INVITED PAPER

CMOS and Memristor-Based Neural Network Design for Position Detection

By Idongesit E. Ebong, *Member IEEE*, and Pinaki Mazumder, *Fellow IEEE*

ABSTRACT | Most hardware neural networks have a basic competitive learning rule on top of a more involved processing algorithm. This work highlights two basic learning rules/behavior: winner-take-all (WTA) and spike-timing-dependent plasticity (STDP). It also gives a design example implementing WTA combined with STDP in a position detector. A complementary metal–oxide–semiconductor (CMOS) and a memristor-MOS technology (MMOST) design simulation results are compared on the bases of power, area, and noise handling capabilities. Design and layout were done in 130-nm IBM process for CMOS, and the HSPICE model files for the process were used to simulate the CMOS part of the MMOST design. CMOS consumes 2.9×10^{-4} cm^2 area, 55-μW max power, and requires a 3-dB SNR. On the other hand, the MMOST design consumes 6×10^{-5} cm^2, 15-μW max power, and requires a 4.8-dB SNR. There is a potential to improve upon analog computing with the adoption of MMOST designs.

KEYWORDS | Neural network applications; neural networks; spike-timing-dependent plasticity (STDP); unsupervised learning; winner-take-all (WTA)

I. INTRODUCTION

Neuromorphic engineering is not a new approach to information processing systems. It particularly gained momentum in the 1980s with the amalgamation of learning rules and very large scale integration (VLSI) technology [1]. The growing transistor integration density in complementary metal–oxide–semiconductor (CMOS) enabled better simulation of neural systems in order to verify models and nurture new bio-inspired ideas. Since then, the neuromorphic landscape has changed and neuromorphic chips and programs are now available that cater to specific applications and tasks.

Technological advancement has always been both friend and foe to neuromorphic networks. Neuromorphic networks are essentially more valuable in instances where parallel computing is necessary. In order to perform neuromorphic computing effectively, a large number of processing elements (PE) are needed [1]. In current CMOS technology, the density and connectivity required for more sophisticated neuromorphic systems does not exist. This has led many neuromorphic chips to implement various schemes that utilize virtual connectivity between processing elements.

The shortcomings of CMOS in terms of density and parallel computing encouraged more complex neuromorphic system designs. Although design complexity increased, the number of neurons, synapses, and connections that can be simulated is orders of magnitude below the integration density of neurons in the human brain. Human beings, possessing neurons that operate in the millisecond range, can perform arbitrary image recognition tasks in tens to hundreds of milliseconds, while very powerful computers would take hours if not days to perform similar tasks. This lapse between digital computing and biology (specifically, the human brain) gives motivation for exploring technologies with connection densities that surpass anything CMOS can offer.

Manuscript received September 25, 2010; revised September 20, 2011; accepted October 5, 2011. This work was supported by HRL Laboratories and Defense Advanced Research Projects Agency (DARPA) (900279-BS).
The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: idong@umich.edu; mazum@eecs.umich.edu).
Digital Object Identifier: 10.1109/JPROC.2011.2173089

In addition to processing capability, reduction of power consumption also drives the development of neuromorphic circuitry, because CMOS approaches scaled to perform neuromorphic functions consume too much power. From a power perspective, the best neuromorphic hardware is no comparison against the human brain (weighing about 1.5 kg). The brain can handle driving during rush hour traffic with a power budget of about 20 W. IBM's state of the art supercomputer on the other hand, weighing in at 227 metric tons and taking up 5500 ft^2 of area [2], requires close to 3 MW to simulate a few seconds of rush hour driving. The amount of processing the human brain can perform in a short time with low power consumption compared to the CMOS digital computer shows that the neuromorphic paradigm is very much worth exploring, if not to expand the field of neurocomputing, then at least to help illuminate various methods that may be incorporated into the digital computing world to bridge the gap between processing capability, speed, and power requirements. The idea of a dedicated power plant for one brain simulator seems a waste of energy resources.

Low power and high device integration in nanotechnology have reignited a spark in the advancement of neuromorphic network in hardware as shown by Türel [3] and Zhao [4]. The "Crossnets" approach shown in [3] provides evidence of the design problems and methods of incorporation of resistive nanoscale devices in crossbar topology with CMOS circuitry to design neuromorphic circuitry. Nanotechnology, specifically memristors as postulated by Chua [5], shows much promise in this area because it may overcome the inability to reach densities found in biological systems. This inability is reduced by two factors: the first is the small size of the memristors with respect to their functionality, and the second is the ability to connect the memristors with crossbars. Connecting these nanodevices (memristors) with nanowires (crossbars) has been shown to increase device integration significantly [6]. Device integration in memristor-MOS technology (MMOST) is expected to improve in the age of memristors and crossbar scaling. A hypothetical study of a cortex-scale hardware performed in [7] shows that the use of nanodevices in a crossbar structure has the potential of implementing large-scale spiking neural systems. More complex algorithms like Bayesian inference [8] have also been studied for crossbar implementation, but these studies limit the crossbar array to digital storage. Analog use of the array would be ideal to reap its full benefits.

Neuromorphic networks derive their behavior from learning rules [9]. The networks have inherent governance that maintains relationships between neurons and synapses. Based on the myriad combinations of synaptic weights and neuron behavior, the network at any given point in time is unique. The focus of this paper will be on two functional blocks, commonly found in neuromorphic hardware implementations, used to determine these synaptic weights and neuronal outputs. Specifically, the two functional blocks are the winner-take-all (WTA) and coincidence/synchrony detection.

This paper presents two core blocks of neuromorphic computing algorithms prevalent in hardware implementations and contrasts a specific example of an MMOST design with a CMOS design. This work shows the advantages of choosing an MMOST implementation for a simple WTA architecture that utilizes spike-timing-dependent plasticity (STDP). A new method of realizing STDP with the crossbar structure is also presented.

II. NEUROMORPHIC BLOCKS

A. Winner-Take-All

WTA is an algorithm whereby one neuron clearly inhibits its neighbors in order to take the prize. This algorithm is ubiquitous in neural network design applications [10]–[14]. In addition, there exists a WTA variation, which allows for design flexibility, called the *k*-WTA. In *k*-WTA, two or more neurons might end up winning the prize. The concept remains the same: inhibit and disrupt the firing patterns of your neighbors in hope of spiking more than every other neuron.

