nature electronics

ARTICLES

https://doi.org/10.1038/s41928-019-0331-1

# Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing

Loai Danial[1], Evgeny Pikhay[2], Eric Herbelin[1], Nicolas Wainstein[1], Vasu Gupta[1], Nimrod Wald[1], Yakov Roizin[2], Ramez Daniel[3] and Shahar Kvatinsky[1]*

**Metal–oxide memristive integrated technologies for analogue neuromorphic computing have undergone notable developments in the past decade, but are still not mature enough for very large-scale integration with complementary metal–oxide–semiconductor (CMOS) processes. Although non-volatile floating-gate synapse transistors are a more advanced technology embedded within CMOS processes, their performance as analogue resistive memories remains limited. Here, we report a low-power, two-terminal floating-gate transistor fabricated using standard single-poly technology in a commercial 180 nm CMOS process. Our device, which is integrated with a readout transistor, can operate in an energy-efficient subthreshold memristive mode. At the same time, it is linearized for small-signal changes with a two-orders-of-magnitude resistance dynamic range. Our device can be precisely tuned using optimized switching voltages and times, and can achieve 65 distinct resistive levels and ten-year analogue data retention. We experimentally demonstrate the feasibility of a selector-free integrated memristive array in basic neuromorphic applications, including spike-time-dependent plasticity, vector-matrix multiplication, associative memory and classification training.**

The decline of Moore's law and the end of Dennard scaling signal the need to develop computing approaches beyond traditional von Neumann architectures. Recently, there has been a renewed interest in neuro-inspired computing[1–6]. Such neuromorphic systems are designed to accelerate data-intensive applications and handle large amounts of data by mimicking the adaptivity, interconnectivity, noise tolerance and energy efficiency of the human brain. The building blocks of these architectures are synapses, which can be trained to store weights, and neurons, which collectively interact to transmit information. Deep neural networks are modularly constructed using many massively interconnected layers trained to perform inference[7]. A layer is an atomic neural entity comprising trainable synapses (matrix) and neurons (vector), abstracted by vector-matrix multiplication (VMM). This dot product is a typical computational bottleneck in artificial neural networks (ANNs). Training commonly relies on machine learning (ML) optimization techniques, such as stochastic gradient descent, or neuro-inspired heuristics, such as spike-time-dependent plasticity (STDP).

Unfortunately, even in custom-designed digital hardware, data movement between the memory and processing units impedes the computationally intensive arithmetic operations[2]. One alternative is to use integrated circuits based on analogue non-volatile memory devices. These analogue devices possess adjustable conductance, which could mimic synaptic transmission by multiplying the input neuron signal (encoded, for example, as the applied voltage to the device) by the corresponding weight (conductance) and passing the multiplication product (the resulting current) to the output neuron. Dense, fast and power-efficient VMM computation would thus be inherently enabled at the physical level (using Ohm's and Kirchhoff's laws).

Non-volatile memories in analogue and mixed-signal neuromorphic networks, which were first implemented 30 years ago[8], relied mostly on floating-gate 'synapse transistor' technology. However, the technology used devices with relatively large area and low retention time, which led to long time delays and high power consumption[9–13]. Developments in alternative nanoscale non-volatile resistive switching memory devices (such as phase-change, magnetic and resistive random access memory (RRAM)) has led to a resurgence of interest in the field[14,15]. The memristor (or memristive device), predicted in 1971[16,17], was originally defined as the fourth passive circuit element and has many valuable circuit properties. Memristive theory was applied to two-terminal resistive switching devices only a decade ago[18]. Fundamentally, resistive switching is the physical basis of memristive devices. Synaptic plasticity is implemented by adjusting the device conductance, by controlling the voltage (or current) stimuli through it in proportion to the duration, pulse rate and amplitude of the applied signals[19] (see Methods). These devices are considered promising candidates for future non-volatile memory applications and neuromorphic analogue circuits, thanks to their low power consumption, fast write and read, low fabrication costs, high density and scalability.

