# Vector-Matrix Multiply and Winner-Take-All as an Analog Classifier

Shubha Ramakrishnan, *Student Member, IEEE,* and Jennifer Hasler, *Senior Member, IEEE*

***Abstract*— The vector-matrix multiply and winner-take-all structure is presented as a general-purpose, low-power, compact, programmable classifier architecture that is capable of greater computation than a one-layer neural network, and equivalent to a two-layer perceptron. The classifier generates event outputs and is suitable for integration with event-driven systems. The main sources of mismatch, temperature dependence, and methods for compensation are discussed. We present measured data from simple linear and nonlinear classifier structures on a 0.35-$\mu$m chip and analyze the power and computing efficiency for scaled structures.**

***Index Terms*— Analog computing, classifiers, computing efficiency, reconfigurable.**

## I. Efficient Analog Classifiers

ENERGY efficiency is a key concern in information processing in low-power smart sensors and mobile devices [1]. A typical information processing chain usually involves a refinement stage that reduces the processing load on the following stages. In embedded systems that receive sensory inputs, process and classify them to take decisions, it is essential to take a low-power approach for enabling such structures in robots and other mobile platforms. Classifiers are typically used in the information refinement stage and it is often essential that besides being low power, they also produce very few events. Events are generated when a certain class has been detected, triggering further circuitry dependent on this decision.

In highly integrated systems, an increased number of events often leads to increased power consumption, which is required to transmit events over interconnects between blocks that have significant capacitances. This strategy of minimizing the number of events is observed in biology, where the nervous system processes several sensory inputs and refines the information before transmitting them along large distances. The high power efficiency of the nervous system observed in biological organisms is achieved by maintaining a low rate of spiking in the neurons, which is on average 100 Hz or less [2]. In the past, significant effort in hardware classifiers has been through the rise of the artificial neural network (ANN) community since the 1980s, which solidified a framework of neural models that resulted in a variety of techniques to solve problems in many applications. Many of these techniques are considered standard and taught in most universities. The neural network (NN) approach has its early roots in the perceptron [3] and adaptive filter models [4] that then extend to multilevel network models, Hopfield models, and other related computational approaches. A simple NN has inputs being multiplied by a weight vector, added together at the soma compartment, where a linear or nonlinear function is applied to generate the output. ANN approaches include having continuous-valued (e.g., tan$h$) functions that approximate the spike frequency versus current input (f-I) characteristic of neurons with an analog voltage, or spiking (integrate-and-fire neurons, rate-encoded neurons), feedforward or feedback stages.

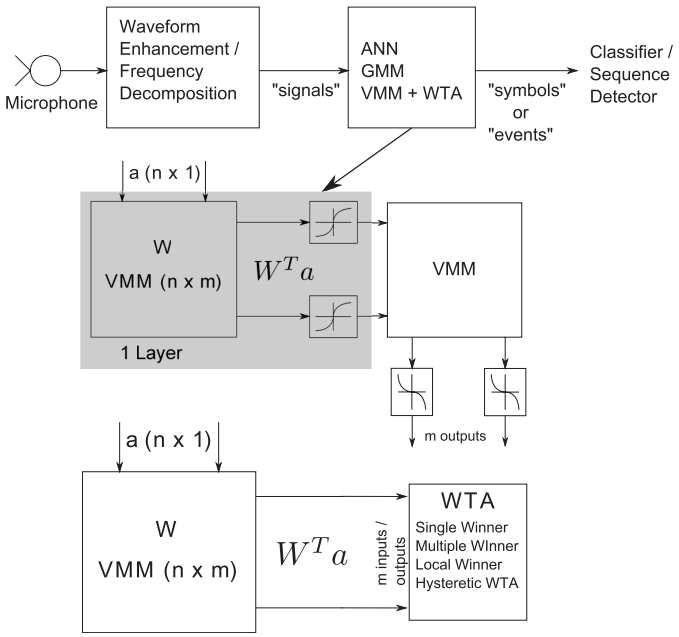


Fig. 1. Application in analog speech recognizer chain. Speech input undergoes frequency decomposition or enhancement resulting in sub-band signals. These signals undergo first-level information refinement in the feature detection stage, resulting in a sparse "symbols" or "event" representation. The following stage detects sequences of symbols/events to identify words or syllables. The feature detect stage may be implemented as an ANN, GMM, or a VMM + WTA classifier. A typical two-layer NN has synaptic inputs represented by the VMM and the sigmoid modeling the soma of a point-neuron. Alternatively, we can have synaptic computation followed by a competitive network modeled by the WTA. We investigate computational advantages to using the VMM + WTA over the ANN/GMM approach.

In this paper, we consider an analog classifier consisting of a vector-matrix multiply (VMM) terminated with a winner-take-all (WTA), shown in Fig. 1, that is versatile and has more computing power than a one-layer NN. The VMM block performs a multiply operation between a vector and a matrix of weights, resulting in a vector and forming a core component of many signal processing algorithms. The VMM + WTA, which we use as the base classifier, compares

Manuscript received April 19, 2012; revised October 5, 2012; accepted December 2, 2012.

The authors are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-250 USA (e-mail: shubha@gatech.edu; jennifer.hasler@ece.gatech.edu).

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/TVLSI.2013.2245351

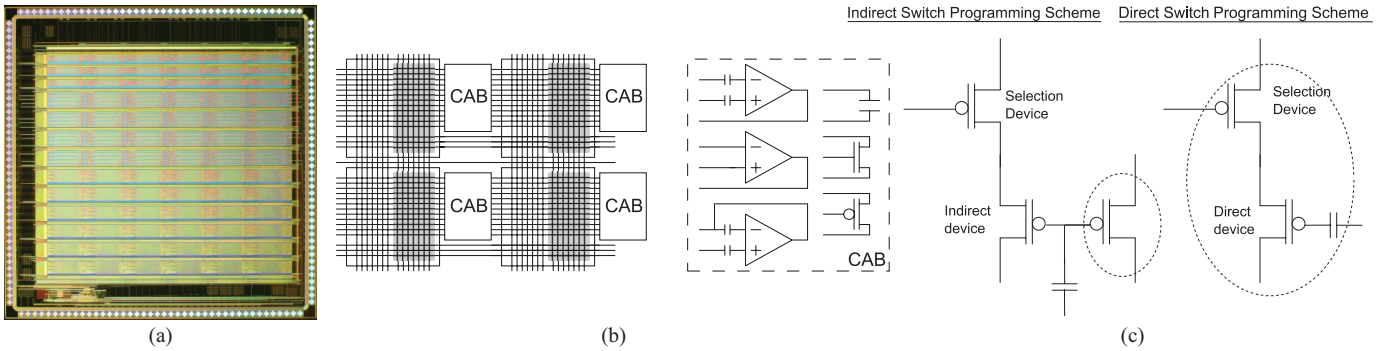


Fig. 2. (a) Field programmable analog array. The FPAA used in this paper consists of 78 CABs embedded in reprogrammable routing enabled by floating-gate switches [8]. Each CAB consists of capacitors, transistors, and OTAs that have programmable bias currents. Some OTAs have floating-gate inputs that allow cancellation of input offsets. (b) Two types of routing elements. (c) Switch programming schemes. The device(s) within the dashed circle appear in the signal path, while other devices are used for programming and selection. The indirect programming scheme minimizes parasitic in the signal path by using a separate device that shares the floating gate with the actual device. The selection device is required for isolation. The indirect scheme can result in inaccuracies due to a mismatch between the programmed device and actual device, but can be characterized. The direct scheme, where the programmed device and actual device are the same, requires no additional characterization. However, there is an extra selection device in the signal path, which reduces switch conductance at low voltages.

favorably against the one-layer NN in terms of the number of components as well. We show a direct translation of a one-layer NN to a VMM + WTA, where the WTA acts as a current comparator. In a different formulation, the WTA can perform an analog max function, selecting the largest (or smallest) of its inputs. With minor modifications, the WTA can be designed to allow multiple winners, local winners, or exhibit hysteresis [5]–[7], leading to classifiers that allow multiple winners with spatial responses, which can be useful in image processing, or exhibit hysteresis, which makes the classifier immune to noisy inputs.

