nature electronics

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41928-019-0270-x

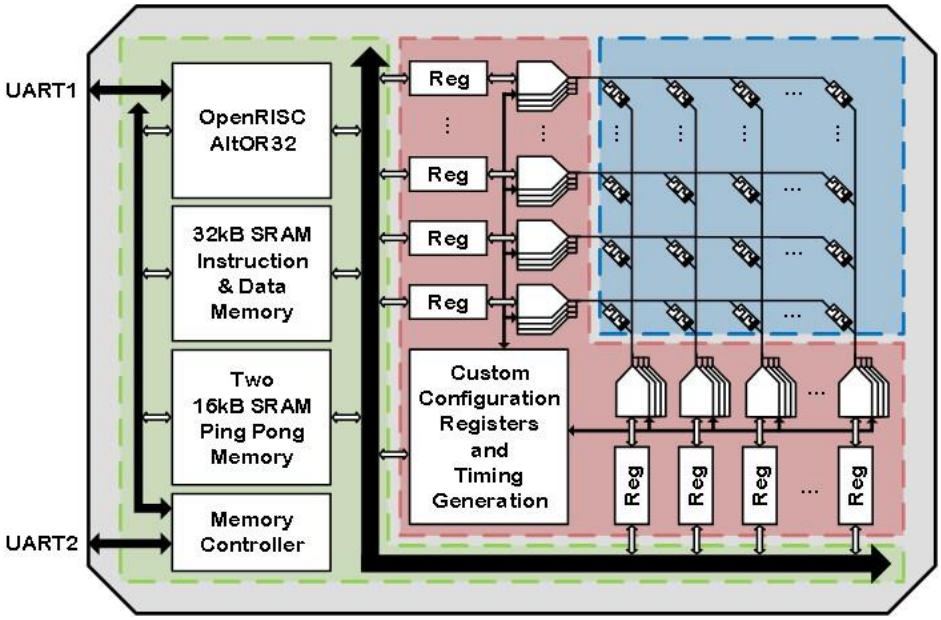
In the format provided by the authors and unedited.

A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations

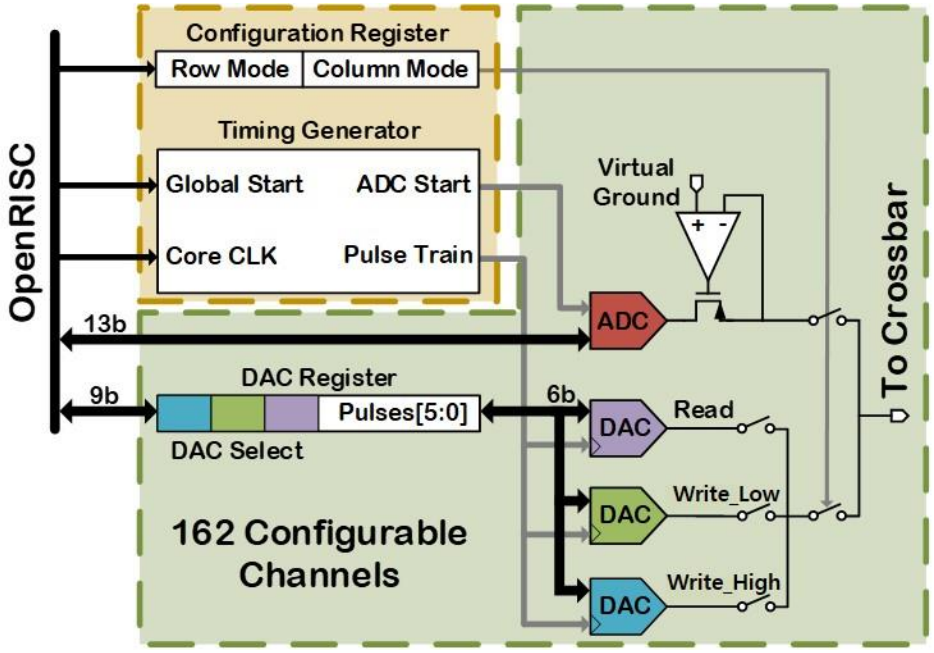
Fuxi Cai[1,3], Justin M. Correll[1,3], Seung Hwan Lee[1,3], Yong Lim[1,2], Vishishtha Bothra[1], Zhengya Zhang[1], Michael P. Flynn[1] and Wei D. Lu[1]*

[1]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. [2]Present address: Samsung Electronics, Yongin, Korea. [3]These authors contributed equally: Fuxi Cai, Justin M. Correll, Seung Hwan Lee. *e-mail: wluee@umich.edu

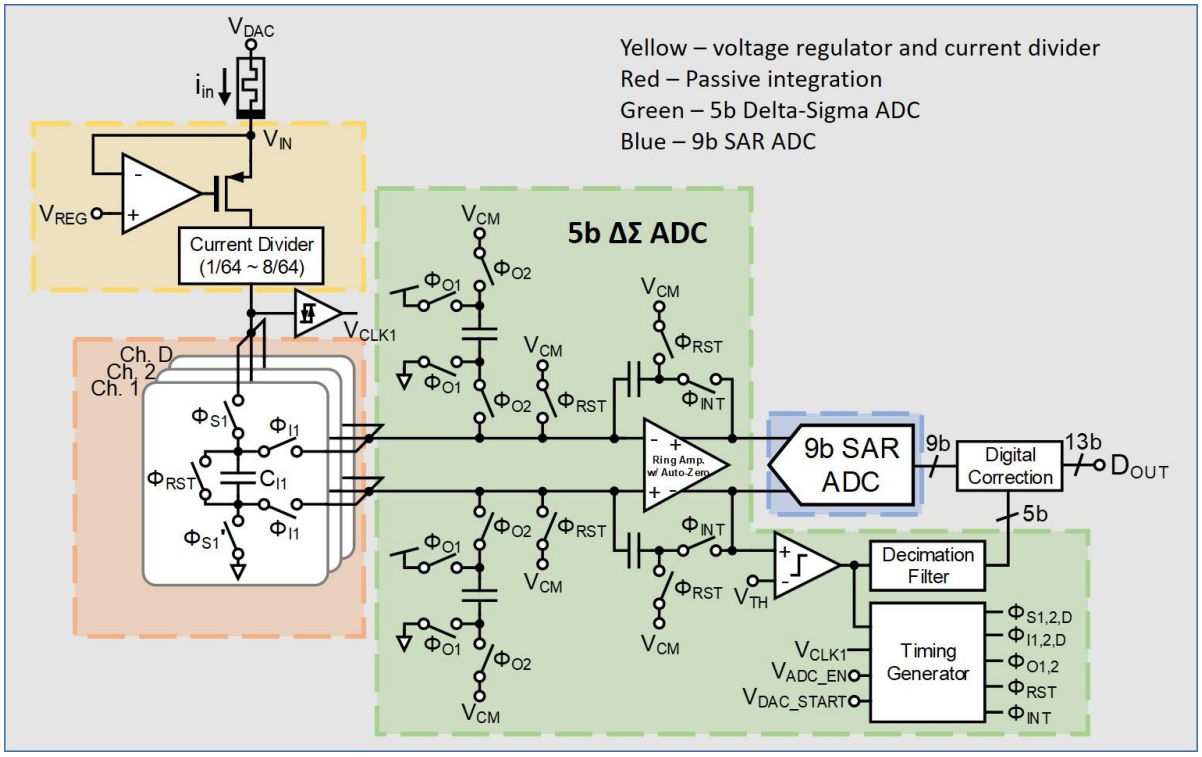
Supplementary Figures



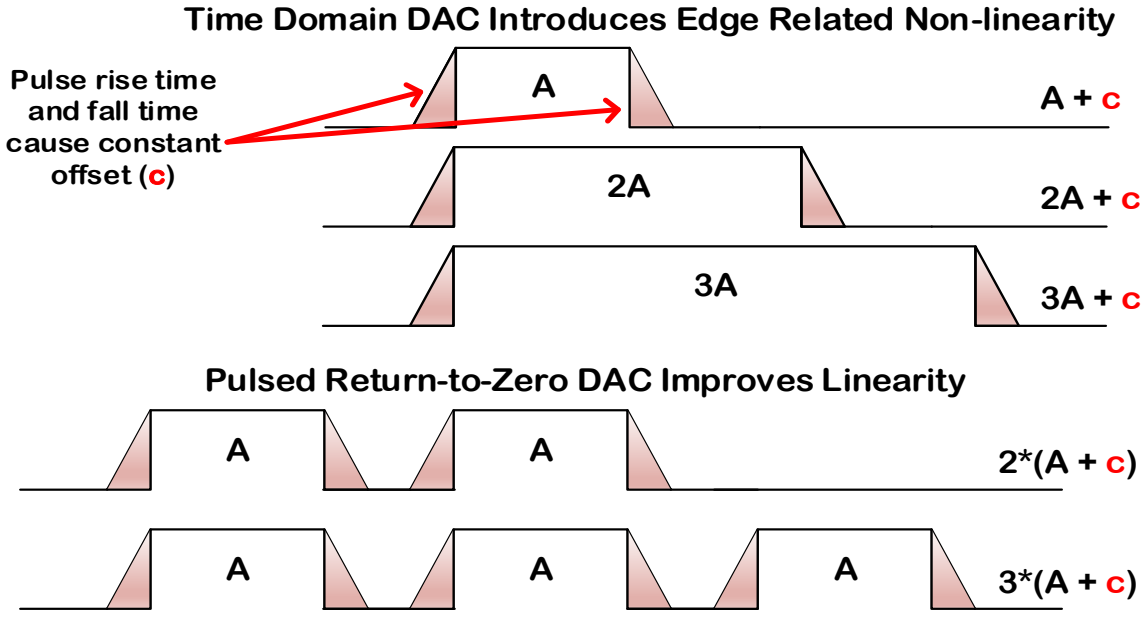
Supplementary Figure 1 | Chip System Architecture. The integrated memristor/CMOS chip is comprised of the digital controller and bus shown in green, the mixed-signal interface shown in red, and the memristor crossbar shown in blue.



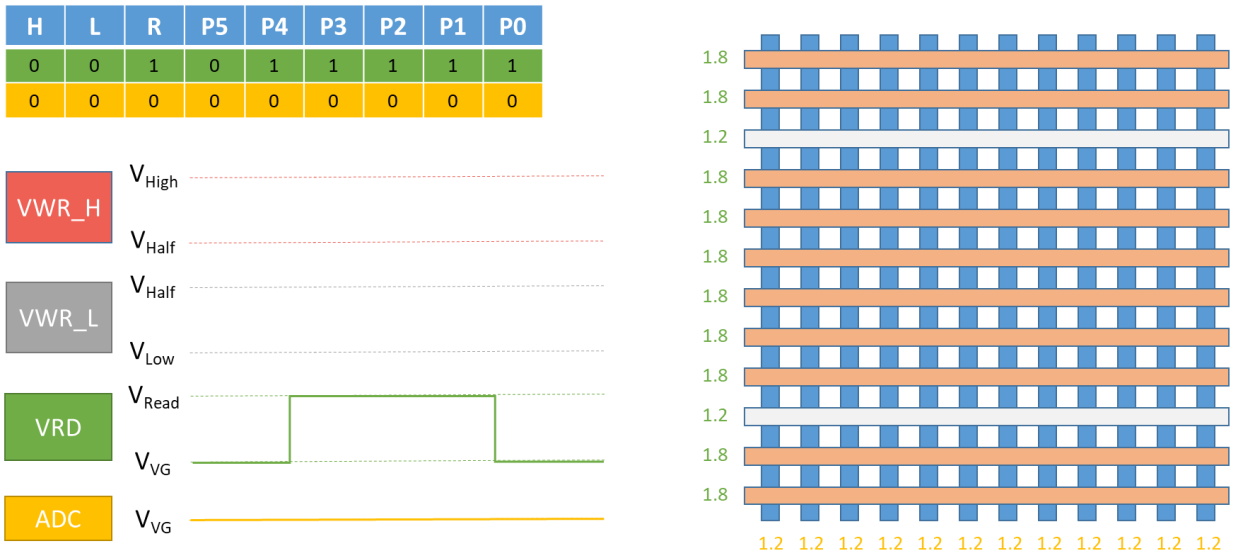
Supplementary Figure 2 | Mixed Signal Interface design. The mixed-signal interface is comprised of global configuration registers and timing generation shown in brown, and 162 configurable channels (shown in green) to provide input and measure output to and from each row and column of the memristor crossbar.



