Exploiting Local Connectivity of CMOL Architecture for Highly Parallel Orientation Selective Neuromorphic Chips

Melika Payvand, Luke Theogarajan
Department of Electrical and Computer Engineering
University of California Santa Barbara
Santa Barbara, CA, California
{melika01, ltheogar}@ece.ucsb.edu

***Abstract*—Biological neural networks exploit local connectivity to solve complex image recognition tasks. While CMOS scaling has enabled packing more transistors and functionality into a given area, connectivity still remains an unsolved problem. The vast interconnectedness required in a neural network further exacerbates this problem. Recently memristors have emerged as viable on-chip synaptic mimics. However, the two terminal nature of these devices requires a crossbar network to enable individual addressing, in turn precluding large connectivity domain required for neural networks. In this paper, we explore the use of large fan-in locally connected spiking silicon neurons readily available in CMOL architecture to solve edge recognition in images via unsupervised learning. We show the system level simulation of an edge classifying network using Simulink employing self-inhibition and Spike Timing Dependent Plasticity. Transistor level simulation of the system blocks in Cadence Spectre is also included. We derive the constraints on nanowire length given a particular choice of memristor implementation, resulting in a maximum kernel size.**

Keywords—CMOL; Neuromorphic Circuits; Image Processing; Memristor; Spiking Neural Networks; Local Receptive Field

I. INTRODUCTION

Hardware implementation of human brain has been the dream of circuit designers for many years now. Since the visual cortex is a very well-studied part of the brain, many different groups have worked on the neuromorphic implementation of this specific part of the brain with applications in pattern recognition, edge detection, etc. Typically, a first layer, namely V1, extracts orientation edges, the next layer combines these edges to more complex features and as we one goes further in the layers features combine to more and more complex shapes until an object is recognized [1]. This computation happens in the connections between the neuron cells. Synaptic connections change while learning the patterns and their learned state is remembered. Taking inspiration from biology, [2] and [3] use an event driven sensor chip connected to a processing chip through an AER bus. In these works off chip RAM memory is used to save the synaptic connections. As soon as an event is sensed, the appropriate connections are read from the off chip RAM and the corresponding computation is applied. In [4] the authors present an event-driven chip for edge enhancement through analog biases saved on in-pixel capacitors. Flash memory has also been used as a method to remember the state of the synapse as the injected charge in the floating gate of the flash memory. [5]–[7] show how current computation using flash memory can be an efficient way of doing neural computation.

The beauty of the neuronal computation however is the massive parallel processing which enables us as humans to recognize objects with a speed that no hardware has ever reached. A multi-chip solution was presented by Choi et. al [8] in order to detect the edge orientation of the input image. This design although interesting is not very area efficient since for each edge one chip is employed.

With the emergence of memristors as nano-devices which can be 3D integrated on top of CMOS chips [9]–[11], there is a new wave of creativity in the neuromorphic VLSI field to achieve recognition performance closer to the brain. [12] uses the idea of the 3D integrated memristors and Spike Timing Dependent Plasticity (STDP) [13] to emulate orientation selectivity in V1. Moreover, in a study done in HRL labs, 3D-integrated memristors are used as analog memory which will be read to change the state of the CMOS synapse. This work employs heavy time division multiplexing in order to save hardware and also for routing spikes through the chip [14]. Sheridan et al. in [15] used the STDP method and crossbar size of 784x10 to recognize MNIST handwritten images, however, the limiting factor is the size of the crossbar. The parasitic resistance and capacitances of the nanowire will limit programming and speed of the operation respectively. Strukov et al. in [16] suggested alleviating this problem by using CMOL architecture in which nanowires are segmented into fractions.

Taking further inspiration from biology, information representation in the form of time of arrival of the spikes has gained a lot of attention [17]–[19]. In this mechanism, namely rank order coding [17], the neurons which are activated more strongly fire sooner and the spike propagates through the network. For the very specific application of pattern recognition, one can say more intense image pixels cause the input neurons to fire sooner and this spike propagates to the next neuron layers. Unsupervised synaptic learning can be implemented by employing lateral inhibition and Spike Timing Dependent Plasticity. Repetition of the input pattern will therefore result in the convergence of the network to different clusters each of which containing similar input patterns [20].

This paper combines the idea of rank order coding and CMOL to realize a highly dense, fully parallel and potentially low power computing architecture to imitate the first layer of visual cortex (V1) in which cells become responsive to selective edges [1]. Since neurons have local receptive fields they can be easily mapped to the CMOL architecture. Exploiting the local connectivity of CMOL results in the following advantages:

1) The whole input image can be scanned in parallel and the salient orientation edges at each part of the image can be classified. 2) The computational complexity does not scale with the size of the image since all the computation is done locally and there is no need for a global controller. 3) A very large size neural network is not required and hence less complexity and shorter nanowires are needed. The desired edge kernels will be trained using a local and rather small neural network. We use a