We see this structure being used in an analog speech recognizer as shown in Fig. 1. The speech input undergoes frequency-decomposition or signal-enhancement in the front-end, resulting in input features, such as sub-band energies. These signal inputs are transformed into symbols or events with ANN, Gaussian mixture model (GMM), or VMM + WTA in the first stage of information refinement. This can be followed by higher level refinement or by a sequencing block to detect syllables or words.

This paper is organized as follows. We briefly discuss the computational efficiency and circuit complexity comparisons of VMM + WTA versus NN implementations in Section II. In Section III, we describe the hardware platform used for implementing our classifiers. Next, in Section IV, we discuss the multiple-winner WTA circuit. In Section V, we describe our VMM implementation, which is more compact and has lower noise and power than the previously described VMMs. In Section VI, we present measured results from classifier circuits that integrate the VMM and WTA to yield linear, multiclass, and nonlinear classifier systems. Finally, we discuss mismatch, computing efficiency, and temperature effects in Section VII.

## II. Implementation and Efficiency Overview

A one-layer NN requires the computation of a VMM + neuron. The addition of various weighted inputs is achieved through Kirchoff's current law (KCL) at the soma node. We define synaptic computation as the multiplication of inputs with synaptic weights, and neuron computation as a nonlinear threshold function. Assuming we have $n$ synapses per neuron and $m$ neurons, we expect a complexity of $m * n$ for synaptic computation. The computation at the neuron is governed by the choice of complexity in the model. A simple neuron model [$\tanh(.)$] would require four multiply-accumulate (MAC) units per neuron computation, as seen from a Taylor series expansion with four terms

$$\begin{aligned}\tanh(x) &\approx x - \frac{x^3}{x} + \frac{x^5}{5} - \frac{x^7}{7} \\ &\approx x\left(1 - \frac{x^2}{3}\left(1 - \frac{3}{5}x^2\left(1 - \frac{5}{7}x^2\right)\right)\right). \quad (1)\end{aligned}$$

Usually, for a moderate size of $n$, the synaptic computation dominates the neuron computation. The VMM + WTA classifier topology has the advantage of being highly dense and low power. Each multiply is performed by one single transistor that stores the weight as well, and each WTA unit has only two transistors, providing very high circuit density. Custom analog VMMs have been shown to be 1000× more power efficient than commercial digital implementations [9]. The nonvolatile weights for the multiplier can be programmed allowing flexibility. The transistors performing multiplication are biased in a deep sub-threshold regime of operation, resulting in high computing efficiency. We combine these advantages of VMMs with the reconfigurability offered by field-programmable analog array (FPAA) platforms to develop simple classifier structures. VMMs on FPAA with high power efficiency have already been demonstrated in core signal processing applications viz. Image transforms and orthogonal frequency-division multiplexing receivers [10], [11]. In this paper, we discuss the computing power of the VMM + WTA classifier, and show that we can implement any two-layer perceptron with modifications to the WTA.

## III. Hardware: FPAA Implementation

The hardware platform used for implementing the classifier is among the family of FPAA chips specifically geared toward building large VMMs. A detailed description of this chip and its features can be found in [8]. However, for the sake of completeness, we provide a short discussion on the architecture of this chip.

FPAAs have the general structure of a computational analog block (CAB) with routing infrastructure to make reprogrammable connections between the components. The CAB

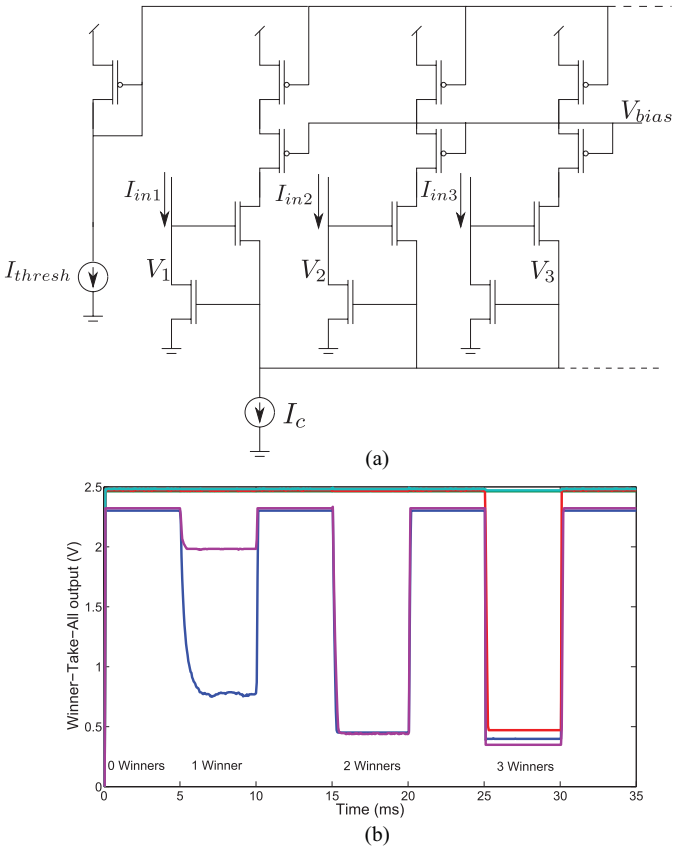


Fig. 3. (a) Traditional WTA modified to a $k$-WTA with a current threshold at each output, realized using a cascaded pFET. The current flowing through the winning branch is constrained, allowing other inputs to the WTA to win. The voltage outputs from the WTA are inverted and a node wins when its output is below mid-rail. (b) WTA with 0, 1, 2, and 3 winners. The choice of current threshold determines the number of winners.

consists of circuit blocks commonly used in analog design, as shown in Fig. 2. The re-programmability is enabled using floating-gate transistors that can be programmed ON or OFF by operations known as injection and tunneling, respectively, similar to programming Electrically Erasable Programmable Read-Only Memories. The programming infrastructure that includes selecting specific switches and injecting them is integrated on-chip, as discussed in [12].