The MMOST array may be used to implement WTA quite easily. Synapses can be modeled with memristors and network connectivity can be attained with the crossbars as shown in Fig. 1. The neurons, realized on CMOS, can connect to every other neuron through memristor synapses on the crossbar structure. This structure, as proposed for resistive memories, fits just as well for the WTA algorithm. A nonvolatile recurrent network implementing WTA can be built with MMOST to take advantage of the density that MMOST has to offer.

B. Coincidence Detection

Coincidence detection occurs when two spiking events are linked and coded for in a certain way. This algorithm is usually found in pattern recognition or classification systems whereby the neuromorphic network codes differently an input train of pulses or spikes. Based on the level of coincidence between different inputs to the network, the

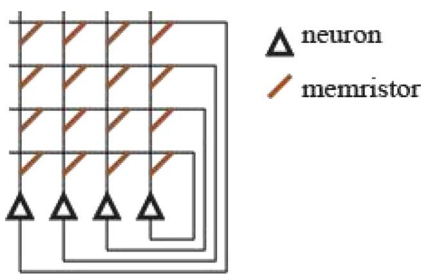


Fig. 1. *Recurrent network architecture showing an example of how a WTA network can be connected using crossbars.*

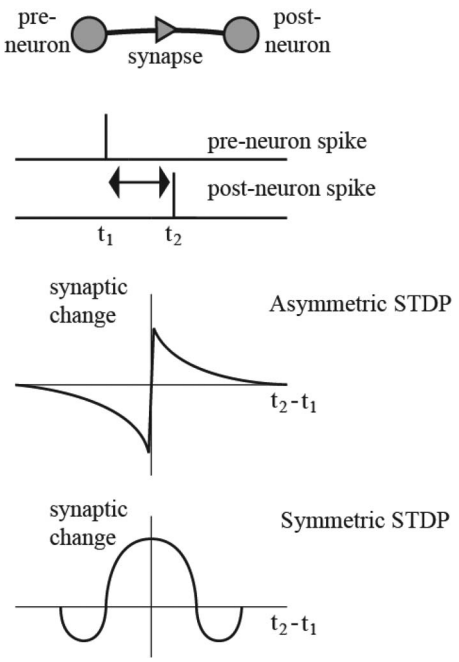


Fig. 2. *STDP curves showing relationship between synaptic weight change and the difference in spike times between the preneuron and the postneuron. Symmetric STDP and asymmetric STDP are both found in nature [19].*

neural network responds appropriately. This global network response is not the only way to use coincidence detection.

Another way to use coincidence detection is to update synaptic weights based on coincidence. This relates to the plasticity of the synapse and governs the learning rule of the synapse locally. In this form, the coincidence detection is known as STDP [15]. There are two main forms of STDP: symmetric STDP and asymmetric STDP (Fig. 2). Symmetric STDP performs the same weight adjustments independent of the spike order between the preneuron and the postneuron while asymmetric STDP reverses weight adjustment based on the spike time difference between the preneuron and the postneuron.

STDP implementations utilizing the crossbar structure have been proposed [16]–[18]. In their current state, they do not provide much density gains when comparing MMOST to CMOS. The implementations require pulse/signal generations in the positive and negative directions across the memristor. Snider [16] proposes a decaying pulse width while Linares-Barranco and Serrano-Gotarredona [17] and Afifi *et al.* [18] propose decaying signal amplitudes. All of them suggested that implementations rely on the additive effect of the signals across the memristor to control the synaptic weight changes. The STDP synaptic weight implementation in this work makes a linear approximation of the STDP curve in order to reduce the size of the neuron.

The proposed STDP implementations are usually of the form in Fig. 2. These synaptic behaviors, both asymmetric and symmetric, have been implemented in CMOS [20]–[22]. In the asymmetric STDP case, if the preneuron spikes before the postneuron, the synaptic weight is increased. If the order of spikes is reversed, the synaptic weight is decreased. In both cases, the larger is the duration between the preneuron and the postneuron spikes, the lesser is the magnitude of the synaptic change. Most circuit implementations take advantage of the asymmetric implementation.

The STDP implementation in this work is asymmetric and is based on the equation in the form of

$$\Delta W(t_2 - t_1) = \begin{cases} A_+ e^{-(t_2-t_1)/\tau_+}, & t_2 - t_1 > 0 \\ -A_- e^{(t_2-t_1)/\tau_-}, & t_2 - t_1 < 0. \end{cases} \quad (1)$$

The change in synaptic weight ΔW is dependent on spike time difference between the preneuron and the postneuron $t_2 - t_1$. A_+ is the maximum change in the positive direction, A_- is the maximum change in the negative direction, and both changes decay with time constants τ_+ and τ_-, respectively.

Most implementations use capacitors and weak inversion transistors to adjust τ_+ and τ_- in order to obtain decay times in the hundreds of milliseconds [23]. An alternate way to realize STDP in CMOS, when working under a lower area budget, is to incorporate digital storage units that can help remember spike states instead of using huge analog capacitors to set time constants.

The total change in weight for a given synapse is the summation of all positive and negative weight changes. Over the learning period, the synapse will converge to a certain weight value and will remain stable at that value. The STDP concept was tested through Verilog simulations whereby STDP was pitted against digital computation to do a comparison under noisy conditions.

The network of interest for simulation was that of a 1-D position detector whereby the location of an object is determined by the two-layered neural network presented in Fig. 3. The network consists of an input neuron layer (neurons labeled n_{11} through n_{15}) connected through feedforward excitatory synapses to an output neuron layer (neurons labeled n_{21} through n_{25}). At the output layer, each output neuron is connected to every other output neuron through inhibitory synapses.