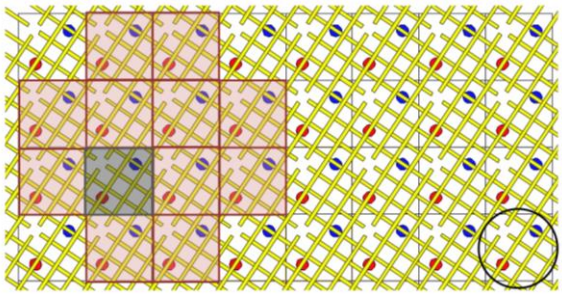

Fig. 1. CMOL concept. The circle highlights the CMOS cell. The pink cells are the connectivity domain of the gray cell. Figure is from [29].

feed forward network and employ Spike Timing Dependent Plasticity in order to change the memristive synapses locally in crossbars in an unsupervised manner. Accurate modeling of Memristor was enabled by using device models from University of Michigan's Pd/WO/W devices [21].

This paper is organized as follows: In section II we touch upon the background on rank order coding and also a brief explanation of CMOL architecture. Section III explains the mapping of an image processing problem to CMOL architecture and how we can use many rather small neural networks to do computation on a much bigger scale problem. Section IV describes the details of the transistor level design of these local neural networks. The network is also fully modeled in Simulink for system level simulations and the results of successful classification of edges are reported in part V.

II. Background

A. Rank order Coding

Spiking neural networks are inspired by brain computation and the way the neurons interact between each other in the brain. Spiking Neural Network (SNN) gains its computational power from time of spike firing. Because of their dynamic event-driven processing, they have the potential for a high memory capacity and fast adaptation. Thorpe et al. in [17] argue that the speed in which the visual processing happens in the brain suggests that the information is encoded in the time of spike rather than the rate of firing. Moreover, the implementation of spiking neural networks is intrinsically much more power efficient, since it's only using a single spike to convey the information, rather than multiple in the case of rate coding. When used in a feed-forward network employing shunting inhibition, the neuron which fires faster reduces the probability of firing from other neurons. This feature coupled with STDP to change the synapse's states enables each neuron to respond to a specific pattern. [22]

B. CMOL Architecture

The term "CMOL" comes from the combination of CMOS and Molecular scale devices and was conceived to mitigate the density scaling issues of CMOS. The main idea of CMOL is to construct 3D-integrated crossbars, i.e. mutually perpendicular layers of parallel nanowires (electrodes) forming two terminal, memristive devices at the cross-points. However, high- density memory requires long crossbars, causing excessive capacitances and resistances on the lines, limiting the speed and functionality of the system. By segmenting these long nanowires into shorter fragments, some of these issues are alleviated.

Exploiting the intrinsic nanoscale dimensions of this memory requires decoupling the underlying CMOS feature size from the device. One method of decoupling is to rotate the nanowires [16]. Shown in Fig. 1 is the CMOL concept. Red and Blue pins represent the area-distributed interface connecting the underlying CMOS to the integrated top and bottom crossbar nanowires, respectively. Each Red and Blue pair forms a so-called "CMOS cell," highlighted in Fig. 1. Every memristor provides a means of connecting one CMOS cell to surrounding ones (its "connectivity domain") via nanowire-memristor-nanowire path. There have been multiple attempts to fabricate memristors and rotated crossbars on the CMOL area-distributed-interface chips [10],[23],[11].

III. Edge Classifiction In CMOL

Our goal is to process an image and extract the salient edges at each part of it as is done in V1. Fig. 2a shows the idea of convolving four feature maps over an image, each of which is representing an edge as is shown in the figure. Each "neuron" on the feature map has a receptive field from which it receives the information. Neurons with the same coordinate on different feature maps have the same receptive field. In other words, the information from a certain receptive field goes to four neurons on different feature maps [1], [24]-[25]. The key architectural idea is processing of aggregated and localized information from the receptive field of each group of four neurons. In image processing terms, the receptive field translates to the kernel window scanned over the image. Fig. 2b shows the architectural realization of this idea. Assume an image is being scanned with a kernel of size 3x3. What we desire is the information of these 9 pixels (3x3) to go to 4 output neurons which are representing 4 classes of orientation edges. As is shown in Fig. 2b, the spikes travel from pixel numbers 1,2,3,6,7,8,11,12 and 13 to the 4 red output neurons in the middle. As we keep scanning the image with the kernels, 4 output neurons located in the middle of the 9 pixels collect the information. The neurons receiving the strongest weights*inputs will fire faster and as explained in the previous section it will inhibit all the other neurons.

Overlapping between different kernel windows is desirable and will result in shifting tolerance [25]. As a result of overlapping, some of the input neurons are connected to more than 1 output neuron. As is shown in Fig. 2c, pixel number 8 is connected to 36 output neurons which is the largest connectivity of any neuron in this system using a kernel size of 3x3. The huge "connectivity domain" of these neurons pointed us to the CMOL architecture. Fig. 3 depicts the same structure put together to look more like a standard CMOL structure. Every CMOS "unit cell" contains one presynaptic and four postsynaptic neurons. Transistor level design of the "unit cells" are described in section IV. In order to keep the CMOL structure as is mentioned in [16], we connect the four blue pins together as they are representing one presynaptic neuron. Four postsynaptic neurons shown in red pins remain the same.

