Analog/Mixed-Signal Circuits for Machine Learning Applications

EE451 Supervised Research Exposition (Guide: Prof. Rajesh Zele)

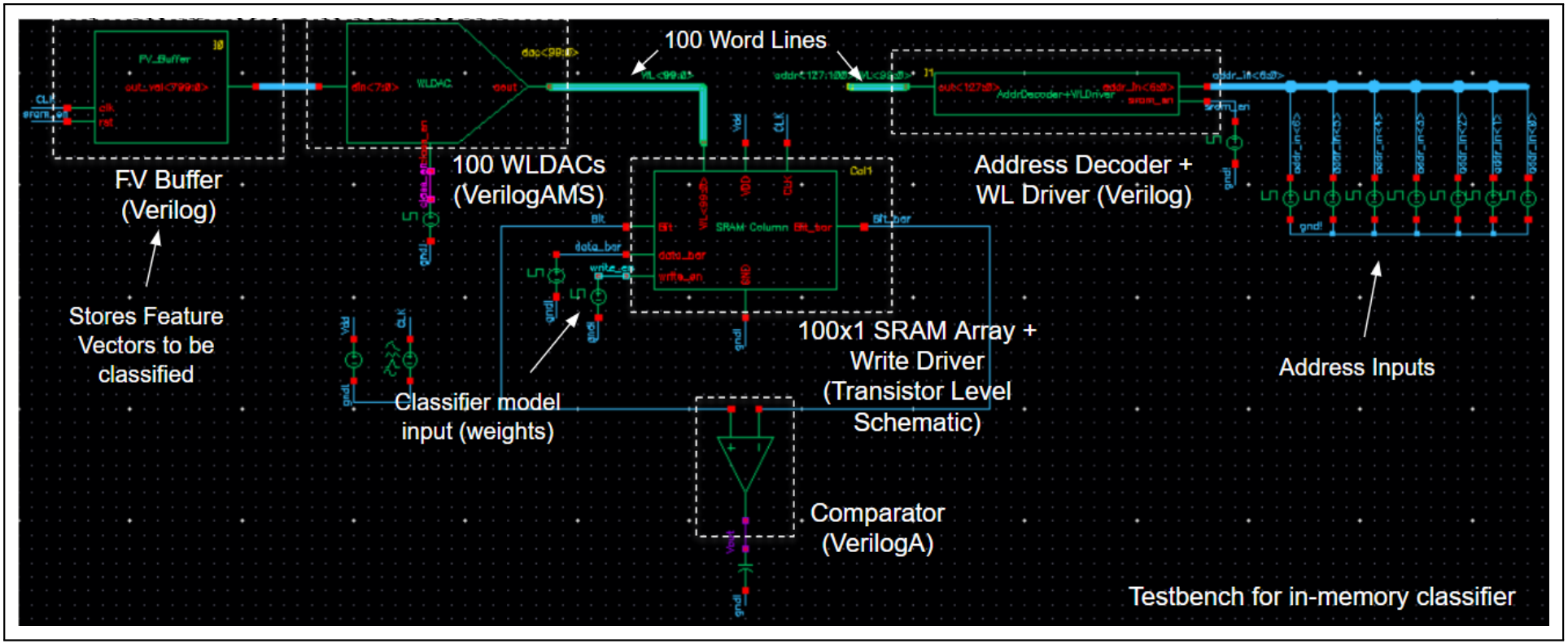
Mihir Kavishwar (17D070004)
Advanced Integrated Circuits and Systems Lab
Electrical Engineering, IIT Bombay

Introduction

- Over the past few years, there have been several efforts to design Analog/Mixed-Signal circuits which can implement Machine Learning algorithms
- One of the main driving factors for this research is the **high energy efficiency** and **low latency** of analog designs when compared to their digital counterparts, which is critical for applications like Edge Devices
- In this presentation I will discuss some literature that I reviewed and show some simulation work which was carried out during the course of this semester

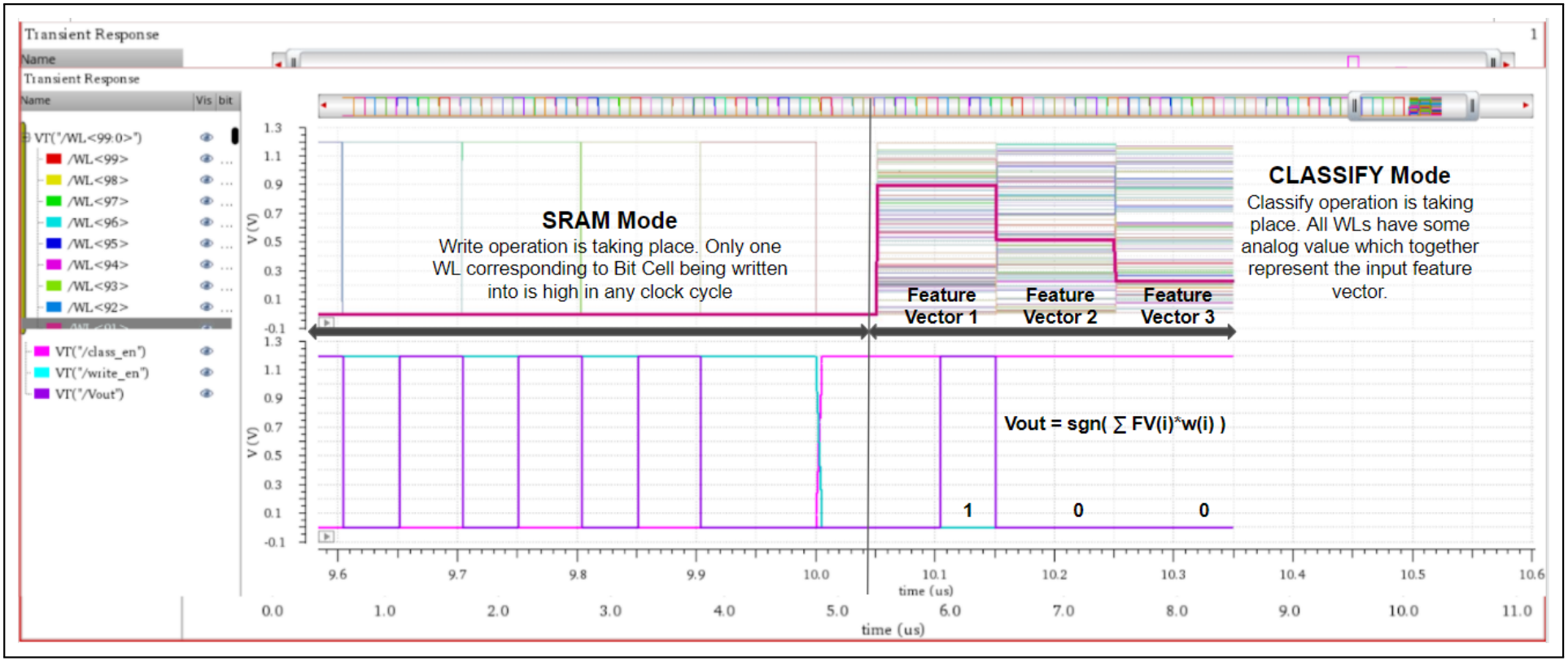
Outline

Simulation Setup



Ref: J. Zhang, Z. Wang and N. Verma, "**In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 4, pp. 915-924, April 2017

Transient Simulation Results



Ref: J. Zhang, Z. Wang and N. Verma, "**In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 4, pp. 915-924, April 2017

