6.1-31.6.4.

[50] G. C. Adam, B. D. Hoskins, M. Prezioso, F. Merrikh-Bayat, B. Chakrabarti, and D. B. Strukov, "3-D memristor crossbars for analog and neuromorphic computing applications", *TED*, vol. 64, pp. 312-318, 2017.

[51] B. Chakrabarti, M. A. Lastras-Montaño, G. Adam, M. Prezioso, B. Hoskins, K.-T. Cheng and D. B. Strukov, "A multiply-add engine with monolithically integrated 3D memristor crossbar/CMOS hybrid circuit", accepted to *Nature Scientific Reports*, 2017.

[52] P. Chi, L. Shuangchen, Z. Qi, P. Gu, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory", in: *Proc. ISCA'16*, Seoul, South Korea, June 2016, pp. 27-39.

[53] S. B. Eryilmaz, D. Kuzum, S. Yu, and H. P. Wong, "Device and system level design considerations for analog-non-volatile-memory based neuromorphic architectures", *IEDM'15 Tech. Dig*, Washington, DC, Dec. 2015, pp. 4.1.1-4.1.4.

[54] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strach, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars", in: *Proc. ISCA'16*, Seoul, Korea, June 2016, pp. 14-26.

[55] Y. Nishitani, Y. Kaneko, and M. Ueda, "Supervised learning using spike-timing-dependent plasticity of memristive synapses", *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26 (12), pp. 2999-3008, 2015.

[56] C. Diorio, P. Hasler, B. A. Minch, and C. Mead, "A single-transistor silicon synapse", *IEEE Trans. Elec. Dev.*, vol. 43, pp. 1972-1980, 1996.

[57] S. M. Jung, J. Jang, W. Cho, H. Cho, J. Jeong, Y. Chang, J. Kim, Y. Rah, Y. Son, J. Park, M. S. Song, K. H. Kim, J. S. Lim, and K. Kim, "Three dimensionally stacked NAND flash memory technology using stacking single crystal Si layers on ILD and TANOS structure for beyond 30 nm node," in: International Electron Device Meeting (IEDM), 2006. 37–40.

[58] S. George S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan, "A programmable and configurable mixed-mode FPAA SoC", *TVLSI*, vol. 24 (6), pp. 2253-2261, 2016.

[59] "Superflash Technology Overview", SST, Inc,; available online at www.sst.com/technology/sst-superflash-technology .

[60] F. Merrikh Bayat, X. Guo, H.A. Om'mani, N. Do, K.K. Likharev, and D.B. Strukov, "Redesigning commercial floating-gate memory for analog computing applications", in: *Proc. ISCAS'15*, Lisbon, Portugal, May 2015, pp. 1921-1924.

[61] F. Merrikh Bayat, X. Guo, M. Klachko, N. Do, K. Likharev, and D. Strukov, "Model-based high-precision tuning of NOR flash memory cells for analog computing applications", in: *Proc. DRC'16*, Newark, DE, June 2016, pp. 1-2.

[62] X. Guo, F. M. Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do, and D. B. Strukov, "Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells", *ArXiv*:1611.03379, Nov. 2016, accepted to *CICC'17*.

[63] F. Merrikh Bayat*, X. Guo*, M. Klachko, M. Prezioso, K.K. Likharev, and D.B. Strukov, "High-Performance Analog Neurocomputing with Nanoscale Floating-Gate Memory Cell Arrays", submitted to *Science Advance*. *these authors have equal contribution.

[64] X. Guo*, F. Merrikh-Bayat*, L. Gao, B. D. Hoskins, F. Alibart, B. Linares-Barranco, L. Theogarajan, C. Teuscher, and D.B. Strukov, "Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits", *Frontiers in Neuroscience*, art. 488, Dec. 2015. *these authors have equal contribution.