Using short rotated nanowires will connect the presynaptic neuron (blue pin) in cell number 8, highlighted in yellow, to all the 36 postsynaptic neurons (red pins) in the yellow and gray cells. If we scale the size of the kernel by $n=P^2$, the connectivity domain will scale by 4n. Theoretically, the size of the connectivity domain can scale indefinitely. However there are limitations on the length of the crossbar, since the parasitic of the nanowires start dominating with continued scaling. We address this limitation in section IV.D.

IV. Neural network

Fig. 4a depicts the neural network used in this paper. Four output neurons are selected in order to classify the four desired edges of vertical, horizontal, 45 and 135 degrees as was discussed before. Training image pixels are mapped to input neurons with a 1-1 correspondence as is shown in the inset of Fig. 4a. Input neurons shown in blue circles send their spikes down the horizontal nanowire in each row and the spikes travel

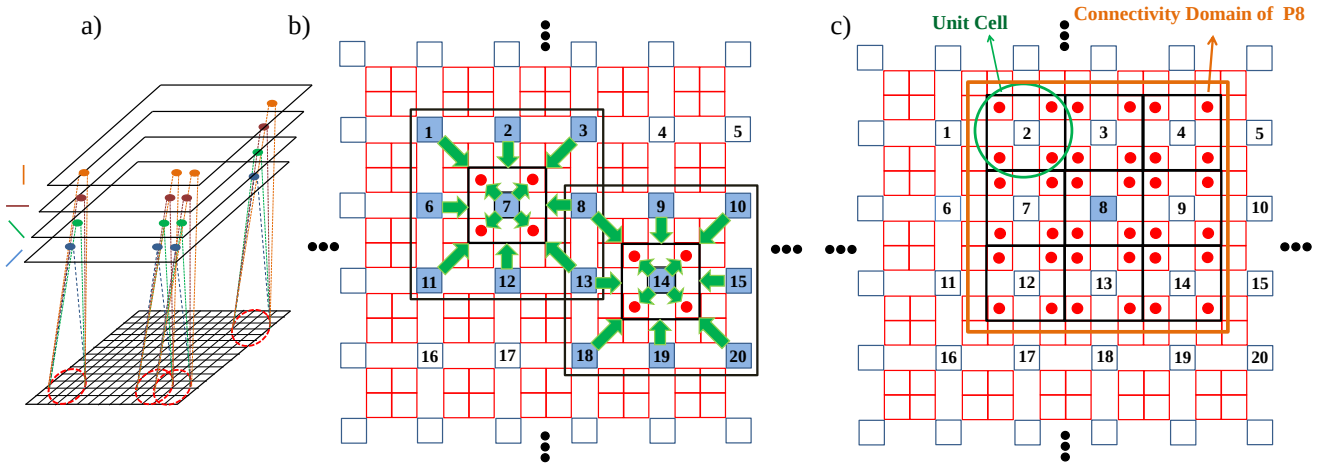

Fig. 2. Local processing of an image. a) scanning the image with the feature maps, each of which representing a specific salient feature. b) Scanning the image using a kernel size of 3x3. The pixels lying inside the kernel represent the receptive field of the four postsynaptic neurons shown in red and their information will gather to the four postsynaptic neurons in the middle. The image pixels are correspondent to the presynaptic neuron connecting to multiple postsynaptic neurons. c) The "connectivity domain" of the input pixel 8. One presynaptic and four postsynaptic neurons are the unit cell of the structure, which if repeated will construct the full image with postsynaptic neurons in the middle collecting the information of their receptive field.

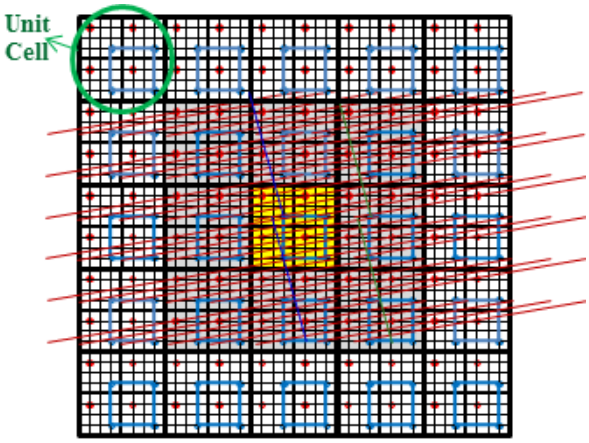

Fig. 3. Mapping the connectivity domain to CMOL architecture. The presynaptic neuron (Blue pin) inside the yellow CMOS cell is connected to all the postsynaptic neurons in the yellow and gray cells. This connectivity is enabled through the rotated crossbars passing through the red and blue pins. [29]

through the memristors in the crossbar joints. The output neurons shown in red circles at the bottom receive the sum of the spikes from their "receptive field" in each column. The weights of these memristors will slowly move towards the input patterns and when the network converges, they will determine to which edge each output neuron is selective.