Winner-Take-All Circuits

- Many classification algorithms in supervised learning use a winner-take-all (WTA) approach to produce the final outputs. Given N input-output pairs, say

$$(input_k, output_k) \quad k \in \{1, 2, \ldots, N\}$$

$$\text{then,} \quad output_k = \begin{cases} 1, & if \;\; k = m \\ 0, & if \;\; k \neq m \end{cases}$$

$$\text{where,} \quad m = \underset{k \in \{1,2,\ldots,N\}}{\arg\max} \; input_k$$

- That is, only the output corresponding to the largest input is high while all other outputs are low

Ref: J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead, "**Winner-take-all networks of O(N) complexity**", Advances in Neural Information Processing Systems 1, Morgan Kaufmann Publishers, San Francisco, CA, 1989

Winner-Take-All Circuits

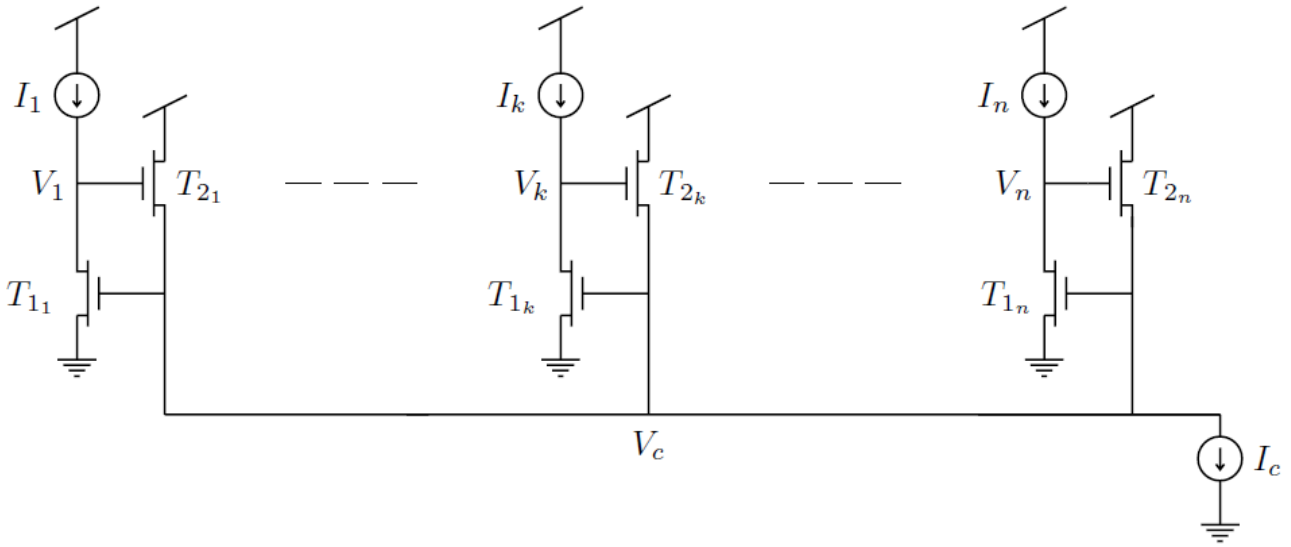


Figure 1. Schematic diagram of the winner-take-all circuit. Each neuron receives a unidirectional current input I_k; the output voltages $V_1 \ldots V_n$ represent the result of the winner-take-all computation. If $I_k = \max(I_1 \ldots I_n)$, then V_k is a logarithmic function of I_k; if $I_j \ll I_k$, then $V_j \approx 0$.

Ref: J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead, "**Winner-take-all networks of O(N) complexity**", Advances in Neural Information Processing Systems 1, Morgan Kaufmann Publishers, San Francisco, CA, 1989

Winner-Take-All Circuits

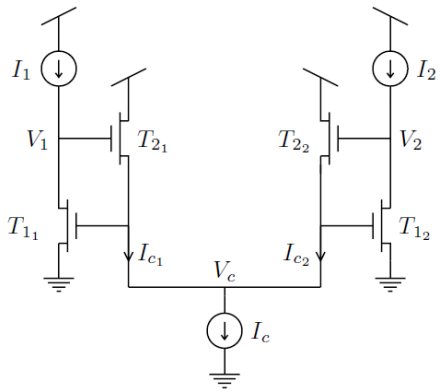


Figure 2. Schematic diagram of a two-neuron winner-take-all circuit.

Subthreshold Current Equation:

$$I_{ds} = I_0 e^{\frac{\kappa V_{gs}}{V_T}} \left(1 - e^{-\frac{V_{ds}}{V_T}}\right)$$

If $I_1 = I_m + \delta_i$ and $I_2 = I_m$, we can show

$$V_1 \approx \frac{V_T}{\kappa} ln\left(\frac{I_m + \delta_i}{I_0}\right) + \frac{V_T}{\kappa} ln\left(\frac{I_c}{I_0}\right)$$

$$V_2 \approx 0$$

Ref: J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead, "**Winner-take-all networks of O(N) complexity**", Advances in Neural Information Processing Systems 1, Morgan Kaufmann Publishers, San Francisco, CA, 1989

Winner-Take-All Circuits

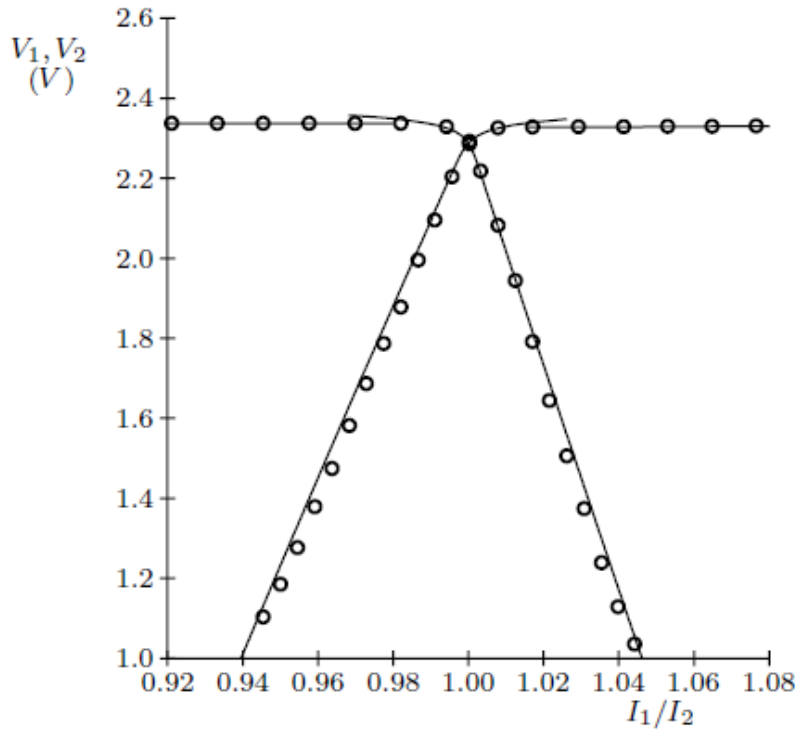


Figure 3. Experimental data (circles) and theory (solid lines) for a two-neuron winner-take-all circuit. I_1, the input current of the first neuron, is swept about the value of I_2, the input current of the second neuron; neuron voltage outputs V_1 and V_2 are plotted versus normalized input current.