The majority of reported memristive neural networks[20,21] owe their success to one-transistor–one-resistor (1T1R) technology, in which every memory cell is coupled to a select transistor fabricated in the front end of the complementary metal–oxide–semiconductor (CMOS) process. When introduced into memristive crossbar arrays, these selectors provided read/write-disturb immunity (from sneak paths or half-select, for example). However, they considerably increase the overall area, power and control overhead. Passive selecting devices (such as diodes), fabricated with memristive devices in the back-end of the CMOS process, are a promising scalable

[1]Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel. [2]TowerJazz, Migdal HaEmek, Israel. [3]Department of Biomedical Engineering, Technion – Israel Institute of Technology, Haifa, Israel. *e-mail: shahar@ee.technion.ac.il

alternative to select transistors in eliminating sneak paths[21]. However, functional neuromorphic networks based on pure passive memristive devices have been integrated only on a small scale[22,23]. As for commercial products based on memristive devices, existing ones lack CMOS integration. Commercialization of memristive devices has been further hindered by reliability issues, including local heating, and parameter spread variability[24,25], as well as by the lack of a standard process flow for most of the materials used to fabricate them[26,27]. The fabrication process is therefore still too immature for high-yield large-scale integration[28].

Non-volatile floating-gate memories can now be feasibly embedded in CMOS mixed-signal integrated circuits after recent optimizations and downscaling of the cells[29,30]. Floating-gate devices have been used in various ways to implement artificial synapses in neuromorphic systems, most frequently as analogue memory cells for offline programmable synaptic weights and/or digital parameter storage or as trainable analogue synapse implementations that usually obey a learning rule such as STDP[8–13]. Redesigned floating-gate synapse transistors enable individual fast and high-precision tuning of their memory state as well as energy-efficient, high-endurance and temperature-insensitive analogue operation[31–33]. However, their feasibility in neuromorphic applications is limited by the costly double-poly process. Novel floating-gate devices with memristive operation mode (for example, MemFlash)[34,35] have been proposed as a step forward to potentially substitute two-terminal memristive devices in large-scale CMOS-compatible dense neuromorphic systems[36,37]. Yet, such devices operate in a different conduction mode than that of the floating-gate synapse transistors[8–13,29–33]. (A summary of the technological, structural and functional properties of previously reported floating-gate and memristive devices for neuromorphic computing is provided in Supplementary Section 6 and Supplementary Table 10.)

In this Article, we bridge the gap between the emerging computational capabilities of memristive devices in neuromorphic systems and the technological maturity of floating-gate transistors by using a standard CMOS technology. We propose a power-efficient memristive device based on an optimized two-terminal single-poly floating-gate transistor, optionally connected in parallel to a readout transistor[38]. Our cell, which is called the Y-flash, operates in a subthreshold memristive mode[8,34] and is linearized for small-signal changes. We apply memristive techniques recently employed in small-scale selector-free dense integrated ANNs for STDP, VMM, associative memory and classification training. With this approach, we theoretically and experimentally demonstrate a practical memristive device for high-performance neuromorphic computing.

## Floating-gate memristive device

The asymmetric Y-flash device was originally proposed as a two-terminal n-channel metal–oxide semiconductor (NMOS) transistor (injection transistor) with a floating gate (FG). It is manufactured using the standard CMOS process flow and requires no additional masks. A readout NMOS transistor is optionally added to the Y-flash cell to mitigate reliability issues (for example, read disturb), while a common FG is shared with the injection transistor (see Methods). The FG potential is controlled by a capacitive coupling between the FG and the common drain terminal, as shown in Fig. 1a. For this purpose, the Miller capacitance of the drain is made larger than that of the source with a customized layout (Fig. 1b). Top-view and cross-section images of the Y-flash taken by a scanning electron microscope (SEM) are shown in Fig. 1c. The operation modes of the Y-flash as a digital memory element are specified (see Methods and Supplementary Section 1). Accordingly, different memory states are recognized by a shift of the $I$–$V$ curves of the Y-flash when conducting in the saturation mode, corresponding to different threshold voltage levels ($V_{th}$; Fig. 1d). However, this shift produces a limited current dynamic range (slightly different charge is stored in the FG at different $V_{th}$) of up to one order of magnitude only at the same applied voltage, as shown by the $I$–$V$ slope in the inset of Fig. 1d. We thus operate the Y-flash in subthreshold mode to achieve zero-crossing $I$–$V$ curves with highly dissimilar slopes, a wide current dynamic range and continuously varying resistance of different memory states (Fig. 1e). The subthreshold drain current is determined mainly by the read transistor current, thanks to its low $V_{th}$ (see Methods and Fig. 1f).