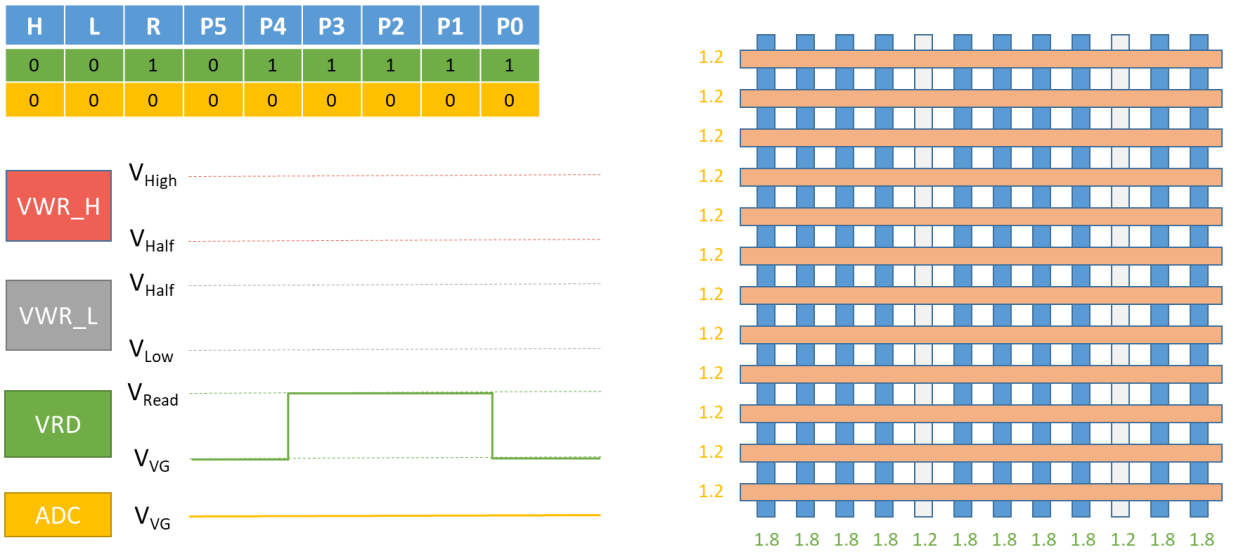
Supplementary Figure 3 | Schematic of 13b current-integrating ADC. The 13 bit hybrid integrating ADC architecture is comprised of a 5b first stage first-order incremental ADC and a 9b second stage SAR ADC with 1b stage redundancy for high resolution, small capacitance and simplified clocking.



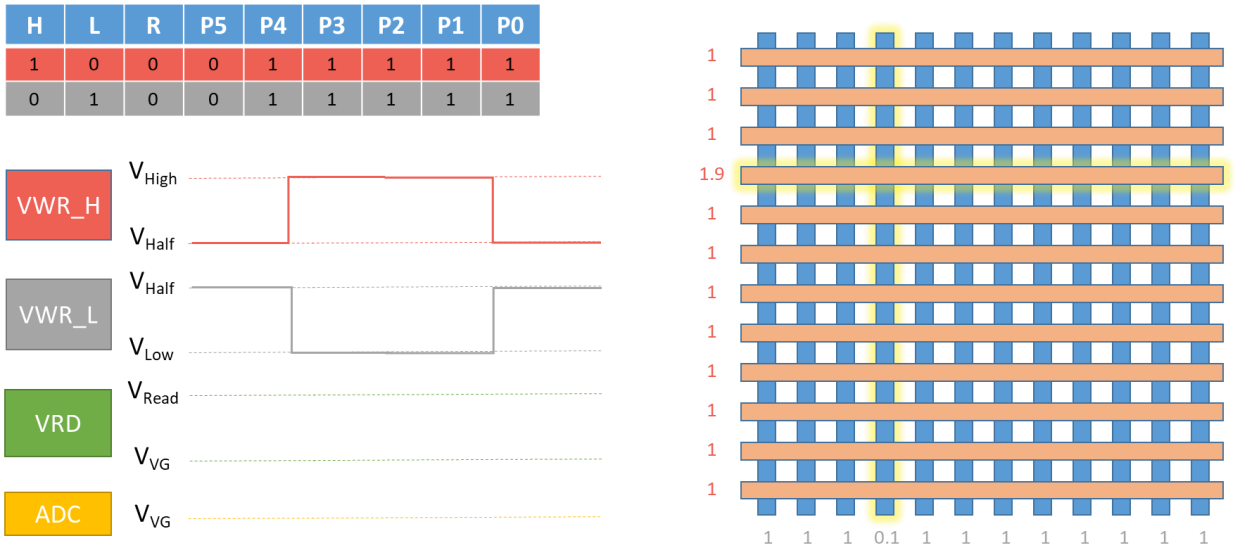
Supplementary Figure 4 | Pulsed-mode DAC scheme. Instead of directly modulating the pulse width, we use the number of pulses to represent the input amplitude (effectively modulating the pulse width in discrete time domain) to eliminate errors due to pulse rise and fall time.



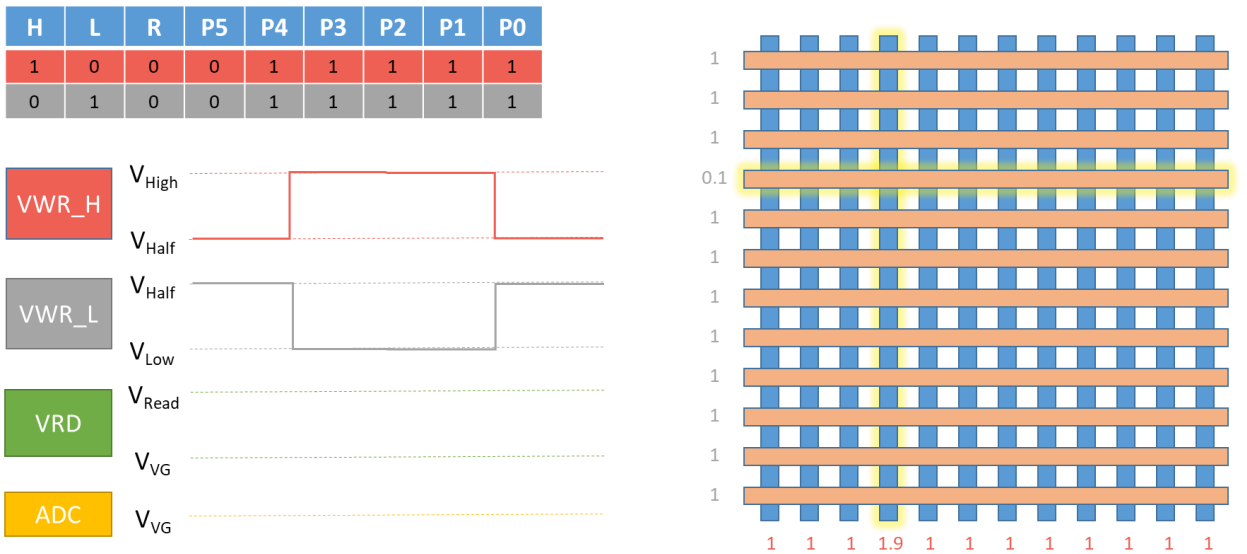
Supplementary Figure 5 | Forward pass mode. The forward pass mode is used to perform VMM to update the output neurons. In this configuration, all rows are connected to the 'Read' DACs, which pulses between 1.2V(V_{VG}) -1.8V(V_{read}) with the input data represented by different pulse widths, and all columns are connected to 1.2V ADC virtual ground (V_{VG}). The colour table lists the DAC register configuration used to control the "Read" DACs and ADCs.



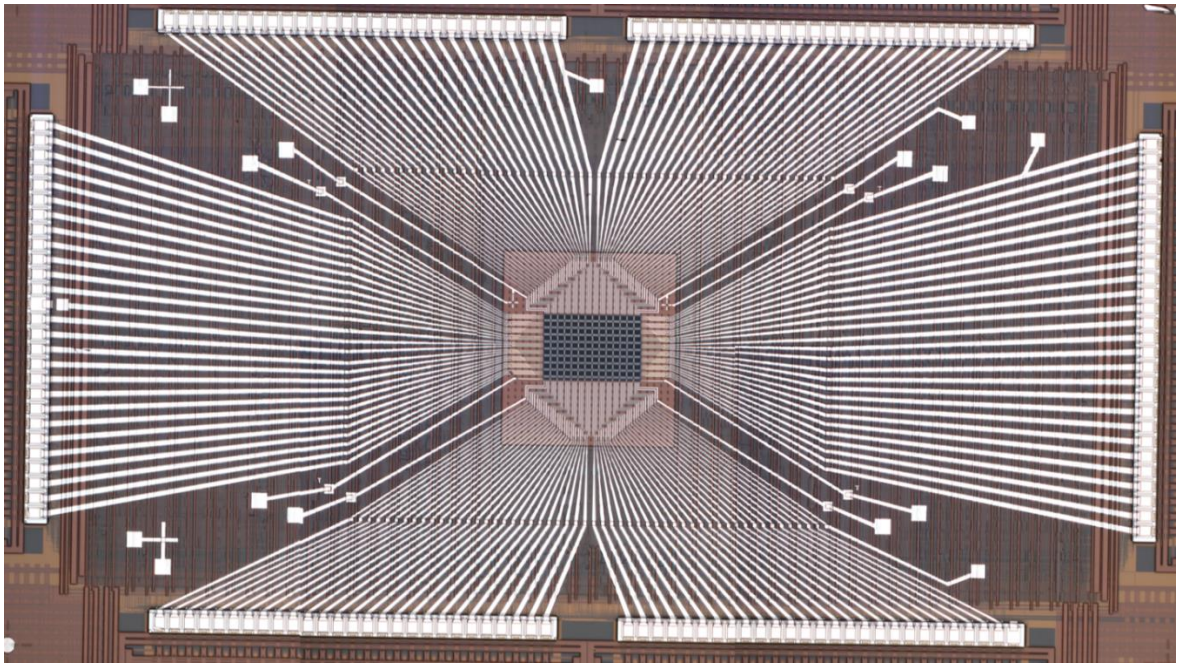
Supplementary Figure 6 | Backward pass mode. The backward pass mode is used for calculating the vector-transposed matrix multiplication. In this configuration, all columns are connected to the 'Read' DACs, which pulses between 1.2V(V_{VG}) -1.8V(V_{read}) with the input data represented by different pulse widths, and all rows are connected to 1.2V ADC virtual ground (V_{VG}). The colour table lists the DAC register configuration used to control the "Read" DACs and ADCs.



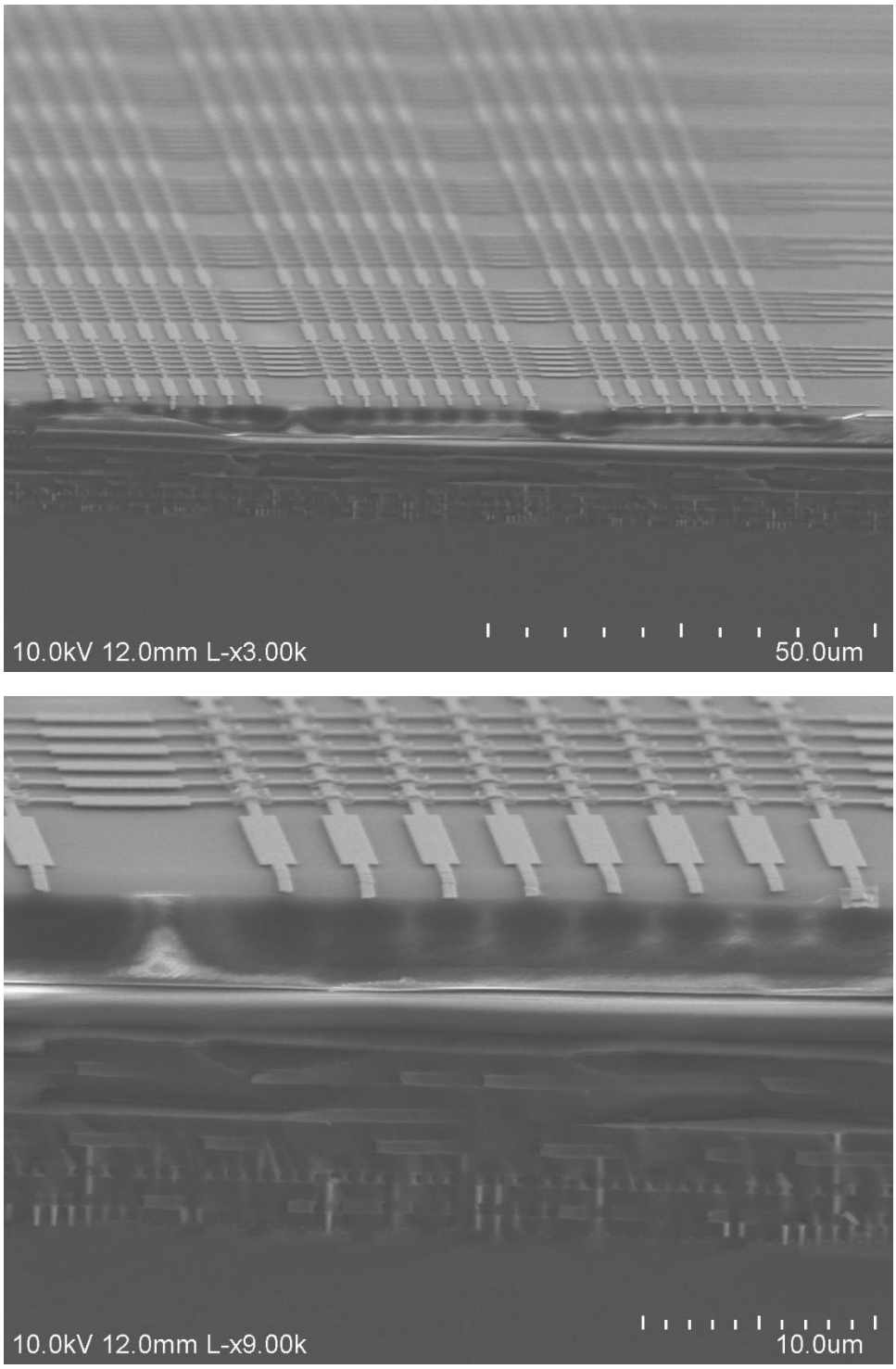
Supplementary Figure 7 | Write mode. The write mode is used for programming the memristor devices to higher conductance states. In this configuration, all rows are connected to the "Write High" DACs (with range 1V(V_{Half})-1.9V(V_{High})) and all columns are connected to the "Write Low" DACs (with range 1V(V_{Half})-0.1V(V_{Low})). The voltage difference 1.8V of the two DACs cross the selected device is used to program the device. The colour table lists the DAC register configuration used to control the "Write High" DACs and "Write Low" DACs.