Ref: J. Lazzaro, S. Ryckebusch, M. A. Mahowald and C. A. Mead, "**Winner-take-all networks of O(N) complexity**", Advances in Neural Information Processing Systems 1, Morgan Kaufmann Publishers, San Francisco, CA, 1989

Bump and Anti-Bump Circuits

- The notion of **similarity** is commonly used in many ML tasks:
  - Clustering algorithms such as K-means clustering use Euclidean distance to compute similarity between two data points
  - Radial Basis Function (RBF) Kernel which is commonly used in Support Vector Machines is actually a similarity function
- **Bump circuits** implement a gaussian-like **similarity function** which "bumps" if the input voltages are close to each other and dies down if they are far apart
- **Anti-bump circuits** implement a **dis-similarity function** which is like an inverse gaussian - it gives high output if input voltages are far apart and low output if they are close by

Ref: T. Delbruck, "**'Bump' circuits for computing similarity and dissimilarity of analog voltages**," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991

Bump and Anti-Bump Circuits

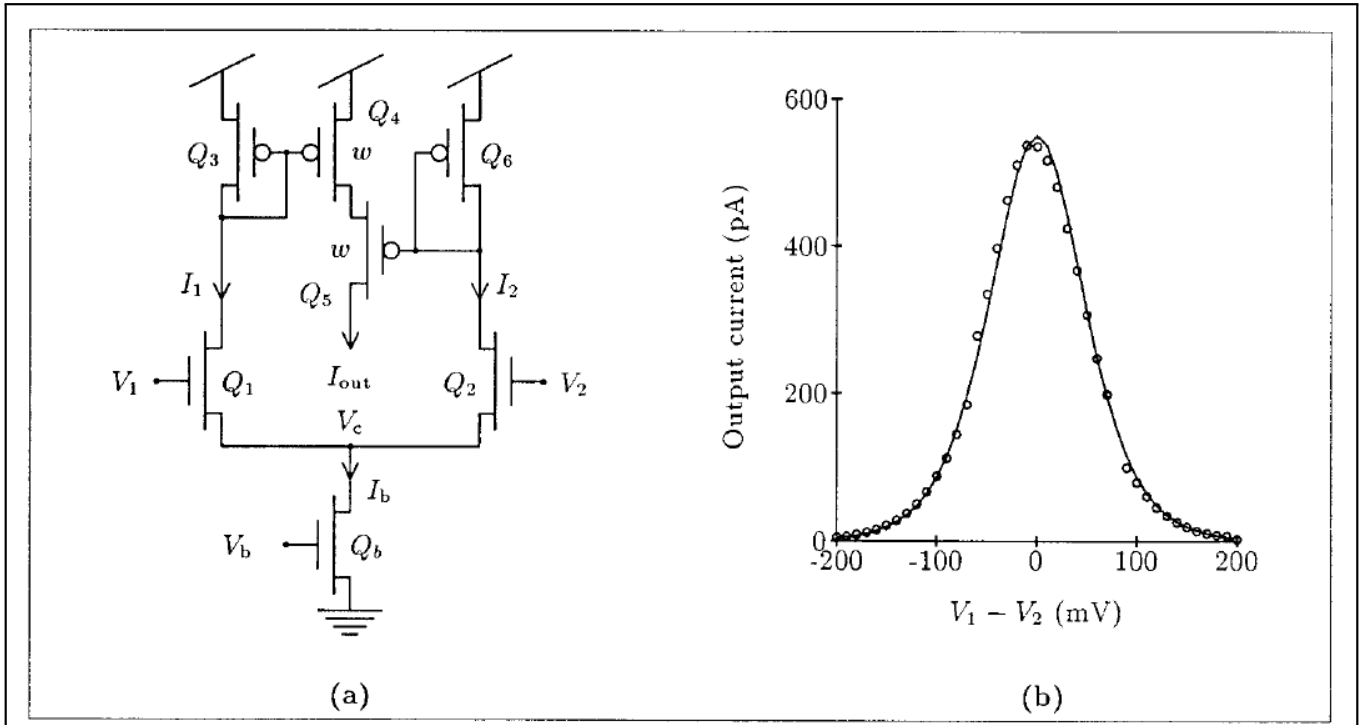
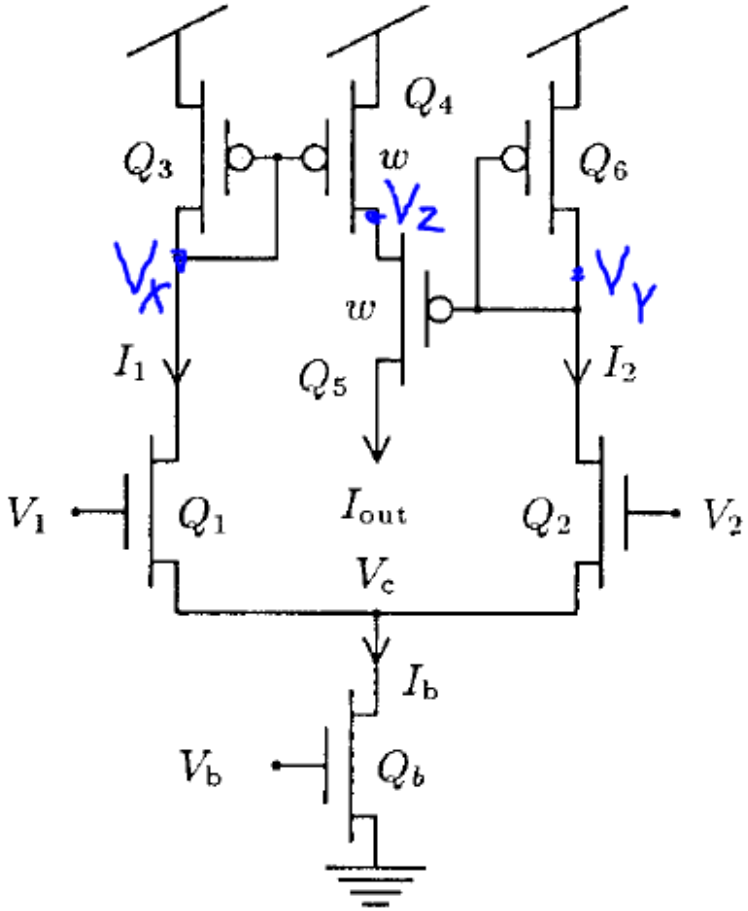


Figure 1: First bump circuit. (a) Bump circuit. (b) Resulting output.