The memristive device was basically implemented using an injection transistor[36]. A read transistor was optionally added in parallel to alleviate the performance constraints[38] (Fig. 2a). The equivalent two-transistor Y-flash cell operates in a memristive mode when the sources of both transistors are connected[34]. The large-signal $I$–$V$ model of the Y-flash cell, when operating in the subthreshold mode, is approximated as a multiplication of two separable and dimensionless factors ($V$ and $G$, as functions of $V_{DS}$ and $V_{th}$, respectively):

$$I_{DS} \approx I_{read}\,\underbrace{e^{\frac{CR\times V_{DS}}{mV_T}}}_{V(V_{DS})}\,\underbrace{e^{-\frac{V_{th}}{mV_T}}}_{G(V_{th})} \qquad (1)$$

where $m$ is a technology constant called the subthreshold slope factor, $V_T$ is the thermal voltage constant equal to 26 mV, CR is the coupling ratio between the gate–drain and gate–source capacitances of the asymmetric Y-flash, $V_{th}$ is the equivalent threshold voltage that represents the internal memory state variable of the device, and $I_{read} = 1$ nA is empirically determined as the current at $V_{DS} = V_{th}$. The total subthreshold current, $i_{DS} = I_{DS} + i_{ds}$, in response to the total applied voltage, $v_{DS} = V_{DS} + v_{ds}$, in equation (1) is linearized using Taylor series expansion around a fixed read voltage $V_{DS}$ (bias, defined also as $V_r$) as

$$\underbrace{i_{DS} - I_{DS}}_{i_{ds}} = \underbrace{\left.\frac{dI_{DS}}{dV_{DS}}\right|_{V_{DS}=V_r}}_{1/R_{mem}}\,\underbrace{(v_{DS} - V_{DS})}_{v_{ds}} \qquad (2)$$

where we adopt the convention that voltages/currents with lower-case symbols and upper-case subscripts refer to total voltages/currents, those with upper-case symbols and subscripts refer to pure large-signal values, and those with lower-case symbols and subscripts refer to pure small-signal values. Equation (2) determines the small-signal current product linearized in an Ohm's law-like manner, and defines the small-signal resistance of the device (dependent only on $I_{DS}$), as approximated by the derivative of $I_{DS}$ at $V_r$; it is known as the dynamic or incremental resistance for small-signal changes[39] (see Methods). The complete small-signal model, schematic and analysis are discussed and validated in Supplementary Section 2a,b. The resistance is highly state-dependent and exponentially correlated with the internal state variable (see Methods). Therefore, the theoretical current dynamic range spans three orders of magnitude [1 nA:1 μA], while the state variable dynamic range was measured (see Methods) as [1 V:2 V], corresponding to $R_{mem} \in [145\,\text{k}\Omega{:}145\,\text{M}\Omega]$ (Fig. 2b). The input dynamic range (IDR = $2v_{ds}$) of the device was measured at three different values of $V_{th}$ (Fig. 2c). The linearization described in equation (2) is valid under certain conditions ($v_{ds} \ll mV_T/\text{CR}$), where currents in response to positive and negative small-signal voltages are cancelled at IDR = 100 mV ($v_{ds} = 50$ mV), as shown in the inset of Fig. 1c; otherwise, it becomes nonlinear ($V_{th} = 2$ V, for example). Consequently, the current dynamic range trades off with the input dynamic range.

Precise intermediate resistance values are gradually tuned by applying short program/erase pulses with specific parameters (such as pulse duration and voltage amplitude). The Y-flash programming/erasing model imitates the RESET/SET process of a voltage-controlled memristive device while adjusting $V_{th}$ as a function of