Patterns are presented to the network every 1/(*Frame Rate*) seconds. The patterns' information is converted to time of arrival of spikes through the presynaptic neuron and is sent to the postsynaptic neurons via the memristive synapses. Within the time window of the 1/*Frame Rate*, the process on the presented pattern is done through one single spike per pixel and the weight changes are applied if needed. The network is then restarted with the arrival of the *Frame Rate* signal and waits for the next pattern.

Parts A, B and C explain the details of the transistor level design of the pre and postsynaptic neurons and the memristor crossbar used as the synaptic connections between them. Part D describes how to combine the pre and postsynaptic neurons in a modular CMOS "unit cell". This modular "unit cell" can be repeated to generate an image processing neural chip.

A. Presyanptic Neuron

Fig. 4c shows the schematic of the presynaptic neurons. The higher the intensity of the input pixels, the higher is the current charging the 1pF neuron capacitor (I_{image}). Therefore, the faster the membrane potential reaches the threshold voltage and the faster the input neuron fires. It is worth mentioning that the input capacitor does not discharge until the *Frame Rate* signal resets the value of the capacitor. This way, with every image presentation, only one spike per pixel gets generated. Each image is processed within the *Frame Rate* time period. We then use an SR latch in order to convert the information in the time of arrival of the spike to the pulse width. This will imitate the presence of neurotransmitter at the synaptic cleft upon firing of the presynaptic neuron, since the time constant of the neurotransmitters are much longer than the electrical pulse arriving at the synaptic cleft [26]. Using this method, the earlier the spike is generated, the wider the pulse width and hence the longer the postsynaptic neuron receives the signal. This will result in a faster firing of the postsynaptic neuron.

An Integrator-Inhibitor is employed as a self-inhibition scheme in order to reduce the effect of the "mutual information" of the input patterns. The mutual information appears in the form of commonly strong pixels between the randomized input patterns causing the corresponding presynaptic neurons to fire more frequently. For classification purposes unique information of each pattern is desired, thus it is useful for the network convergence to cancel the mutual information between the input patterns. A circuit realization for this idea is shown in Fig. 4b. Upon the firing of the neuron, Transistor M2 starts conducting and charge from current source M1 will get pumped to the capacitor C. The voltage developed across C results in more current ($I_{Inhibit}$) being taken away from the neuron capacitor. To prevent constant inhibition there is a slow leakage path for this charge through M3 (I_{leak}). Neuron firing and inhibition can be described mathematically as:

$$V_{inhibit} = \frac{1}{C}\int_0^t (I_{bias}\sum_{t_{fire}}\delta(t - t_{fire}) - I_{leak})dt \quad (1)$$

$$I_{inhibit} = k(V_{inhibit} - V_{th})^2 \quad (2)$$

$$C\frac{V_{th,neuron}}{t_{fire}} = I_{image} - I_{inhibit} \quad (3)$$

Equation (2) assumes that the inhibition current starts being dominant when transistor M4 enters the moderate inversion although it is even conducting before then, in the subthreshold regime. Fig. 5 a-d shows the frame rate signal, neuron's membrane potential, presynaptic spikes and also the pulse width converted signal. For this example we are assuming the image intensity is high in every cycle just to show how the inhibition