## IV. Winner Take All

WTA networks of neurons were an early area in which VLSI and neuroscience positively interacted with each other, providing a unique and efficient means of computation. The classic circuit in [5] was based on continuous-valued approximations to neurons, and utilized transistor device physics to build an efficient circuit. Later, others built multiple spike-based representations to complete the connection between the analog VLSI approach and biological computation [13]. Several modifications to this circuit exist that allow local winners, hysteresis behavior that stabilize the outputs, temporary winners that fatigue after a period of winning and allow other inputs to win and multiple winners [6], [14], [15]. The CAB components in the FPAA support several WTA implementations, but we implement the classic WTA from [5].

### A. Multiple Winners

Often, we require classifiers that generate not just one output, but multiple outputs. In pattern classification, we can expect the classifier to indicate that a certain pattern matches two categories instead of just one. The WTA circuit does not preclude multiple winners and this can be achieved by modifying the circuit in [5], as shown in Fig. 3. For a $k$-WTA, or a WTA with $k$ winners, we use the current outputs from the WTA and apply a current threshold at the output. The modified implementation is shown in Fig. 3(a). A current threshold $I_{\text{thresh}}$ is mirrored to each of the current outputs from the WTA. By constraining the current in the winning branch, we allow other inputs of the WTA to continue winning after the first winner. The choice of $I_{\text{thresh}}$ determines the number of winners. For $k$ winners, the relation between $I_{\text{thresh}}$ and $I_c$ is given by

$$\frac{I_c}{k+1} \leq I_{\text{thresh}} < \frac{I_c}{k}. \tag{2}$$

The distribution of input currents also determines the number of winners for a fixed $I_{\text{thresh}}$. When the inputs are close to each other, $I_{\text{thresh}}$ needs to be closer to $I_c/(k+1)$ than $I_c/k$. The value of $I_{\text{thresh}}$ required to guarantee $k$ winners is given by the lower limit of (2). Fig. 3(b) shows the measured results from a five-input WTA, with different current thresholds to obtain multiple winners. The cascading p-type field-effect transistor (pFET), devices were inserted to improve the Early voltage of the current threshold, thereby constraining the current through the winning branches to $I_{\text{thresh}}$ more effectively than if cascades were not present. The $k$-WTA produces inverted voltage outputs that are taken at the drain of the thresholding pFET. Compared to the $k$-WTA circuit in [14], this implementation does not require any additional power/circuitry.

## V. Compact VMM Implementations

VMMs can be implemented in a power-efficient and compact manner using floating gates. The multiplication weights are stored as charge on the floating node and can be precisely programmed and controlled. The weight can be expressed as

$$w = e^{\kappa Q/C_T U_T} \tag{3}$$

where $Q$ is the charge programmed on the floating-gate node and $C_T$ is the total effective capacitance seen at the floating node. A single floating gate stores the weight as well as performs a multiply function. The programming accuracy of the VMM weights has been well characterized and in one application, has been shown to be 1.5% accurate in [8]. Examples of the different VMM topologies that we can implement are discussed in [9].

For a VMM with voltage inputs, we require a structure shown in Fig. 4. This structure converts a voltage input linearly into a current, using an operational transconductance amplifier (OTA) as a transconductance stage. The current is log-compressed on the source terminal using a logarithmic transimpedance amplifier and broadcast. Hence, the voltage input into the VMM and the weight of the input is encoded in the source voltage using the transconductance amplifier and logarithmic transimpedance amplifier. The signal-conditioning block that maps the input voltage to a broadcast source voltage is shown in the dashed box in Fig. 4.

We note that the number of OTAs required in the signal-conditioning block scales linearly with inputs, and is $2n$ for single-quadrant multipliers, and $4n$ for four-quadrant

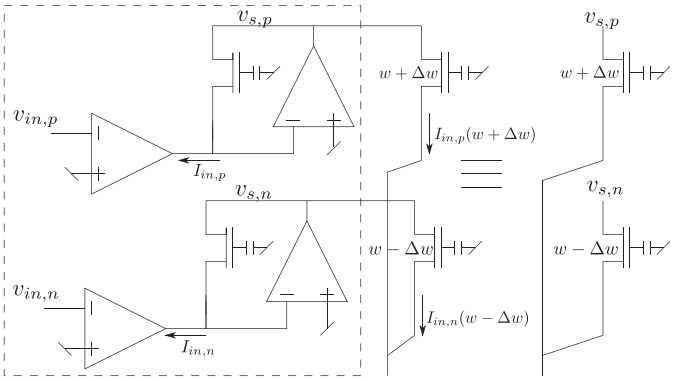


Fig. 4. Equivalence between VMM topologies (see Appendix). The VMM structure shown on the left transforms voltage inputs into currents using a transimpedance stage. The multiplication is achieved in current mode using a source-driven floating-gate current mirror where the weights are a result of differences in charge programmed on the floating gates. $v_{\text{in},p} - v_{\text{in},n}$ is the differential input to the VMM. The structure shown on the right shows a source-driven VMM where the input signal is applied directly to the source of the floating gates. $v_{s,p} - v_{s,n}$ is the differential input to the source-driven VMM.

multipliers, where $n$ is the number of inputs. Hence, for classifiers with a large number of inputs, a significant portion of the power budget is spent in the signal-conditioning block. In addition to the power overhead due to the amplifiers, we see effects of mismatch and noise added on the inputs. The main sources of mismatch are input offsets in the V-I, mismatch between bias currents of the V-I, and input offsets in the I-V. Cancellation of these effects often requires a lengthy characterization process. In this paper, we choose a source-driven VMM topology as shown in Fig. 4 to build low-power compact structures that minimize the added noise and mismatch effects by eliminating 4 OTAs per VMM input. Here, the voltage inputs are directly applied to the source terminal of the weighted current sources. We derive the equivalence between the two topologies in the Appendix.

We assume that the inputs to the classifier are from the set $\{x : |x| \leq 1\}$, which is reasonable for normalized inputs. It can be shown that for small $x$, there is a linear relation between the differential inputs for the two different VMM topologies shown in Fig. 4

$$v_{\text{in},p} - v_{\text{in},n} \propto v_{s,p} - v_{s,n}. \tag{4}$$

The differential input at the source is a compressed linear representation of the inputs to the V-I, and the attenuation factor is inversely related to the input linear range of the transconductance stage. The two voltage inputs for different values of $x$ are plotted in Fig. 5. The equivalence of the two structures in Fig. 4 shows that we can achieve compact VMM structures using just the routing infrastructure in the FPAA. From (4), the voltage inputs to the VMM can be applied directly to the source of the FG transistors. We note that the differential voltage inputs to the source-driven VMM need to be constrained to a range of $2\ U_T \approx 50$ mV for linear operation of the VMM. The stage driving the VMM also needs to supply the current required for the VMM. The output current can be expressed as

$$I_{\text{out}} = I_{\text{bias}}(2w + x\,\Delta w) \tag{5}$$

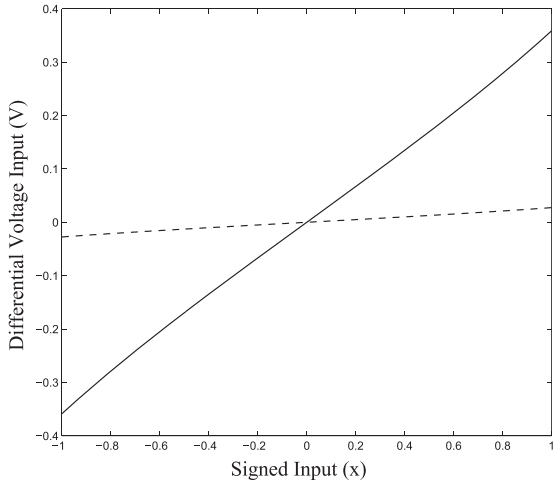


Fig. 5. The differential input voltages for the two topologies versus signed input $x$, as calculated in (24). $x$ is the normalized input to the classifier. We notice that the input for the source-driven topology (dashed) is a compressed form of the input for the standard VMM topology (solid). In this simulation, we use $\kappa_{\text{eff}} = 0.1$.