The network shown in Fig. 3 updates its synaptic weights through STDP. Both excitatory (gray triangles) and inhibitory (red triangles) synaptic weights are modified through STDP. The inherent competition resulting when the output neurons spike help establish the weights for all 20 inhibitory synapses. An object is presented to the line of input neurons shown in Fig. 3. The object's presence generates signals that affect the closest neurons to its

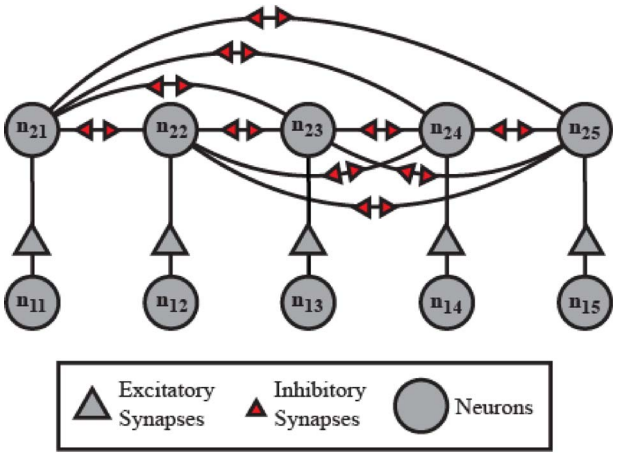


Fig. 3. ***Neural network implemented in Verilog in order to determine noisy performance of STDP in comparison to digital logic.***

position. For example, if the object is directly in front of n_{13}, then only n_{13} receives the object's generated signals, but if the object lies between n_{13} and n_{14}, then both n_{13} and n_{14} receive the input signals. The object's position was deciphered from the output neuron based on the relative spiking frequency (or period) of the output neurons.

The 1-D position detection was simulated for two noise conditions—noise-free condition and noisy condition—with different object locations. The noise-free case results are trivial. If there is no noise in the input of the system, then the output neuron results can be reduced to binary outputs—spike or no spike. For example, in the noise-free case, an object placed next to n_{13} causes n_{23} to spike while the other input or output neurons do not spike. In this noise-free case, the implementation of this position detection function could have been accomplished with digital logic where input signals exceeding some threshold would provide the desired output. In the noise-free case, when the object is placed between n_{12} and n_{13}, both n_{22} and n_{23} spike but the relationship between their spiking frequencies is proportional to the input object's exact location between both n_{12} and n_{13}. If the object is closer to n_{13}, then the spiking frequency of n_{23} is a little greater than n_{22}. The noise-free condition provides direct mapping of either a spike or a no spike with neurons involved in receiving the object's input and those not receiving the object's input. The noisy condition case is a bit more interesting, and the results are summarized in Table 1.

Table 1 provides results for the noisy case whereby all neurons in the output layer spike due to the noise background effect fed in through the input layer. The units in the simulation are time units or simulation time steps. Period is determined after weight stabilization has occurred and the time between successive spikes becomes fairly regular. The object's position can be determined in all three cases presented in the table. When the object is at

Table 1 Verilog STDP Output Neuron Results for an Object Placed at Different Locations on the 1-D Position Detection Line

Output Neurons	Period (time between successive spikes)		
	Object at n_{13}	Object between n_{12}&n_{13} but closer to n13	Object midway between n_{12}&n_{13}
n_{21}	1746	2046	1014
n_{22}	786	**684**	**660**
n_{23}	**636**	**642**	**660**
n_{24}	786	3030	1506
n_{25}	1746	7242	7266

n_{13}, n_{23} spiking period is the lowest (n_{23} is spiking the most). When the object is between n_{12} and n_{23} but closer to n_{13}, n_{23} spikes the most but its spiking period is comparable to n_{22}. A second level processing can compare these two neurons' spiking period to determine the object's location relative to the two neurons that spike the most. Last, when the object is exactly midway between n_{12} and n_{13}, then both n_{22} and n_{23} spike with the same spiking period.

An extension of these results may be used for motion detection. Looking at the spiking response of n_{23}, we may conclude that the spiking period decreases as the object moves away from n_{13}. The advantages therefore seen in using STDP are that by determining the object's position using the spiking frequency, the neural network can withstand the effects in a noisy background while digital threshold logic fails.

Two algorithms have been briefly described, and an application showing a WTA example with STDP plasticity in a 5 × 5 position array detector will be discussed in Section IV. Two position detectors—an MMOST version and a CMOS version—are simulated in SPICE in IBM 130-nm technology. Section III presents the memristor model used for the MMOST simulation.

III. MEMRISTOR MODELING

Many groups have shown memristive behavior in their devices [24]–[26], but only a few have characterized their devices for SPICE modeling and adaptation. The memristor model used in simulation is similar to that developed for the device made in HP Labs [27]–[30]. The nanodevice made in HP Labs has two layers TiO_2 thin film material. One of the thin film layers is doped with oxygen which reduces the resistivity of this layer, while the other layer is left undoped. The total resistance of the nanodevice depends on the resistance combination of both TiO_2 layers. The memristor's resistance (memristance) can be modulated by electrically biasing (current or voltage) the device. The current through the device moves oxygen dopants laterally, thereby widening (narrowing) the doped region depending on bias direction [28].

The memristor model of HP labs gives rise to a device whose resistance change is proportional to applied bias. If

applied bias is relatively low for a certain time span, then the change in memristance is very small and can be neglected. This idea allows for the establishment of a device threshold whereby the memristor's resistance is assumed to be unchanged when bias is below this threshold value. This memristor behavior is seen in Jo *et al.*'s a-Si memristor [31]. Jo *et al.*'s memristor shows conformity to the idea of a built-in threshold thereby allowing the authors to use different voltage biases for read/write interpretation. This memristor can withstand low current without resistance change, and this quality is important for the MMOST circuit design.

The memristor behavior already described allowed for the creation of a threshold-based SPICE model proportional to conductance change magnitude Δ_C that follows

$$\Delta_C = -M \times \sqrt[3]{(V_{ab} - V_{thp})(-V_{ab} - V_{thn})} + V_{\text{off}} \qquad (2)$$

where M is an amplitude correcting factor, V_{ab} is the applied bias across the terminals of the memristor, and V_{thp} and V_{thn} are both threshold voltages of the memristor with a positive and negative applied bias, respectively. V_{off} corrects and maintains a zero change with no applied bias. Equation (2) works really well for a symmetric device, and the simulation done in this work uses a device with the same magnitude in threshold voltage for both positive and negative directions. This threshold behavior, in conjunction with the linear-drift model presented in [30], is used to implement a memristor with threshold characteristics.