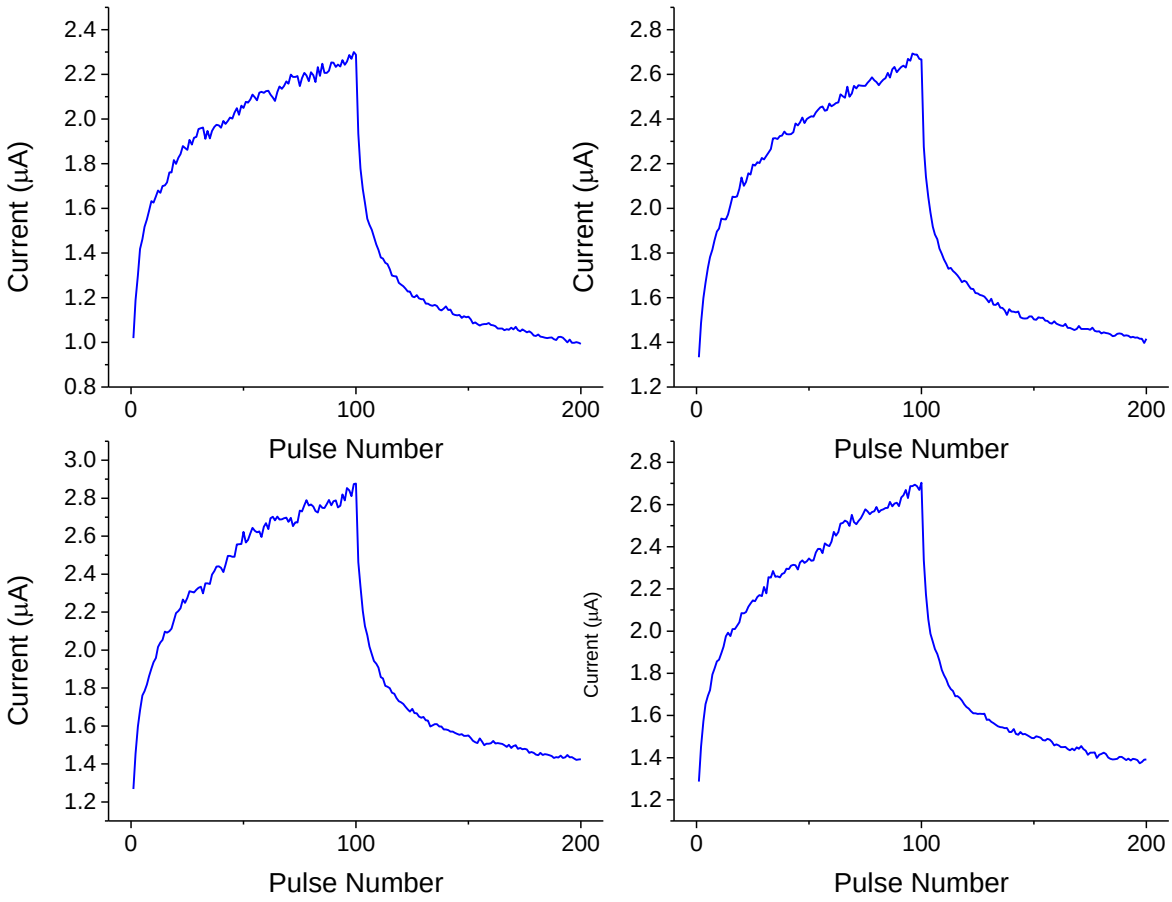
Supplementary Figure 8 | Erase mode. The erase mode is used for programming the memristor devices to lower conductance states. In this configuration, all columns are connected to the "Write High" DACs (with range 1V(V_{Half})-1.9V(V_{High})) and all rows are connected to the "Write Low" DACs (with range 1V(V_{Half})-0.1V(V_{Low})). The voltage difference -1.8V of the two DACs cross the selected device is used to erase the device. The colour table lists the DAC register configuration used to control the "Write High" DACs and "Write Low" DACs.



Supplementary Figure 9 | Top view optical microscope of the integrated chip. The 54×108 crossbar array (center region) and the extension lines are visible. Every row and column is connected to a specific landing pad left open during the CMOS fabrication process.

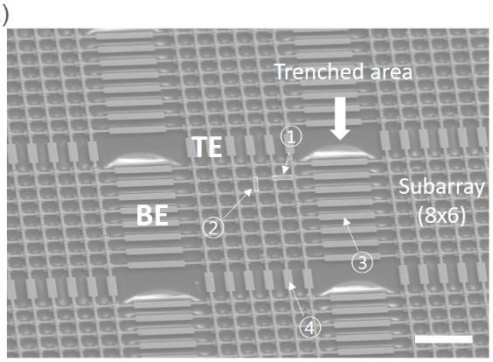


Supplementary Figure 10 | Cross-sectional view SEM images of the integrated chip. The memristor crossbar array is fabricated on top of the CMOS circuits. The different CMOS wiring layers are also visible underneath the memristor array.



Supplementary Figure 11 | Programming memristors on chip. Weight update curves from several memristor devices measured from the crossbar array in the integrated chip. The devices were programmed with 100 write pulse at 1.8V and 100 erase pulses at -1.8V with 82μs pulse width, using the on-chip processor and the integrated DAC/ADC circuitry.

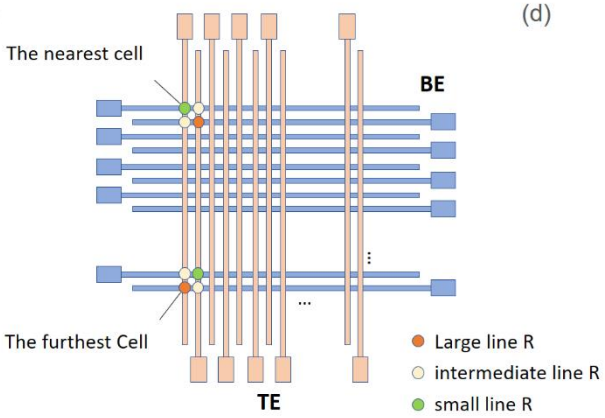
(a)



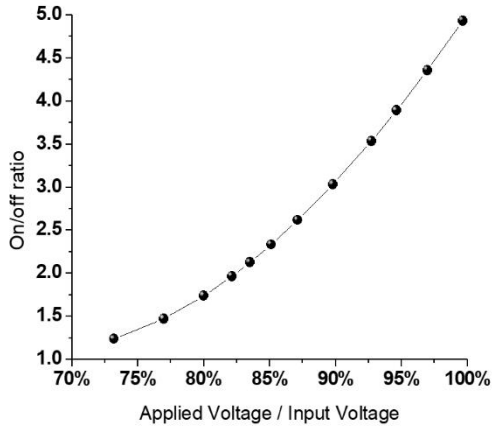
(b)

	Ex-situ integration	Monolithic integration
① BE line R / cell	21.7Ω	2.0Ω
② TE line R / cell	8.4Ω	2.0Ω
③ BE subarray line R	65.1Ω	-
④ TE subarray line R	10.3Ω	-
⑤ BE extension (+ contact) R	300Ω	150Ω
⑥ TE extension (+ contact) R	200Ω	150Ω

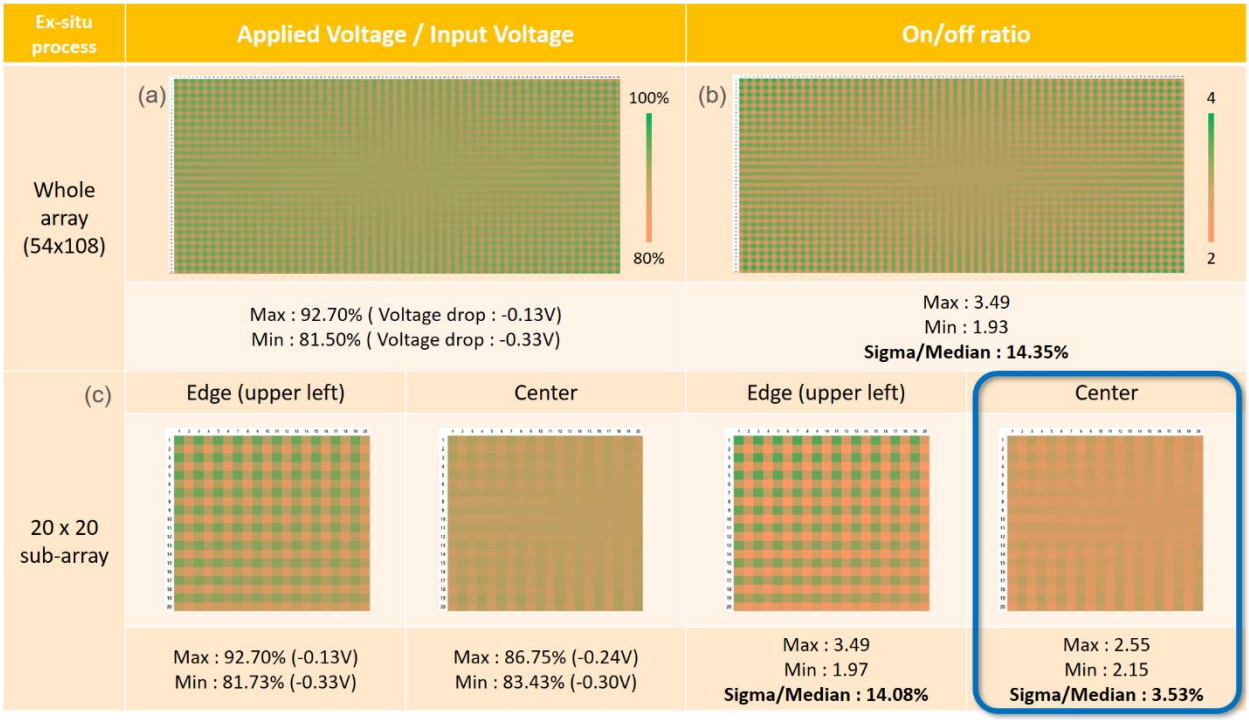
(c)



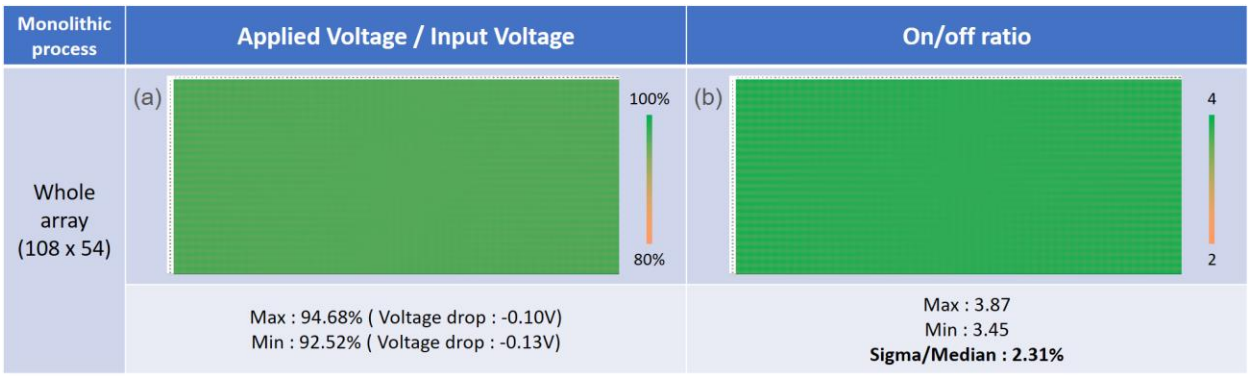
(d)



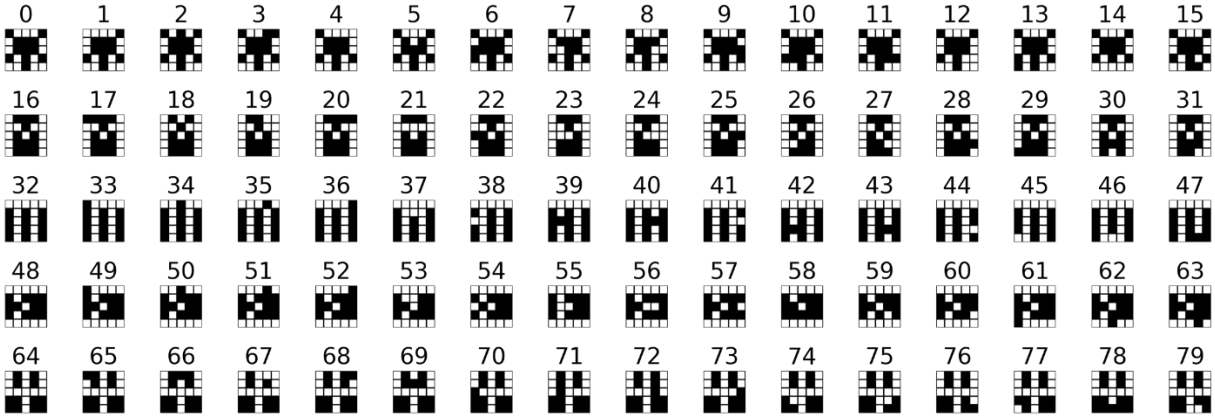
Supplementary Figure 12 | Line resistance effects. (a) Zoomed-in SEM image of the memristor array integrated on top of the CMOS chip. The 54×108 array used in this study is composed of 126 8×6 subarrays to work around the periodic deep trenched areas (~2000Å deep) created from the design rule. (b) Measured BE and TE line resistance values from the integrated array (labeled as ex-situ integration). The line resistances can be significantly reduced if the memristor array can be integrated at the local or intermediate interconnect levels (labeled as monolithic integration). (c) Schematic of the wiring patterns. The total line resistance seen by the device can change significantly even among neighboring devices due to the even/odd arrangement of the pad patterns. (d) Simulated results using the memristor model, showing the measured on/off ratio can be strongly affected by the voltage loss due to the line resistance. The on/off is calculated after 50 consecutive programming pulses.