Ref: T. Delbruck, "**'Bump' circuits for computing similarity and dissimilarity of analog voltages**," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991

Bump and Anti-Bump Circuits



All voltages are in units of $\frac{KT}{q}$. Analysis of differential pair gives us

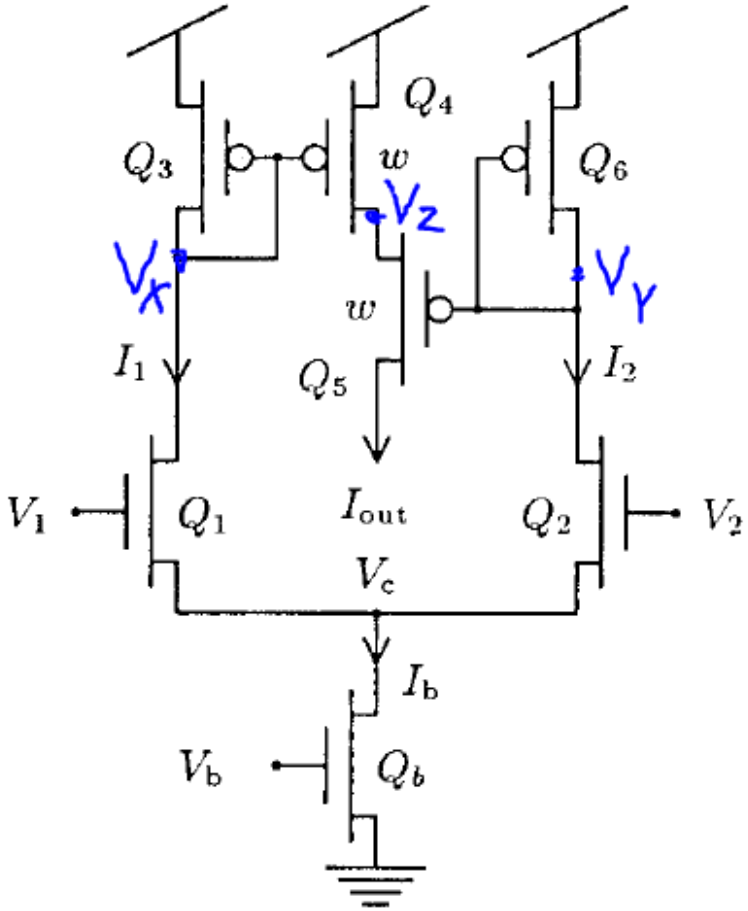
$$I_1 = I_b \frac{e^{\kappa V_1}}{e^{\kappa V_1} + e^{\kappa V_2}}$$

$$\textit{and} \quad I_2 = I_b \frac{e^{\kappa V_2}}{e^{\kappa V_1} + e^{\kappa V_2}}$$

If $|\Delta V| = |V_1 - V_2|$ is larger than a few $\frac{KT}{q}$ then current in one of the two legs will shut off

Ref: T. Delbruck, "**'Bump' circuits for computing similarity and dissimilarity of analog voltages**," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991

Bump and Anti-Bump Circuits



Analysis of current correlator gives us

$$I_{out} \approx wI_b \frac{e^{(V_{dd}-\kappa V_x)} e^{(V_{dd}-\kappa V_y)}}{e^{(V_{dd}-\kappa V_x)} + e^{(V_{dd}-\kappa V_y)}}$$

$$\approx w \frac{I_1 I_2}{I_1 + I_2}$$

Substituting I_1 and I_2 gives us

$$I_{out} \approx w \frac{I_b}{2} sech^2 \left(\frac{\kappa(V_1 - V_2)}{2} \right)$$

where $sech(x) = \frac{2}{e^x + e^{-x}}$

Ref: T. Delbruck, "**'Bump' circuits for computing similarity and dissimilarity of analog voltages**," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991

Bump and Anti-Bump Circuits

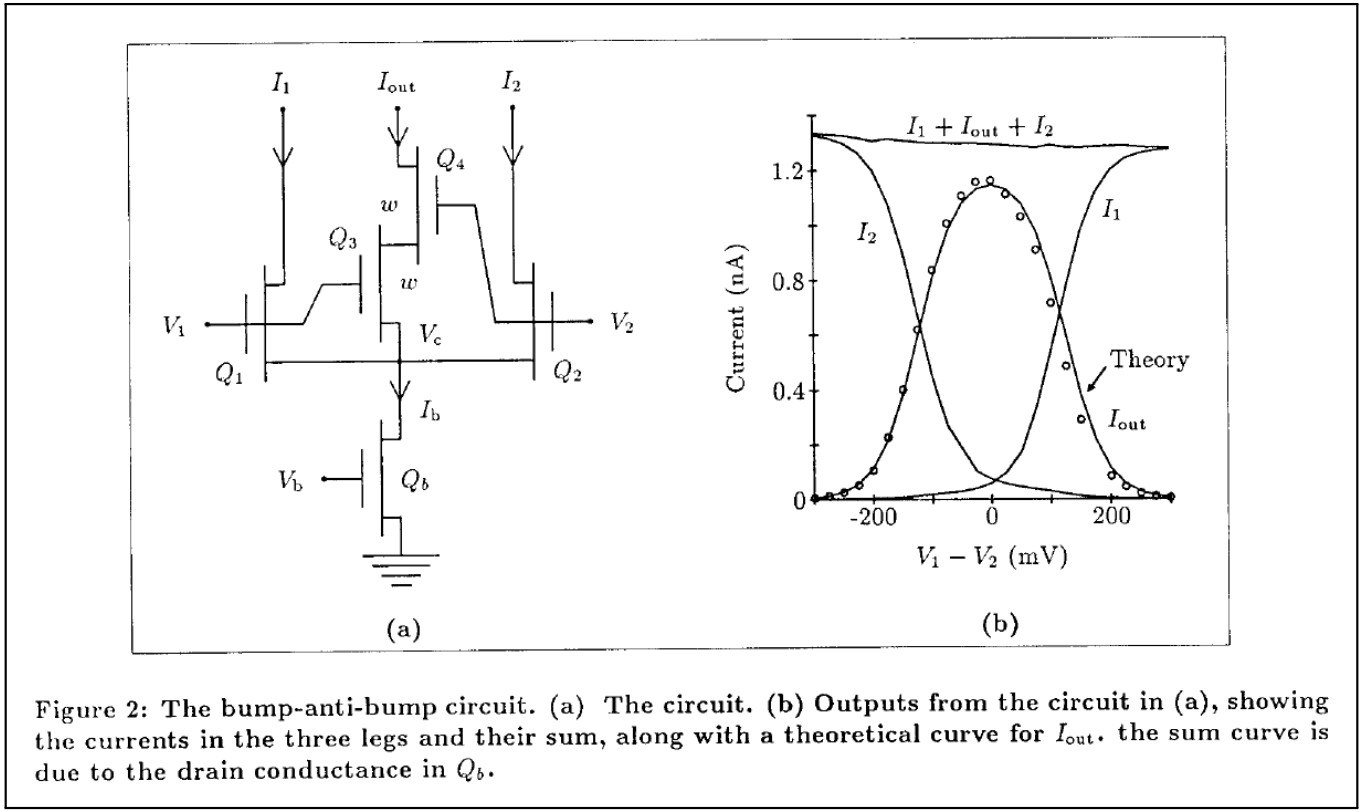


Figure 2: The bump-anti-bump circuit. (a) The circuit. (b) Outputs from the circuit in (a), showing the currents in the three legs and their sum, along with a theoretical curve for I_{out}. the sum curve is due to the drain conductance in Q_b.

$$I_{out} \approx \frac{I_b}{1 + \frac{4}{w}\cosh^2\left(\frac{\kappa(V_1 - V_2)}{2}\right)} \quad \text{where} \quad \cosh(x) = \frac{e^x + e^{-x}}{2}$$

Ref: T. Delbruck, "**'Bump' circuits for computing similarity and dissimilarity of analog voltages**," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991