The memristor threshold model does not assume zero change below the applied threshold voltage. The change is minimal but not negligible to some above threshold changes as shown in a normalized plot of Δ_C versus V_{ab} in Fig. 4. In circuit design, depending on application, the voltage choices between read and write pulses will determine how the memristive device is used. The read pulse is chosen to not cause drastic change in memristance, while the write pulse is chosen to encourage higher levels of conductance change than the read pulse.

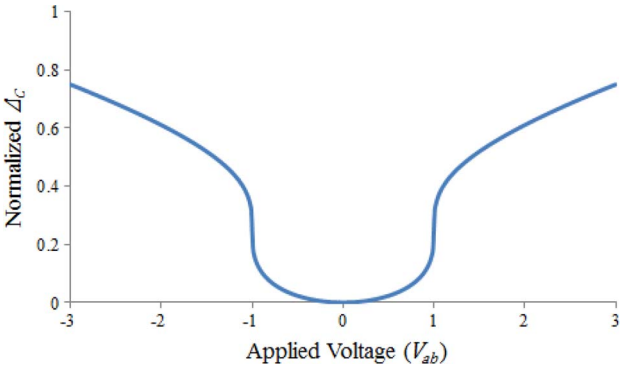


Fig. 4. *Normalized Δ_C versus V_{ab} showing proportional magnitude of conductance change as a function of applied bias. ± 1 V can be viewed as threshold voltages.*

Following the linear-drift model result (see [30] for derivation details), memristance as a function of flux is

$$M_T = R_0 \sqrt{1 - \frac{2 \cdot \eta \cdot \Delta R \cdot \phi(t)}{Q_0 R_0^2}} \qquad (3)$$

where M_T is the total memristance, R_0 is the initial resistance of the memristor, η can be viewed as memristor pin configuration ($+1$ for as is and -1 for switching the memristor polarity), ΔR is the memristor's resistive range (difference between maximum resistance and minimum resistance), $\phi(t)$ is the total flux through the device, and Q_0 is the charge required to pass through the memristor for dopant boundary to move a distance comparable to the device width. The memristance M_T cannot be larger than the maximum resistance or smaller than the minimum resistance of the device of interest.

For hand design purposes, it is useful to determine appropriate pulse widths and approximate memristance changes, because the change in memristance for each pulse is very important. The exact role of the thresholding factor Δ_C needs to be quantified. By taking the derivative of (3) with respect to $\phi(t)$, the approximation of the change of memristance is

$$\Delta M_T = \frac{-R_0 \cdot \eta \cdot \Delta R \cdot \Delta\phi / \left(Q_0 R_0^2\right)}{\sqrt{1 - 2 \cdot \eta \cdot \Delta R \cdot \phi(t) / (Q_0 R_0^2)}} . \Delta_C . \qquad (4)$$

Equation (4) shows us that for successive small changes in $\Delta\phi$ whereby $\phi(t)$ is not affected significantly, then the change in memristance ΔM_T will respond with almost constant step changes. The STDP design voltages for the MMOST design take advantage of this localized constant stepping for a range of $\phi(t)$ values. The concept is represented in Fig. 5 by graphing (3) with respect to $\phi(t)$.

The plot in Fig. 5 suggests an analog mode and a digital mode for the memristor. The modes of operation are strongly linked to the concept of localized constant stepping range previously discussed. In Fig. 5, the decrease in memristance seems nearly linear at first and then exponentially increases. The nearly linear part of operation is where we want the memristors to operate for the analog neural network functionality. In this region of operation, $\phi(t) <\sim 2.6$ Wb, the memristance decreases by about 2–3 MΩ in response to every 1-Wb change in $\phi(t)$. This operating region is a design choice to allow for better flexibility in choosing voltage levels and pulse widths. Designs that desire higher changes with respect to chosen

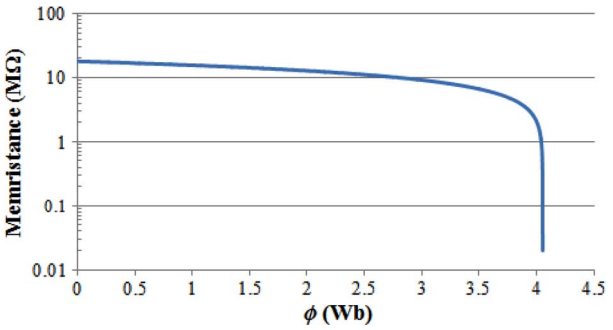


Fig. 5. *M_T versus ϕ showing two regions of operation for the memristor. In the slowly changing region, the magnitude of memristance change ranges from ~2 to 3 M for every 1-Wb flux change. The change in memristance increases drastically when ϕ is >~2.5 Wb. (Parameters used to simulate the analog memristor: $R_0 = 18$ M, $Q_0 = 5 \times 10^{-7}$ C, $\Delta R \approx 20$ M.)*

applied biases will most likely operate in the region closer to the digital device characteristics.

IV. POSITION DETECTOR APPLICATION

A. Architecture

Given a 2-D area, split up the area into a 5 × 5 grid (Fig. 6). Each square on the grid represents the resolution for the detector. A neuron resides at the center of each square on the grid. The detector has a 2-D layer of neurons. Each neuron is connected to its immediate neighbor through synapses. Each synaptic connection is unidirectional, so by having two connections, there is a bidirectional information flow between neighboring neurons. Each neuron is a leaky-integrate-and fire (LIF) neuron. Each has a leaky capacitor that stores integrated input information.

The architecture in Fig. 6 is an extension of the 1-D detector earlier discussed in Fig. 3. The difference is that the 1-D case had a fully connected output layer whereby all output neurons were connected to one another. The proposal in Fig. 6 however only has local connections between output neurons. This simplification was made because the CMOS neuron with STDP synapses would consume too much area. The simplification therefore limits the effectiveness of the detector to local detection, but local detection within the fabric does not invalidate the comparison between CMOS and MMOST made in this study.