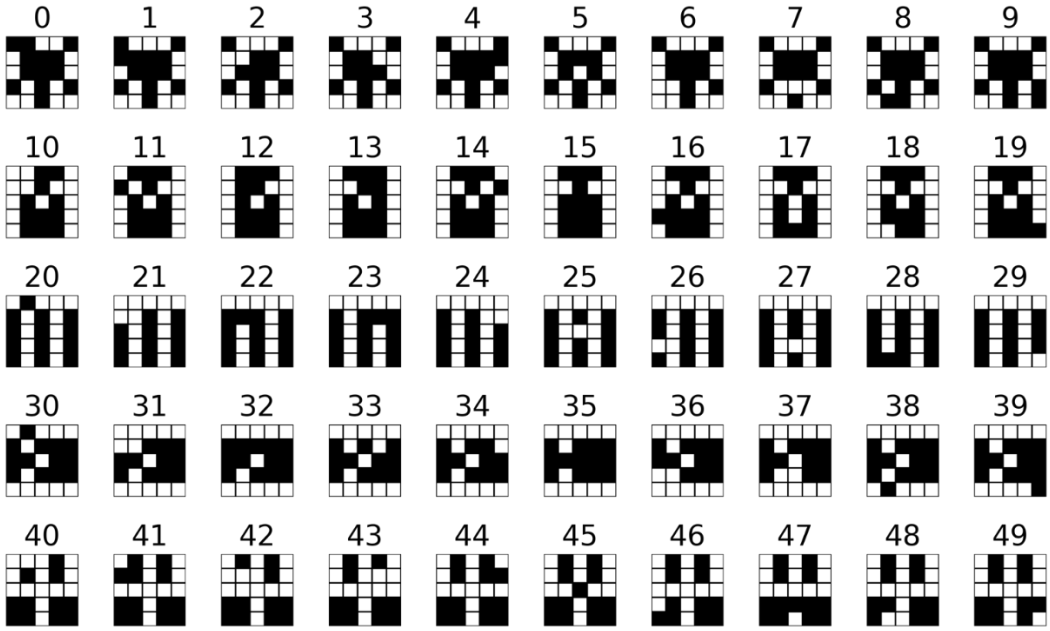
Supplementary Figure 13 | Voltage loss effects in the memristor array. (a) The effect of voltage loss in the integrated 54×108 array, and (b) the resulting on/off ratio affected by the voltage loss effect, obtained from SPICE simulations. The voltage loss and on/off ratio values are represented by the colour labels shown on the right. (c) The voltage loss effect at different regions in the array. The large line resistance effect, combined with the even/odd electrode design, leads to large variations among neighboring cells for cells near the edge. Better uniformity can be obtained for cells in the center of the array, with a reduced on/off ratio.



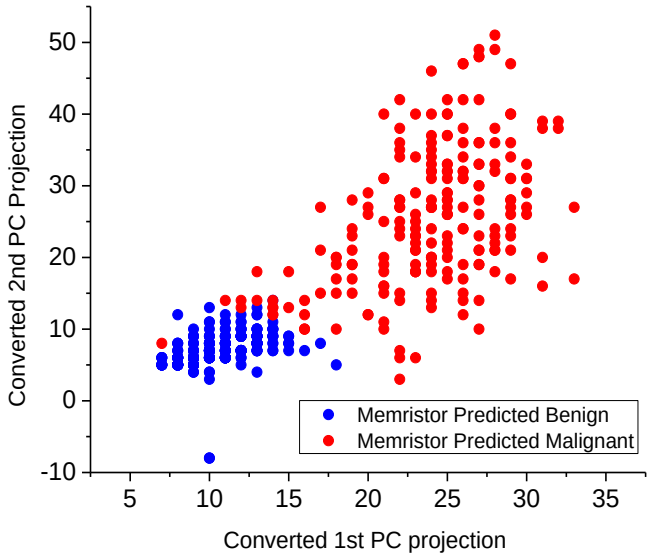
Supplementary Figure 14 | Improvements with monolithic integration. SPICE simulations showing (a) the effect of voltage loss and (b) the expected on/off ratio from a 54×108 array integrated at the local interconnect level, showing much reduced line resistance effect and improved device uniformity.



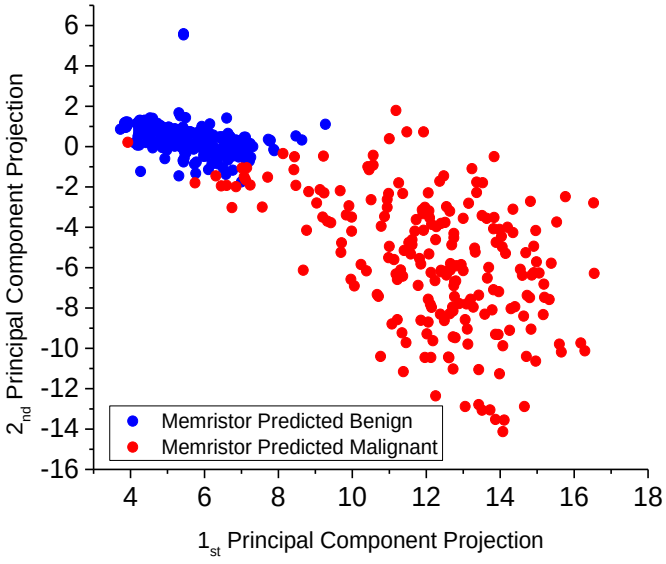
Supplementary Figure 15 | Noisy training data set for the SLP. Training data for 5 different Greek letters used in the SLP implementation. The training data set for each class includes the original image and 15 out of the 25 noisy images created by flipping 1 pixel in the original image.



Supplementary Figure 16 | Noisy testing data set for the SLP. Testing data for the Greek letters used to test the SLP operation after training. The testing data set includes the 10 noisy images not in the training set, created by flipping 1 pixel in the original image for each class.

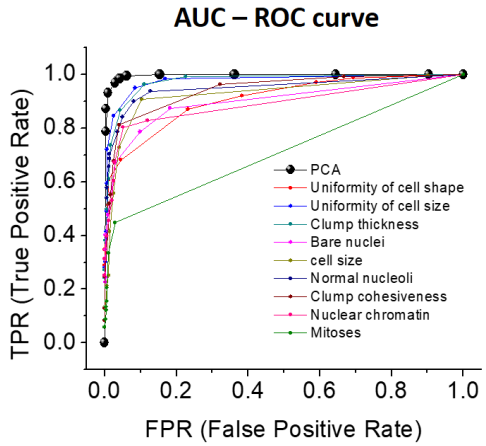


Supplementary Figure 17 | Classification of the quantized PCA data. Outputs from the PCA layer were quantized and scaled to the range of 0~63, and used as input to the second perceptron layer. The quantized data were then classified by the perception layer, with blue points representing the network classified benign data and red points representing the network classified malignant data.



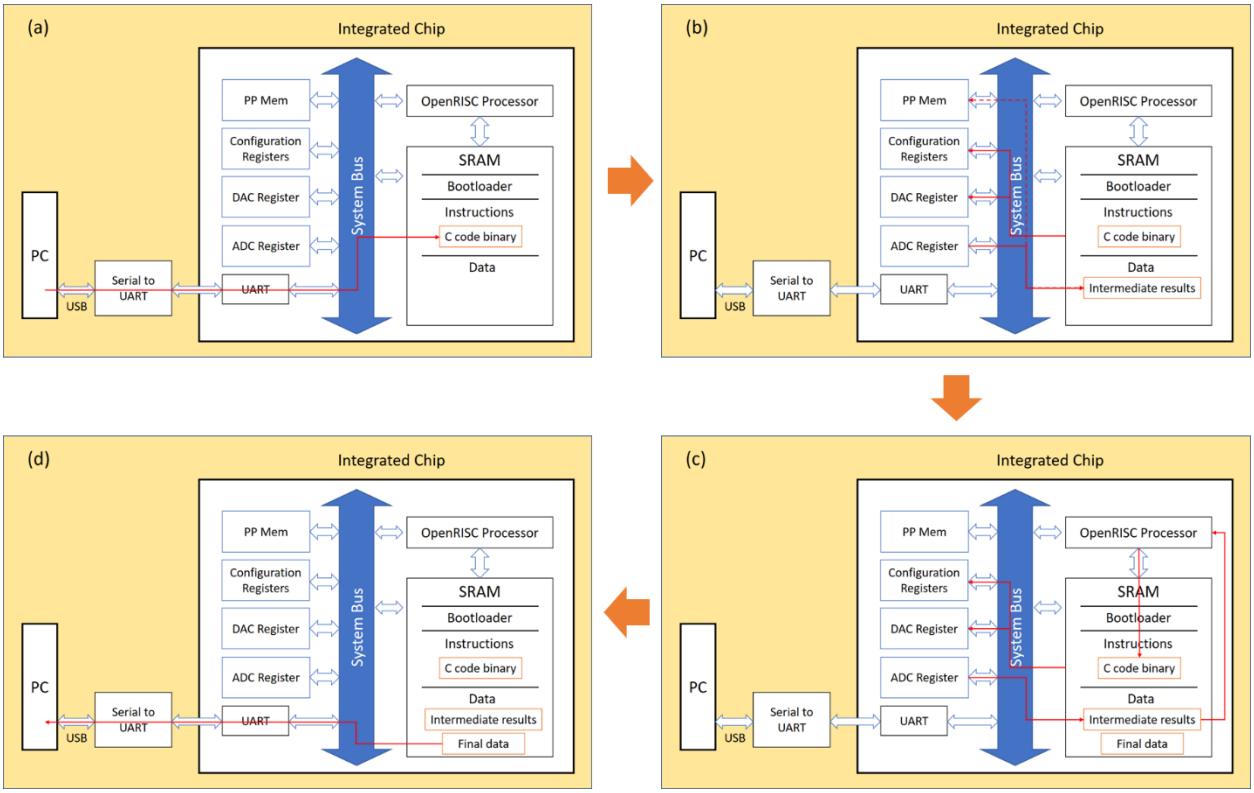
Supplementary Figure 18 | Replotted classification results in the original space. The classified results in Supplementary Figure 17 were then replotted in the original output space

from the PCA layer, with blue points representing the network classified benign data and red points representing the network classified malignant data.