## VI. Capability of VMM + WTA Classifiers

We now integrate the VMM and WTA circuits to build simple classifier structures. In this section, we first describe measured results from system compilations of linear, multi-class, and nonlinear classification problems.

### A. Linear Classifiers

We start by considering a perceptron, which is a simple linear classifier with a binary output that can be implemented with a one-layer NN. A linearly separable set of inputs can be classified using a perceptron trained to weights $w_i$ and bias $b$ with the equation

$$z = \begin{cases} 1, & \text{if } \sum_i w_i x_i - b \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

A VMM + WTA classifier can be trained as a generalized single-layer perceptron by using a fixed current source as an additional bias input to the WTA, shown in Fig. 6. The WTA functions as a current comparator and detects the largest input. When $\sum_i w_i x_i > b$, the first input wins. By using a 1-WTA circuit implemented with the current threshold at the WTA output, we obtain inverted voltage outputs. Hence, the first output is low when $\sum_i w_i x_i > b$ and high otherwise.

We measured results from two different linear classifier boundaries programmed on the VMM + WTA circuit, for multiple bias values. For a linear decision boundary, we train a perceptron using MATLAB's NN toolbox and apply the weight and bias values directly to the VMM + WTA classifier. We restricted ourselves to a two-input case for ease of visualization. The structure in Fig. 6(a) only supports positive values for the bias. Since our implementation required signed weights and bias values, we chose a topology with fully differential inputs. The classifier was tested over all inputs from the set $\{(x, y) : |x| \leq 0.8, |y| \leq 0.8\}$. We plot the inverted WTA voltage output in Fig. 6(c) and (d). The output makes a sharp transition at the desired decision boundary, which is marked by the solid line in the plots. Since our VMM implementation consisted of directly programmed floating-gate transistors, we were able to directly apply the weights obtained

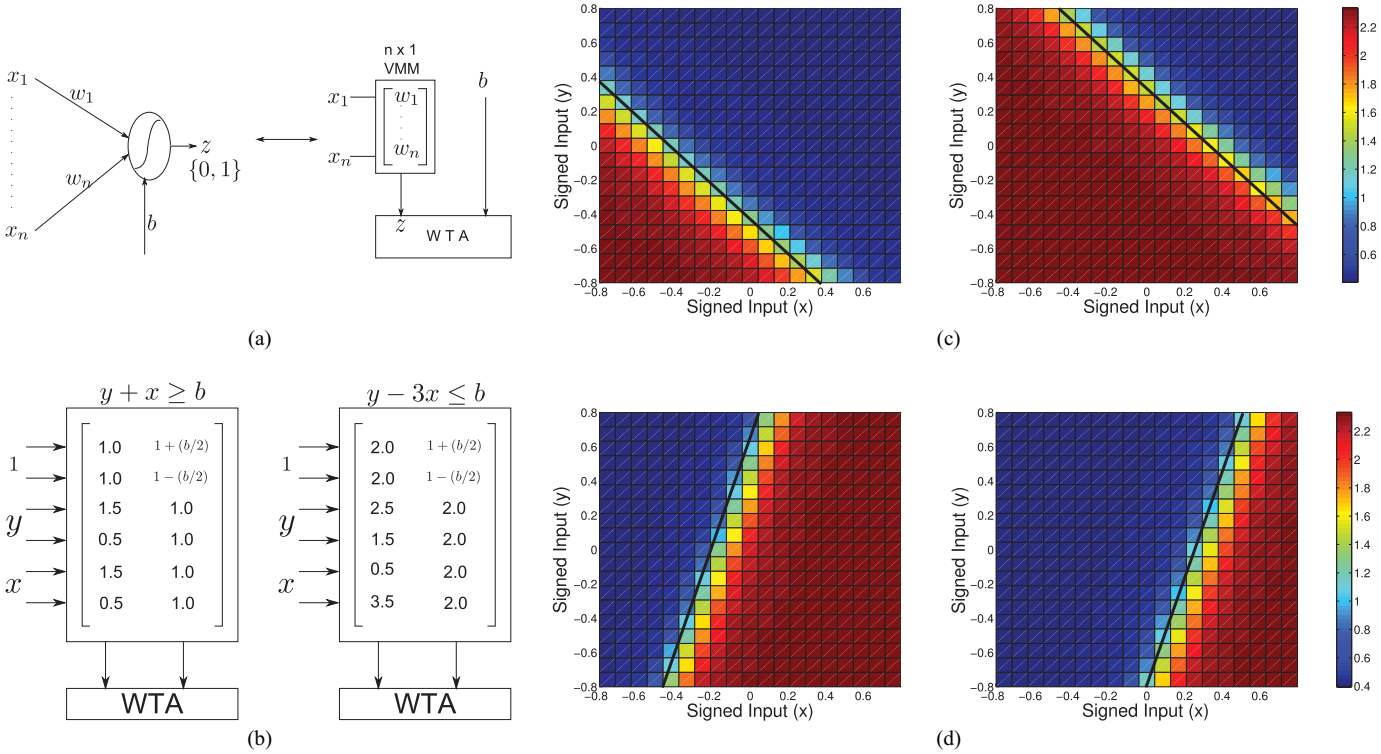


Fig. 6. Linear classifiers: Simple perceptron or a one-layer feed-forward network can be implemented using a VMM + WTA structure. (a) Input multiplication can be implemented using VMMs. The bias $b$ is the second input to the WTA, implemented as a fixed current source. (b) Differential implementation of a linear separator. The bias is programmed as a differential weight with a fixed input. Measured results: (c) VMM + WTA classifier trained to have a decision boundary of $y + x \geq b$, for bias values $b = 0.25, -0.25$. (d) VMM + WTA classifier trained to have a decision boundary of $y - 3x \leq b$, for bias values $b = 0.75, -0.75$. The black solid line represents the theoretical decision boundary.