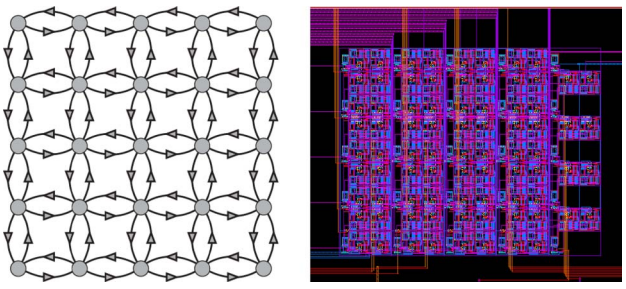


Fig. 6. *Neuron layer connectivity showing position detector architecture (circles are neurons and triangles are synapses). The left figure shows the connectivity matrix while the right figure shows the CMOS layout (190 μm × 152 μm).*

Table 2 Design Summary for Both Proposed WTA CMOS and MMOST 5 × 5 Position Detector Arrays

	CMOS	MMOST
Timing	Asynchronous	Clocked (1 kHz)
Power (Static, Dynamic max)	0.2μW, 55μW	5.28μW, 15.6μW
Chip Area	2.89×10^{-4} cm^2	6.1×10^{-5} cm^2
Input Noise (0.3V noise level)	> 3 dB SNR	> 4.8 dB SNR

Two design methodologies were taken in order to achieve STDP. The first is the CMOS design which is based on work with previous implementations in order to give a basis for the state of the art, and the second is the MMOST design used to specifically provide a new way of achieving STDP with area-conscious neuron design. The CMOS design will be explained briefly because the implementation is not exactly new, and the MMOST design decisions will be expanded upon to show that STDP really can be implemented in a way that does not consume too much area. Last, the comparison results will be explicated in context so apples are not compared to oranges due to different design decisions. The design summary is given in Table 2.

1) CMOS Design: The CMOS design has a LIF neuron with multiple inputs depending on the location within the position detection fabric. The neuron is inspired by designs with complimentary inputs, which has PMOS (pull-ups) for excitatory inputs and NMOS (pull-downs) for inhibitory inputs. Each neuron has only one pull-up and multiple pull-downs depending on the location in the position detector fabric, e.g., four pull-downs for neurons surrounded by four neighbors.

The STDP synapse approach is similar to those already presented in literature [20], [23] and the synapse schematic is shown in Fig. 7. When the preneuron spikes, S_{Pre} activates a switch that charges C_1. When S_{Pre} deactivates, C_1 discharges exponentially, but the capacitor C_{Weight} is not updated until there is a postneuron spike event. A postneuron spike event would activate S_{Post}, therefore allowing the evaluated output of the top comparator to see C_{Weight}. This explained sequence describes long-term potentiation (LTP). The postspiking before the prespiking would entail long-term depression (LTD). To reduce area, the capacitors C_1 and C_2 were implemented with diode connected NMOS transistors operating in weak inversion. The voltage range between V_{Charge} and V_{Q} is made to be about 100 mV. The decay shape of the voltages across C_1

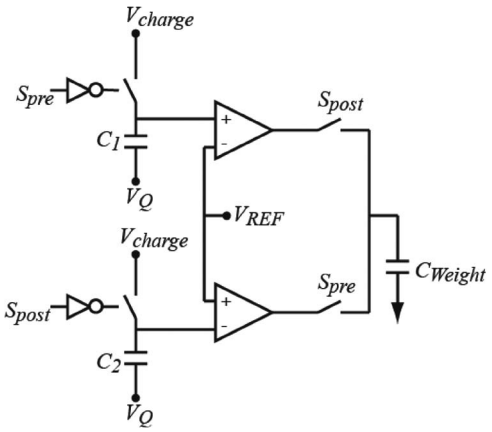


Fig. 7. *CMOS synapse block diagram used to perform STDP. The capacitors in this implementation take up the most area and control the STDP time window.*

and C_2 from V_{Charge} to V_Q is a function of the difference between V_{Charge} and V_Q. By reducing the voltage range, the decay seems to look more linear than exponential.

2) MMOST Design: The MMOST design will be delved in more detail than the CMOS design. The design goal is to take advantage of the memristor crossbar thereby simplifying the synapse and making it a fraction of the size of the CMOS synapse. The synapse itself is a simple memristor whose changes respond to pulses of equal widths provided through the neurons. Comparing the CMOS and MMOST designs, the STDP mechanism is moved from the synapse to the neuron.

The neuron design utilizes a new way of realizing STDP by striking a tradeoff between neuron area and asynchrony. The neuron implementation of STDP is depicted graphically in Fig. 8.

Fig. 8 shows the spike patterns between a preneuron's output and a postneuron's input (the memristor lies between these two terminals). In Fig. 8, the preneuron spikes right before time $t0$, so at time $t0$, the preneuron's output is at 0 V. The 0-V level is held for four clock cycles (from $t0$ to $t3$), and then pulses are allowed to pass for another four clock cycles (from $t4$ to $t8$). Afterwards, the preneuron's output rests at a reference voltage V_{REFX}. The postneuron's input exhibits a similar behavior as the preneuron's output, but instead of spiking before time $t0$, it spikes sometime in the interval from $t2$ to $t3$. The postneuron's input is pulled to 0 V at time $t3$, as opposed to time $t0$ as the preneuron's output.

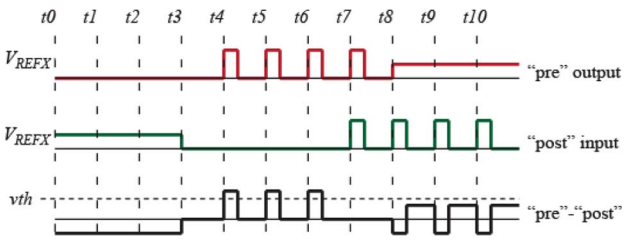


Fig. 8. *Preneuron and postneuron spiking diagram showing three pulses above the memristor's threshold. The below threshold pulses do not affect conductance.*

The preneuron's output and the postneuron's input spiking patterns present a difference across the memristor's terminals, and this difference is shown in Fig. 8 as "pre"–"post." As explained earlier, the utilized memristor is a threshold device, meaning its conductance experiences greater change when a voltage greater than its threshold voltage *vth* is met. The threshold is exceeded only by the three pulses shown in Fig. 8. The neuron circuit that can implement the spiking patterns depicted in Fig. 8 is shown in Fig. 9.