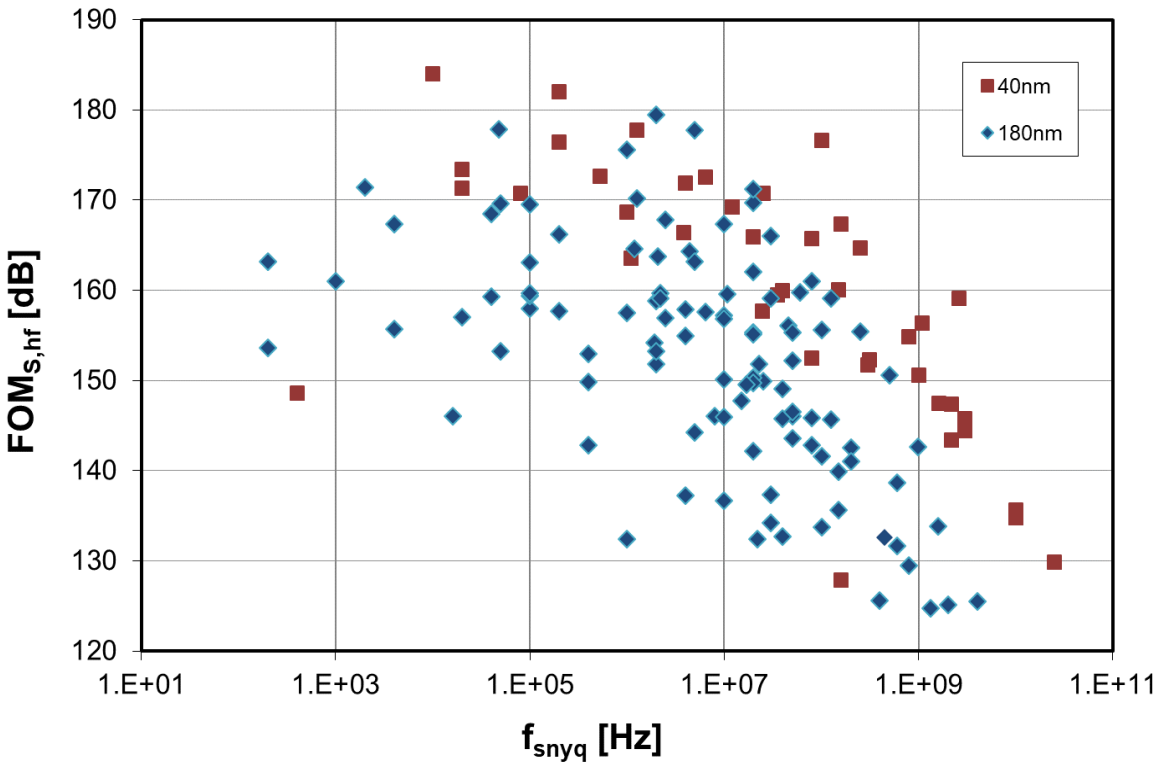


	AUC (Area Under The Curve)	F1 score (best value)
PCA	**0.996**	**0.960**
Uniformity of cell shape	0.909	0.773
Uniformity of cell size	0.976	0.901
Clump thickness	0.975	0.890
Bare nuclei	0.901	0.798
Cell size	0.927	0.865
Normal nucleoli	0.947	0.878
Clump cohesiveness	0.942	0.860
Nuclear chromatin	0.891	0.846
Mitoses	0.712	0.596

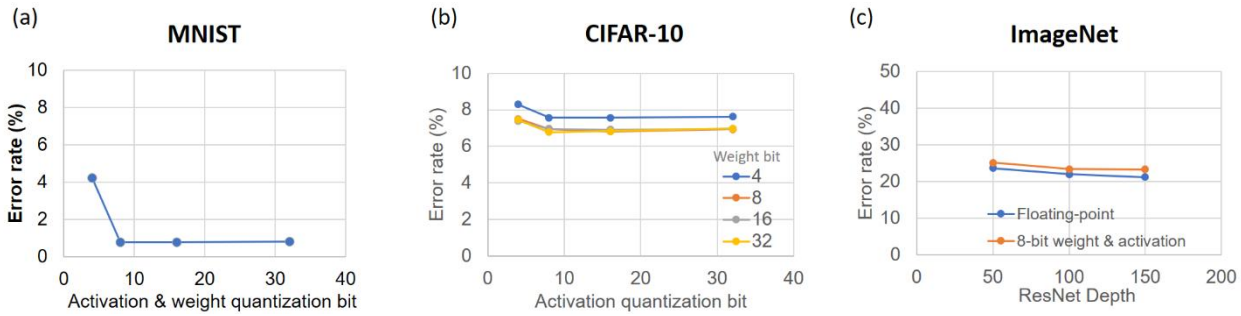
Supplementary Figure 19 | AUC-ROC curve and F_1 score of the breast cancer task. The experimentally PCA + classifier network show excellent AUC value of 0.996 and high F_1 score of 0.960, corresponding to sensitivity, specificity and accuracy values of 93.1%, 99.0% and 94.6%, respectively.



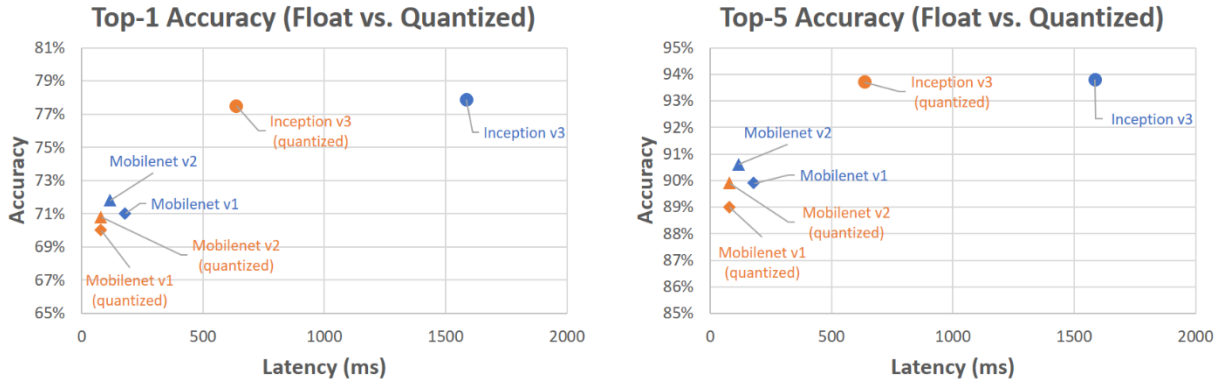
Supplementary Figure 20 | Data path during training and inference. (a) All instructions for necessary mathematical calculations and data storage are first programed in C code and compiled. The entire compiled code is then loaded into the on-chip 32kB SRAM data memory by a bootloader. (b) The binary program instructions are executed through the OpenRISC processor to run the training and inference algorithms. The vector-matrix multiplication (VMM) results are read as charge values from the ADCs. (c) During training, the required weight updates are calculated through the OpenRISC processor. Afterwards, the DACs are configured to supply the desired pulse widths to the update the memristors. (d) The final output can be transferred to a personal computer and accessed by the user.



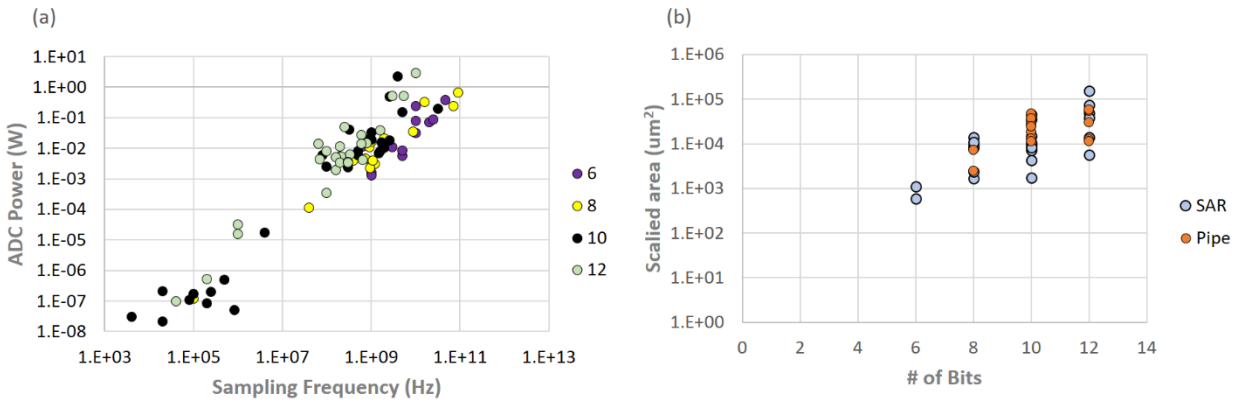
Supplementary Figure 21 | Schreier FOM for 180nm and 40nm ADCs. The Schreier FOM is used for comparing high-resolution ADC performance across architectures. The data are collected from all ADCs published in ISSCC and VLSI conferences from 1997-2018.



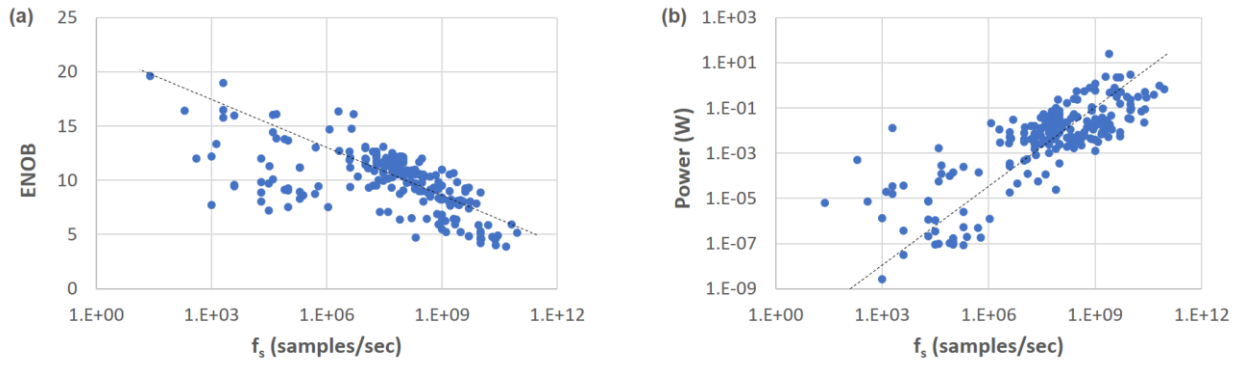
Supplementary Figure 22 | Error rates from different models as a function of weight and activation quantization effects. (a) Error rates obtained using quantized weight and activation, for 32-bit, 16-bit, 8-bit, 4-bit fixed-point for the MNIST data set [S9]. (b) Error rates as a function of activation quantization effect, for weights quantized at 32-bit, 16-bit, 8-bit, 4-bit, for CIFAR-10 [S10]. (c) Floating-point vs 8-bit quantized network error rates for various network depths of the ResNet model, tested on ImageNet [S11].



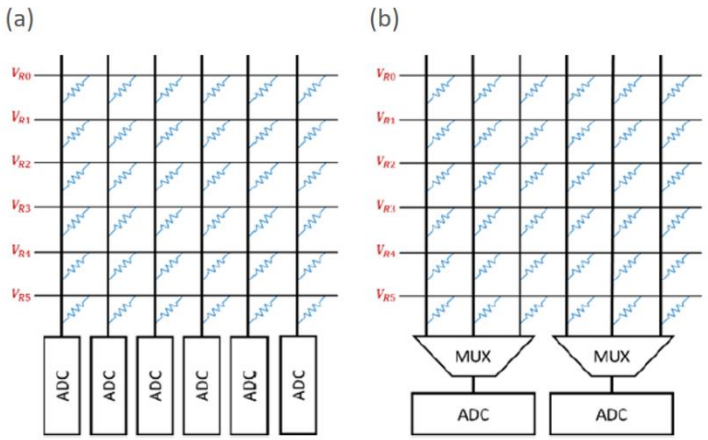
Supplementary Figure 23 | Quantization effect for common CNN models. The models are implemented using 8-bit weight and 8-bit activation. All latency numbers are measured on Pixel 2 cell phones using a single core, adapted from tensorflow.org (https://www.tensorflow.org/lite/performance/model_optimization)



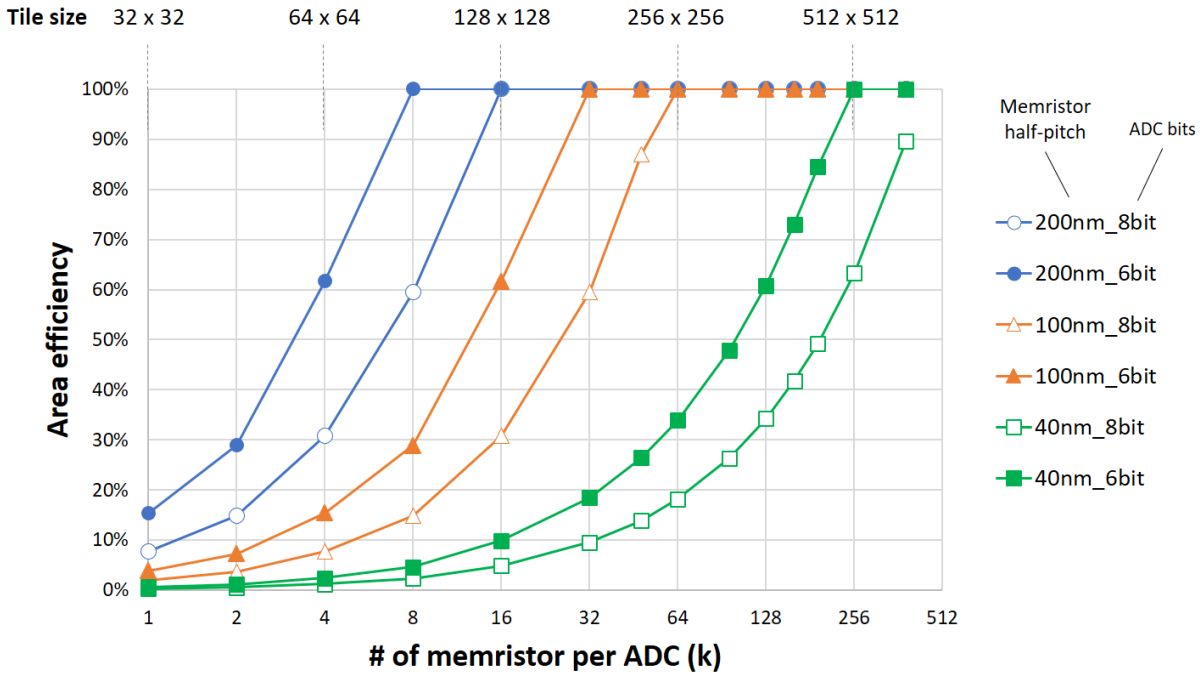
Supplementary Figure 24 | ADC design choices. (a) Speed and power tradeoffs, and (b) resolution and area tradeoffs for reported ADC designs. The data are collected from publications at key technology conferences, and scaled to the 40nm technology node.



