Specifications of Nanoscale Devices and Circuits for Neuromorphic Computational Systems

Bipin Rajendran, *Senior Member, IEEE*, Yong Liu, Jae-sun Seo, *Member, IEEE*, Kailash Gopalakrishnan, Leland Chang, *Senior Member, IEEE*, Daniel J. Friedman, *Member, IEEE*, and Mark B. Ritter

***Abstract*—The goal of neuromorphic engineering is to build electronic systems that mimic the ability of the brain to perform fuzzy, fault-tolerant, and stochastic computation, without sacrificing either its space or power efficiency. In this paper, we determine the operating characteristics of novel nanoscale devices that could be used to fabricate such systems. We also compare the performance metrics of a million neuron learning system based on these nanoscale devices with an equivalent implementation that is entirely based on end-of-scaling digital CMOS technology and determine the technology targets to be satisfied by these new devices. We show that neuromorphic systems based on new nanoscale devices can potentially improve density and power consumption by at least a factor of 10, as compared with conventional CMOS implementations.**

***Index Terms*—CMOS, hybrid integrated circuits, neural network hardware, resistive random access memory (RRAM).**

I. Introduction

HUMAN brains are the most power-efficient computational engines known to man—they perform the many complex computations that underlie cognition, action, and thought yet consume a mere 20 W [1]. Our brain consists of approximately 10^{11} neurons, each of which creates action potentials (voltage spikes) at an average rate of 10 Hz [2]. These communication signals are then sent to other neurons through synapses; the typical fan-out of a neuron is in the range of 5000–10 000, resulting in a total of about 10^{15} synapses [2]. The tokens of information processing in the brain are the action potentials, whereas the strength of communication between neurons is encoded in the effective conductance of the synapse. The communication strength of synapses can change with activity, and it is believed that our ability to learn and form new memories are based on this synaptic plasticity [3], [4].

Silicon-based computational systems, on the other hand, are based on the von Neumann model of computation [5], where data storage and processing typically happens in physically distinct entities (i.e., memory and CPU). Computation involves multiple steps in which data are fetched from the memory to the CPU, where it is modified according to the algorithms specified by the programs being executed, and the processed data are sent back to the memory for storage. Such computational systems derive their processing power by leveraging the fast switching speed of transistors used to build the CPU, memory, and communication infrastructure. In contrast to the brain, there are only few channels through which information is sent back and forth at any point in time, although data transfer and processing happens at least a million times faster.

The superior efficiency of the brain in performing fuzzy and fault-tolerant computation has motivated engineers to mimic the key algorithmic and computational features of the brain in software and silicon-based hardware [6], [7]. Neuromorphic computation aims to mimic the key operating principles, algorithms, and architecture of the brain and holds promise to deliver the next generation of systems capable of tackling a wide variety of unstructured computational problems. Although reverse engineering the brain remains a grand challenge of this century, the neuroscience community has made significant progress in the past 50 years and has provided enough clues for engineers to start mimicking some of these features in nanoscale electronic devices [8]–[13]. There are also many efforts all over the world to develop large neuromorphic systems that can interpret fuzzy and noisy inputs and make intelligent decisions, but these primarily employ CMOS technology [14]–[16]. However, a clear understanding of the requirements and targets for operating conditions of nanoscale devices that are being developed to mimic the key features of biological neurons and synapses has been lacking.

In this paper, we present a scalable integration scheme for implementing large learning systems that could potentially be superior in performance over conventional digital CMOS implementations. In this approach, an interconnected network of crossbar arrays is used to implement integrate-and-fire neuron circuits and plastic synapse circuits. The synapse devices are built using nanoscale memristive resistive random access memory (RRAM) devices [11], whereas analog CMOS circuits could be used to implement neuron behavior [7], [17]. We determine the operating characteristics and technology targets of these new devices necessary to develop power- and area-efficient learning systems. We also compare the characteristics of equivalent systems that are designed using conventional digital CMOS technology to quantify the relative benefits. In the digital implementation, the synapses are implemented using static RAM (SRAM) cells—each synapse consists of 4 SRAM bits for storage [18]. The neuron spiking and integration behavior is also implemented in digital CMOS technology using latches, counters, and adders.

Manuscript received August 3, 2012; revised September 24, 2012; accepted October 18, 2012. Date of publication December 5, 2012; date of current version December 19, 2012. The review of this paper was arranged by Editor Y.-H. Shih.

B. Rajendran was with IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA. He is now with the Indian Institute of Technology, Bombay 400-076, India (e-mail: bipin@ee.iitb.ac.in).

Y. Liu, J. Seo, K. Gopalakrishnan, L. Chang, D. J. Friedman, and M. B. Ritter are with IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.

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

Digital Object Identifier 10.1109/TED.2012.2227969

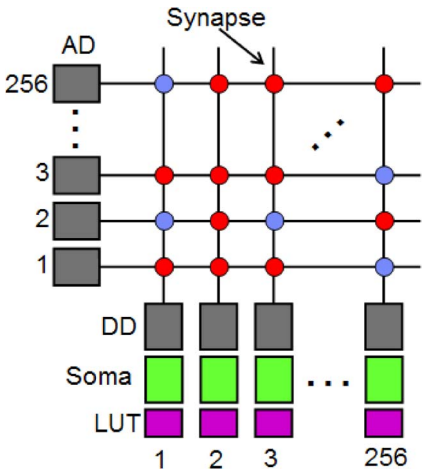


Fig. 1. Architecture of the neurosynaptic core consisting of 256 neuron circuits and 256^2 synapses. Each neuron consists of its axon driver (AD), dendrite driver (DD), soma (LIF), and look-up table (LUT) that stores the connectivity information.