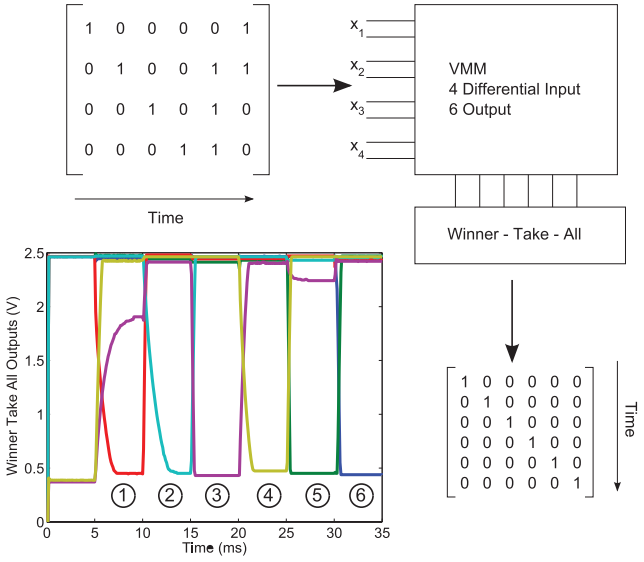


Fig. 7. Multi-dimensional classifiers: a four-input six-output VMM+WTA classifier constructed to classify input sequences. The weights are computed using the psuedo-inverse method. The trained classifier responds to sequence of input patterns.

from the training algorithm and target them to the hardware, without any calibration or offset correction procedure and still match the theoretical decision boundary.

### *B. Multiclass Classifiers*

As the name suggests, multiclass classifiers have several outputs, and classify data into multiple classes. The competitive behavior modeled in the VMM + WTA circuit allows building of such classifiers with multiple outputs that can detect regions of interest. We demonstrate the capability of the VMM + WTA circuit to build a region detector in Fig. 8. We train a two-input, three-output classifier to detect regions of inputs defined as shown in Fig. 8. Again, for simplicity of visualization, we chose only two differential inputs. We constructed a classifier with three outputs and the region boundaries specified in Fig. 8(a). From this theoretical construction, we obtained the weights for the VMM using the pseudo-inverse method. We generate random inputs in MATLAB and multiply them by the weight matrix obtained. We then do a max function on the transformed inputs to generate the theoretical classifier output in Fig. 8(a). Since the theoretical weights were signed, we constructed a fully differential implementation and targeted the weights to the VMM circuit. We then applied 1000 inputs randomly from the set $\{(x, y) : |x| \leq 0.8, |y| \leq 0.8\}$. Since the WTA voltage outputs are inverted, we found the winning output by finding WTA voltages below inverter threshold (mid-rail) and recording its position. In Fig. 8(b), we denote the winning position for each of the random inputs by a different colored dot. Our three-output classifier was programmed with weights obtained directly from MATLAB and matched desired classifier response quite well. Multiclass classifiers are often used as pattern recognizers. We constructed a simple pattern recognizer consisting of four inputs and six outputs, shown in Fig. 7, where the input sequence produces an identity matrix at the output. Each column in the identity matrix represents an output of the WTA and each bit of the four-bit input pattern

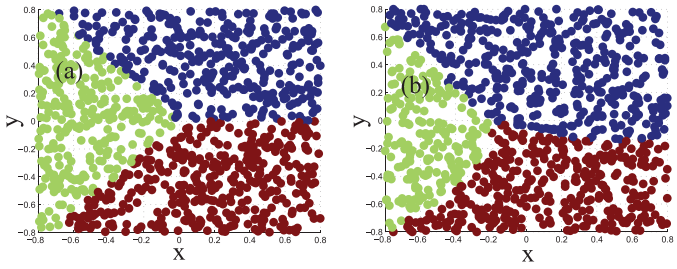


Fig. 8. Multi-dimensional classifiers: (a) a two-input three-output VMM+WTA classifier constructed to have the theoretical decision boundaries shown. Each color represents a different winner. (b) Measured results from the VMM+WTA classifier compiled.

represents a differential input. We obtained the weights using the pseudo-inverse method and programmed the classifier. We tested the classifier by generating a repeating sequence of the inputs. Each pattern was held for 5 ms before the next pattern was presented to the classifier. The transient response measured from each output of the WTA shows that the system classifies the patterns correctly. Since the outputs from the WTA were unbuffered and saw a pin capacitance of $\approx$ 10 pF, we see a slow transition between states of the classifier. This was also the reason why we presented inputs for a long time before switching, and can be avoided by buffering the output nodes.

### C. Nonlinear Classifiers

Nonlinear classification boundaries required in most real-world problems are usually very computationally intensive. Single-layer NNs can only implement classifiers for linearly separable data, but a two-layer NN can approximate any function [16]. A two-layer NN has an input layer, hidden layer, and an output layer. An analog VLSI implementation would require two VMMs for the synaptic computation and two layers of threshold blocks for the hidden and the output layers. This considerably increases the complexity and power consumption of the circuitry. In [17], Maass showed that any Boolean function with analog or digital inputs and one binary output can be approximated with a VMM+$k$-winner-take-all classifier. He showed that the weights for the VMM + WTA classifier are a linear combination of weights of the two-layer perceptron, and further, they are all positive, requiring only single-ended inputs in our implementation. This result provides additional support to the computational power of the VMM + WTA classifier, by halving the computing resources required.

One of the most computationally challenging problems for NNs is the XOR problem. We use the algorithm provided in [17] to compute weights for our VMM+$k$-winner-take-all structure to implement a nonlinear classification boundary for an XOR circuit. One possible implementation of the XOR gate with a two-layer NN and its equivalent VMM + WTA implementation is shown in Fig. 10. The VMM + WTA XOR circuit requires only a single-winner WTA. The position of the WTA output computing the XOR function is marked $z$ in Fig. 10. We tested the XOR circuit by generating inputs from the set $\{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$ and recording the voltage at the third output. The VMM weights are biased at 10 nA, resulting in 95 nA drawn in the VMM when both inputs are active. The WTA is biased at 100 nA, resulting in 0.47 $\mu$W

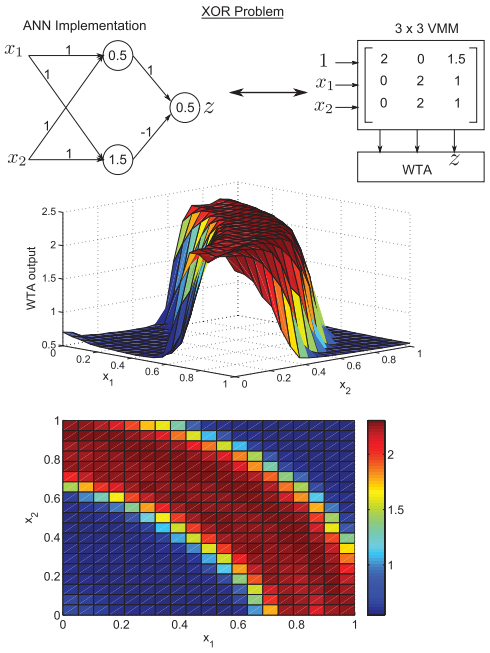


Fig. 9. Nonlinear classifiers. VMM + WTA structure is powerful enough to implement any Boolean function with one digital output. A solution for the XOR problem using a two-layer NN can be translated to a VMM + WTA implementation. Measured results from an XOR implementation using the VMM + WTA structure are plotted.

drawn at 2.4 V, when all inputs are active. The XOR gate is also the simplest case of the $N$-input parity function. Here, we demonstrate the implementation of a four-input parity function using the VMM + WTA classifier. Starting with a two-layer NN implementation [18], we obtain weights for the VMM + WTA classifier using the procedure detailed in [17]. This implementation requires a two-WTA, with five inputs, with the fifth output computing the parity function. The theoretical NN and VMM + WTA implementation is shown in Fig. 10. We obtain a two-WTA by setting $I_{\text{thresh}}$ to $I_c/3$, as shown in Fig. 3, where $I_c$ is the WTA bias current. We test the four bit parity circuit by setting input patterns using DACs on our test platform. We compute the expected parity (marked in the figure) and plot the fifth output from the WTA in Fig. 10. The slow transition at the WTA output is due to the large capacitance at the node and can be avoided by buffering the output. The WTA output does not swing all the way up to the rail for the case of all zeros and all ones, since the VMM output for those cases is very close to the second winner. However, the transition from this state to the winning case is large enough that it can be detected by a logic gate.