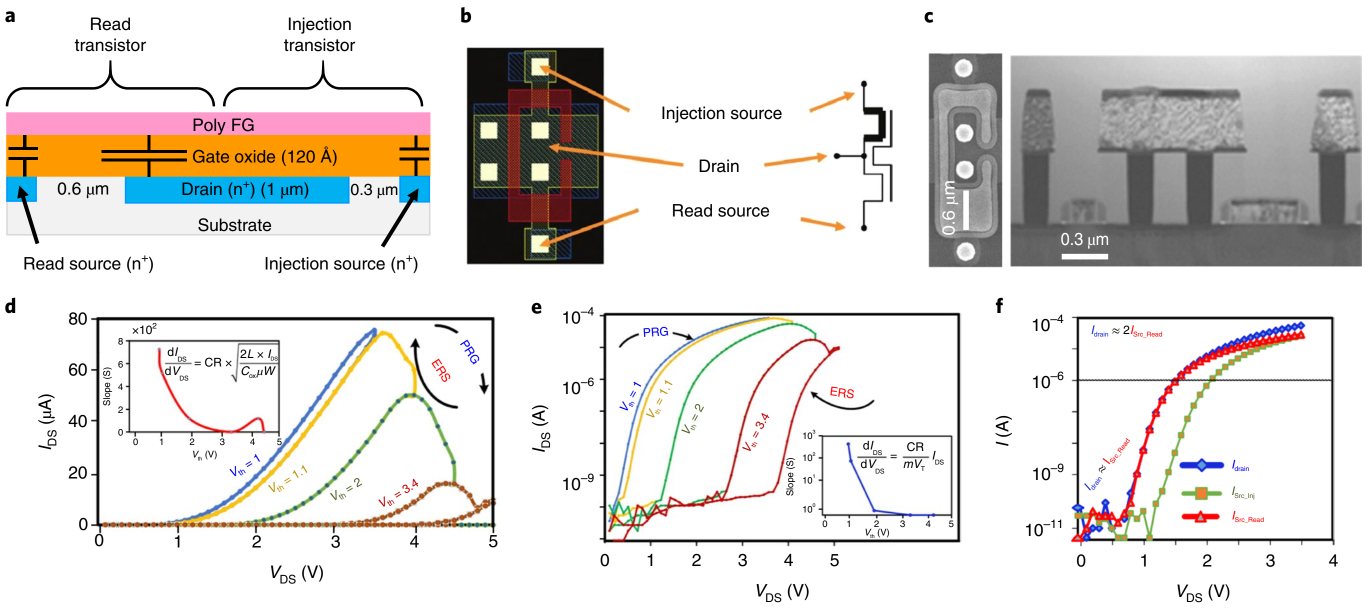

**Fig. 1 | Y-flash non-volatile memory device.** **a**, Cross-sectional schematic of the single-poly floating-gate Y-flash device produced using a standard CMOS process. The device comprises two two-terminal NMOS transistors—injection and read—with an asymmetrical (increased) coupling ratio (CR) between the common drain and sources to the floating gate. The readout transistor can be optionally added and optimized to mitigate read disturb issues when the device is operating in saturation. **b**, Y-flash cell layout mapped to an electrical schematic by each terminal. **c**, SEM images of the Y-flash device fabricated in a 180 nm CMOS process: top view (left) and cross-section (right). **d**,**e**, *I*–*V* curves for different programming (PRG) or erasing (ERS) voltage values in a linear scale in saturation conduction mode (**d**) and in a log-linear scale at subthreshold conduction mode (**e**). The insets describe the slope of the *I*–*V* curves, showing one (**d**) to three (**e**) orders of magnitude dynamic range. Below 1 nA, the noise is dominant. Above 1 µA, the Y-flash conducts in saturation mode. **f**, Read transistor current versus injection transistor current in subthreshold mode (sub-1 µA), showing that the Y-flash total current is dominated by the read transistor. While in saturation, the read current is equal to the injection transistor current.

the applied voltage, pulse duration and the current value of $V_{\mathrm{th}}$ (see Methods). The physical mechanism, the device's energy diagram, the state variable dynamics and the corresponding resistance level in RESET/SET are illustrated in Fig. 2d–i. An exponential hysteresis in response to a large-signal voltage sweep is shown in Fig. 3a. Similarly, a linear hysteresis could be extrapolated in response to a small-signal input, where the zero-crossing (bias) point, programming and erasing voltages are all in large-signal. Despite the nonlinear effects, fine-tune program/erase pulses that adjust the resistance precisely are used to achieve an analogue memristive operation mode with more than 65 discrete levels, roughly equivalent to 6 bit precision (see Methods, equation (11)). These features make it possible to emulate synaptic learning and plasticity using a floating-gate memristive device.

## Floating-gate memristive synapse

The analogies between SET/RESET in memristive devices and synaptic long-term potentiation (LTP) and depression (LTD) have been investigated in refs. [19,40]. LTP and LTD are the most important physiological mechanisms at the synaptic level, as they underlie the biological processes of learning, forgetting and memory. As they are associative and cooperative, they can employ Hebbian learning[41], which is used as a basic behavioural rule in ANNs. STDP could be viewed as an asymmetric temporal version of the abstract Hebbian rule in spiking neural networks. In this Article, we implemented STDP using a floating-gate device that operates in a memristive mode in response to small-signal neural spikes (see Methods). The derived synaptic weight update ($\Delta W$) rule (see Methods, equation (10)) inherently implements STDP without time-division multiplexing[19] (Fig. 3b): it has a decaying exponent as a function of the time difference between pre- and post-synaptic spikes ($\Delta t$). The exponential weight update rule is derived from the injection and tunnelling dynamics. It is utilized together with the two-terminal memristive operation to achieve high efficiency for in situ analogue STDP. This nonlinear rule implies state-dependency; switching to the next state is mostly determined by the last memory state. It also implies that no update is required when the spikes are simultaneous, concurring with biological observations[41]. Furthermore, the nonlinear conductance dynamics could extend the memory lifetime of such networks[42].