Analog Programmable Multidimensional RBF Classifier

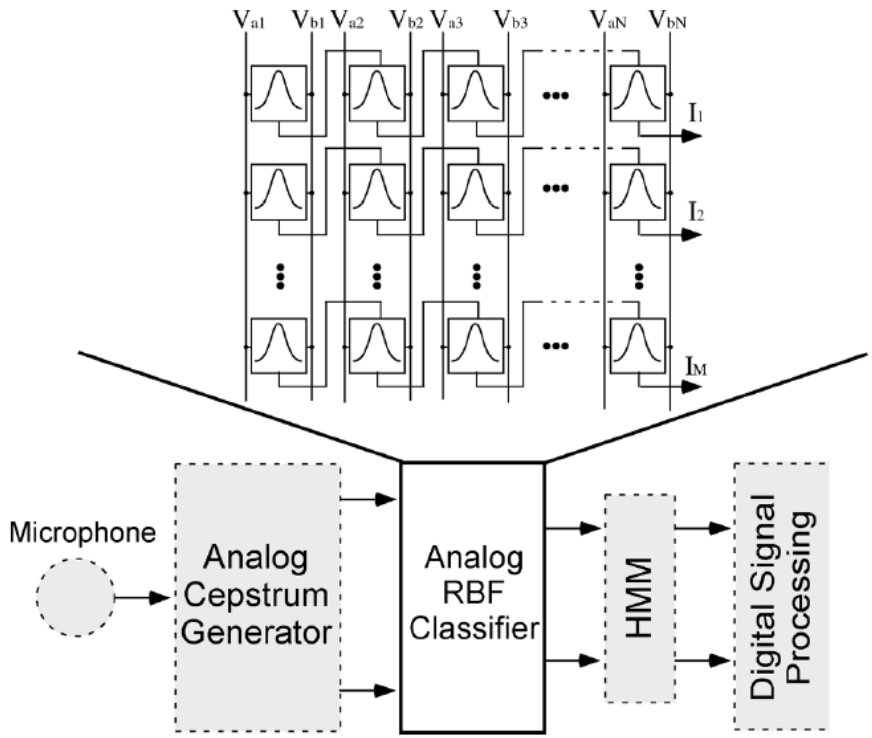


Fig. 1. Analog RBF-based classifier in an analog front-end for speech recognition. The front-end of our current speech recognition system includes a band-pass-filter bank based analog Cepstrum generator, an analog RBF-based classifier, and a continuous-time HMM. Putting the DSP stages behind the analog front-end makes the entire system more efficient.

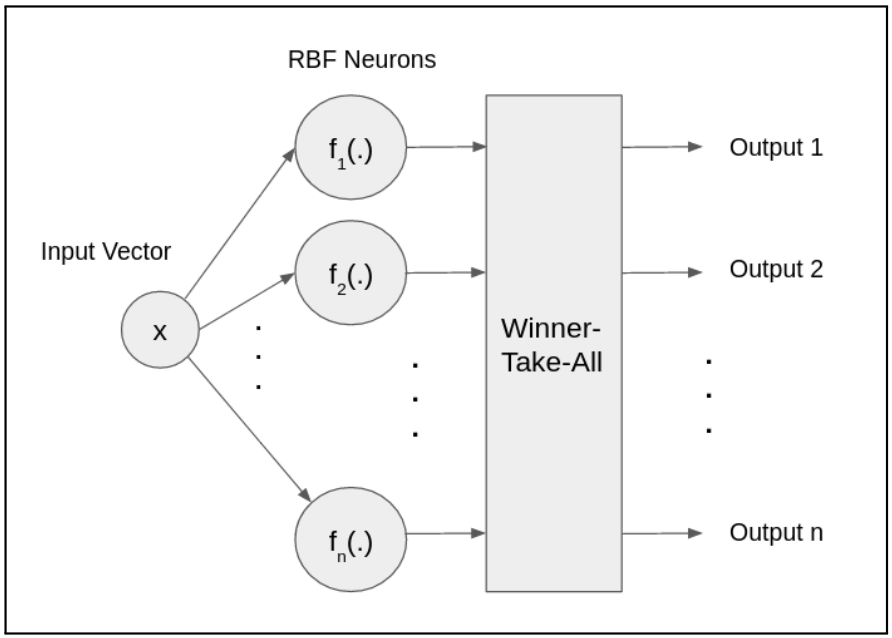
A real-valued function φ whose value depends only on the distance between the input and some fixed point, c, called a center, so that $\varphi(x) = \varphi(\|x - c\|)$ is called Radial Basis Function (RBF)

A classifier which uses RBFs to determine decision boundaries is called an RBF classifier.

Ref: S. Peng, P. E. Hasler and D. V. Anderson, "**An Analog Programmable Multidimensional Radial Basis Function Based Classifier**," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 10, pp. 2148-2158, Oct. 2007

Analog Programmable Multidimensional RBF Classifier

Classification Algorithm



Implementation

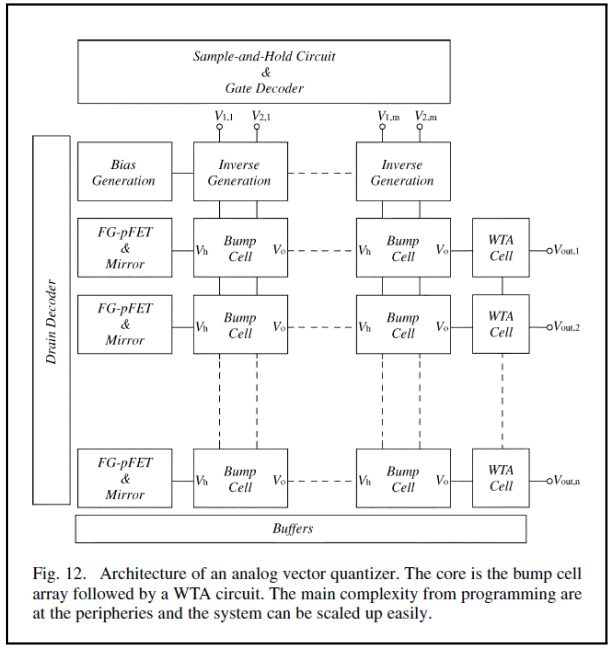


Fig. 12. Architecture of an analog vector quantizer. The core is the bump cell array followed by a WTA circuit. The main complexity from programming are at the peripheries and the system can be scaled up easily.

Here $f_i(.)$ is a multidimensional Gaussian RBF function with diagonal covariance matrix Σ_i, mean vector μ_i , and maximum likelihood K_i

$$f_{\mu_i,\Sigma_i,K_i}(\mathrm{x}) = K_i.exp\left(-\frac{1}{2}(\mathrm{x}-\mu_\mathrm{i})^T{\Sigma_i}^{-1}(\mathrm{x}-\mu_\mathrm{i})\right)$$

Ref: S. Peng, P. E. Hasler and D. V. Anderson, "**An Analog Programmable Multidimensional Radial Basis Function Based Classifier**," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 10, pp. 2148-2158, Oct. 2007

Analog Programmable Multidimensional RBF Classifier

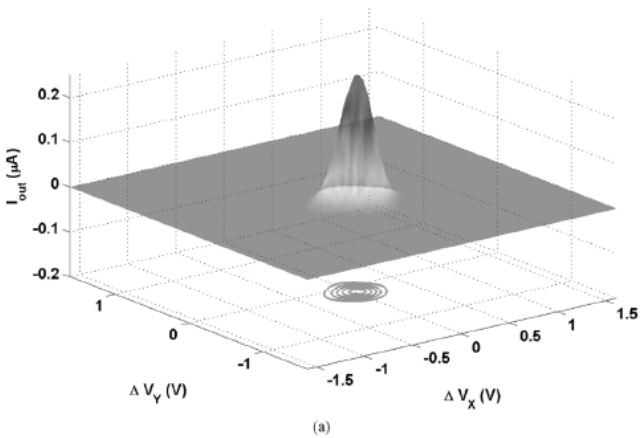


Fig. 7. Measured bivariate RBFs. Measurement results from two cascading floating-gate bump circuits. ΔV_X is the input voltage difference $\Delta V_{in} = V_{in1} - V_{in2}$ of the first stage floating-gate bump circuit and ΔV_Y is the input voltage difference of the second stage. In both stages, $V_{in2} = V_{DD}/2$. The common-mode charges are programmed to different levels to approximate bivariate Gaussian functions with different variance.