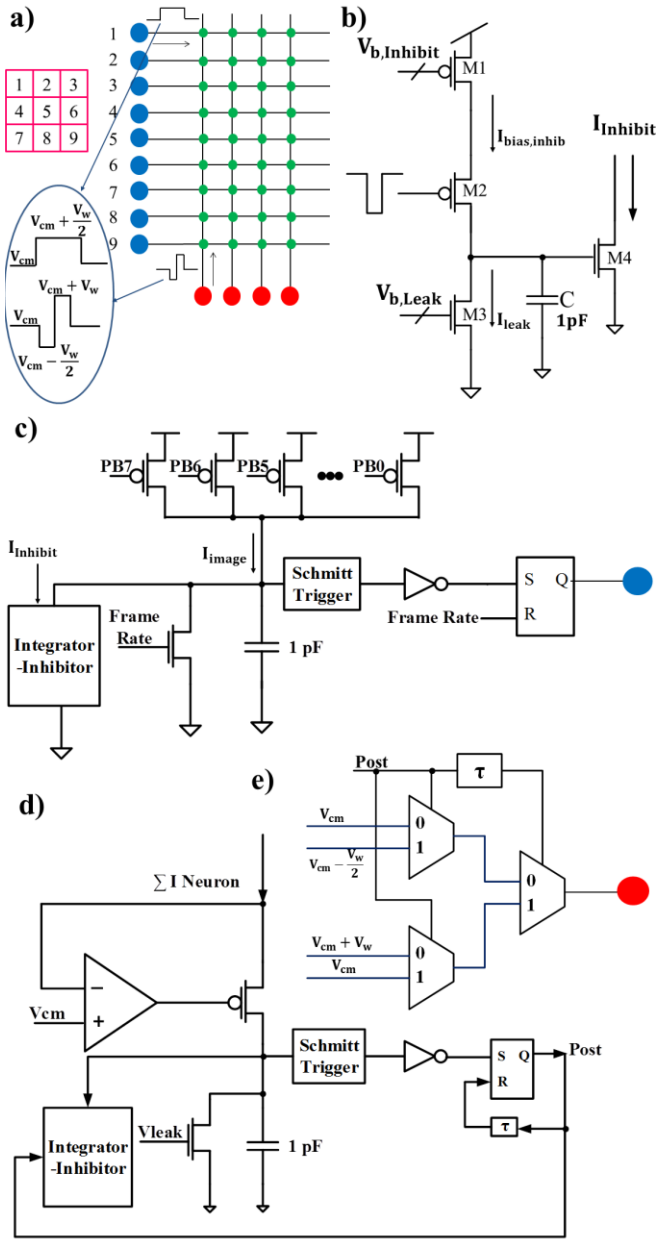

Fig. 4. Neural Circuit. a) Pre (blue pins) and Postsynaptic neurons (red pins) connected together through a memristive crossbar array. The pre and post synaptic spikes are shaped. b) Integrator-Inhibitor circuit. If spikes generate frequently $I_{inhibit}$ increases which will take out current from the neuron capacitor. c) Presynaptic neuron. 8 bit pixels are connected to the binary weighted width transistors to generate the pixel current which triggers the integrate-and-fire neuron. d) Postsynaptic neuron. Weighted input spikes are summed at the input of the postsynaptic neuron and triggers the integrate-and-fire neuron. e) Postsynaptic neuron pulse shaper.

happens. Fig. 5e illustrates the inhibiting voltage developing by the arrival of each spike and leaking away afterwards. It can be seen that as this voltage increases as a result of constant firing of the neuron, it self-inhibits the neuron and decreases the firing rate.

B. Postsynaptic Neuron

The same design is used for the postsynaptic neuron with the only difference that the input voltage to the neuron is pinned at V_{cm} in order to pin one side of the memristor while spikes are propagating in the network. Lateral inhibition on the postsynaptic neurons is employed as a winner-takes-all mechanism. Whichever neuron fires faster, discharges the other three neuron's input capacitor. An Integrator-Inhibitor is also used as a self-inhibition mechanism in order to avoid the network solution in which one postsynaptic neuron responds to all the input patterns. In this case, the postsynaptic spikes are integrating and inhibiting the current going into the neuron. Therefore, if a neuron responds to multiple patterns in a row, the system will automatically reduce the probability of the neuron firing for the next incoming patterns to give other postsynaptic neurons a chance to compete. The same circuit shown in Fig. 4b can be used to realize this mechanism at the postsynaptic neuron. Fig. 5 f-h shows the neuron's capacitor voltage, post synaptic spikes and the inhibition capacitor's voltage respectively. Similar to the presynaptic neuron, voltage development on the inhibitor capacitor decreases the probability of neuron firing.

C. Memristor Crossbar

Every time a presynaptic neuron fires, it generates a spike on the row corresponding to the neuron. All the spikes from the neurons in different rows go through the memristive synaptic connection at the crossbar joints and sum together as the current at the input of the postsynaptic neurons. The postsynaptic neuron sends back a dual polarity spike to facilitate STDP [27]. This dual polarity spike is shown in the inset of the Fig. 4a. If the presynaptic neuron has fired, the voltage across the memristor will be V_w for a short time and increase the synapse's state. This is desired since the pre and postsynaptic neurons are correlated and this correlation should be rewarded. If there is a spike from postsynaptic neuron while the presynaptic neuron has not fired yet, the voltage across the memristor will be $-V_w$ for a short time and the device's state will decrease because of the lack of correlation between the pre and post synaptic neurons. Fig. 4e shows the circuit which can generate this dual polarity pulse. As soon as the postsynaptic neuron fires the circuit will output $V_{cm} - \frac{V_w}{2}$ and it will keep this value until the post spike is on. The delay of τ is chosen to be same as the pulse width of the post signal. As soon as the post signal goes low, the second mux gets activated and outputs a voltage of $V_{cm}+V_w$. This will generate the desired pulse. In the absence of post signal this circuit always produces V_{cm}.

D. CMOS Pixel

As was discussed in section III, each CMOS unit cell (CMOS pixel) contains one presynaptic and four postsynaptic neurons. The estimated area of the CMOS pixel using the design explained in A and B in 0.13µm technology node is 120µmx120µm. Considering the nanowire dimensions mentioned in [21] and also the resistivity of tungsten the wire resistance per length is:

$$\frac{R}{L} = \frac{\rho}{\text{width} * \text{thickness}} = \frac{20 * 10^{-6}\ \Omega.\text{cm}}{130\text{nm} * 60\text{nm}} = 25\ \frac{\text{M}\Omega}{\text{m}}$$

It is worth mentioning that the tungsten resistivity used here is conservatively estimated to be three times its bulk resistivity, because of the surface and grain scattering effects. Moreover because of the topographic hills in crossbars the resistance will a bit higher than estimated in [28].