A critical design choice for a neuromorphic system is the speed of operation. Although most earlier designs maintained one-to-one correspondence with biology and chose the average spike rate f_{avg} of neurons to be 10 Hz, accelerating the rate of learning and decision making is essential to handle large workloads. This is particularly true for many enterprise applications such as fraud detection in online transactions, analysis of financial markets, image and video analysis, speech recognition and tagging, and autonomous navigation. It has been observed that, for large neural systems, the typical spike probability p_{spike} for a neuron at any instant of time is about 0.01 [2]. Based on this, the time step resolution required for accurate neural network simulations can be estimated from the following relation:

$$T_{\text{step}} = \frac{p_{\text{spike}}}{f_{\text{avg}}}. \tag{1}$$

Hence, most computational simulations of large neural systems use a 1-ms time step T_{step} for updating the internal states of neurons and calculating synaptic currents. Alternatively, the emulation of the overall network behavior could be accelerated by appropriately scaling f_{avg} and T_{step} while keeping p_{spike} invariant. In this paper, we choose an acceleration factor of 1000, resulting in the average spike rate of about 10 kHz for each neuron, which fires, on average, 1% of the time. The challenge, then, is to efficiently simulate a large network of such neurons with sufficient accuracy and speed.

II. Architectural Framework

Our implementation of a large learning network consists of repeating units of what are called "neurosynaptic cores" [19]. Each core is composed of a crossbar array with electronic synapses at the junctions and circuits at the periphery of the array to mimic the action of the soma of biological neurons. In addition, there are driver circuits at the periphery to control the voltages on the input (dendrite, vertically drawn) and output (axon, horizontally drawn) wires, as shown in Fig. 1.

The metal lines of the axon/dendrite have a finite electrical resistance, placing a limitation on the size of crossbar arrays that can be built due to IR drop that develops during signaling.

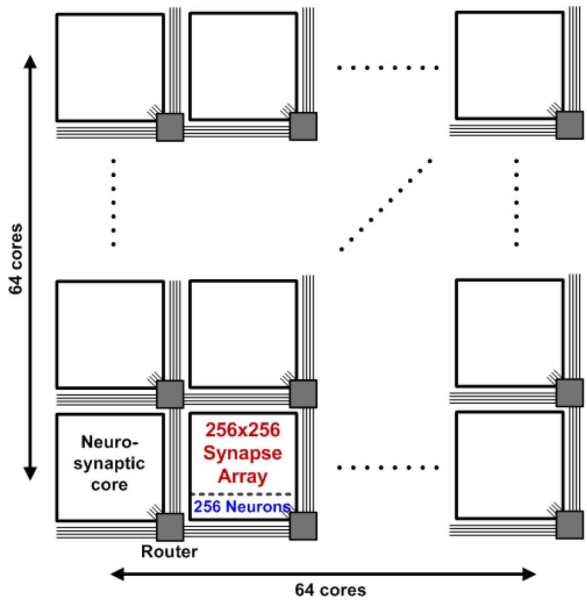


Fig. 2. High-level architecture for learning systems showing a tiled array of neurosynaptic cores that communicate to each other using a packet routing digital mesh network. To realize a system with 1 million neurons, 4096 cores (with 256 neurons) are tiled in a 64×64 array.

In order to allow neurons in one core to have connections to neurons in other cores, a digital packet routing architecture, as shown in Fig. 2, could be utilized. Each core has a local router, which communicates to the routers of other cores through a dedicated network-on-chip (NoC). Every axon wire has a unique binary address, and each neuron maintains a local look-up table (LUT) that stores the addresses of the axons in other cores to which it communicates. For a system with k axons per core and B cores, the address requires $\log_2(k) + \log_2(B)$ bits.

When a neuron within a core spikes, it communicates the Axon addresses in its LUT to a first in, first out (FIFO) queue in the router. The router then communicates the spike event to the destination axons via the NoC. The overall network behavior is correctly emulated as long as the worst case latency of communication through the network is a small fraction of T_{step}. Crossbar arrays can thus mimic the dense local connectivity between neurons, and the mesh network permits sparse long-range connections, which is similar to the architecture of the brain.

With this interconnection scheme, the activity within the core could be implemented using analog or digital circuits. Whether the neuron membrane potential is stored as a continuous valued variable (for instance, as a voltage across a capacitor) or as a discrete multibit variable (whose value is held in CMOS latches), the only information that needs to be transmitted between the cores is the single bit denoting the issue of a spike and the addresses of the destination axons. Thus, this scheme is ideal for implementing spike-driven neural networks [20].

III. Nanoelectronic Implementation

The advent of nanoscale materials has made it possible to develop compact power-efficient devices and circuits that could be used to implement the functionality of neurons and synapses. Here, we discuss the requirements and possible implementations of these devices.