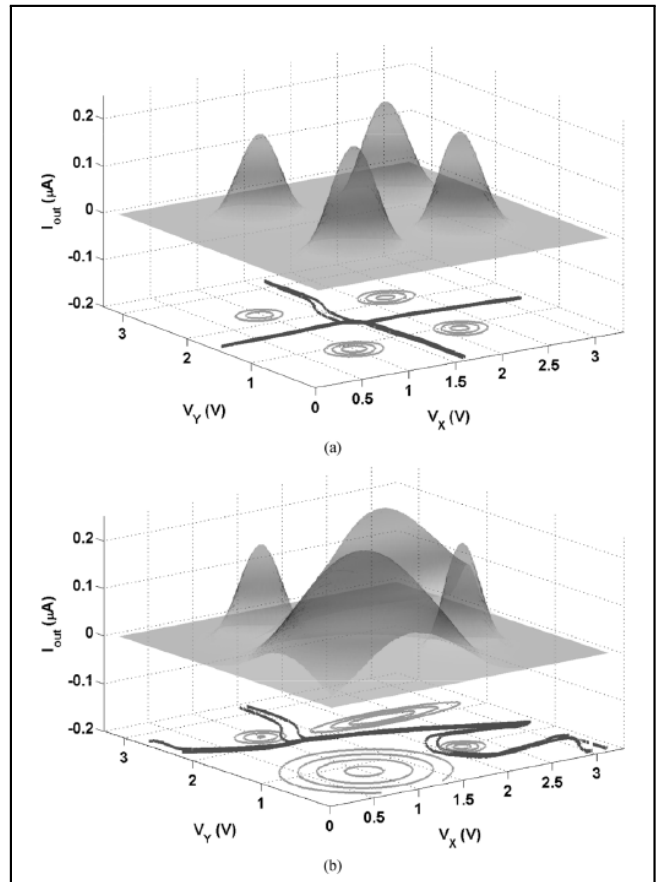


Fig. 13. Configurable classification results. The measured bump output currents (circle contours) and the WTA voltages (thick solid lines at the bottom) of four templates are superposed in a single plot. V_X and V_Y are the V_{in1} in the first stage and the second stage floating-gate bump circuits, respectively. Both of their V_{in2} terminals are fixed at $V_{DD}/2$. (a) Four templates are programmed to have the same variance and evenly spaced means. (b) Four templates are programmed to have different variances with evenly spaced means.

Ref: S. Peng, P. E. Hasler and D. V. Anderson, "**An Analog Programmable Multidimensional Radial Basis Function Based Classifier**," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 10, pp. 2148-2158, Oct. 2007

Analog Programmable Multidimensional RBF Classifier

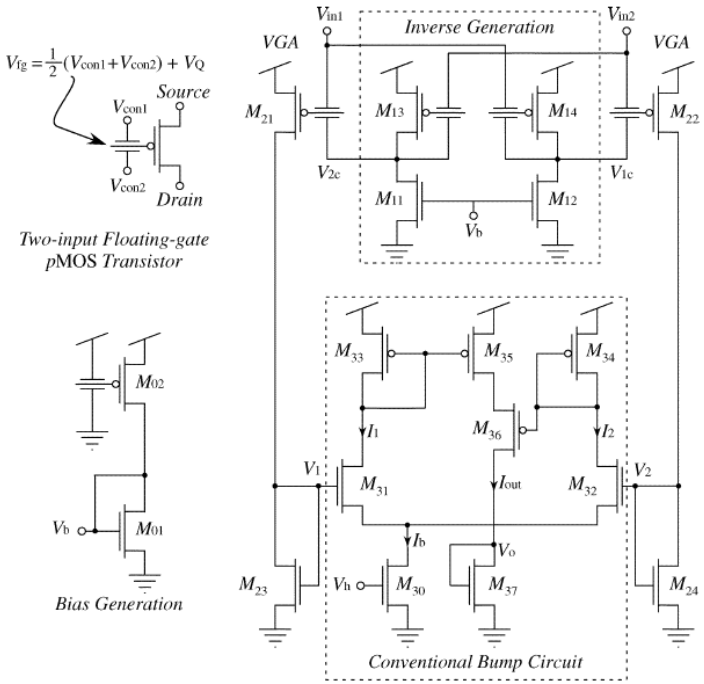


Fig. 2. Schematics of the new floating-gate bump circuit. All floating-gate transistors in the schematics have two inputs with equal weights and the floating-gate voltage can be expressed as $V_{\rm fg} = 1/2(V_{\rm con1} + V_{\rm con2}) + V_Q$, where $V_Q = Q_{\rm fg}/C_{\rm T}$, $Q_{\rm fg}$ is the charge on the floating gate and $C_{\rm T}$ is the total capacitance from the floating gate. The bump circuit is composed of an inverse generation block, a fully differential VGA, and a conventional bump circuit. The width and the center of the bell-shaped transfer function are set by the common-mode and differential charges on M_{21} and M_{22}. The height is controlled by the tail current $I_{\rm b}$. All of them are independently programmable.

In order to achieve programmability of parameters (K_i, Σ_i, μ_i), 2-input Floating Gate Transistors were used
We can show,

$$I_{out} \approx w\frac{I_b}{2}.exp\left(-\eta'(\Delta V_{in} + V_{Q,dm})^2\right)$$

η' is a function of $V_{Q,cm}$
By programming I_b, $V_{Q,cm}$ and $V_{Q,dm}$ we can vary K_i, Σ_i, μ_i respectively

To make this multidimensional we simply cascade the bump cells

Ref: S. Peng, P. E. Hasler and D. V. Anderson, "**An Analog Programmable Multidimensional Radial Basis Function Based Classifier**," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 54, no. 10, pp. 2148-2158, Oct. 2007