[65] C. Farabet, B. Martini, B. Codra, P. Akselrod, E. Culurciello, and Y. LeCun, "NeuFlow: A runtime reconfigurable dataflow processor for vision", in: Proc. CVPRW'11, Colorado Spriongs, CO, June 2011, pp. 109-116.

[66] C. R. Schlottmann and P. E. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation", *IEEE JETCAS*, vol. 1, pp. 403-411, 2011.

[67] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, and B. Brezzo, "A million spiking-neuron integrated circuit with a scalable communication network and interface", *Science*, vol. 345, pp. 668-673, 2014.

[68] F. Alibart, L. Gao, B. Hoskins, and D. B. Strukov, "High-precision tuning of state for memristive devices by adaptable variation-tolerant algorithm", *Nanotechnology*, vol. 23, art. 075201, 2012.

[69] N. Do, L. Tee, S. Hariharan, S. Lemke, M. Tadayoni, W. Yang, et al., "A 55 nm logic-process-compatible, split-gate tlash memory array fully demonstrated at automotive temperature with high access speed and reliability", in *Proc. IMW*, 2015, pp. 46-48.

[70] Y. Tkachev, X. Liu, and A. Kotov, "Floating gate corner-enhanced poly-to-poly tunneling in split-gate flash memory cells", *IEEE Trans. Electron Devices*, vol.29, pp. 5-11, 2012.

[71] F. Alibart, L. Gao, B.D. Hoskins, and D.B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm", *Nanotechnology*, vol. 23, art. 075201, 2012.

[72] J. Li, C. I. Wu, S. C. Lewis, J. Morrish, T. Y. Wang, R. Jordan, T. Maffitt, M. Breitwisch, A. Schrott, R. Cheek, H. L. Lung, and C. Lam, "A novel reconfigurable sensing scheme for variable level storage in phase change memory", *International Memory Workshop (IMW)*, 2011, pp. 1-4.

[73] N. Papandreou, H. Pozidis, A. Pantazi, A. Sebastian, M. Breitwisch, C. Lam, and E. Eleftheriou, "Programming algorithms for multilevel phase-change memory", *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 329-332.

[74] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing", in: *Proc. NIPS'15*, Montreal, Canada, Dec. 2015, pp. 1117-1125.

[75] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[76] S. Wang, H. Hu, H. Zheng, and Puneet Gupta, "MEMRES: A fast memory system reliability simulator", *IEEE Transactions on Reliability*, 2016, 65(4): 1783-1797.

[77] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neruons", *Proceedings of the National Academy of Sciences*, vol. 81, pp. 3088-3092, 1984.

[78] D. W. Tank and J. J. Hopfield, "Simple neural optimization networks - an A/D converter, signal decision circuit, and a linear-programming circuit", *IEEE Transactions on Circuits and Systems*, vol. 33, pp. 533-541, May 1986.

[79] B. W. Lee and B. J. Sheu, "Design of a neural-based A/D converter using modified Hopfield network", *IEEE J. Solid-State Circ*., vol. 24, pp. 1129-1135, 1989.

[80] M. J. S. Smith and C. L. Portmann, "Practical design and analysis of a simple "neural" optimization circuit", *IEEE Trans. Circ. Syst*., vol. 36, pp. 42-50, 1989.

[81] Y. Chigusa and M. Tanaka, "A neural-like feed-forward ADC", In: *Proc. ISCAS'90*, New Orleans, LA, May 1990, pp. 2959-2962.

[82] A. Moopenn, T. Duong, and A. P. Thakoor, "Digital-analog hybrid synapse chips for electronic nerual networks", *Advances in Neural Inofmration Processing Systems*, pp. 769-776, 1990.

[83] L. D. Jackel, H. P. Graf, and R. E. Howard, "Electronic neural network chips", *Applied Optics*, vol. 26, pp. 5077-5080, 1987.