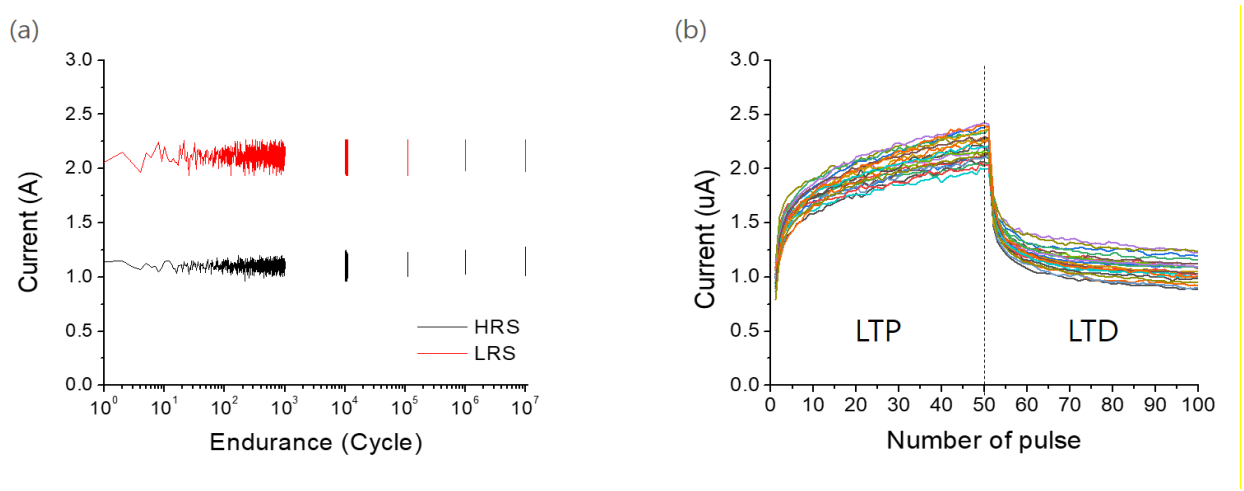
Supplementary Figure 25 | ADC power scaling. Tradeoffs in conventional ADC designs between (a) speed and accuracy, (b) speed and power, as reported in Supplementary Reference [8].



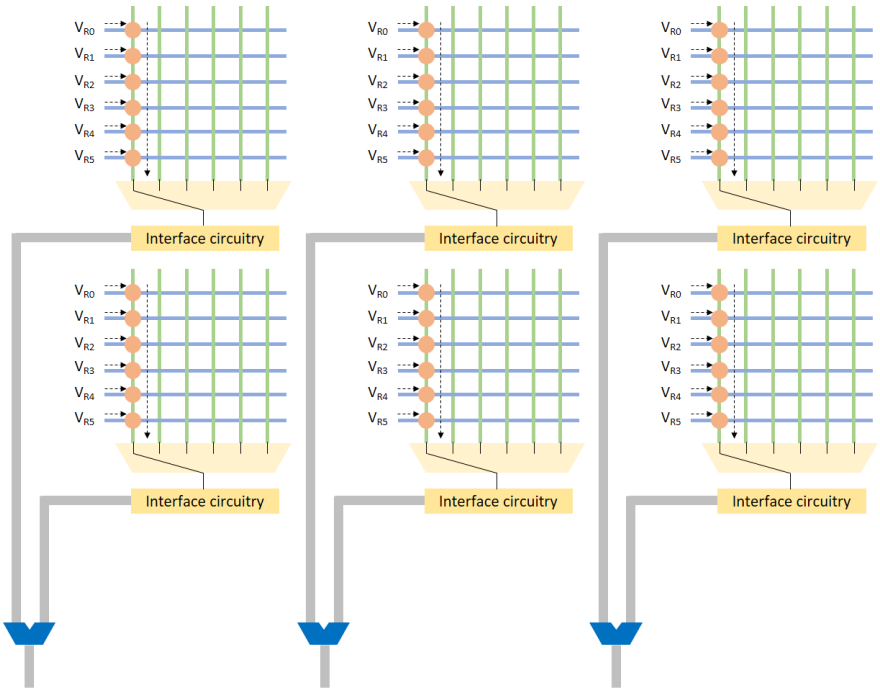
Supplementary Figure 26 | Shared ADC options. Schematics showing dedicated ADC (a) and shared ADC (b) designs.



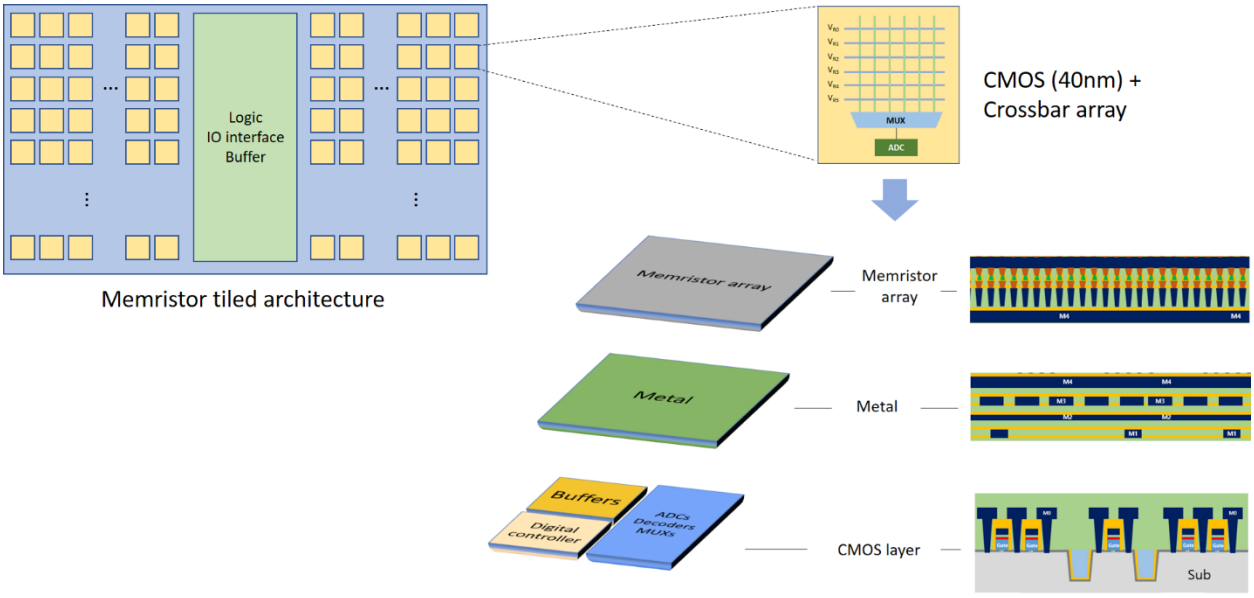
Supplementary Figure 27 | ADC area efficiency. Expected area efficiency for different memristor half-pitch sizes. The ADC is based on conventional 40nm SAR ADC design, for 6 bit [S12] and 8 bit ADCs [S13].



Supplementary Figure 28 | Endurance and variability test of the memristor device. (a) Endurance test, showing HRS and LRS currents during the write/erase processes. After each order of the endurance cycle, the device was programmed and erased another 1000 times, with write/erase pulses of +1.8 V and -1.8 V, 200 μs, respectively. (b) Device to device variability test from 22 stand-alone devices. During the measurement, 50 consecutive programming pulses are applied (LTP test), followed by 50 erase pulses (LTD test). The current is measured at 0.6 V after each pulse.



Supplementary Figure 29 | Tiled architecture based on small crossbar arrays. Every tile consists of a memristor array (e.g. 128 × 128) integrated on top of CMOS circuitry forming a self-contained unit (macro). The tile-to-tile communication is performed in digital domain.



Supplementary Figure 30 | Tiled architecture with high area efficiency. Each macro consists of a small memristor array (128 × 128) integrated with underlying CMOS circuitry (DACs, ADCs, MUXs, register buffer, etc.). Larger systems are built by tiling the macros through digital interfaces.

Supplementary Notes

Supplementary Note 1. Device characteristics

A memristor's resistance is determined by the internal ion configuration [1], which can be modulated by the external electric field and the ion's concentration gradient. The WO_x device's behavior is attributed to the migration of oxygen vacancies (V_Os) inside the device. Specifically, the memristor dynamics can be described by the following equations:

$$I = (1 - w)\alpha[1 - \exp(-\beta V)] + w\gamma \sinh(\delta V) \quad (1)$$

$$\frac{dw}{dt} = \lambda \sinh(\eta V) - \frac{w}{\tau} \quad (2)$$

where Supplementary Equation (1) is the I-V equation which includes a Schottky term (the 1st term) corresponding to conduction in the V_O-poor region and a tunneling-like term (the 2nd term) corresponding to conduction in the V_O-rich region (Supplementary Reference [1]). The two conduction channels are in parallel and the internal state variable *w* represents the relative contribution from the two channels. Supplementary Equation (2) is the dynamics equation which describes the change rate of the state variable *w* with respect to the applied voltage, including the drift effect under an applied electric field (the 1st term) and the spontaneous diffusion (the 2nd term). α,β,γ,δ,λ,η are all positive-valued parameters determined by material properties. τ is the diffusion time constant. In our array-level simulation, a series resistance *R* is also added to the device model.

The gradual conductance changes is due to modulation of *w* through the applied programming voltages, and can be clearly observed by pulse measurements as shown in Supplementary Figure 11. Here 50 write pulses (+1.8 V, 82 µs) were applied to the device, followed by 50 erase pulses (-1.8 V, 82 µs). The device state was monitored by a small read pulse (0.6 V, 82 µs) after each write/erase pulse. All measurements were performed using the on-chip processor and integrated DAC/ADC circuitry.

Supplementary Note 2. System Architecture

The custom CMOS circuitry includes an OpenRISC processor with 64KB SRAM and a mixed-signal interface with 162 configurable channels as shown in Supplementary Figures 1,2. The processor configures the mixed-signal interface through a set of global configuration registers and performs write and read operations through digital to analogue converters (DACs) and analogue to digital converters (ADCs).

Since the processor is mainly used for register manipulations, the reduced instruction set Alternate Lightweight OpenRISC processor (AltOR32) is used in the design to minimize area and power consumption. The SRAM is divided into 3 parts. The processor instruction and data memory are mapped to 32 KB SRAM data memory. The remaining 32KB are dual port SRAMs that support simultaneous input/output and are assigned as two "ping-pong" memory banks for potential data buffering, although in this study the ping-pong memory banks were not used since all data can be fit in the data memory.

The custom mixed-signal interface is shown in Supplementary Figure 2 and includes global timing generation and configuration registers, and 162 configurable channels – all can write/read to/from the Wishbone bus (the shared digital bus in OpenRISC architecture) and mapped to the OpenRISC memory space. Each channel is set to either have an ADC, or 1 of the 3 DACs connected to a row or column of the crossbar. The ADC or DAC connection is set in the mode register and the type of DAC connections is set in the DAC register along with the 6b DAC pulse input. The timing generator handles both the ADC start signal and creates the duty-cycled pulse-train for the DACs.

During operation, the processor is first used to configure the mixed-signal interface by setting the configuration registers. The processor is then paused and the control is handed off to the timing generator of the mixed-signal interface. The timing generator operates for 64 cycles during which VMM operations and memristor weight updates are performed. The control then goes back to the processor.

Supplementary Note 3. Mode configurations on the integrated chip

The integrated chip is highly flexible and can be configured into different modes to facilitate mapping of a complete computing model. Since there are 2 write DACs, 1 read DAC and a 13bit ADC at each row or column, the chip can be configured in four major modes: forward pass mode, backward pass mode, forward write mode (write) and backward write mode (erase).

A global configuration register on the chip controls the write/read mode of all the rows and columns:

MS_GBL_CFG[2]	MS_GBL_CFG[1]	MS_GBL_CFG[0]
GBL_START	ROW_MODE	COL_MODE

For example, at forward pass mode, all rows are configured as DACs and all columns are configured as ADCs, so the ROW_MODE is 1 and COL_MODE is 0. At write mode, all rows and columns are configured as DACs, so the ROW_MODE and COL_MODE are both 1.

The register for the DAC is a 9-bit register that holds the voltage selection data and the read/write pulse durations. The lower 6 bits specifies the pulse durations and the higher 3 bits selects which voltage reference the DAC connects. The register is loaded by the OpenRISC controller via the Wishbone bus. The DAC registers are loaded serially from their associated (addressed) memory locations.