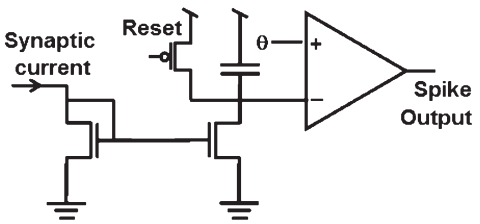


Fig. 3. Schematic of an analog neuron LIF circuit.

A. Analog Neuron

The simplest model used to emulate the behavior of the neuron cell body is the well-known leaky integrate-and-fire (LIF) model, which is given by the following differential equation:

$$C\frac{dV(t)}{dt} = -g_L\left(V(t) - E_L\right) + I_{\text{app}}(t) \tag{2}$$

where g_L is the leakage conductance term, C is the membrane capacitance, and I_{app} is the external input current. When $V(t)$ exceeds threshold θ, a spike is issued, and V is reset to E_L. This is the easiest scheme to implement as it requires only one capacitor, whose voltage represents the membrane potential, a transistor to implement the leakage term, and a threshold detector with a reset mechanism to issue the spike (see Fig. 3). Although there have been many efforts to mimic the various nonlinear phenomena that biological neurons exhibit (refractory periods, spike frequency adaptation, bursting, etc.) [21], the computational merits of these processes are still topics of active research [22]. Hence, in this paper, we will assume that the simple LIF model will suffice to capture the basic characteristics of biological neurons and apply this to both analog and digital implementations.

Once a neuron spikes, this information has to be transmitted to all its fan-out neurons through their respective synapses. The neuron should also issue programming voltages to alter the state of the synapse to implement the timing-dependent plasticity. Many schemes have been proposed to implement various plasticity rules, taking into account the memory programming characteristics of novel RRAM devices [23]–[25]. In this paper, we will assume a very simple and flexible scheme that relies on the rectification property of a diode that is connected in series with the RRAM, as the waveforms needed to implement plasticity are simple exponentials, which can be generated using $R-C$ circuits in a straightforward manner.

The basic idea behind the neuron spike communication scheme is as follows: When a neuron spikes, it generates a voltage waveform on its axon, which exponentially decays with a characteristic time constant. In order to avoid disturbing the state of the RRAM synapse, we will assume that the maximum current (i.e., when the RRAM is in its lowest resistance state) that can flow in this configuration has to be no more than 10% of the minimum programming current.

The neuron also has to create voltage pulses that modify both its upstream and downstream synapses according to timing-dependent plasticity rules. To achieve this, we stipulate that, when a neuron spikes, in addition to the voltage waveform that is applied on the axon, it should also create a voltage waveform on its dendrites, with exponential decay characteristics. Hence, when two neurons closely spike in time, it is the overlap of the two voltage waveforms that causes the synapse to get programmed. The waveforms can be intelligently chosen to result in various spike timing dependent plasticity (STDP) rules [23].

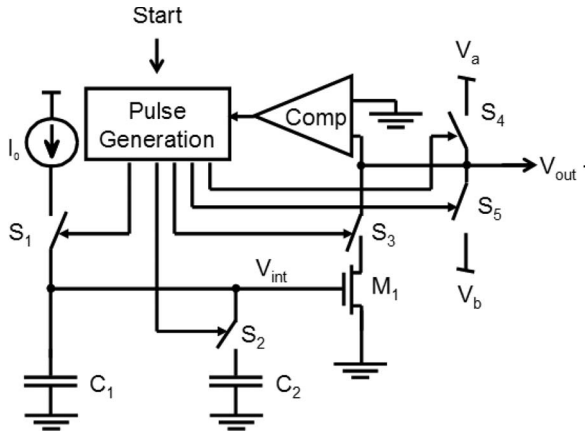


Fig. 4. Schematic of a axon/dendrite driver circuit.

We now discuss the design of a driver circuit to generate the programming waveforms. As will be explained in Section III-B, our chosen waveform consists of two exponential signals, mimicking the shape of the biological action potentials (see Fig. 5). This waveform is generated by the circuit in Fig. 4, which consists of predriver and driver stages. The predriver stage incorporates current source I_0, switches S_1 and S_2, and capacitors C_1 and C_2, which generates the double-exponential waveform. Transmission gates are used for the switches. The two time constants required in the proposed waveform are generated by the current integration on capacitors C_1 and $C_1 + C_2$. The driver stage incorporating transistor M_1 and switches S_3-S_5 drives the axons and dendrites. Transistor M_1 roughly transfers the wave shape of the internal signal V_{int} to output V_{out}. Switches S_4 and S_5 set the peak amplitudes V_a and V_b of the two decaying shapes in the proposed waveform. During the initialization of the driver output, switch S_3 disconnects the predriver stage from the driver output. The comparator and the pulse generation block generate the pulsed control signals for all the switches S_1-S_5 from the STDP enable signal, the start, and the output signal V_{out}. The pulse generation block is constructed using simple digital logic circuits.

B. Nanoscale RRAM Synapse

As shown in Fig. 5, the synaptic element consists of a diode connected in series with the RRAM device, and they obey the following characteristics: We assume that the RRAM is bipolar (i.e., the conductance change depends on the sign of the applied voltage) and can be programmed to intermediate resistance states determined by the magnitude of the programming current. Prior demonstrations have achieved full switching using voltages less than 1 V, currents in the range of 0.1–100 μA, and pulse duration in the range of 50–500 ns [26], [27]. For synaptic learning, only slight resistance changes (and not complete switching) are necessary with each spike.