The neuron in Fig. 9 is composed of an integrate-and-fire circuitry, a path for passing an inhibitory current signal I_{in} to the integrate-and-fire circuitry (pass), paths for pulling the neuron's input and output nodes high (adj1), and paths for pulling both its inputs and output nodes low (adj2). The control signals (pass, adj1, and adj2) to turn each path on is controlled by the finite state machine (FSM) shown in Fig. 10.

In Fig. 10, **Start** is the default state—the neuron is not spiking, the neuron's input and output voltages are at reference voltage (V_{REFX}), pass is ON, adj1 is OFF, and adj2 is OFF. When the neuron receives excitatory inputs from the environment enough to cause a spike, then spike becomes 1, and in the next clock cycle, the neuron moves to the next state **Low**. In the **Low** state, both the input and output ports of the neuron are pulled to 0 V—the neuron has spiked, pass is OFF, adj1 is OFF, and adj2 is ON. The neuron stays in this state for four clock cycles (a counting variable increments from 0 to 3) before moving to the **Pulse** state. The **Pulse** state is the state where the neuron passes the external pulse to both its input and output ports—pass is OFF, adj1 is ON, and adj2 is OFF. In order to move from **Pulse** to **Start**, a counting mechanism is employed for four clock cycles. This internal FSM resides within each neuron. The simulation results that verify STDP using this scheme are presented in Fig. 11.

The connectivity matrix (or architecture) in Fig. 6 is therefore implemented for the MMOST design using this scheme. The triangles that signify synapses are memristors and the circles that signify neurons are implemented with Fig. 9. Section IV-B presents the simulation results with the comparison between CMOS and MMOST designs.

B. Results and Discussion

The CMOS design is an asynchronous design in which minor perturbations on a neuron's excitatory input can cause a spiking event. The MMOST design is a clocked design that synchronizes OFF-chip signals with the ON-chip logic. The MMOST design itself has asynchronous parts to

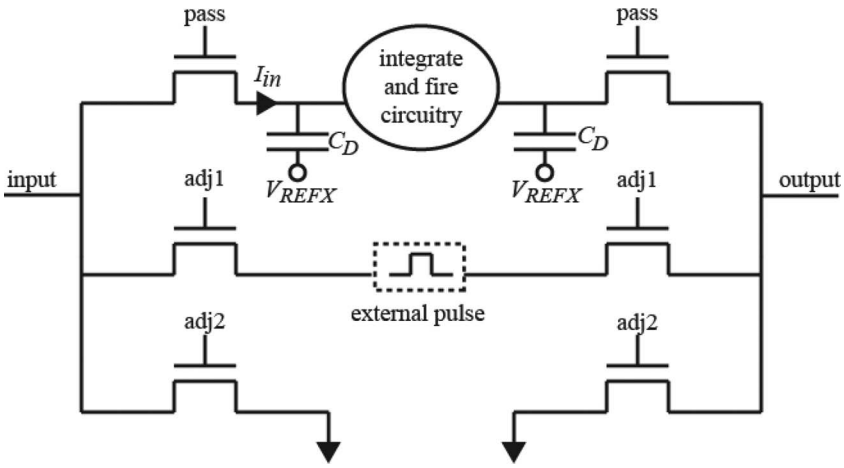


Fig. 9. *Neuron circuit that can provide spiking pattern shown in Fig. 8. The external pulse is an off-chip signal, and the switch signals (pass, adjx) are generated by an FSM.*

it (neuron integration and signal input), but the timing of change in resistance of the memristor is a synchronous event. The WTA algorithm allows for spiking neurons to inhibit one another while changing synaptic weights to strengthen or weaken the inhibition. The change of synaptic weight for both the CMOS and memristor or MMOST design qualifies as the ability for the chip to learn.

The advantage of choosing an STDP design is to capitalize on its noise handling capability. The lower is the noise level, the lower is the difference between signal and noise necessary for position detection. In comparing the CMOS and MMOST designs, the MMOST design has a higher potential because it consumes less area and requires less operating power. The quoted values in Table 2 for the MMOST design for both power and area are overestimations, and yet it still outperforms CMOS on these specs. This is without even considering potential synaptic and neuronal densities that can be achieved.

1) Design Complexity: For the current implementation, the timing of the CMOS circuitry is designed to perform STDP in the tens of microseconds range in order to conserve area. This value can be adjusted by using bigger capacitors (C_1 and C_2 in Fig. 7) to extend the time constant or by putting the synaptic transistors (those that implement switches and comparators) even more into subthreshold. The CMOS design can become very complex when trying to design for its most dismaying feature: volatility. Currently, when the stimulus is removed, the weight decays exponentially to its direct current (dc) steady state in about 100 ms, since synaptic weight is stored on capacitors. A better design would save these weights to memory and incorporate read, write, and restore schemes, which requires careful timing requirements.

The chip area (5×5 array) for the CMOS design is about 2.9×10^{-4} cm^2 from the CMOS layout, while that for the MMOST is about 6×10^{-5} cm^2. The memristor design area is an overestimation so it is likely to be much less than the proposed value. From design automation, the current logic for the memristor design is expected to take about 488 minimum sized transistors. Since this automated design was not simulated for signal integrity, drive, etc., for a worst case scenario, we double this value by two in order to account for various signal buffering, clock signal regeneration, and via spaces to the crossbar structure. This is a gross estimation, but it still shows that the memristor

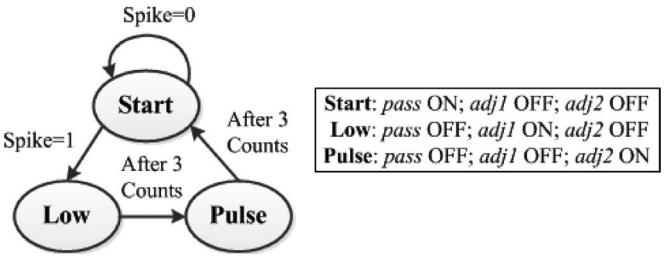


Fig. 10. *FSM showing control signal generation. The switch from low to pulse and back to home is determined by a counter circuit not described in this paper.*