Although STDP is more biologically plausible and can describe a local learning rule for a single synapse, VMM is the basis for parallel computation in ANNs[7]. It is inherently implemented in hardware, where the integration of analogue resistive memory devices in dense arrays enables extremely compact, fast and energy-efficient analogue computation by utilizing Ohm's and Kirchhoff's laws. VMM is represented as $Y_m = \sum_n W_{n,m} X_n$ (where $\mathbf{X}$ is an input vector with $n$ rows, and $W$ is the synaptic weight matrix, with $m$ rows and $n$ columns) and is implemented via the small-signal $I$–$V$ operation described in equation (2) by $i_m = \sum_n W_{n,m} v_n$, as shown in Fig. 3c, where $W_{n,m} = 1/R_{\mathrm{mem}_{\mathrm{n,m}}}$ is the incremental conductance for the small-signal at the $n,m$ node, and $v_n$ is the small-signal voltage (within the IDR). VMM is an atomic operation executed simultaneously (see Methods) in the entire array. As a result of the nonlinear operation and asymmetry of the Y-flash structure, selecting elements are not required to eliminate disturbing effects (such as sneak path currents). Large-scale, dense neuromorphic arrays, comprising up to one million devices, are thus made feasible (Supplementary Section 5a)[38].

We demonstrate two neuromorphic proof-of-concept applications that utilize the floating-gate memristive synapse. Although conceptually simple, these applications constitute technological milestones in the hardware realization of ANNs. In two recent articles[22,43], these applications were used to demonstrate how the