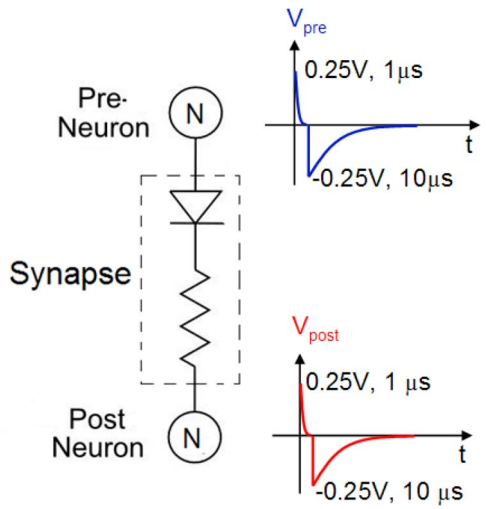


Fig. 5. Programming scheme for implementing spike communication and synaptic plasticity. When a neuron spikes, $V_{\rm pre}$ is applied to the axons, and $V_{\rm post}$ is applied to the dendrites.

Hence, we assume that the synapse RRAM at the 10-nm node can be switched on/off over the full conductance range using a voltage of 500 mV, a current of 1 μA, and pulse duration of 1 μs. To obtain reasonable power for communication of spikes, we also stipulate that the read voltage for the RRAM–diode combination should be no more than 250 mV, and the maximum spike communication current is less than 100 nA.

Reliable diodes based on mixed ionic–electronic conduction materials have been demonstrated for memory applications [28]. The characteristics of the diode needed for the synaptic element are similar—they should conduct current in both directions, with a leakage current of less than $\pm$100 pA at $\pm$50 mV. The diode current typically exponentially increases with voltage with an inverse slope of 60 mV/dec. Hence, for the diode to conduct 100 nA, 230 mV should drop across it. This implies that the resistance of the RRAM in the fully ON-state is $R_{\rm ON} = [250{-}230]\ {\rm mV}/100\ {\rm nA} = 200\ {\rm k}\Omega$. The bipolar diode is necessary in our scheme to ensure that the state of the RRAM is unaltered, if there is no dendritic waveform present when a neuron spikes and sends the axonal waveform.

RRAM conductance is communicated to downstream neurons by the the axonal waveform. We assume that the effective on/off ratio of the diode–RRAM combination should be at least 10—this implies that the OFF-resistance of the RRAM is $R_{\rm OFF} = [250{-}(230{-}60)]\ {\rm mV}/10\ {\rm nA} = 8\ {\rm M}\Omega$. Hence, the full on/off ratio of the RRAM should exceed 40.

The synaptic state is altered when the axonal and dendritic waveforms overlap. In this configuration, a larger voltage drops across the diode, and because of the exponential dependence of the diode current on voltage, a significantly larger current flows through the RRAM, altering its state. Depending on the sign of the effective voltage, a positive or negative current flows through the RRAM, and hence, its conductance increases or decreases, capturing timing-dependent plasticity.

The signals shown in Fig. 5 implement anti-STDP learning in RRAM synapses. We have chosen the $V_{\rm pre}$ and $V_{\rm post}$ waveforms to be identical, consisting of two parts: The initial part has a peak amplitude $V_a = 250$ mV and an exponential decay time constant of $\tau_1 = 1$ μs and then followed by a second pulse of peak amplitude $V_b = -250$ mV and a decay time constant of $\tau_2 = 10$ μs. For positive Δt, a maximum current flows through the RRAM at the instant when the positive peak of $V_{\rm pre}$ coincides with the negative portion of $V_{\rm post}$, and the magnitude of this current depends on the instantaneous value of $V_{\rm post}$ (see Fig. 6). The effective current flowing through the RRAM and, hence, the final state of the synapse depend on the time of the overlap between the waveforms (see Fig. 7). Note that different plasticity rules could be obtained by simple modifications of the characteristics of these waveforms.

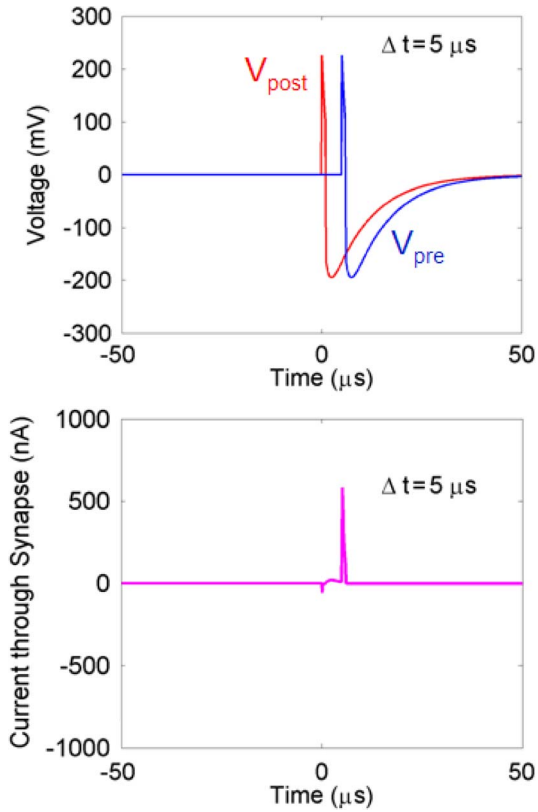


Fig. 6. (Top) Pulse overlap and (bottom) current flow for $\Delta t = +5$ μs. In this example, the postneuron spikes at $t = 0$ and issues the dendritic waveform $V_{\rm post}$. The preneuron spikes at $t = +5$ μs and issues the axonal waveform $V_{\rm pre}$. Δt defined as $t_{\rm pre} - t_{\rm post}$ is positive for acausal firing.