Assuming the maximum allowable parasitic resistance of the nanowire to be an order of magnitude less than the minimum resistance of the memristor (≈ 100kΩ), the maximum nanowire length will limit to 400µm. As is shown in Fig. 3, with a kernel size of 3x3, each nanowire is covering over 3 CMOS unit cells. With some simple calculations based on the estimated area of the CMOS pixel, the size of the nanowire will be about 350µm. Hence we have chosen a kernel of size 3x3 which will result in

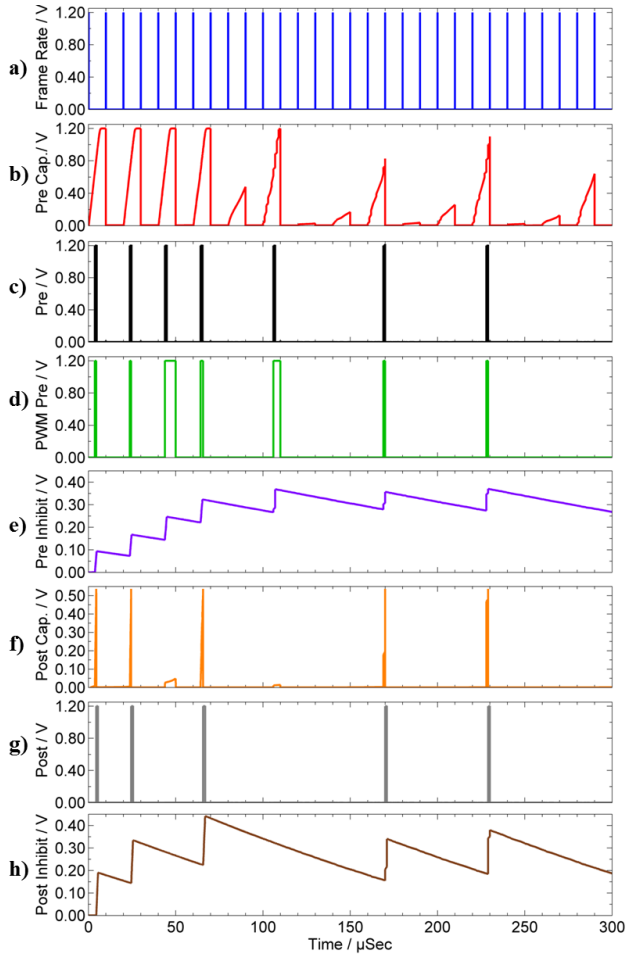

Fig. 5. Circuit results. Pre and post synaptic neuron spikes along with the integrate and inhibitor signals are shown.

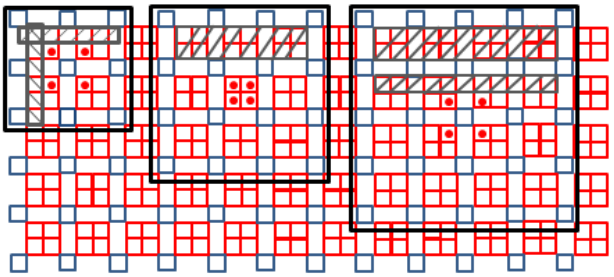

Fig. 6. Scanning the image with kernel sizes of 3,4,5. The shaded gray parts show the edge effect in which postsynaptic neurons are not needed.

a 9x4 neural network. By choosing a more advanced technology node this length can support a much larger connectivity domain since the size of each CMOS pixel would decrease. Also, by engineering the material and nanowire width and thickness, the resistivity of the nanowire could be lowered and therefore the nanowire could be chosen longer.

Although because of the reasons mentioned above we have chosen to work with a 3x3 kernel, it would be interesting to find the optimum kernel size for this image processing task. Scaling the size of the kernel results in a tradeoff between the area of the chip and the length of the nanowire. In order to explain this tradeoff let us consider the MNIST database handwritten digits. At the first glance, since these images have 28x28 pixels one would estimate the area as $28 * 28 * A_{CMOS\ Pixel}$. However, due to the topological properties of the CMOL architecture, the perimeter of the image undergoes an edge effect. Meaning that at the edges of the image, there is no need for the postsynaptic neurons in CMOS pixels and they only contain the presynaptic neuron. As the size of the kernel

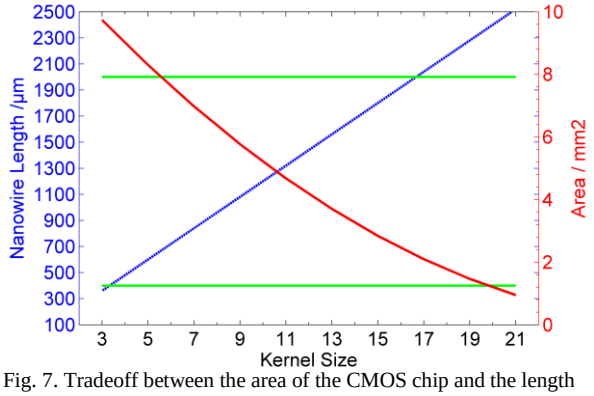

Fig. 7. Tradeoff between the area of the CMOS chip and the length of the nanowire with respect to the kernel size.

increases, this edge effect becomes more noticeable. This phenomenon is explained in Fig. 6. The shaded area shows the unused postsynaptic neurons which should be eliminated at the edges. As a result of the edge effect, the size of the CMOS chip for processing the MNIST database as a function of the kernel size PxP is:

$$A_{Chip} = (28 - P + 1) * (28 - P + 1) * A_{CMOS\ Pixel} + 4 * 27 * \frac{P-1}{2} * A_{prePixel} \quad (4)$$

The first part of the equation is the area of the middle of the chip, and the second part is the area of the pixels in the perimeter which only contain the presynaptic neuron.