DAC_REG [8]	DAC_REG [7]	DAC_REG [6]	DAC_REG [5-0]
VWR_H	VWR_L	VR	PULSE_DUR

The configuration of the DAC registers at four different modes and the corresponding pulse signals are shown in the tables in Supplementary Figures 5-8. The four different colours of the DAC registers (red, gray, green and yellow) reprent the corresponding 3 DACs and 1 ADC the register is used to configure.

When performing VMM, the chip is configured at read mode (forward or backward depends on the eqaution of the operation). During the VMM operation, we apply a discrete-time pulse-train input and measure the accumulated charge from each column (row). The column

(row) ADCs present a 1.2 V virtual ground while the row (column) DACs apply 6-bit programmable train of fixed-amplitude 0.6 V “read” pulses (1.8 V-1.2 V). The integrating ADCs measure the collected charges over the input period.

When performing weight update, the chip is configured at write/erase mode (depends on the need to increase/decrease the weight). During the write operation, we apply discrete-time pulse-trains at both rows and columns. When writing (erasing), the row (column) DACs apply 6-bit programmable trains of fixed-amplitude “High voltage” pulses (1.9 V-1 V) and the column (row) DACs apply 6-bit programmable trains of fixed-amplitude “Low voltage” pulses (0.1 V-1 V) with the same duration, which effectively generate pulses with 1.8 V (-1.8 V) voltage drop across the device. The idle level of both “High voltage” and “Low voltage” are chosen at 1 V, which is the halfway of the voltage drop to provide write protection for unselected devices in the array.

Supplementary Note 4. SPICE simulation for device variability with line resistance

To analyze the effect of the series line resistance on the array operation, we performed detailed SPICE (Simulation Program with Integrated Circuit Emphasis) simulations using the measured line resistance values and a dynamic memristor model (Supplementary Equation (1) and (2)) to estimate the voltage loss effect. The parameters used in the simulation are listed below.

Parameter	Value	Parameter	Value
α	9e-7	β	4
γ	2.8e-7	δ	6
λ	0.045	η	6
τ	10	w_{max}	1
w_{min}	0	R	400Ω

Our SPICE simulations show that although this effect is relatively small during the VMM operation due to the high memristor resistance (~ 120-580 kΩ) at 0.6 V used for the inference stage, the voltage loss can be significant at the write voltage (1.8 V) due to the much lower device (~ 1.7-8.2 kΩ) at the higher voltage due to the non-linear I-V characteristics of the device. In the worst case (i.e. the target cell is furthest from the voltage supply), the voltage loss due to the line resistance is around 0.33 V, corresponding to a 18.5% drop of the write voltage. The reduction of the applied voltage leads to significantly reduced device response, as shown in Supplementary Figure 12d.

Analysis of the integrated array show large device variations can be observed when trying to program the device, due to the different line resistances the devices see, as shown in Supplementary Figure 13. Near the edges, the effect is significant due to the even/odd wiring patterns, as seen in Supplementary Figure 12b and 13c. To achieve more uniform device response, we chose the center area of the array in our studies, where the voltage drop due to line resistance is roughly similar (Supplementary Figure 13c) that allows us to reliably program the devices using the open-loop method.

We emphasize that this choice is a compromise due to the limitation of cost and the university fab facility. Future studies that can allow more direct integration at the local or intermediate interconnect levels (which we termed monolithic integration) will effectively

address this issue. For example, our detailed SPICE simulations show that monolithic integration will greatly reduce the line resistance and minimize measured device variability, and allow larger arrays to be successfully operated, as shown in Supplementary Figure 14.

Supplementary Note 5. Mapping memristor conductance to synaptic weight in PCA

A linear conversion is used to map the memristor conductances in the array to the synaptic weights g_{ij} with range ([-1 1]) used in PCA, by using the relationship:

$$g = \frac{ADC - a}{b} \tag{3}$$

where *ADC* is the unconverted ADC output from the circuit, which is converted to the current/conductance value through factors *a* (ADC shift factor, which is about 1900) and *b* (ADC scaling factor, which is about 1500) in Supplementary Equation (3). The conversion based on Supplementary Equation (3) maps the maximum average current to weight 1 and minimum average current to weight -1.

For each input data, the dot-product of the input data and the *j*th feature, y_j is directly obtained from the ADC output of the j_{th} column in the 9×2 weight matrix. The column's weights are then updated based on Sanger's rule:

$$\Delta g_{ij} = \eta y_j \left(x_i - \sum_{k=1}^{j} g_{ij} y_j \right) \tag{4}$$

During training, the desired weight updates are linearly converted into write pulse widths and applied to the memristor devices, without using nonlinear compensation schemes such as the one discussed Supplementary Reference [2]. Device nonidealities including the nonlinear weight updates (as shown in Supplementary Figure 11) caused the experimentally obtained training results to differ from the software results (Supplementary Reference [3]), as evidenced in Figs. 5b-e in the main text.

Supplementary Note 6. Scaling of the PCA layer output as perceptron layer input

The on-chip DACs offer 6-bit resolution and can generate pulses of range 0~63. However, after the PCA process, the 2-D projected data have analogue and negative values. To use the PCA output data as input to the second, perceptron layer, we need to rescale the data to the range of 0~63 and quantize them into pulse numbers.

As can be observed from Fig. 4g and 4h in the main text, most of the PCA output data are located in the range of 3~25 in the x axis and -15~3 in the y axis. To quantize and scale the data to the range of 0~63 for the perceptron layer, the following formulas are used:

$$\hat{x} = round[2x] \tag{5a}$$

$$\hat{y} = round[-3(y-3)] \tag{5b}$$

Notice that some outlier data points would produce values larger than 63. These few points were mapped to 0 pulses so that all other points can use as much as the dynamic range of 0~63 as possible.

After quantization and scaling, the data are classified using the perception layer, and the results are shown in Supplementary Figure 17. The labels are directly obtained from the neuron outputs in the perceptron layer, without reading the weight values. Finally, data were replotted in the original PCA output space, using the obtained corresponding label for each data point, as shown in Supplementary Figure 18.

Supplementary Note 7. Evaluation of the PCA + classifier network

Beyond classification accuracy, the following statistical parameters can be used to further evaluate the performance of the network:

- Condition positive (P): the number of real positive cases in the data
- Condition negative (N): the number of real negative cases in the data
- True positive (TP): Sick people correctly identified as sick
- False positive (FP): Healthy people incorrectly identified as sick
- True negative (TN): Healthy people correctly identified as healthy
- False negative (FN): Sick people incorrectly identified as healthy
- Sensitivity (the true positive rate (TPR) or recall): The proportion of actual positives that are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition). i.e. TP/(TP+FN)
- Specificity (true negative rate (TNR)): The proportion of actual negatives that are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition). i.e. TN/(TN+FP)
- Precision (Positive Predictive Value (PPV)): measure of the classifier exactness. i.e. TP/(TP+FP)

The receiver operating characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR, which equals to (1 – specificity)) at various threshold settings. The area under curve (AUC) represents the degree or measure of separability. It shows how much the model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. By analogy, the higher the AUC, the better the model is at distinguishing between

patients with disease and no disease. Therefore, the AUC - ROC curve is an important performance measure for a classification problem at various thresholds settings.

The F_1 score is a measure of a test's accuracy. It considers both the precision (PPV) and the recall (TPR) of the test to compute the score: PPV is the number of correct positive results divided by the number of all positive results returned by the classifier, and TPR is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F_1 score is the harmonic average of the precision and recall, where an F_1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

The sensitivity, specificity and accuracy for our experimentally implemented PCA + classifier network were calculated to be 93.1%, 99.0% and 94.6%, respectively. From the AUC-ROC curve shown in Supplementary Figure 19, the AUC of classification using the learned principal components is 0.996, suggesting the network is almost perfect at distinguishing between the positive class and the negative class. The F_1 score of the network is 0.960, proving the network has excellent precision and sensitivity for breast cancer evaluation.

Supplementary Note 8. Data path during training and inference and on-chip implementation of the neuron function

As shown in Supplementary Figure 20, at the first step, we program all instructions for necessary mathematical calculations and data storage in C code, and compile the C code into binary machine code. This process is performed in a personal computer. Note the programming and compilation only need to be performed once for every task. The binary code will be used to run the training and inference algorithms, and allocate the memory space for the necessary input & output data and all the internal intermediate variables.

Afterwards, we load the final binary code to the chip. Specifically, the binary code is sent through the USB port of the computer, and a Serial-to-UART (Universal Asynchronous Receiver-Transmitter) board converts the serial data in the USB to the UART protocol and sends the compiled code to the memristor chip through the UART interface. The entire compiled binary code is loaded into the on-chip SRAM by a bootloader.

After the binary code is loaded into the on-chip SRAM, the binary program instructions are executed, and the entire application will be run on chip. The instructions set the circuit configurations and DAC registers (i.e. row/column configurations and the pulse widths for weight updates and VMM operations). The VMM results are read as charge values from the ADCs and stored in the data memory.

Then, the VMM outputs are processed by the on-chip OpenRISC processor to calculate the required weight updates or for running other operations of the algorithm. The processor executes the algorithm as programmed by the C code and sends the results (i.e. batch update values) as instructions to the on-chip registers, and the DACs are configured to the new configurations for the next operations.

Finally, after all the instructions have been executed, the output can be supplied to the user through the Serial-to-UART interface.

We note in the networks we run most of the operations are multiplication and summations that can be readily implemented in the memristor array and the OpenRISC processor. There are two functions that require more complex calculations, namely the Sigmoid (for neurons in the PCA classification layer) and Softmax (for neurons in the SLP classification

layer) activation functions. The Sigmoid is a one-to-one mapping function with can be approximated by a piece-wise linear function. As a result, we have successfully implemented the Sigmoid function on chip using the OpenRISC processor, and all experiments for the LCA network and the entire PCA + Classification network are implemented on chip, without communication with the computer during the learning and testing process. However, the Softmax function is a n-to-n mapping that computes the ratio of an exponential output of the current class over the sum of the exponential outputs of all classes, which is challenging to implement without floating point operations.

In this prototype integrated chip, the OpenRISC processor we use is a striped down version called "AltOR32" (Alternative Lightweight OpenRISC 32-bit version) without the floating point unit, to reduce power, area and design costs since most of the calculations performed in our algorithms are multiplication and summations. The decimal values of the parameters were approximated with fixed point representation in the integer domain.

Therefore, in the Greek letter classification experiment that uses the Softmax function during the training process, we made the compromise of calculating the Softmax activation off chip in software (Python), the activation value was then passed back to the chip for batch-gradient descent calculations. We note the current version of the system is developed as a prototype that demonstrates the integration of memristor arrays with functional periphery circuitry. Not all functions, such as floating point support, were targeted. Designs that include floating point unit supports, as well as improved training protocols and more efficient processor designs [S4-S6] will be incorporated in future studies.

Supplementary Note 9. Power Estimate

The power consumption of the integrated system consists of three parts: the digital OpenRISC core, the mixed signal interface, and the power consumed by the passive crossbar array.

Both the digital processor power and the mixed signal interface power are directly measured experimentally, by measuring the root mean square (RMS) current with a Fluke meter while running the chip. At the maximum frequency of 148 MHz, the digital power reads 235.3 mW and the total analogue power reads around 64.4 mW. The crossbar array power is obtained from the average device current at the read voltage, which yields ~7 mW for the 54×108 array. The total power at 148 MHz clock is thus 306.7 mW for the current chip based on 180 nm CMOS technology.

The energy efficiency is estimated using the 148 MHz clock speed and an average 4-bit input during inference, which gives around 9.87M VMM operations per second. Multiplying this number by 54×108 leads to 5.75×10^{10} operations per second. Therefore, the energy efficiency can be derived by dividing the number of ops/second with the total power, which results in 187.6 GOPS/W for the current memristor/CMOS chip.

The custom circuitry was designed in 180nm CMOS and features a generic digital processor along with a full set of mixed signal analogue to digital converters (ADC) and digital to analogue converters (DAC). Two different approaches were used to estimate the power dissipation at the 40 nm technology node. The digital power was estimated using generalized scaling (Supplementary Reference [7]) and the mixed-signal power was estimated using a figure of merit (FOM) approach.

In digital circuits, the length scaling factor S, and the supply voltage scaling factor U are different during scaling. Specifically, from 180 nm to 40 nm, S = 180 nm/40 nm = 4.5, U = 1.8 V/1.0 V = 1.8. As a result, the digital power is reduced by a factor of $1/U^2 = 0.32$, while the circuit speed is improved by a factor of S = 4.5. Note the faster processor allows the same process to control more channels, so normalizing to the same number of 162 channels, the digital power at 40 nm is estimated to be

$$P_{40nm} = \frac{P_{180nm}}{U^2 S} \tag{6}$$

Using the measured digital power at 180nm, the estimated digital power at 40 nm is then 235.3 mW/((1.8)^2×4.5), which is about 16.1 mW.