Switched-Capacitor Matrix Multiplier

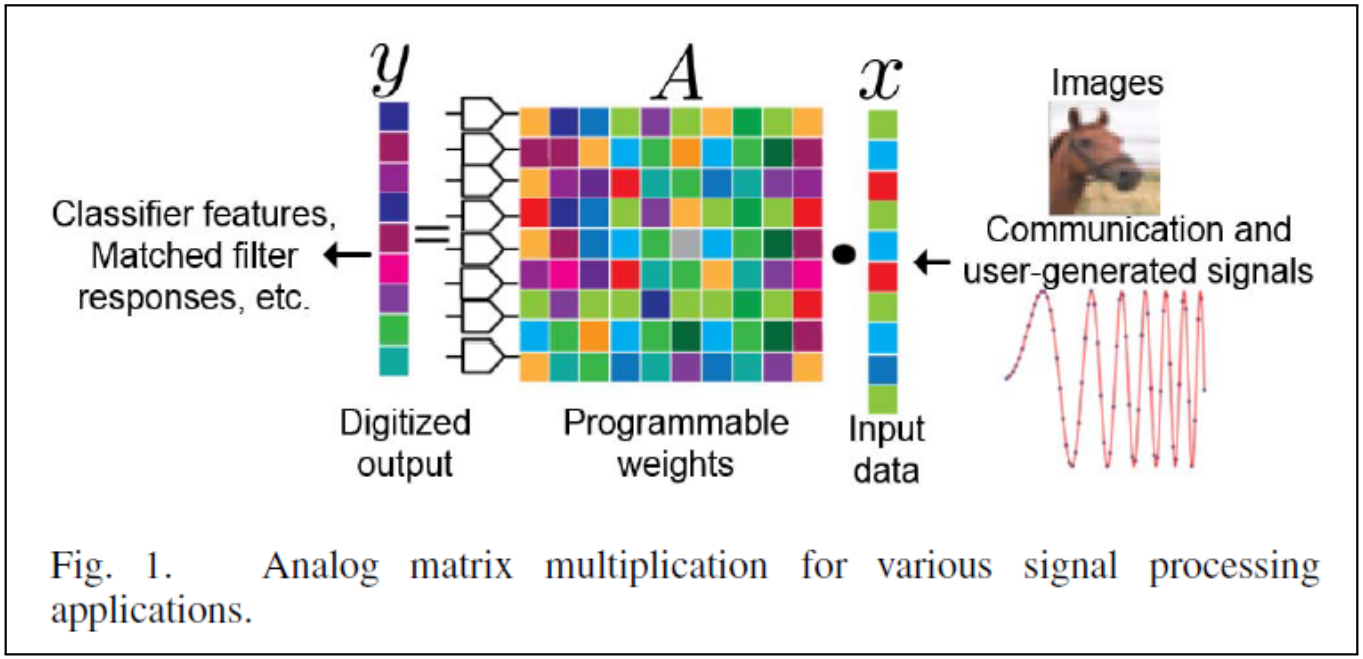


Fig. 1. Analog matrix multiplication for various signal processing applications.

$$y[j] = \sum_{i=1}^{n} A[j,i].x[i]$$

Ref: E. H. Lee and S. S. Wong, "**Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017

Switched-Capacitor Matrix Multiplier

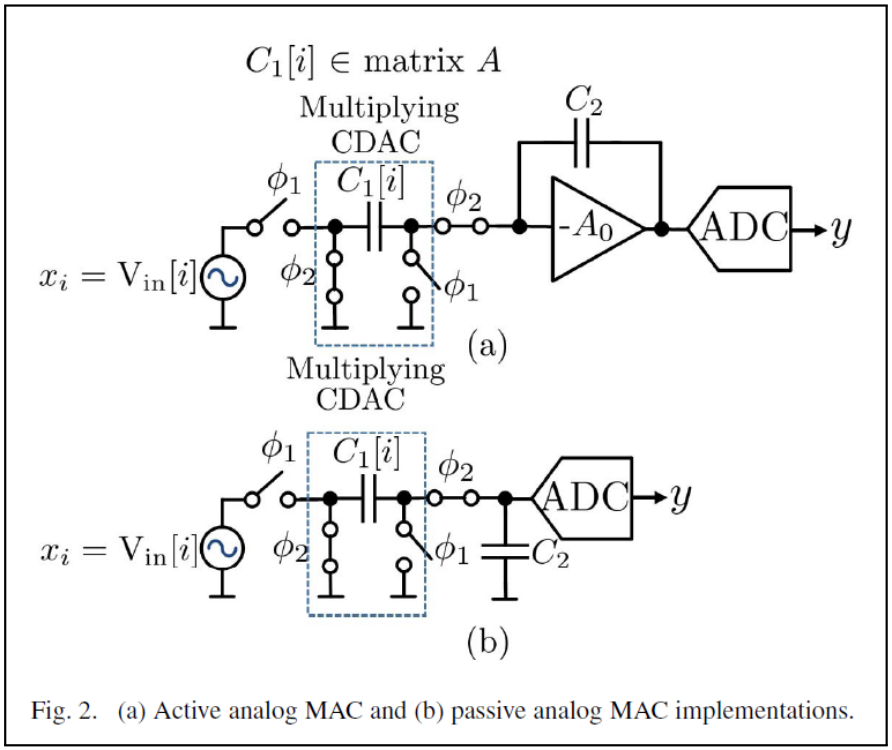


Fig. 2. (a) Active analog MAC and (b) passive analog MAC implementations.

Active

$$V_{C_2}[i] = \frac{C_1[i].(1+A_0)}{C_1[i]+C_2.(1+A_0)} V_{in}[i] + \frac{C_2.(1+A_0)}{C_1[i]+C_2.(1+A_0)} V_{C_2}[i-1]$$

Passive

$$V_{C_2}[i] = \frac{C_1[i]}{C_1[i]+C_2} V_{in}[i] + \frac{C_2}{C_1[i]+C_2} V_{C_2}[i-1]$$

$$V_{C_2}[i] = \mu[i].k[i].V_{in}[i] + k[i].V_{C_2}[i-1]$$
$$\approx \mu[i].V_{in}[i] + V_{C_2}[i-1] \quad ; \quad C_2 \gg C_1[i]$$

Ref: E. H. Lee and S. S. Wong, "**Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017

Switched-Capacitor Matrix Multiplier

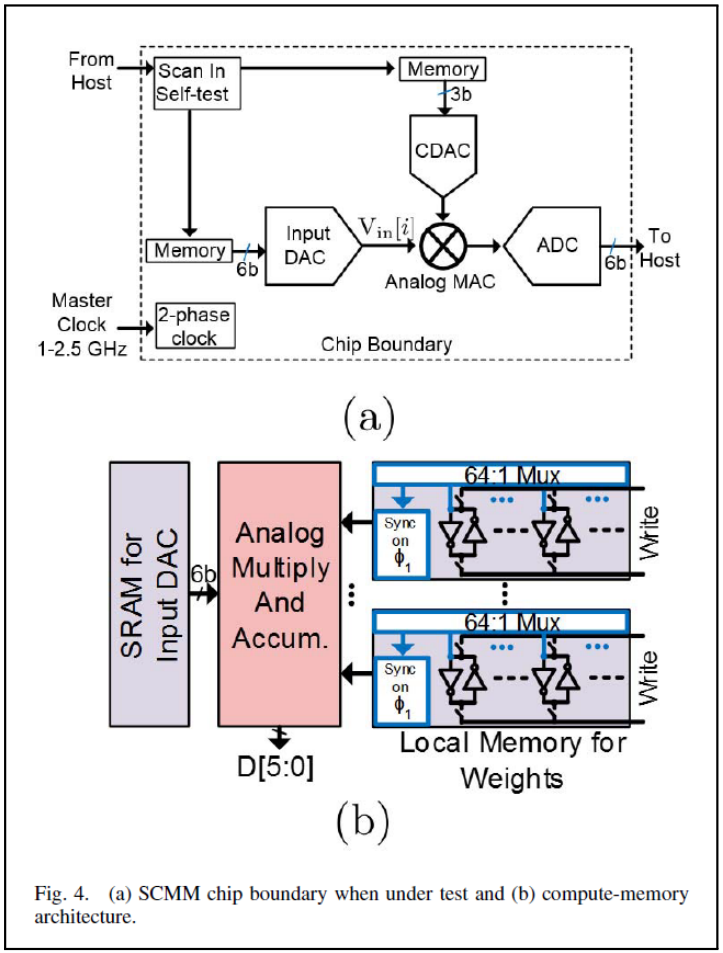


Fig. 4. (a) SCMM chip boundary when under test and (b) compute-memory architecture.