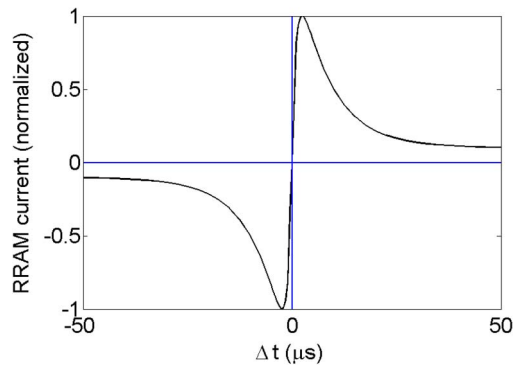


Fig. 7. Effective current flow through the RRAM device for various overlap times (and hence, the maximum conductance change) as a function of spike timing difference.

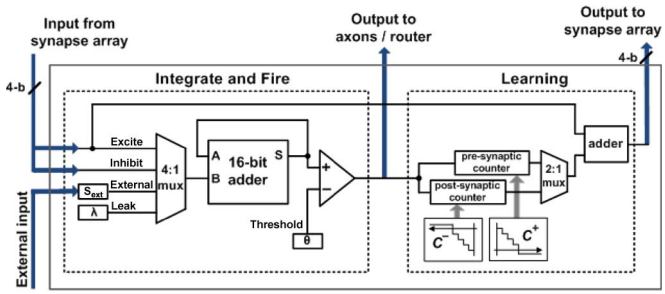


Fig. 8. Block diagram of a digital neuron implementing integrate-and-fire and learning behavior [18].

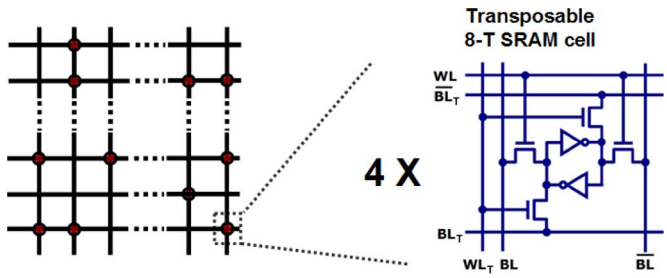


Fig. 9. Block diagram of a digital synapse capable of implementing timing-dependent plasticity [18].

IV. Digital CMOS Implementation

If digital circuits are employed to emulate the functionality of neurons and synapses, one can leverage existing CMOS logic with its scaling and reliability. Furthermore, many design tools that are available for digital circuit design could be used for performing design and verification. In [18], a digital scheme was proposed, where CMOS logic circuits are used to build neurons and transposable SRAM arrays are used as synapses. We describe the salient features of this approach in the succeeding discussion.

A. Digital Neuron

The basic idea behind the digital implementation of neuron behavior is to discretize (2) as

$$C\frac{V(n+1)-V(n)}{T_{\text{step}}} = -g_L\left(V(n)-E_L\right)+I_{\text{app}}(n) \quad (3)$$

and this equation could be then implemented using logic circuits such as adders, multipliers, and counters, as shown in Fig. 8. Once a spike is detected, the neuron circuit accesses the digital synapse array and reads the value of the synapses connected to the axon of the spiking neuron. In order to implement the timing-based plasticity on its synapses, the neuron circuit employs counters to track the time that has elapsed since the last spike event and use this information to read and modify the value of the synapse. All this computation can be locally done within the neuron circuit, and synaptic communication and plasticity are reduced to memory array read and programming.

B. Digital Synapse

One way to implement the synapses in digital circuits is to employ SRAM cells with slight modifications for reading and programming from both bit line (BL) and word line (WL) directions, as shown in Fig. 9. For this paper, we assume that at least 4 bits are required to represent each synaptic weight; thus, four SRAM cells are used to represent each synapse.

V. System-Level Analysis

In order to evaluate the performance of a large learning system, we chose a network with 10^6 neurons, with each neuron having a maximum fan-out of 1024, as our design point. We further assume that the entire emulation is done at a 1000× speedup over biology. In our design, each core has 256 neurons and 256^2 synapses. This immediately implies that there are 4096 cores, which could be laid out in a square tile of 64 × 64. We also chose the 10-nm node as the design point for our comparison. Our calculations are based on scaling projections of IBM's 22-nm CMOS technology (including the deep trench capacitors) at the 10-nm node.

