SUPPLEMENTARY INFORMATION

Hybrid Architecture based on Two-Dimensional Memristor Crossbar Array and CMOS Integrated Circuit for Edge Computing

Pratik Kumar[1], Kaichen Zhu[2], Suidong Wang[2], Xu Gao[2], Mario Lanza[4*], Chetan Singh Thakur[1*]

[1]Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru, India

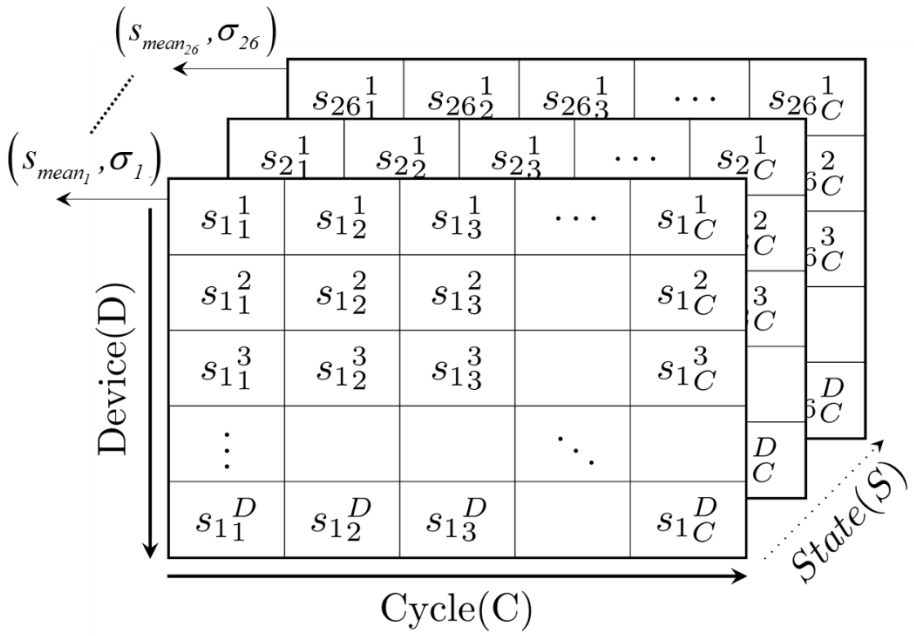
[2] MIND, Department of Electronic and Biomedical Engineering, Universitat de Barcelona, Martí i Franquès 1, E-08028 Barcelona, Spain

[3] Institute of Functional Nano & Soft Materials, Collaborative Innovation Center of Suzhou Nanoscience and Technology, Soochow University, 199 Ren-Ai Road, Suzhou 215123, China

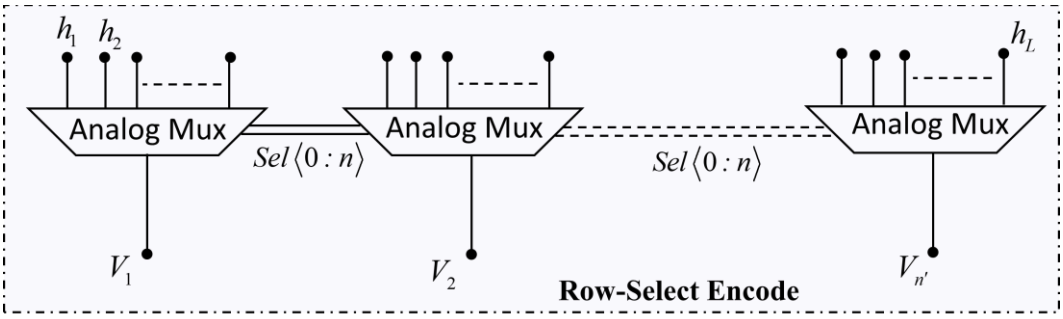
[4] Physical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia

* Correspondence and requests for materials should be addressed to

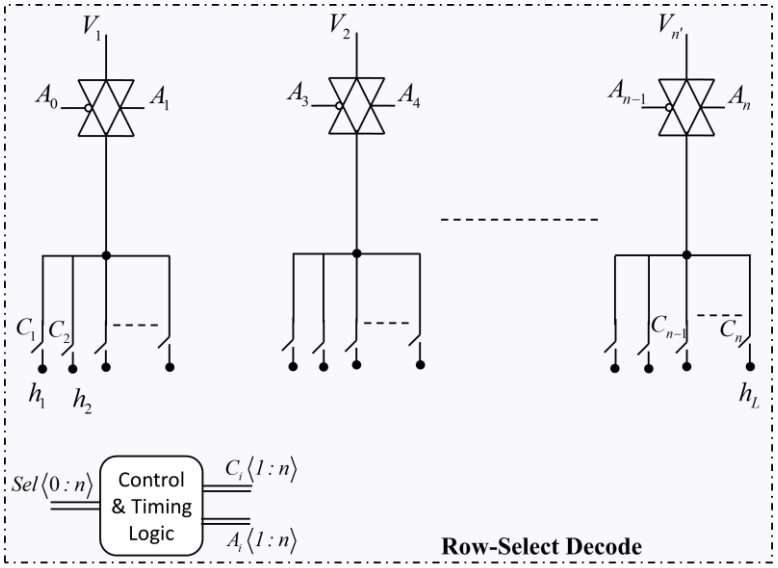
Chetan Singh Thakur (csthakur@iisc.ac.in) or Mario Lanza (Email: mario.lanza@kaust.edu.sa)