## VII. System Performance Characterization

In the following section, we characterize the system performance by considering mismatch effects, power consumption, computing efficiency, and speed of computing. We also discuss the temperature dependence of the classifier output.

### A. Mismatch Compensation

In this section, we investigate effects of mismatch in the WTA circuit, and techniques to compensate for them. We will ignore effects of mismatch in $(W/L)$ and $\kappa$. The dominant source of mismatch in analog design is the threshold voltage mismatch $\Delta V_T$ [19], which is true in sub- and near-threshold

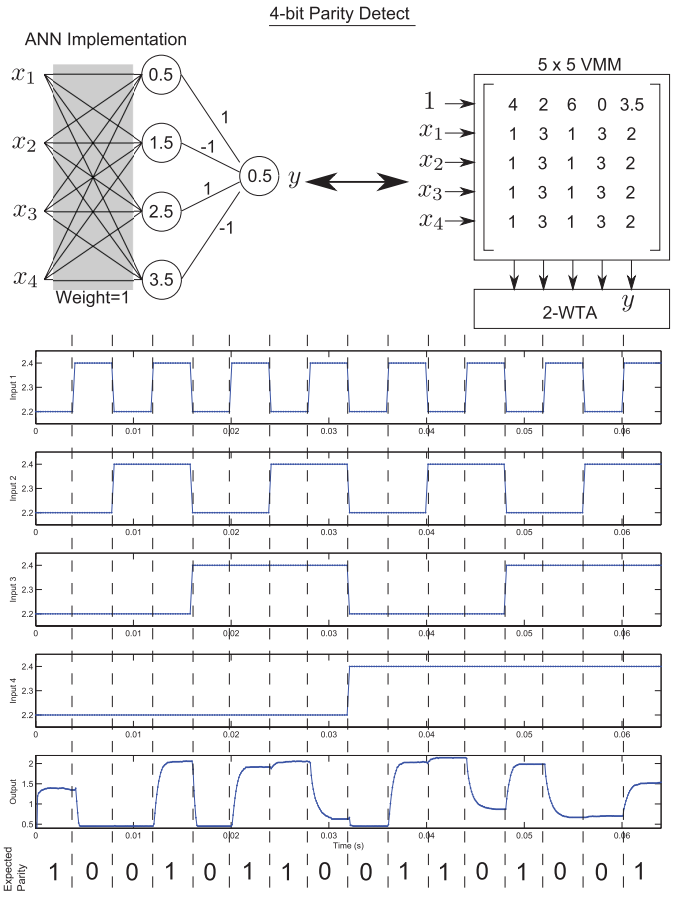


Fig. 10. Nonlinear classifiers. Four-bit parity problem implementation using a two-layer NN and its equivalent VMM + WTA implementation and measured results from the parity detect block.

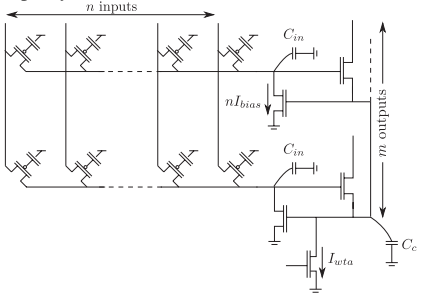


Fig. 11. Schematic of VMM + WTA circuit. Node capacitance at the WTA input scales with the VMM inputs, the common node capacitance scales with WTA outputs. We assume single-quadrant multiplication in the VMM.

regions. In particular, effects of mismatch are worse in the sub-threshold mode of operation since $\delta I/I = -\kappa \Delta V_T / U_T$.

In the WTA shown in Fig. 3(a), we assume that $I_{\text{in1}} = I_{\text{in2}} = I_{\text{in}}$. Then, we expect that $V_1 = V_2$. Both $M_1$ and $M_2$ share the same gate voltage $V_c$. The equation for the drain current through $M_1$, assuming sub-threshold saturation is

$$I = 2I_{\text{th}}(\frac{W}{L})e^{\frac{\kappa(V_g - V_{\text{T0}})}{U_T}} e^{V_d/V_A} \tag{7}$$

where $I_{\text{th}}$ is the threshold current of the device, $V_g$ is the gate voltage, and $V_d$ is the drain voltage. In the balanced case, the difference between $V_1$ and $V_2$ can be expressed as

$$V_1 - V_2 = \frac{\kappa V_A \Delta V_{\text{T1}}}{U_T} + \Delta V_{\text{T3}} \tag{8}$$

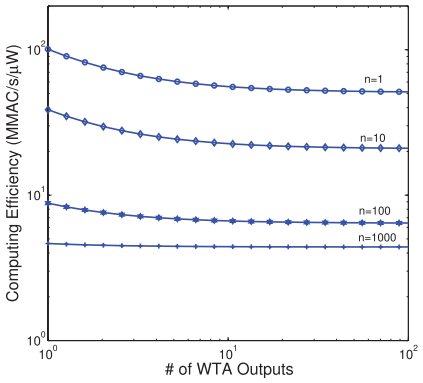


Fig. 12. Computing efficiency versus classifier size. Inverse of the power-delay product in (13) is approximately the computing efficiency in MMAC/s/$\mu$W, which is fixed and scales with inputs and outputs.

where $\Delta V_{\text{T1}}$ is the mismatch between $M_1$ and $M_2$ and $\Delta V_{\text{T3}}$ is the mismatch between $M_3$ and $M_4$. A difference in the input currents $I_{\text{in1}} = I_{\text{in}} + \Delta I_{\text{in}}$ and $I_{\text{in2}} = I_{\text{in}}$ results in a difference in output voltages given by

$$V_1 - V_2 = V_A ln\left(1 + \frac{\Delta I_{\text{in}}}{I_{\text{in}}}\right) \approx V_A \frac{\Delta I_{\text{in}}}{I_{\text{in}}}. \tag{9}$$

This difference in input currents can be programmed in the VMM bias currents to cancel offsets present in the WTA. Another technique for mismatch compensation is including floating-gate transistors in the WTA circuit ($M_1$ and $M_2$), which would require floating-gate nFET devices. Our current chip does not include floating-gate nFET transistors, but this is possible in future versions of this chip. A detailed treatment of mismatch characterization and its automation on the FPAA is presented in [20].