TABLE I
Estimates for Area (in Square Micrometers) at the 10-nm Node for the Neurosynaptic Core

	256 Neurons	256^2 Synapses
Analog	1434	1770
Digital	15360	28300

A. Area Comparison

The area required for the analog neuron circuit is about 5.6 μm^2—the integrator and comparator circuits implementing the LIF function occupy 5 μm^2, and the area of the capacitors for membrane potential and timekeeping can be limited to about 0.6 μm^2 by using deep trench capacitors similar to those used for embedded DRAM technology. In the digital implementation, the area of the LIF circuits is about 26 μm^2, the learning circuits occupy ∼24 μm^2, and an additional 10 μm^2 is reserved for control circuits. Thus, the analog neuron area can be at least ten times smaller than the digital neuron area at the 10-nm node (see Table I).

The digital synapse circuits are implemented using SRAM (0.3 μm^2 at 10 nm), and the area of the programming and driver circuits is estimated, assuming effective array efficiency of about 70%. For the analog implementation, the synaptic element can be implemented in a crossbar array with a cell area of $4F^2$ ($F = 29$ nm) resulting in 220 μm^2 for the 256^2 synapses in a core. Each driver circuit may have to provide, at worst, 256 μA, and this results in a total area of about 774 μm^2 for the axon and dendrite drivers. Although the driver circuits are about seven times larger than the RRAM array itself, the overall area of the analog implementation is 16 times smaller compared with the digital implementation. The LUTs could be implemented using $4F^2$ RRAM devices programmed to store binary data, and hence, for a maximum fan-out of 1024, the total area needed for the LUTs per core is about 70 μm^2, which is a small fraction of the total core area. Since there is only one router (implemented in CMOS) in each core, the area of the router is also an insignificant fraction of the aforementioned areas.

TABLE II
ESTIMATES FOR POWER (IN MICROWATTS) AT THE 10-nm NODE FOR THE NEUROSYNAPTIC CORE

	256 Neurons	256^2 Synapses		Inter-core comm
		Comm	Learning	
Analog	3.84	27	190	0.5
Digital	29.4	23	54	1.8

B. Power Comparison

We limit ourselves to the active power consumed by the circuits (see Table II), as the threshold voltage of the transistors used in both implementations can be modified to reduce the leakage power. This is possible only because of the low switching speed requirement (in the megahertz range) that is unique to this particular application and the learning acceleration factor of 1000.

The active power of a 4-bit CMOS SRAM learning system at the 45-nm node was reported in [18] to be 234 nW/MHz. From this, we estimate the active power at the 10-nm node based on technology scaling (and $V_{dd} = 0.5$ V) to be 15 nW/MHz. On average, each neuron spikes every 100 μs and requires three cycles to process a spike, translating to about 770 cycles for the core in a time window of 100 μs. This leads to active power of about 115 nW per neuron (64 nW for the LIF function, 36 nW for learning, and 15 nW for control) in the digital implementation. For the analog implementation, based on the average current and neuron spiking rates, we calculate the active power to be on the order of 15 nW (4.5 nW for the integrator and 7.5 nW for the comparator). Thus, the subthreshold analog implementation results in at least a 7$\times$ reduction in active power for implementing the basic neuron function.

The digital neuron circuit implements learning by keeping track of the time that has elapsed since the last spike and using this to read and modify data stored in the SRAM. We estimate, based on bitline load capacitance, the average read and programming energy to be about 34 and 82 fJ per bitline—scaling this by the average number of update events per time step $(.01 \times 256^2)$ leads to 23 and 54 μW, respectively, for the communication and the learning power for the digital synapse.

For the analog implementation, the power for spike communication (equivalent to RRAM read when only the axonal waveform is present) and synaptic learning (equivalent to RRAM programming that results due to the overlap of axonal and dendritic waveforms) depends on the current magnitude and duration (approximately τ_1 in Fig. 5) through the driver circuits and the RRAM. By integrating the $V \times I$ product, we calculate the power per read and programming event to be about 41.2 and 290 fJ, respectively, assuming that the maximum programming current is 1 μA and $\tau_1 = 1$ μs. This translates to about 27 μW (read) and 190 μW (write) per core at the 10-nm node. The RRAM programming currents and durations assumed in this analysis result in the analog implementation consuming approximately four times higher power than the digital implementation.

Finally, the active power required to communicate spikes between cores is estimated based on the average hop distance between cores by assuming that each spike routing will require an average of five hops to go from the source to the destination core. The analog hop distance is about 57 μm, as compared with 200 μm in the digital implementation. The total power required for inter-core communication for the entire system is about 0.5 μW for the analog implementation, which is about three and a half times smaller than in the digital counterpart (1.8 μW).

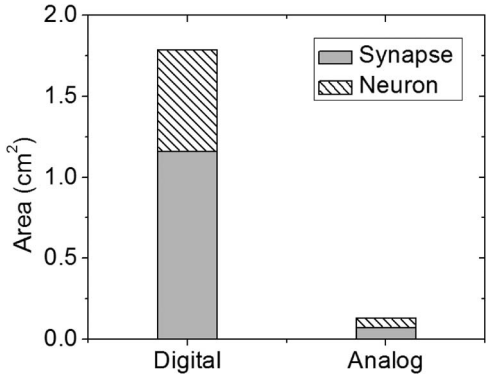


Fig. 10. Area comparison of the 1 million neuron learning system. The analog implementation could potentially lower the area by at least a factor of 10.