[84] H. P. Graf, L. D. Jackel, R. E. Howard, B. Straughn, J. S. Denker, W. Hubbard, D. M. Tennant, and D. Schwartz, "VLSI implemetnation of a nerual network memory with several hundreds of neurons", *AIP Conference Proceedings*, vol. 151, pp. 182-187, 1986.

[85] D. B. Schwartz, R. E. Howard, J. S. Denker, R. W. Epworth, H. P. Graf, W. Hubbard, L. D. Jackel, B. Straughn, and D. M. Tennant, "Dynamics of microfabriacted eletronic neural networks", *Applied Physics Letters*, vol. 50, pp. 1110-1112, 1987.

[86] S. B. Eryilmaz1, D. Kuzum, R. Jeyasingh, S. Kim, M. BrightSky, C. Lam and H.-S. P. Wong, "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array", *Frontiers in neuroscience*, vol. 8, art.205,2014.

[87] A. Wu, S. Wen, and Z. Zeng, “Synchronization control of a class of memristor-based recurrent neural networks”, *Information Sciences*, vol. 183, pp. 106-116, 2012.

[88] G. Zhang, Y. Shen, and J. Sun, “Global exponential stability of a class of memristor-based recurrent neural networks with time-varying delays”, *Neurocomputing*, vol. 97, pp. 149-154, 2012.

[89] E. Lehtonen, J. H. Poikonen, M. Laiho, and P. Kanerva, “Large-scale memristive associative memories”, *IEEE Trans. VLSI*, vol. 22, pp. 562-574, 2014.

[90] T. J. Walls and K. K. Likharev, “Self-organization in autonomous, recurrent, firing-rate CrossNets with quasi-hebbian plasticity*”, IEEE Trans. Neural Networks and Learning Systems*, vol. 25, pp. 819-824, 2014.

[91] R. Waser, R. Dittmann, G. Staikov, and K. Szot, “Redox-based resistive switching memories–nanoionic mechanisms, prospects, and challenges”, *Advanced Materials*, vol. 21, pp. 2632-2663, 2009.

[92] R. Rakkiyappan, A. Chandrasekar, S. Laksmanan, and J. H. Park, “State estimation of memristor-based recurrent neural networks with time-varying delays based on passivity theory”, *Complexity*, vol. 19, pp. 32-43, 2014.

[93] L. Gao, F. Merrikh-Bayat, F. Alibart, X. Guo, B.D. Hoskins, K.-T. Cheng, and D.B. Strukov, "Digital-to-analog and analog-to-digital conversion with metal oxide memristors for ultra-low power computing", in: *Proc. NanoArch'13*, New York, NY, July 2013.

[94] S.G. Hu, Y. Liu, Z. Liu, T. P. Chen, J. J. Wang, Q. Yu, L. J. Deng, Y. Yin, and S. Hosaka, “Associative memory realized by a reconfigurable memristive Hopfiled neural network”, *Nature Communications*, vol. 6, art. 7522, 2015.

[95] F. Merrikh-Bayat, F. Alibart, L. Gao, and D.B. Strukov, “A reconfigurable FIR filter with memristor-based weights”, in: *Proc. ISCAS’15*, June 2014, Melbourne, Australia, 2014.

[96] L. Gao, F. Alibart, and D. Strukov, “Programmable CMOS/memristor threshold logic”, *IEEE Trans. Nanotechnology*, vol. 12 (2), pp. 115-119, 2013.

[97] R. J. van de Plassche, CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, Kluwer Academic Publishers: Norwell, MA, 2nd Ed., 2003.

[98] L. Gao, F. Alibart, and D. B. Strukov, “A high resolution nonvolatile analog memory ionic devices”, in: *Proc. Non-Volatile Memories Workshop*, San Diego, CA, Mar. 2013.

[99] M. Prezioso, I. Kataeva, F. Merrikh-Bayat, B. Hoskins, G. Adam, T. Sota, K. Likharev, and D. Strukov, “Modeling and implementation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al_2O_3/TiO_{2-x}/Pt memristors”, accepted to *IEDM'15*, Dec. 2015.