### B. Speed, Power, and Efficiency

We observe the classic power-speed tradeoff in the performance of the VMM + WTA classifier. The power consumption of the VMM is $\mathcal{O}(mn)$, while the WTA power is $\mathcal{O}(n)$. The settling time of the WTA is dominated by the input capacitance $C_{\text{in}}$. The settling time can be reduced by increasing the VMM bias current, which also increases the power consumption.

The dynamic response of the system is determined by the capacitance at the common node in the WTA, shown in Fig. 11. From [5], we get first-order behavior from the circuit when

$$I_{\text{wta}} > 4nI_{\text{bias}}(C_c/C_{\text{in}}) \tag{10}$$

which gives us the WTA bias current to avoid ringing at the winning outputs. Then, the winning node has a time constant $\tau = C_{\text{in}}U_T/(nI_{\text{bias}})$. Since $C_{\text{in}}$ scales with the number of inputs $n$, we write $C_{\text{in}} = nC_{\text{in0}}$. Hence, the settling time for the winning node is independent of $n$ and can be written as

$$\tau = \frac{C_{\text{in0}}U_T}{I_{\text{bias}}}. \tag{11}$$

The power consumption for our classifier, when all inputs are active can be expressed as

$$\begin{aligned} P &= P_{\text{VMM}} + P_{\text{WTA}} \\ &= mnI_{\text{bias}}V_{\text{dd}} + I_{\text{wta}}V_{\text{dd}} \\ &= I_{\text{bias}}V_{\text{dd}}\left(mn + 4m\frac{C_{\text{c0}}}{C_{\text{in0}}}\right) \end{aligned} \tag{12}$$

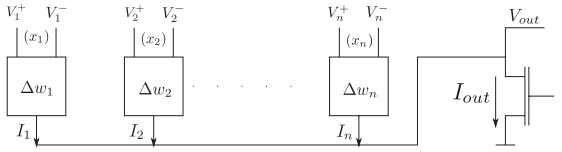


Fig. 13. Temperature dependence. Classifier with a differential VMM can be compensated for temperature.

where $m$ is the number of outputs. $P_{\text{VMM}}$ scales linearly with the number of inputs and outputs, while $P_{\text{WTA}}$ scales with the number of outputs only. This is because the common node capacitance scales with the number of WTA outputs as $C_c = mC_{\text{c0}}$.

We assume a settling time of $4\tau$ to calculate the computation performed by the classifier. The VMM computation is $m * n$ MAC. The WTA computation is more involved, and is equivalent to solving dynamical equations at the $m$ input nodes and the common node. For an equivalent ordinary differential equation (ODE) simulation using Runge–Kutta fourth- and fifth-order adaptive integrator (RK45), we need approximately 60 MAC per node. Thus, the effective computation performed by the classifier can be approximated as $C = (m * n) + 60 * (m + 1)$ MAC. The power per unit computation can be calculated as

$$\frac{P * (4\tau)}{C} = \frac{4m(n + 4\frac{C_{\text{c0}}}{C_{\text{in0}}})C_{\text{in0}}U_T V_{\text{dd}}}{(m * n) + 60 * (m + 1)}. \tag{13}$$

The computing efficiency is plotted in Fig. 12. We assume that $C_{\text{in0}} = C_{\text{c0}} = 1pF$ for this calculation. For a large number of inputs and outputs, the VMM efficiency (which is constant) dominates. For smaller outputs from the classifier, the WTA efficiency dominates.

### C. Temperature Effects

The programmed weights have a direct temperature dependence due to $U_T$, as seen in (3). In a classifier with a differential VMM implementation, as seen in Fig. 13, it is possible to compensate for temperature effects [9]. To derive the temperature dependence, we first note that the WTA output voltage $V_{\text{out}}$ in Fig. 13 is directly proportional to $U_T$. This is true, whether in the balanced case (gain determined by Early voltage) or in the winning case (gain determined by diode-connected nFET), and only the proportionality constants differ. We use the exponential formulation for Early voltage in the nFET drain current equation given by

$$I_{\text{nfet}} = 2I_{\text{th}}\frac{W}{L}e^{(\kappa(V_g - V_{\text{T0}}) + \sigma V_d)/U_T} \tag{14}$$

where $\sigma = U_T/V_A$.

We determine the current from a single differential cell in terms of a reference temperature $T_0$

$$I_1 = I_b w^{T_0/T} e^{V_b/U_T}\left[\left(1 + \frac{\Delta w}{2w}\right)^{T_0/T}\left(1 + \frac{x}{2}\right) + \left(1 - \frac{\Delta w}{2w}\right)^{T_0/T}\left(1 - \frac{x}{2}\right)\right] \tag{15}$$

which can be approximated, by ignoring higher order terms, as

$$I_1 = 2I_b w^{T_0/T} e^{V_b/U_T}\left[1 + x_1\frac{\Delta w_1}{4}\frac{T_0}{T}\right] \tag{16}$$

where $V_b$ is the common mode input voltage, $x$ is the differential input normalized to $U_T$, and $w$ is the bias weight. We assume that the bias weight $w = 1$ and express the total WTA input current as

$$\begin{aligned} I_{\text{out}} &= 2I_b e^{V_b/U_T}\sum_{\text{k}=1}^{n}\left[1 + x_k\frac{\Delta w_k}{4}\frac{T_0}{T}\right] \\ &= 2nI_b e^{V_b/U_T}\left[1 + (1/n)\sum_{\text{k}=1}^{n} x_k\frac{\Delta w_k}{4}\frac{T_0}{T}\right]. \end{aligned} \tag{17}$$

For small increases in temperature, we can assume that $V_g$ remains fixed, resulting in the WTA output voltage

$$V_{\text{out}} = \frac{U_T}{\sigma}\log\left(\frac{2nI_b}{I_{\text{th}}}\right) + \frac{V_b}{\sigma} + \frac{U_{\text{T0}}}{4n\sigma}\sum_{\text{k}=1}^{n} x_k\Delta w_k. \tag{18}$$

The WTA output voltage consists of a bias term, which is temperature-dependent and the signal term, which is temperature independent. We note that the signal term contains $x$, which shows no temperature dependence when the differential input to the VMM scales with temperature.

## VIII. Conclusion

Analog classifiers can provide low-power alternatives to digital signal processing (DSP) techniques for low-precision applications [21]–[23]. We presented results from a powerful reprogrammable classifier that can implement linear and nonlinear decision boundaries. The classifier architecture combines two power efficient circuits to provide an analog signal processing (ASP) alternative to traditional approaches. The system is extremely compact, allowing scaling to a large number of inputs. One of the disadvantages of ASP is fixed functionality. The reconfigurability of the chip allows programmable weights, which enable off-line training, modifications to the size and topology of the WTA to generate different behavior. As an extension to this paper, we can implement local and hysteretic WTAs for certain applications. We have seen that the VMM + WTA is roughly equal to a one-layer NN in circuit complexity, but has computing power equivalent to a two-layer NN. We demonstrated this by implementing classic small-scale nonlinear classification problems.