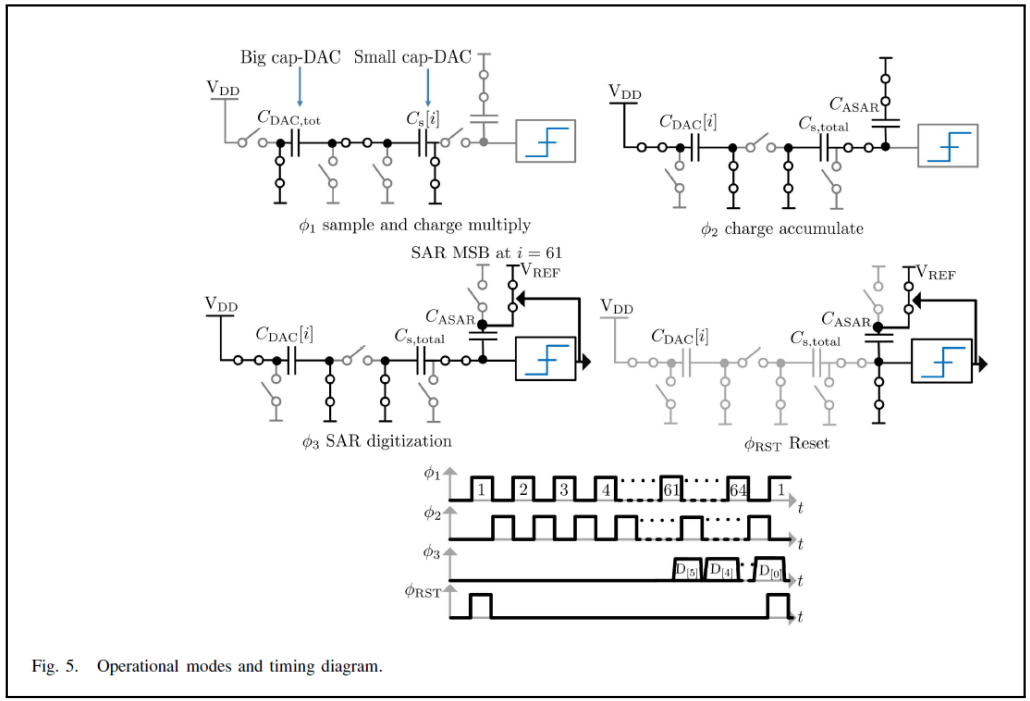


Fig. 5. Operational modes and timing diagram.

Ref: E. H. Lee and S. S. Wong, "**Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017

Switched-Capacitor Matrix Multiplier

$$y = \tilde{A}x$$

$$\tilde{A} = \begin{bmatrix} \mu[1,1]\prod_{j=1}^{n} k[1,j] & \mu[1,2]\prod_{j=2}^{n} k[1,j] & \dots & \mu[1,n]k[1,n] \\ \mu[2,1]\prod_{j=1}^{n} k[2,j] & \mu[2,2]\prod_{j=2}^{n} k[2,j] & \dots & \mu[2,n]k[2,n] \\ \vdots & \vdots & \ddots & \vdots \\ \mu[m,1]\prod_{j=1}^{n} k[m,j] & \mu[m,2]\prod_{j=2}^{n} k[m,j] & \dots & \mu[m,n]k[m,n] \end{bmatrix}$$

To correct for the non-ideality, multiply the output by another matrix B which is derived by solving

$$\min_{B \epsilon \Omega_B} ||A - B\tilde{A}||_F$$

Ref: E. H. Lee and S. S. Wong, "**Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017

Switched-Capacitor Matrix Multiplier

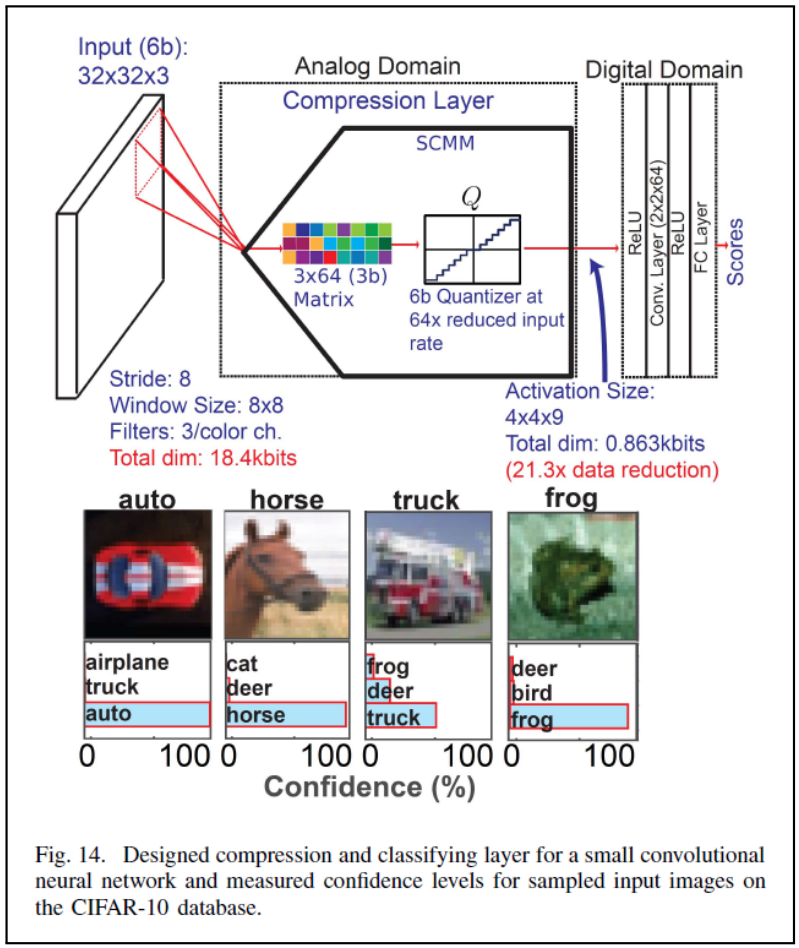


Fig. 14. Designed compression and classifying layer for a small convolutional neural network and measured confidence levels for sampled input images on the CIFAR-10 database.

TABLE I
PERFORMANCE OF THE COMPRESSION LAYER COMPARED WITH THE CONVENTIONAL

	Conventional	This work
Top-3 Accuracy (%)	86	85
Layer's Energy/Op (fJ) at 1GHz	145*	13
# of A/Ds per image at 6b	3072	144
Resolution	6b/4b/6b	Analog/3b /6b
NMSE of feature outputs (avg. over all batches)	0.0033	0.0054
*Energy estimated by synthesis in 40nm Resolution Notation: Input/Weights/Output		

Ref: E. H. Lee and S. S. Wong, "**Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing**," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017

Conclusion

- Depending on the ML algorithm that we want to implement, different circuits can be used
  - For non-linear computations, **sub-threshold** designs are very effective as they are not only energy efficient but also have an exponential equation governing their behaviour. Use of **floating gate transistors** allows us to program such systems.
  - For linear computations such as Multiply and Accumulate (MAC), standard **SRAM memory** can be modified to combine the functionality of storage and computation. Alternatively, **switched-capacitor** designs can be used. While the later may not offer performance boost, it can save a lot of energy.
- These circuits may be useful in emerging style of analog VLSI design, where numerical precision is sacrificed for massive parallelism, collective computation and exploitation of nonlinear circuit properties[1]

[1] T. Delbruck, 1991