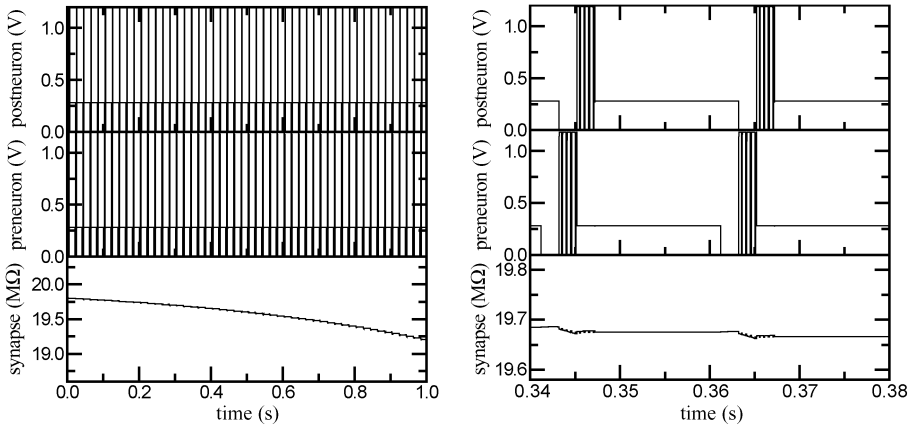


Fig. 11. ***Verification of the STDP scheme. (Left) After multiple preneuron to postneuron spiking, the synapse resistance decreases in value. (Right) Zoomed-in version showing multiple pulses and reference voltage.***

design consumes ∼5 times lower area than the CMOS design. This value can only improve because a custom design would use fewer transistors. The area estimation assumes that the crossbar array area will be fully contained over the CMOS area.

The area estimations, of course, are implementation dependent. The assumptions here are that both CMOS and MMOST detectors are operating without defects, the problems associated with accessing specific devices in the crossbar array are not addressed, and the crossbar array stacks fit nicely on top of the CMOS circuitry. These assumptions are implementation dependent because a denser connectivity for the MMOST case would mean a more complex connection scheme from CMOS neurons to the crossbar array. For the CMOS case, denser connectivity would mean much larger area (due to additional STDP synapses) and more complex connection scheme due to finite number of metal layers.

2) Power: The CMOS design consumes less static power than the memristor design mostly due to the fact that both designs are operating under different supply voltages (1 V for CMOS, 1.5 V for MMOST), and the memristor design has only a few transistors operating in the weak inversion region. The operating voltage difference is due to the fact that memristors will need to exceed a threshold voltage in order to change resistance, and the largest voltage across the memristor with under the 1.5-V power supply is about 0.9 V. The static power can be reduced for later generations of the design by having a lower voltage supply and using charge pumps to achieve required threshold voltages.

Although the static power consumption for CMOS is lower, its maximum dynamic power is higher than that of the memristor design. The memristor design consumes 15.6 μW while the CMOS design consumes 55 μW. The memristor logic and comparators take up most of the power due to heavy switching during spiking events. In the case of CMOS, as neurons begin to inhibit one another, they create or strengthen paths to ground allowing larger current draw especially when both excitatory and inhibitory inputs are activated. This current adds up pretty quickly as the array size increases.

3) Noise: Both the CMOS and memristor designs were tested with a jitter noise background between 0.1 and 0.3 V. The conclusion for testing under CMOS is that as noise level increases, the required signal level to counter this noise also increases. For example, at a noise level of 0.2 V, as long as the signal is at least 0.3 V, the neuron of interest will spike accordingly. This is a 100-mV difference between signal and noise. This value changes to 125 mV while the noise level increases to 0.3 V. In real-world computing, we do not expect the noise to be quite that high, but as long as the signal level is above 0.425 V, the neural network will work as designed.

For the memristor design, the noise level is actually used to randomly assert the memristors at different conductance states. Once the network is stabilized under a certain noise level, the signal input is capable of tuning the memristors around its signal level for the detecting purpose. The noise levels used for simulation are similar to that of the CMOS design (0.1, 0.2, and 0.3 V). At 0.3 V, as long as the input is about 200 mV greater than the noise level, then the signal is discernible.

V. CONCLUSION

We have explored the benefits of moving to an MMOST design for STDP circuit implementation on the bases of circuit area, power, and noise. The area considerations are implementation dependent but scaling to denser networks favor the MMOST design because a CMOS implementation will require more STDP synapses which greatly limit connectivity. The power considerations show a mixed

result because moving to synchronous STDP for the MMOST implementation may actually waste more power in the idle state than the CMOS implementation. Dynamic power numbers are better for MMOST so a more active circuit would take advantage of the MMOST design. The noise considerations show that both designs are comparable. This may change however with device scaling as both memristors and CMOS transistors become more susceptible to noise. ■

Acknowledgment

The authors would like to thank N. Garegrat for his assistance with the STDP Verilog simulations.

REFERENCES

[1] P. C. Treleaven, "Neurocomputers," *Neurocomputing*, vol. 1, no. 1, pp. 4–31, 1989.

[2] K. Koch*Roadrunner System Overview*," Los Alamos Nat. Lab., Albuquerque, NM.

[3] Ö. Türel, J. H. Lee, X. Ma, and K. K. Likharev, "Neuromorphic architectures for nanoelectronic circuits: Research articles," *Int. J. Circuit Theory Appl.*, vol. 32, no. 5, pp. 277–302, 2004.

[4] W. S. Zhao, G. Agnus, V. Derycke, A. Filoramo, J.-P. Bourgoin, and C. Gamrat, "Nanotube devices based crossbar architecture: Toward neuromorphic computing," *Nanotechnology*, vol. 21, no. 17, 2010175202.

[5] L. O. Chua, "Memristor—Missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.

[6] D. B. Strukov and R. S. Williams, "Four-dimensional address topology for circuits with stacked multilayer crossbar arrays," in *Proc. Nat. Acad. Sci. USA*, 2009, vol. 106, no. 48, pp. 20155–20158.