## Appendix

Consider a four-quadrant VMM cell, shown in Fig. 4. We start with the signed input $x$ and the desired multiplication $y = w * x$, where w is a signed weight. The core of the VMM is a current multiplication with the input current being expressed as $I_{\text{in}} \propto x$. In our multiplier structure, currents are unidirectional, but we desire four quadrant behavior. This is achieved by using differential input currents. The signed input $x$ is encoded as

$$I_{\text{in},p,n} = I_{\text{in,bias}}(1 \pm (x/2)). \tag{19}$$

The output of the transimpedance stage implementing the I-V stage can be calculated by writing the sub-threshold current equation for the transistor in feedback. We assume that the transistor bulk is tied to the power supply

$$I_{\text{pfet}} = I_o e^{\kappa(V_{\text{DD}} - V_{\text{FG}})/U_T} e^{-(V_{\text{DD}} - V_S)/U_T} = I_{\text{in,bias}} w e^{V_S/U_T}.$$

The constraint on the input range can be seen from (20). $x$ is a dimensionless input and (1+ (x/2)) expresses the ratio of the input current to the bias current. Since the voltage input is applied to the source, and due to the exponential dependence of the drain current on source voltage, the linear approximation only holds for a small range. Using (20), the output of the transimpedance stage that sets the source voltage of the input device, which has a weight $w = 1$, we get

$$\begin{aligned} v_{s,p} &= U_T \ln \frac{I_{\text{in},p}}{I_{\text{in,bias}}} \\ v_{s,n} &= U_T \ln \frac{I_{\text{in},n}}{I_{\text{in,bias}}} \\ v_{s,p} - v_{s,n} &= U_T \ln \frac{I_{\text{in},p}}{I_{\text{in},n}} = U_T \ln \frac{1 + (x/2)}{1 - (x/2)}. \end{aligned} \tag{20}$$

For small values of $x$, i.e., $-1 \leq x \leq 1$, $\ln(\frac{1+x/2}{1-x/2}) \approx x$ and hence, $v_{s,p} - v_{s,n} \approx U_T x$. To generate the current inputs to the VMM, $v_{\text{in},p}, v_{\text{in},n}$ are applied to the negative terminal of an OTA with bias current $I_{\text{otabias}}$, used here as a V-I block. To allow values for $-1 \leq x \leq 1$, we require $I_{\text{otabias}} \geq 2I_{\text{in,bias}}$. As a result, the input currents are

$$I_{\text{in},p,n} = I_{\text{otabias}} \tanh(\kappa_{\text{eff}}(v_{\text{in},p,n} - v_{\text{ref}})/2U_T). \tag{21}$$

By using small inputs or a highly linear input stage that has capacitive dividers at the inputs, we can make a linear approximation of (21)

$$I_{\text{in},p,n} = \kappa_{\text{eff}} I_{\text{otabias}}(v_{\text{in},p,n} - v_{\text{ref}})/2U_T. \tag{22}$$

The differential voltage input can be expressed as

$$v_{\text{in},p} - v_{\text{in},n} = \left(\frac{2U_T}{\kappa_{\text{eff}}}\right) \frac{I_{\text{in},p} - I_{\text{in},n}}{I_{\text{otabias}}}. \tag{23}$$

By choosing $I_{\text{otabias}} = 2I_{\text{in,bias}}$, we obtain the relation between voltage inputs to the two VMM topologies as a function of the input $x$

$$v_{\text{in},p} - v_{\text{in},n} = \frac{U_T}{\kappa_{\text{eff}}} x = \frac{v_{s,p} - v_{s,n}}{\kappa_{\text{eff}}} \tag{24}$$

where $\kappa_{\text{eff}}$ denotes the effective coupling from the OTA input to the channel of the differential pair transistors and includes any linearizing factor applied to the OTA to obtain a wide linear input range. The output current can be calculated using the pFET subthreshold (20) as

$$\begin{aligned} I_{\text{out}} &= I_{\text{in,bias}}(w + \Delta w)e^{v_{\text{sp}}/U_T} + I_{\text{in,bias}}(w - \Delta w)e^{v_{\text{sn}}/U_T} \\ &= (w + \Delta w) * I_{\text{in},p} + (w - \Delta w) * I_{\text{in},n} \\ &= 2I_{\text{in,bias}} w + I_{\text{in,bias}} x \Delta w. \end{aligned} \tag{25}$$

The first and second terms in (25) represent the bias and the four quadrant multiplication terms, respectively, since $x$ and $\Delta w$ can be signed.

## REFERENCES


[1] B. Marr, B. Degnan, P. Hasler, and D. Anderson, "Scaling energy per operation via an asynchronous pipeline," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 147–151, Jan. 2013.

[2] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA, USA: Addison-Wesley, 1989.

[3] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[4] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *Computer*, vol. 21, no. 3, pp. 25–39, 1988.

[5] J. Lazzaro, "Winner-take-all networks of $O(N)$ complexity," DTIC, Fort Belvoir, VA, USA, Tech. Rep. A664154, 1988.

[6] 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.

[7] T. Morris, T. Horiuchi, and S. DeWeerth, "Object-based selection within an analog VLSI visual attention system," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 12, pp. 1564–1572, Dec. 1998.

[8] C. Schlottmann, S. Shapero, S. Nease, and P. Hasler, "A digitally enhanced dynamically reconfigurable analog platform for low-power signal processing," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2174–2184, Sep. 2012.

[9] C. Schlottmann and P. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 403–411, Sep. 2011.

[10] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz, 128 × 32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Proc. IEEE Custom Integr. Circuits Conf.*, Oct. 2004, pp. 651–654.

[11] S. Suh, A. Basu, C. Schlottmann, P. Hasler, and J. Barry, "Low-power discrete fourier transform for OFDM: A programmable analog approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 290–298, Feb. 2011.

[12] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *IEEE J. Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sep. 2010.

[13] G. Indiveri, "Modeling selective attention using a neuromorphic analog VLSI device," *Neural Comput.*, vol. 12, no. 12, pp. 2857–2880, 2000.

[14] 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.

[15] W. Kruger, P. Hasler, B. Minch, and C. Koch, "An adaptive WTA using floating gate technology," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1997, pp. 720–726.

[16] K. Siu, V. Roychowdhury, and T. Kailath, *Discrete Neural Computation: A Theoretical Foundation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[17] W. Maass, "On the computational power of winner-take-all," *Neural Comput.*, vol. 12, no. 11, pp. 2519–2535, 2000.

[18] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation*, vol. 1. Boulder, CO, USA: Westview, 1991.

[19] P. Kinget, "Device mismatch and tradeoffs in the design of analog circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 6, pp. 1212–1224, Jun. 2005.

[20] S. Shapero, and P. Hasler, "Mismatch characterization and calibration for accurate and automated analog design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 3, pp. 548–556, Mar. 2013.

[21] T. Yamasaki and T. Shibata, "Analog soft-pattern-matching classifier using floating-gate MOS technology," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1257–1265, Sep. 2003.

[22] M. Gu and S. Chakrabartty, "Synthesis of bias-scalable CMOS analog computational circuits using margin propagation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 2, pp. 243–254, Feb. 2012.

[23] M. Yldz, S. Minaei, and I. Goknar, "A CMOS classifier circuit using neural networks with novel architecture," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1845–1850, Nov. 2007.


**Shubha Ramakrishnan**, (S'–) photograph and biography are not available at the time of publication.

**Jennifer Hasler**, (SM'–) photograph and biography are not available at the time of publication.