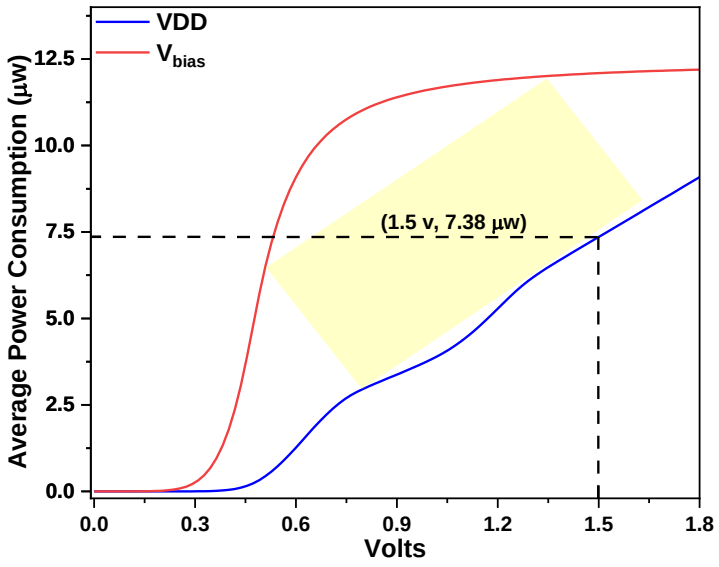
Supplementary Figure 1. Memristor data matrix $M_{Data} \in \mathbb{R}^{D \times C \times S}$ where D is the number of identical fabricated memristor devices C is the number of cycles and S shows the number of memristor states. Here, $\{S_{mean_k}, \sigma_k\}_{k=1}^{26} \in \mathbb{R}^{1X1}$ are the mean and the variance value obtained from the matrix corresponding to each state. For e.g., s_{1C}^{D} is the memristors' conductance data corresponding to D^{th} device and C^{th} cycle.



Supplementary Figure 2. Row-Select Encode circuit architecture for CMOS Encoder chip in Figure 1b of the main text.



Supplementary Figure 3. Row-Select Decode circuit architecture for Memristor Decoder chip in Figure 1b of the main text.



Supplementary Figure 4. Average power consumption plot for one hidden node implementing 9D Gaussian cell as shown in Figure 2a of the main text as a function of VDD and V_{bias}.

Supplementary Note 1: Memristor Data Matrix

To incorporate both device to device and cycle to cycle variability, we include the statistical device variabilities $\{\sigma_k\}_{k=1}^{26}$ while training the algorithm. Supplementary Figure 1 shows the memristor data matrix $M_{Data} \in \mathbb{R}^{D\times C\times S}$ where D is the number of identical fabricated memristor devices C is the number of cycles and S shows the number of memristor states. Here, $\{S_{mean_k}, \sigma_k\}_{k=1}^{26} \in \mathbb{R}^{1X1}$ are the mean and the variance value obtained from the matrix corresponding to each state. For e.g., s_{1C}^{D} is the memristors' conductance data corresponding to D^{th} device and C^{th} cycle.

Supplementary Note 2: Row-Select Encode Circuit

Supplementary Figure 2 shows the CMOS Row-Select Encode unit for the CMOS Encoder chip. The circuit consists of an array of analog multiplexers[1] designed using transmission gates. Here, $h_1,\ldots\ldots,h_L$ denotes the output of L hidden layer neurons and m output (target) nodes. $Sel\langle 0:n\rangle$ is the common select signals to this multiplexer network generated by timing and control unit (see Figure 1b of main text). Depending on the select signals, n' hidden node outputs from $h_1,\ldots\ldots,h_L$ are latched to $V_1,\ldots\ldots,V_{n'}$. With each change in select signal $\left(Sel\langle 0:n\rangle\right)$, the analog multiplexer array scans n' new hidden nodes outputs and latches it to $V_1,\ldots\ldots,V_{n'}$. This process continues till all the hidden nodes outputs are scanned out.