In addition, as the size of the kernel grows the connectivity domain of the neurons increases. This will result in longer nanowires which is not desirable. The connectivity domain of the CMOS cells and the length of the nanowire are related as [29]:

$$M \simeq r^2 \quad (5)$$

In this equation r is a topological parameter representing the number of CMOS cells upon which a nanowire crosses. Parameter r can be estimated as the ratio of the nanowire length to the CMOS cell width. Moreover, the connectivity domain as a function of the kernel size P is $M=4P^2$. The nanowire length is therefore:

$$\text{Length} = \sqrt{M} * \text{CMOS Cell width} \quad (6)$$

However, the maximum length is limited by the parasitic resistance of the nanowire. Fig. 7 plots equations (1) and (3) with respect to the kernel size. Green lines show the maximum length limited to one tenth of the minimum memristance of memristors in [21] and [30]. The bottom green line is the limit on the memristor model used in this paper. As is shown in the figure a kernel size of 3 is the desired solution.

V. Simulation Results

The network has been modeled and simulated in Simulink. As we mentioned before memristor's model in [21] is used. Images are presented at the rate of 100 Hz (*Frame Rate* signal). Random initial memristor states are chosen using a Gaussian distribution with a mean weight of 0.2 and a standard deviation of 0.01. 5000 noisy patterns are generated using a MATLAB code and are randomly given to the network. Although memristors' initial conditions are very close, the network weights converge to the desired values. 3 different initial conditions with the same mean and standard deviations are examined and in all of them the output neurons get selective to the 4 orientation edges. Fig 8 shows the memristors weights after 18s of training. As can be seen in each "cluster", the weights corresponding to the unique pixels of the patterns become strong. Weight correspondent to the common pixel

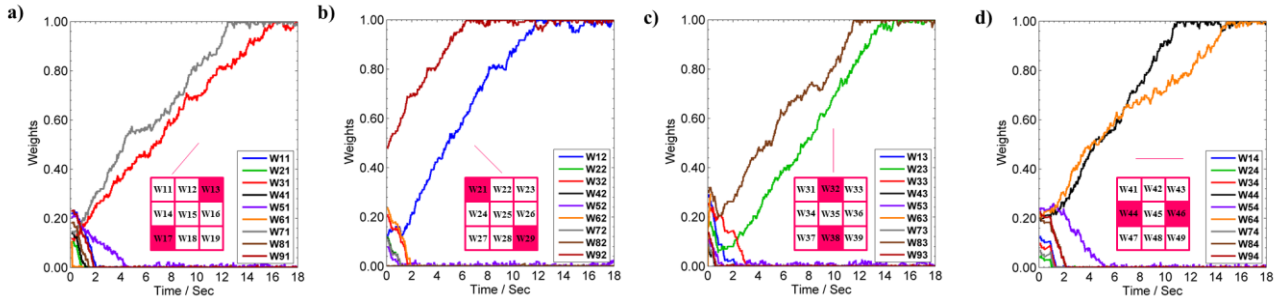

Fig. 8. Weights converging to the correct values in order to classify 4 different edges. a-d) Four classes of orientation edges are realized by the network. The mutual information between the patterns (Middle pixel, purple curve) is canceled. Wmn is the weight between input neuron m and output neuron n. The inset shows the state of the weights when the network is converged. In each cluster there are two strong weights which uniquely classify the orientation edge.

(Pixel 5 in the inset of Fig. 4a) will decrease after some time as the mutual information is getting cancelled out.

VI. Conclusion and Future Work

We have proposed a novel approach to classify orientation edges using CMOL architecture. This architectural idea can implement the first layer of convolutional neural networks for pattern recognition applications. This architecture enables massively parallel and high throughput advantages. Image processing is done locally and hence the computation complexity does not scale with the size of the problem.

Work is on-going to process the handwritten digits of MNIST database with an array of the pre and postsynaptic neurons arranged in the CMOL architecture as is discussed.

Acknowledgment

This work is funded by the Air Force Office of Scientific Research (AFOSR) under the MURI grant FA9550-12-1-0038.

References

[1] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognit.*, vol. 15, no. 6, pp. 455–469, 1982.

[2] L. Camuñas-Mesa, C. Zamarreño-Ramos, A. Linares-Barranco, A. J. Acosta-Jiménez, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven multi-kernel convolution processor module for event-driven vision sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 2, pp. 504–517, 2012.