The analogue to digital converter performance is evaluated using a figure of merit (FOM) approach. An ADC FOM combines several converter performance metrics into one number for comparison across ADC architectures. The Schreier FOM is typically used for high-resolution converters and is given by the following equation:

$$FOM_s = SNDR + 10\log\left(\frac{BW}{P}\right) \quad (7)$$

where SNDR is the signal to noise ratio and distortion, BW is one-half of the sampling frequency, and P is the power dissipation. To estimate the power scaling from 180nm to 40nm, the ADC Performance Survey (Supplementary Reference [8]) was used which aggregates all ADCs published in the ISSCC and VLSI circuits conferences from 1997 - 2018. A subset of data for 180 nm and 40 nm ADCs is shown in Supplementary Figure 21. The mean FOM between sampling frequencies from 100 kHz to 10 MHz for each technology was determined and compared. The mean FOM for 40 nm was determined at 172 dB and a conservative estimate of 165 dB was used for power estimation. Using an estimated 165 dB, which corresponds to 176 μW per ADC, we can estimate the new total analogue power at 40nm to be 19 mW, leading to a total system power of 42.1 mW, assuming the 54×108 crossbar power remains at 7 mW. Therefore, by simply scaling the system to 40nm technology node, the estimated OPS/W at 148 MHz is 1.37 TOPS/W. The power efficiency can be further improved by further scaling the CMOS circuit to more advanced technology nodes. Additionally, the digital power can be further improved by using a more custom controller design instead of using a generic processor, while the analogue power can be improved by further optimizations of the ADC circuitry (e.g. replacing the fast and high-precision 13-bit ADC with simpler interface circuits), along with memristor device optimizations to reduce the crossbar power.

Supplementary Note 10. Analysis of area efficiency

As the first prototype, we intentionally made the circuit design flexible to serve as an evaluation platform, and we were limited to a relatively old CMOS technology (due to cost and tool compatibility limitations in a university cleanroom). As a result, the area efficiency is not ideal. The total chip area is 6.7 mm × 9.2 mm = 61.64 mm^2. The area of the memristor array is 475 μm × 292 μm ~ 0.14 mm^2. Therefore, the memristor array's area efficiency (the percentage of the array over the entire chip) is about 0.23%.

A large reason of the low area efficiency can be traced to the conservative design choice. To make the chip flexible, we designed 2 write DACs (for positive and negative writes), 1 read DAC and a 13bit ADC at each row and each column, so input data can be applied at every row and the output can be collected at every column and processed in parallel. The reverse operation is also supported for algorithms such as the LCA. The size of the three DACs and the ADC is around 74 μm × 1800 μm ~ 0.1332 mm^2 for each such reconfigurable channel. The total area of these components for the 162 channels (54 rows and 108 columns) occupy 21.57 mm^2, which is around 35% of entire chip area. The rest of the chip areas includes the 64 kB SRAM (32 kB data memory and 32 kB ping pong memory) and the OpenRISC core, which are 3.79 mm^2 and 10.04 mm^2, respectively.

There are a number of approaches we can take to optimize the chip design and improve the area efficiency.

For many applications we can replace the 13-bit ADC used in this evaluation chip with 8-bit ADCs without loss of accuracy. For example, Supplementary Figures 22 and 23 show simulation results comparing floating point activations (corresponding to the ADC outputs) vs. 8 bit activations for commonly used algorithms, as well as weight quantization effects for some models [S9-S11]. Reducing the required ADC precision to 8 bit will significantly decrease the ADC area, as shown in Supplementary Figure 24b. The chip area will also be naturally reduced with more advanced technology nodes [S12, S13]. For instance, state-of-the-art 40 nm 8-bit ADCs occupy ~1650 μm^2 [S13], which will reduce the ADC area by a factor of 20 compared with the current chip. The DAC area is relatively much smaller. For example, the total DAC area for a 128 ×128 memristor crossbar array would be 173 μm^2 with the 40 nm technology node.

The ADC area can be further reduced through optimization of the array interface structure. Instead of using a dedicated ADC at each crossbar column and row, multiple columns can share a single ADC by sequentially sampling one column at a time, e.g. via time-multiplexing, as shown in Supplementary Fig. 26. This approach is feasible since ADCs can operate at speeds of several GHz while the read operation through the memristor takes longer time (e.g. 10 ns). Using this shared ADC approach, the physical ADC area can be significantly reduced. Note here we are trading the system speed for the peripheral circuitry area. However, due to the in-memory operations and high level of parallelism, the memristor-based systems do not need to operate at very high frequency to achieve the desired outputs. For example, our previous analysis has shown that even at a system speed of 10 Mhz, very high energy efficiency of 60.1 TOPS/W can still be achieved [S14].

Another potentially promising approach for area efficiency is to exploit a concept we termed “array ADCs”, where the core component of the ADC is shared among different columns but each column maintains a dedicated latch to accumulate data locally so the column outputs can still be processed in parallel. This approach can potentially reduce the overall ADC area without having to reduce the system clock. Similar approaches have been used in column-parallel CMOS sensor arrays [S15-S18] or multi-channel biomedical and physical detectors [S19-S25]. These ADCs are designed for parallel multi-node sampling applications, which is an excellent match for the tasks discussed here, although circuit optimizations have not been extensively studied since these prior applications do not require high-speed operations. Optimizing the ADC design using the array ADC concept will be a key direction in our future studies.

With technology scaling, proper ADC resolution selection, and optimized design through shared ADC or array ADC approaches, we expect the area efficiency of the integrated chip to be significantly improved, and can potentially reach 100%, as shown in Supplementary Figure 27.

Supplementary Note 11. Tiled architecture and performance comparison with conventional CMOS systems

To scale up the system, instead of simply increasing the crossbar size, we plan to tile small crossbars together to form large-scale neural networks, as schematically shown in Supplementary Figure 29.

A scalable architecture would be to use the memristor - CMOS integration at a macro level, e.g., a 128 × 128 crossbar with the underlying interfacing circuits form a self-contained, modular unit (macro). The macro can be tiled to construct systems of larger scale. Since the macro has a digital interface, the tiling and scaling up would be both straightforward and robust.

The CMOS circuits underneath the memristor array will consist of decoders and MUXs for the random-access operation of the memristor array, and DACs and ADCs for sampling crossbar input and output signals. The CMOS layer will also host some digital circuitry used for simple signal processing operations. Moreover, a centralized control circuitry may be needed to facilitate the overall system operation and control the data flow. Tile-to-tile data communications will be performed in the CMOS layer in the digital domain, as shown in Supplementary Figure 30, following previous architecture designs [S26][S27].

Previous analyses on the effects of different components in the interface circuitry [S28] have shown that the interface performance parameters such as power, area and latency are dominated by those of the ADCs. The optimization of ADCs is thus critical for memristor-based computing architectures. Fortunately, ADC size and power decrease exponentially as the tile size is reduced, as discussed in Supplementary Note 10 and Supplementary Figure 24. For example, dividing one large crossbar into 2 smaller ones will require 2 ADCs vs 1. However, each of the ADC for the smaller array can have one fewer bit compared with the ADC used for the larger array. In general, reducing the ADC bit by 1 will reduce the area and energy by ½, respectively, as shown in Supplementary Figure 24. As a result, there likely will not be significant increase in ADC area or power when using several smaller crossbar/interface circuitry tiles vs one larger crossbar/interface circuitry.

Future studies that can allow more direct integration at the local or intermediate interconnect levels (which we termed monolithic integration) will significantly alleviate the line

resistance issue and enable larger arrays to be integrated and operated. Combined with more optimized circuit designs (e.g. as discussed in Supplementary Note 10), and the use of the scalable tiled architecture, we expect practical memristor-based in-memory computing systems can be built with high energy efficiency and high throughput [S29].

Supplementary References

1. Chang, T. *et al.* Synaptic behaviors and modeling of a metal oxide memristive device. *Appl. Phys. A* **102,** 857–863 (2011).
2. Choi, S., Sheridan, P. & Lu, W. D. Data Clustering using Memristor Networks. *Sci. Rep.* **5**, 10492 (2015).
3. Chen, P.-Y. *et al.* Mitigating effects of non-ideal synaptic device characteristics for on-chip learning. in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* 194–199 (IEEE, 2015)
4. Le Gallo, M. *et al.* Mixed-precision in-memory computing. *Nat. Electron.* **1**, 246–253 (2018).
5. Boybat, I. *et al.* Neuromorphic computing with multi-memristive synapses. *Nat. Commun.* **9**, 2514 (2018).
6. Ambrogio, S. *et al.* Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* **558**, 60–67 (2018).
7. Taur, Y. & Ning, T. H. Fundamentals of Modern VLSI Devices. (Cambridge University Press, 2009).
8. Murmann, B. ADC performance survey 1997-2018. http://web.stanford.edu/~murmann/adcsurvey.html (2018).
9. Hashemi, S., Anthony, N., Tann, H., Bahar, R. I. & Reda, S. Understanding the impact of precision quantization on the accuracy and energy of neural networks. *2017 Design, Automation & Test in Europe* 1474–1479 (2017).
10. Lin, D. D., Talathi, S. S. & Annapureddy, V. S., Fixed point quantization of deep convolutional networks. *arXiv preprint arXiv:1511.06393.* (2015).
11. Jacob, B., *et al.* Quantization and training of neural networks for efficient integer-arithmeticonly inference. *arXiv preprint arXiv:1712.05877.* (2017)
12. Choo, K. D., Bell, J. & Flynn, M. P., Area-efficient 1GS/s 6b SAR ADC with charge-injection-cell-based DAC, *Proc. IEEE Int. Solid-State Circuits Conf.*, 460-461, (2016).
13. Chen, H., Zhot, X., Yu, Q., Zhang F. & Li, Q., A >3GHz ERBW 1.1GS/s 8b Two-Step SAR ADC with Recursive-Weight DAC, *IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, 97-98 (2018).
14. Zidan, M.A. *et al.* A general memristor-based partial differential equation solver, *Nat.*

Electron. **1**, 411-420, (2018).

15. Le-Thai, H., Chapinal, G., Geurts, T. & Gielen, G. G. E., A 0.18- μm CMOS Image Sensor With Phase-Delay-Counting and Oversampling Dual-Slope Integrating Column ADCs Achieving 1e−rms Noise at 3.8-μs Conversion Time. *IEEE J. Solid-State Circuits* **53,** 515-526 (2017).
16. Xu, R., Liu, B. & Yuan, J., Digitally Calibrated 768-kS/s 10-b Minimum-Size SAR ADC Array With Dithering. *IEEE J. Solid-State Circuits* **47,** 2129-2140 (2012).
17. Li, T.-L. *et al.* A Column-Parallel Hybrid Analog-to-Digital Converter Using Successive-Approximation-Register and Single-Slope Architectures with Error Correction for Complementary Metal Oxide Silicon Image Sensors. *Jpn. J. Appl. Phys.* **52,** 04CE04-1 - 04CE04-7 (2013).
18. Jeon, B.-K., Hong, S.-K. & Kwon, O.-K., A Low-Power 12-bit Extended Counting ADC Without Calibration for CMOS Image Sensors. *IEEE Trans. Circuits Syst. II, Exp. Briefs http://dx.doi.org/10.1109/TCSII.2017.2717044* (2017).
19. Kim, C. *et al.* A 92dB dynamic range sub-μVrms-noise 0.8μW/ch neural-recording ADC array with predictive digital autoranging. *IEEE International Solid-State Circuits Conference (ISSCC)*, 470-472 (2018).
20. Kassiri, H. *et al.* All-Wireless 64-Channel 0.013mm2 /ch Closed-Loop Neurostimulator with Rail-to-Rail DC Offset Removal. *IEEE International Solid-State Circuits Conference (ISSCC)*, 452-453 (2017).
21. Gagnon-Turcotte, G., Ethier, C., Köninck, Y. D. & Gosselin, B., A 0.13μm CMOS SoC for Simultaneous Multichannel Optogenetics and Electrophysiological Brain Recording. *IEEE International Solid-State Circuits Conference (ISSCC)*, 466-468 (2018).
22. Bugiel, S. *et al.* Ultra-Low Power Fast Multi-Channel 10-Bit ADC ASIC for Readout of Particle Physics Detectors. *IEEE Trans. Nucl. Sci.* **63,** 2622-2631 (2016).
23. Gao, W., Gao, D., Wei, T., Hu-Guo, C. & Hu, Y., A 12-bit Low-Power Multi-Channel Ramp ADC Using Digital DLL Techniques for High-Energy Physics and Biomedical Imaging. *IEEE International Conference Solid-State and Integrated Circuit Technology (ICSICT)*, 227-229 (2010).
24. O'Driscoll, S., Shenoy, K. V. & Meng, T. H., Adaptive Resolution ADC Array for an Implantable Neural Sensor. *IEEE Trans. Biomed. Eng* **5**, 120-130 (2011).

25. Gao, W., Gao, D., Hu-Guo, C. & Hu, Y., Design of a 12-Bit 2.5 MS/s Integrated Multi-Channel Single- Ramp Analog-to-Digital Converter for Imaging Detector Systems. *IEEE Trans. Instrum. Meas.* **60,** 1942-1951 (2011).
26. Zidan, M. A. *et al.* Field-programmable crossbar array (FPCA) for reconfigurable computing. *IEEE Trans. Multi-Scale Comput. Syst. https://doi.org/10.1109/TMSCS.2017.2721160* (2017).
27. Mikhailenko, D., Liyanagedera, C., James, A. P. & Roy, K. M2CA: Modular memristive crossbar arrays. *Circuits and Systems (ISCAS) 2018 IEEE International Symposium on. IEEE*, 1-5, (2018).
28. Shafiee, A. *et al.* ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* 14–26 (2016).
29. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).