[100] L. Gao, F. Alibart, and D.B. Strukov, "Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices", in: *Proc. VLSI-SoC'12*, Santa Cruz, CA, Oct. 2012, pp. 88-93.

[101] M. Milev, and M. Hristov, "Analog implementation of ANN with inherent quadratic nonlinearity of the synapses," *Neural Networks, IEEE Transactions on*, 14(5), 1187-1200.

[102] J. B. Lont, and W. Guggenbühl, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses," *Neural Networks, IEEE Transactions on*, 3(3), 457-465.

[103] S. Yu, Z. Li, P.-Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu, and H. Qian, "Binary neural network with 16 Mb RRAM macro chip for classification and online training", *IEDM'16 Tech. Dig*, San Francisco, CA, Dec. 2016.

[104] L. Gao, I. T. Wang, P. Y. Chen, S. Vrudhula, J. Seo, Y. Cao, T. H. Hou, and S. Yu, "Fully parallel write/read in resistive synaptic array for accelerating on-chip learning", *Nanotechnology*, vol. 26 (45), art. 455204, 2015.

[105] K. Park, S. Nam, D. Kim, P. Kwak, D. Lee, Y. Choi, M. Choi, D. Kwak, D. Kim, M. Kim, H. Park, S. Shim, K. Kang, S. Park, K. Lee, H. Yoon, K. Ko, D. Shim, Y. Ahn, J. Ryu, D. Kim, K. Yun, J. Kwon, S. Shin, D. Byeon, K. Choi, J. Han, K. Kyung, J. Choi, and K. Kim, "Three-dimensional 128 Gb MLC vertical NAND Flash memory with 24-WL stacked layers and 50 MB/s high-speed programming", *JSSC*, vol. 50 (1), pp. 204 – 213, 2015.

[106] "3D Xpoint Technology", 2015, available online at https://www.micron.com/about/emerging-technologies/3d-xpoint-technology

[107] W. Gerstner, R. Ritz, and J. Leo van Hemmen, "Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns", *Biological Cybernetics*, vol. 69, pp. 503-515, 1993.

[108] W. Gerstner and W. Kistler, *Spiking Neuron Models*, Cambridge U. Press, 2002.

[109] M. Prezioso, F. Merrikh Bayat, B. Hoskins, K. Likharev, and D.B. Strukov, "Self-adaptive spike-time-dependent plasticity of metal-oxide memristors", *Nature Scientific Reports*, vol. 6, art. 21331, 2016.

[110] S. Ramakrishnan, P.E. Hasler, and C. Gordon, "Floating gate synapses with spike-time-dependent-plasticity", *TBCAS*, vol. 5 (3), pp. 244-252, 2011.

[111] J.S. Seo, B. Brezzo, Y. Liu, B.D. Parker, S.K. Esser, R.K. Montoye, B. Rajendran, J.A. Tierno, L. Chang, D.S. Modha, and D.J. Friedman, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons", in: *Proc. CICC'11*, San Jose, CA, Sep. 2011, pp. 1-4.

[112] B. Zhang, Z. Jiang, Q. Wang, J. Seo, and M. Seok, "A neuromorphic neural spike clustering processor for deep-brain sensing and stimulation systems", in: *Proc. ISLPED15*, Rome, Italy, July 2015, pp. 91-97.

Appendix

LIST OF CHIPS FABRICATED

Chip Description	Applications
180-nm ESF1 Modified Array	Exponential-Weight Multilayer Perceptron (MLP)
180-nm CMOS Circuits	Neuron and Digital Periphery Designs
180-nm Mixed Signal Test	Small MLP; Vector-by-matrix Multiplier
180-nm Mixed Signal MLP	MNIST Hand Digits Recognition
180-nm Mixed Signal Convolution Neuron Network (CNN)	MNIST Hand Digits Recognition
55-nm ESF3 Modified Array	Vector-by-matrix Multiplier