[3] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike-based event processing vision systems," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 53, no. 12, pp. 2548–2566, 2006.

[4] P. Venier, A. Mortara, X. Arreguit, and E. a. Vittoz, "An integrated cortical layer for orientation enhancement," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 177–185, 1997.

[5] P. Hasler, B. A. Minch, and C. Diorio, "Adaptive circuits using pFET floating-gate devices," in *Proceedings 20th Anniversary Conference on Advanced Research in VLSI*, 1999, pp. 215–229.

[6] P. Hasler, A. G. Andreou, C. Diorio, B. A. Minch, and C. A. Mead, "Impact ionization and hot-electron injection derived consistently from Boltzmann transport," *VLSI Des.*, vol. 8, no. 1–4, pp. 455–461, 1998.

[7] R. R. Harrison, P. Hasler, and B. A. Minch, "Floating-gate CMOS analog memory cell array," in *ISCAS '98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (Cat. No.98CH36187)*, 1998, pp. 204–207.

[8] T. Y. W. Choi, P. a. Merolla, J. V. Arthur, K. a. Boahen, and B. E. Shi, "Neuromorphic implementation of orientation hypercolumns," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 52, no. 6, pp. 1049–1060, 2005.

[9] K. H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Lett.*, vol. 12, no. 1, pp. 389–395, 2012.

[10] M. Payvand, A. Madhavan, M. A. Lastras-montaño, A. Ghofrani, J. Rofeh, D. Strukov, and L. Theogarajan, "A Configurable CMOS Memory Platform for 3D- Integrated Memristors.", *ISCAS'15*, Libson, Portugal

[11] P. Lin, S. Pi, and Q. Xia, "3D integration of planar crossbar memristive devices with CMOS substrate", *Nanotechnology*,vol. 25, no. 40,2014

[12] C. Zamarreño-Ramos, L. a. Camuñas-Mesa, J. a. Perez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Front. Neurosci.*, vol. 5, no., pp. 1–22, 2011.

[13] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type.," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, 1998.

[14] J. M. Cruz-Albrecht, T. Derosier, and N. Srinivasa, "A scalable neural chip with synaptic electronics using CMOS integrated memristors.," *Nanotechnology*, vol. 24, no. 38, p. 384011, 2013.

[15] P. Sheridan, W. Ma, and W. Lu, "Pattern Recognition with Memristor Networks," *IEEE Int. Symp. Circuits Syst.*, pp. 1078–1081, 2014.

[16] K. K. Likharev and D. B. Strukov, "CMOL : Devices , Circuits , and Architectures." *Introd. Molecular Elect.*, vol. 680, pp. 447–77, 2005

[17] S. Thorpe, A. Delorme, and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural Networks*, vol. 14, no. 6–7, pp. 715–725, 2001.

[18] R. Guyonneau, R. VanRullen, and S. J. Thorpe, "Temporal codes and sparse representations: A key to understanding rapid processing in the visual system," *J. Physiol. Paris*, vol. 98, no. 4–6 SPEC. ISS., pp. 487–497, 2004.

[19] T. Masquelier and S. J. Thorpe, "Learning to recognize objects using waves of spikes and spike timing-dependent plasticity," *Proc. Int. Jt. Conf. Neural Networks*, 2010.

[20] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, pp. 0247–0257, 2007.

[21] T. Chang, S. H. Jo, K. H. Kim, P. Sheridan, S. Gaba, and W. Lu, "Synaptic behaviors and modeling of a metal oxide memristive device," *Appl. Phys. A Mater. Sci. Process.*, vol. 102, no. 4, pp. 857–863, 2011.

[22] L. Vanneschi and E. Systems, *Handbook of Natural Computing*. 2012.

[23] J. Rofeh, A. Sodhi, M. Payvand, M. A. Lastras-monaño, A. Ghofrani, A. Madhavan, S. Yemenicioglu, K. Cheng, and L. Theogarajan, "Vertical Integration of Memristors onto Foundry CMOS Dies using Wafer-Scale Integration," *ECTC*, 2015 IEEE 65th., in press

[24] K. Fukushima, "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements," *IEEE Trans. Syst. Sci. Cybern.*, vol. 5, no. 4, 1969.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] M. Häusser, N. Spruston, and G. J. Stuart, "Diversity and dynamics of dendritic signaling.," *Science*, vol. 290, no. 5492, pp. 739–744, 2000.

[27] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," *Proc. Int. Jt. Conf. Neural Networks*, pp. 1775–1781, 2011.

[28] D. Choi, C. S. Kim, D. Naveh, S. Chung, A. P. Warren, N. T. Nuhfer, M. F. Toney, K. R. Coffey, and K. Barmak, "Electron mean free path of tungsten and the electrical resistivity of epitaxial (110) tungsten films," *Phys. Rev. B - Condens. Matter Mater. Phys.*, vol. 86, no. 4, 2012.

[29] D. B. Strukov and K. K. Likharev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," Nanotechnology, vol. 16, no. 6, pp. 888–900, 2005.

[30] 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, 2010.