[7] M. S. Zaveri and D. Hammerstrom, "Performance/price estimates for cortex-scale hardware: A design space exploration," *Neural Netw.*, vol. 24, no. 3, pp. 291–304, 2010.

[8] M. S. Zaveri and D. Hammerstrom, "CMOL/CMOS Implementations of Bayesian polytree inference: Digital and mixed-signal architectures and performance/price," *IEEE Trans. Nanotechnol.*, vol. 9, no. 2, pp. 194–211, Mar. 2010.

[9] G. Cauwenberghs, *Neuromorphic Learning VLSI Systems: A Survey*. Norwell, MA: Kluwer, 1998.

[10] K. Urahama and T. Nagao, "K-winners-take-all circuit with O(N) complexity," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 776–778, May 1995.

[11] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, L. Shih-Chii, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45k neuron, 5M Synapse, 12G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[12] G. Indiveri, "A current-mode hysteretic winner-take-all network, with excitatory and inhibitory coupling," *Analog Integr. Circuits Signal Process.*, vol. 28, no. 3, pp. 279–291, 2001.

[13] P. O. Pouliquen, A. G. Andreou, and K. Strohbehn, "Winner-takes-all associative memory: A hamming distance vector quantizer," *Analog Integr. Circuits Signal Process.*, vol. 13, no. 1, pp. 211–222, 1997.

[14] V. A. Pedroni, "Inhibitory mechanism analysis of complexity O(N) MOS winner-take-all networks," *IEEE Trans. Circuits Syst. I, Fund. Theory Appl.*, vol. 42, no. 3, pp. 172–175, Mar. 1995.

[15] Y. Dan and M.-M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 23–30, 2004.

[16] G. S. Snider, "Spike-timing-dependent learning in memristivenanodevices," in *Proc. IEEE Int. Symp. Nanoscale Architectures*, 2008, pp. 85–92.

[17] B. Linares-Barranco and T. Serrano-Gotarredona, "Memristance can explain spike-time-dependent-plasticity in neural synapses," *Nature Precedings*. [Online]. Available: http://hdl.handle.net/10101/npre.2009.3010.1

[18] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nanocircuits," in *Proc. Eur. Conf. Circuit Theory Design*, 2009, pp. 563–566.

[19] C. Vassilis, C. Stuart, and P. G. Bruce, "A CA2 + dynamics model of the STDP symmetry-to-asymmetry transition in the CA1 pyramidal cell of the hippocampus," in *Proc. 18th Int. Conf. Artif. Neural Netw. II*, Prague, Czech Republic, 2008, pp. 627–635.

[20] H. Tanaka, T. Morie, and K. Aihara, "A CMOS spiking neural network circuit with symmetric/asymmetric STDP function," *IEICE Trans. Fund. Electron. Commun. Comput. Sci.*, vol. E92A, no. 7, pp. 1690–1698, Jul. 2009.

[21] G. M. Tovar, E. S. Fukuda, T. Asai, T. Hirose, and Y. Amemiya, "Analog CMOS circuits implementing neural segmentation model based on symmetric STDP learning," *Neural Information Processing*, Berlin, Germany: Springer-Verlag, 2008, pp. 117–126.

[22] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1296–1304, Sep. 2004.

[23] T. J. Koickal, A. Hamilton, S. L. Tan, J. A. Covington, J. W. Gardner, and T. C. Pearce, "Analog VLSI circuit implementation of an adaptive neuromorphic olfaction chip," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 60–73, Jan. 2007.

[24] A. Beck, J. G. Bednorz, C. Gerber, C. Rossel, and D. Widmer, "Reproducible switching effect in thin oxide films for memory applications," *Appl. Phys. Lett.*, vol. 77, no. 1, pp. 139–141, 2000.

[25] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," *Nature Mater.*, vol. 6, no. 11, pp. 833–840, 2007.

[26] J. Scott and L. Bozano, "Nonvolatile memory elements based on organic materials," *Adv. Mater.*, vol. 19, no. 11, pp. 1452–1463, 2007.

[27] Z. Biolek, D. Biolek, and V. Biolkova, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, Jun. 2009.

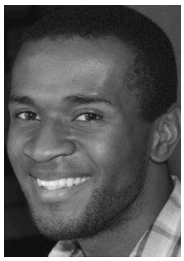
[28] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.

[29] D. B. Strukov, J. L. Borghetti, and R. S. Williams, "Coupled Ionic and electronic transport model of thin-film semiconductor memristive behavior," *Small*, vol. 5, no. 9, pp. 1058–1063, 2009.

[30] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, no. 4, pp. 661–675, Jul. 2009.

[31] S. H. Jo and W. Lu, "CMOS compatible nanoscale nonvolatile resistance, switching memory," *Nano Lett.*, vol. 8, no. 2, pp. 392–397, 2008.

ABOUT THE AUTHORS



Idongesit E. Ebong (Member, IEEE) received the B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2006. He is currently working towards the Ph.D. degree in electrical engineering at the University of Michigan, Ann Arbor.

His research interests include digital/analog integrated circuit design, focused primarily on new devices and low power applications.

Pinaki Mazumder (Fellow, IEEE) received the Ph.D. degree in computer engineering from the University of Illinois at Urbana-Champaign, Urbana, in 1988.

Currently, he is a Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. He has served as the lead Program Director of the Emerging Models and Technologies Program at the U.S. National Science Foundation, and worked for six years in industrial R&D centers that included AT&T Bell Laboratories, where in 1985 he started the CONES Project—the first C modeling-based very large scale integration (VLSI) synthesis tool. He has published over 200 technical papers and four books on various aspects of VLSI research works. His research interests include current problems in nanoscale CMOS VLSI design, computer-aided design (CAD) tools and circuit designs for emerging technologies including quantum MOS and resonant tunneling devices, semiconductor memory systems, and physical synthesis of VLSI chips.

Dr. Mazumder was a recipient of Digital's Incentives for Excellence Award, BF Goodrich National Collegiate Invention Award, and DARPA Research Excellence Award. He is an American Association for the Advancement of Science (AAAS) Fellow (2008) and an IEEE Fellow for his contributions to the field of VLSI.