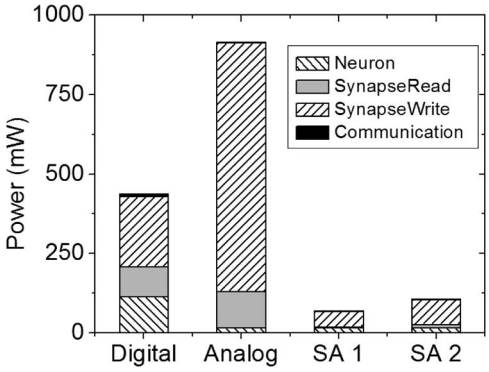


Fig. 11. Power comparison of the 1 million neuron learning system. The RRAM programming conditions assumed are Analog: 1 μA $\times$ 1 μs; Scaled Analog 1 (SA 1): 0.1 μA $\times$ 1 μs; and Scaled Analog 2 (SA 2): 1 μA $\times$ 0.1 μs.

C. Analog versus Digital Implementation: Conclusions

Having determined the area and the power for the component circuits, we can now compare the overall merits of the two approaches (see Figs. 10 and 11). The total area of the 1 million neuron analog learning system is about 14 times smaller than the area of the digital implementation, at the 10-nm node, whereas its power is approximately twice as high when compared with the digital implementation.

There are two approaches to reduce the power for the analog implementation without sacrificing its area advantage. One approach is to reduce the RRAM programming and read current requirement, and the other is to shorten the duration of the initial positive spike of the programming pulse. We estimate that a 10$\times$ reduction in the programming current magnitude of the RRAM will result in an overall system power reduction by a factor of 6 (denoted by Scaled Analog 1), whereas a 10$\times$ reduction in the programming current duration ($\tau_1 = 100$ ns) will result in an overall power reduction by a factor of 4 (denoted by Scaled Analog 2), as compared with the digital implementation.

VI. Conclusion

We have presented a scalable integration scheme for implementing neuromorphic learning systems based on electronic neurons and synapses that are inspired by biology. Our calculations show that there is a definite advantage in the overall system area by implementing these circuits using analog circuits and scaled RRAM devices. However, a surprising conclusion of our work is that an analog implementation, even with aggressively scaled RRAM devices that are being targeted for nonvolatile memory applications, consumes more power than the equivalent digital implementation at the 10-nm node. To obtain a clear advantage in terms of power in comparison with a purely digital implementation, RRAM devices that can be programmed to exhibit incremental conductance changes with maximum currents no more than 100 nA at voltages less than 0.5 V and times on the order of 100 ns are required. Neuromorphic systems based on such new nanoscale devices can, hence, potentially improve density and power consumption by at least a factor of 10, as compared with conventional CMOS implementations. This paper provides a quantified analysis of how developing nanoscale devices might benefit the implementations of components of neuromorphic learning systems. We show that there is a case for investment and research into new devices and architectures, which can provide significant area and power improvements for neuromorphic circuits over silicon CMOS at the end of scaling.

References

[1] G. J. Siegel, B. W. Agranoff, and R. W. Albers, Eds., *Basic Neurochemistry: Molecular, Cellular and Medical Aspects*. Philadelphia, PA: Lippincott-Raven, 1999.

[2] P. Lennie, "The cost of cortical computation," *Curr. Biol.*, vol. 13, no. 6, pp. 493–497, Mar. 2003.

[3] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: Taming the beast," *Nat. Neurosci.*, vol. 3, pp. 1178–1183, Nov. 2000.

[4] E. R. Kandel, *Nobel Lecture, Physiology or Medicine*, 2000. [Online]. Available: http://www.nobelprize.org/nobel_prizes/medicine/laureates/2000/kandel-lecture.pdf

[5] J. von Neumann, "First draft of a report on the EDVAC," *IEEE Ann. Hist. Comput.*, vol. 15, no. 4, pp. 27–75, Oct. 1993. [Online]. Available: http://dx.doi.org/10.1109/85.238389

[6] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[7] G. Indiveri and T. K. Horiuchi, *Frontiers in neuromorphic engineering*, vol. 5, no. 118, 2011. [Online]. Available: http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00118/fulltext

[8] A. K. Friesz, A. C. Parker, C. Zhou, K. Ryu, and J. M. Sanders, "A biomimetic carbon nanotube synapse circuit," in *Proc. BMES Annu. Fall Meeting*, 2007, pp. 1–5.

[9] H. Choi, H. Jung, J. Lee, J. Yoon, J. Park, D. J. Seong, W. Lee, M. Hasan, G.-Y. Jung, and H. Hwang, "An electrically modifiable synapse array of resistive switching memory," *Nanotechnology*, vol. 20, no. 34, pp. 345 201-1–345 201-5, Aug. 2009.

[10] S. D. Ha and S. Ramanathan, "Adaptive oxide electronics: A review," *J. Appl. Phys.*, vol. 110, no. 7, pp. 071101-1–071101-20, Oct. 2011. [Online]. Available: http://link.aip.org/link/?JAP/110/071101/1

[11] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, Apr. 2010.