Supplementary Note 3: Row-Select Decode Circuit

Supplementary Figure 3 shows the CMOS Row-Select Decode unit for the Memristor Decoder chip. The circuit consists of CMOS pass transistor switches, charge based control switches with control inputs $C_i\langle 1:n\rangle$, and a control and timing logic. Outputs from CMOS Encoder chip $(V_1,\ldots\ldots,V_{n'})$ are passed as inputs to CMOS pass transistors, which is then propagated to one of the control switches connected to a pass transistor row. Only one control signal $C_i\langle 1:n\rangle$ corresponding to each pass transistor output row is enabled with each select inputs $Sel\langle 0:n\rangle$. Here, control and timing logic generates necessary control signals for CMOS pass transistor and the charge-based switches depending on the select signals $Sel\langle 0:n\rangle$ from CMOS Encoder chip. Depending on the select signals, n' inputs are latched to memristor crossbar array. With each change in select signal $(Sel\langle 0:n\rangle)$, the row-select decoder scans new n' inputs and latches it to the memristor crossbar. This process continues till all the new inputs are scanned in. It can further be noted that the select inputs for Row-Select Encode of CMOS Encoder chip and Row-Select Decode of Memristor Decoder chip are in sync with each other and with the clock input in Memristor chip.

Supplementary Note 4: Maximum Operating Frequency & Throughput

Several high-end applications such as machine learning and edge computing require systems with high throughput to carry a vast number of computations at extremely high speed[2-4]. In general, throughput (in a digital context) is defined as the number of n-bit operations the system can perform

in one second. By definition, the throughput in any genric system will depend on the time it takes to execute given computation per second. This time, in turn, depends on the speed at which the input and output can change and/or respond to this change. The rate of response further depends on the settling time of the circuit. The settling time depends on the designed architecture and the region in which the circuit is operated. It includes dead time, slew time, and the recovery time[5]. It is this settling time that decides the maximum operating frequency at which the MOS part of the system can operate (assuming all the operations are performed parallelly) and is be given by

$$f_{\max} = \frac{1}{\max\left(t_{settling,rise}, t_{settling,fall}\right) + \Delta t} \quad (1)$$

Where $f_{\max}$ is the maximum operating frequency, $t_{settling,rise}$ denotes the time required for the output to settle at either at a rising edge of the reference pulse/clock, $t_{settling,fall}$ denotes the time required for the output to settle at either at a falling edge of the reference pulse/clock. It can be noted that since the designed gaussian circuit is symmetric, the time is taken at the rising and falling edge are assumed to be equal. Here, Δt in (S.1) denotes the margin for the unexpected error that can arise due to parasitics and interconnect. It can safely be assumed to be between 5% of settling time. We obtained the post-layout settling time required to observe an stable change at the output of the MOS gaussian cell implemnted in Fig. 2(a) of the main text and was found to be

$$t_{settling9D} = 706ns \quad (2)$$

To verify the above, we applied a pulse whose on time-period was equivalent to $t_{settling9D}$ at all the 9 inputs in a 9D gaussian cell, and the corresponding settling response at the output was noted and

compared with the expected result. For our case, we fixed the input feature map to 9 in an LRF-ELM architecture. It can further be noted that as the number of cascaded gaussian blocks increases the effective time for the input to reach the output increases and hence the settling time increases. This happens because of the increase in equivalent parasitic capacitance of the system, which decreases the slew rate of the circuit and thus the settling time increases, hence the maximum operating frequency decreases. Using (1) and (2), the maximum operating frequency $f_{\max}$ can be obtained as

$$f_{\max} = \frac{1}{706n + 35.3n} = 1.34MHz \tag{3}$$

Supplementary Note 5: Task Energy Efficiency

Energy per operation and average power consumption are of paramount importance for edge devices where systems having constrained energy requirements. Maintaining low power along with high throughput is a challenge of the present era. Supplementary Figure 4 shows the average power consumption plot as a function of VDD for a fixe bias V_{bias} of 975 mV and as a function of V_{bias} for a fixed VDD of 1.8 V. It can be analyzed that as the bias voltage increases, the overall power consumption of the circuit increases. Hence, both VDD and V_{bias} plays a significant role in deciding systems energy efficiency. For a V_{bias} of 950 mV, power consumption of implemented gaussian cell was found to be 7.38 uW for a VDD of 1.5 V. This, when multiplied with the minimum time required by a node (here 9D gaussian) for executing a task gives task energy efficiency as

$$Energy = p_{avg} \times \frac{1}{f_{\max}} = 7.38\mu \times 741.3n = 5.47\,pJ \tag{4}$$

Supplementary References

1. Mishra, M., & Akashe, S., High performance, low power 200 Gb/s 4: 1 MUX with TGL in 45 nm technology. *Applied Nanoscience*, ***4*(3)**, 271-277, (2014).
2. Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A., Edge computing: A survey. *Future Generation Computer Systems*, ***97***, 219-235, (2019).
3. Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., & Temam, O. (2014). Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. *ACM SIGARCH Computer Architecture News*, ***42*(1)**, 269-284.
4. Narayanan, S., Shafiee, A., & Balasubramonian, R., INXS: Bridging the throughput and energy gap for spiking neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2451-2459). IEEE, (2017, May).
5. Allen, P. E., & Holberg, D. R., *CMOS analog circuit design*. Elsevier, pp.248-249, (2011).