[12] T. Hasegawa, T. Ohno, K. Terabe, T. Tsuruoka, T. Nakayama, J. K. Gimzewski, and M. Aono, "Learning abilities achieved by a single solid-state atomic switch," *Adv. Mater.*, vol. 22, no. 16, pp. 1831–1834, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1002/adma.200903680

[13] B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla, A. G. Schrott, R. S. Shenoy, B. N. Kurdi, C. H. Lam, and D. S. Modha, "Nano-scale electronic synapses using phase change devices," *ACM J. Emerg. Technol. Comput.*, to be published.

[14] Systems of Neuromorphic Adaptive Plastic Scalable Electronics—SyNAPSE. [Online]. Available: http://www.darpa.mil/dso/solicitations/baa08-28.html

[15] Framework Application for Core-Edge Transport Simulations Project—FACETS. [Online]. Available: http://facets.kip.uni-heidelberg.de

[16] A Universal Spiking Neural Network Architecture—SPINNAKER. [Online]. Available: http://apt.cs.man.ac.uk/projects/SpiNNaker/

[17] K. Hynna and K. Boahen, "Silicon neurons that burst when primed," in *Proc. IEEE ISCAS*, 2007, pp. 3363–3366.

[18] J. Seo, B. Brezzo, Y. Liu, B. Parker, S. Esser, R. Montoye, B. Rajendran, J. Tierno, L. Chang, and D. Modha, "A 45 nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE CICC*, 2011, pp. 1–4.

[19] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45 pJ per spike in 45 nm," in *Proc. IEEE CICC*, Sep. 2011, pp. 1–4.

[20] W. Maas, "Networks of spiking neurons: The third generation of neural network models," *Trans. Soc. Comput. Simul. Int.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[21] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Hafliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Front. Neurosci.*, vol. 5, no. 73, 2011. [Online]. Available: http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00073/abstract

[22] P. DSouza, S.-C. Liu, and R. H. R. Hahnloser, "Perceptron learning rule derived from spike-frequency adaptation and spike-time-dependent plasticity," *Proc. Nat. Acad. Sci.*, vol. 107, no. 10, pp. 4722–4727, Mar. 2010.

[23] B. Linares-Barranco and T. Serrano-Gotarredona, "Exploiting memristance in adaptive asynchronous spiking neuromorphic nanotechnology systems," in *Proc. 9th IEEE Conf. NANO*, Jul. 2009, pp. 601–604.

[24] D. Kuzum, R. G. D. Jeyasingh, B. Lee, and H.-S. P. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano Lett.*, vol. 12, no. 5, pp. 2179–2186, May 2012.

[25] G. Snider, "Spike-timing-dependent learning in memristive nanodevices," in *Proc. IEEE Int. Symp. NANOARCH*, Jun. 2008, pp. 85–92.

[26] H.-S. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. Chen, and M.-J. Tsai, "Metal oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.

[27] C. H. Cheng, A. Chin, and F. S. Yeh, "Novel ultra-low power RRAM with good endurance and retention," in *VLSI Symp. Tech. Dig.*, Jun. 2010, pp. 85–86.

[28] G. Burr, K. Virwani, R. Shenoy, A. Padilla, M. BrightSky, E. Joseph, M. Lofaro, A. Kellock, R. King, K. Nguyen, A. Bowers, M. Jurich, C. Rettner, B. Jackson, D. Bethune, R. Shelby, T. Topuria, N. Arellano, P. Rice, B. Kurdi, and K. Gopalakrishnan, "Large-scale (512 kbit) integration of multilayer-ready access-devices based on mixed-ionic–electronic-conduction (MIEC) at 100% yield," in *VLSI Symp. Tech. Dig.*, Jun. 2012, pp. 41–42.

Bipin Rajendran (S'01–M'07–SM'12) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2006.

He is currently an Assistant Professor with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, India.

Yong Liu received the Ph.D. degree from Harvard University, Cambridge, MA, in 2007.

He is currently with IBM T. J. Watson Research Center, Yorktown Heights, NY, where he is involved with the development of high-speed and low-power data links and emerging packaging technologies.

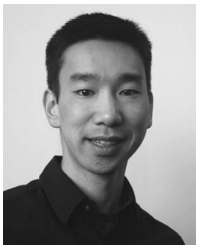
Jae-sun Seo (S'04–M'10) received the Ph.D. degree from the University of Michigan, Ann Arbor, in 2010.

He is currently a Research Staff Member with IBM T. J. Watson Research Center, Yorktown Heights, NY, working on energy-efficient ICs for high-performance processors and cognitive computing chips.

Daniel J. Friedman (S'91–M'92) received the Ph.D. degree from Harvard University, Cambridge, MA, in 1992.

He is currently with IBM T. J. Watson Research Center, Yorktown Heights, NY, where he has been the Manager of the Communication Circuits and Systems Group since 2009.

Kailash Gopalakrishnan He is currently with IBM T. J. Watson Research Center, Yorktown Heights, NY.



Leland Chang (S'99–M'03–SM'12) received the Ph.D. degree from the University of California, Berkeley, in 2003.

He is currently the Manager of the Design and Technology Solutions, IBM T. J. Watson Research Center, Yorktown Heights, NY, where his work focuses on power efficiency in high-performance systems.

Mark B. Ritter received the Ph.D. degree from Yale University, New Haven, CT, in 1987.

He is currently with IBM T. J. Watson Research Center, Yorktown Heights, NY, where he manages a group focusing on high-speed I/O subsystems.