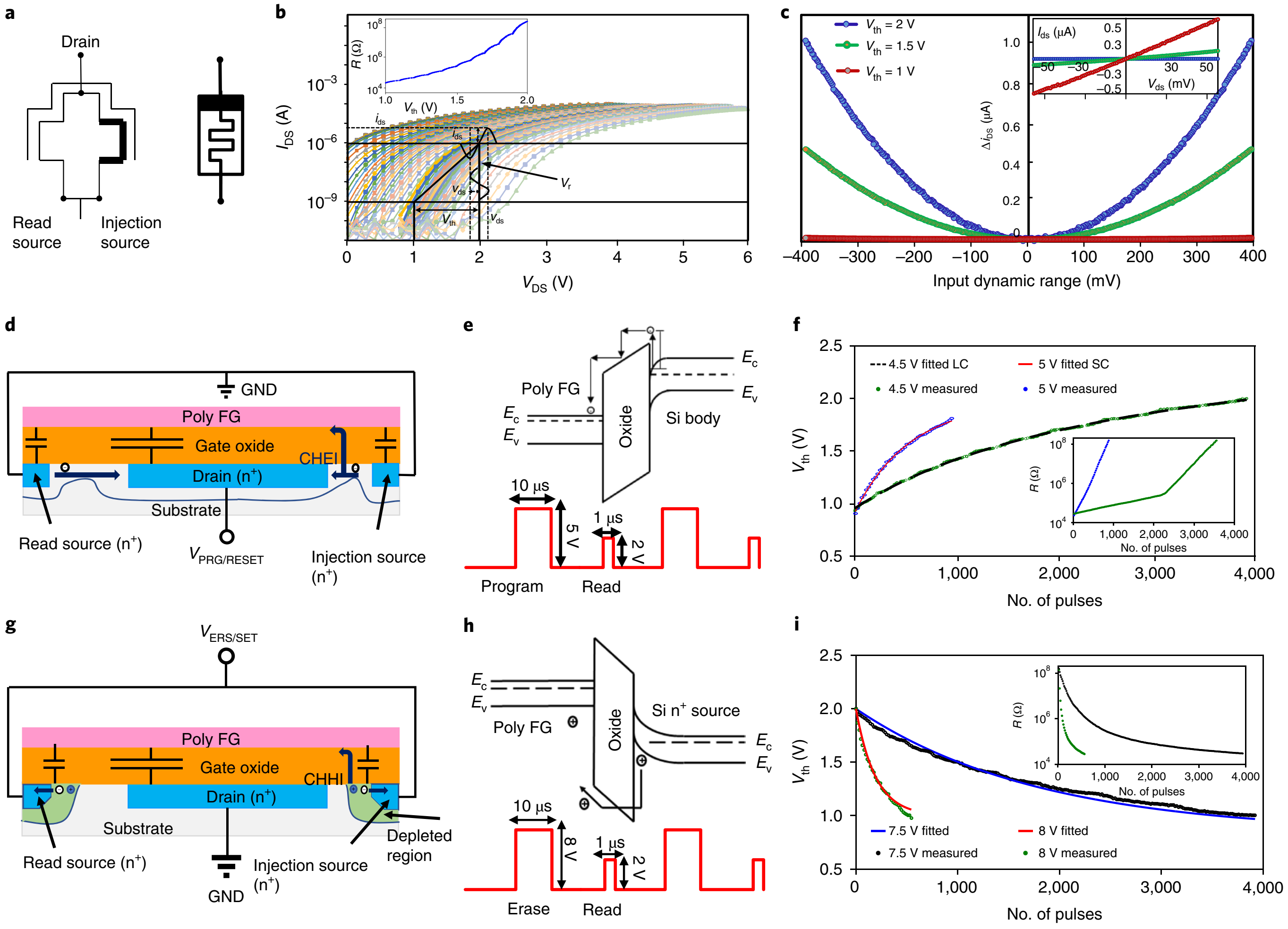

**Fig. 2 | Y-flash memristive device. a**, Schematic of the Y-flash cell composed of a non-volatile, two-terminal NMOS floating-gate (injection transistor) memristive device, with an optional parallel readout transistor added to allow reading of the device memory state when operating in a subthreshold memristive mode at lower voltage and to enhance reliability performance. **b**, *I*–*V* curve shift of the Y-flash cell from 1 nA to 1 µA, in response to a voltage sweep on the drain, corresponding to [1 V:2 V] state variable ($V_{th}$) range. A small-signal analysis is illustrated around the operating read point. Inset: plot of $V_{th}$ versus resistance. **c**, IDR at three threshold states, where small-signal currents in response to positive and negative voltages with the same amplitude are expected to cancel each other. The common IDR between the three states is determined by $V_{th}$ = 2 V and is equal to 50 mV (where the difference is zero). Inset: linearization of the *I*–*V* around the large signal inside the IDR. **d**, Channel hot electron injection (CHEI) to the floating gate from the injection transistor channel. **e**, Energy diagram of the CHEI process, where $E_c$ and $E_v$ describe the conductivity and valence energy levels, respectively, and pulse diagram of the RESET (programming) process. **f**, $V_{th}$ programming as a function of pulse number. Measured data were fitted (see Methods) in the short channel (SC) and the long channel (LC). Inset: resistance as a function of programming pulses. More than 1,000 resistive values were gradually tuned. **g**, Channel hot holes injection (CHHI) generated by band-to-band tunnelling (BTBT) to the floating-gate after applying high voltage on the source. **h**, Energy diagram of the CHHI process and pulse diagram of the SET process. **i**, $V_{th}$ erasing as a function of pulse number and erasing voltage. Inset: resistance as a function of erasing pulses. Approximately 650 resistive values were gradually tuned. The effective number of distinct resistive levels is 65 (see Methods).

applicative potential of memristive RRAM devices might be leveraged beyond their synaptic behaviours. Here, we implement these applications on 12 × 8 selector-free integrated floating-gate memristive arrays, using a standard CMOS design flow.

## Y-flash memristive neural network associative memory

Our first application implements associative memory using a reconfigurable Hopfield network[43]. The Hopfield network has proved useful in content-addressable memories (CAMs) and combinatorial optimization problems, such as the travelling salesman and location allocation problems. A conventional Hopfield network has been realized by constructing power- and area-starved CMOS synapses[44]. A Hopfield network is a dynamic, asynchronous and recursive system consisting of a set of interconnected neurons[43]. The Hopfield-based CAM topology is realized where the digital inputs of the system are used also as its outputs to digitally quantize the intrinsic analogue memory states pre-coded inside the network. Different patterns can be stored into the network by fine-tuning the values of the analogue weights, after which the pre-stored patterns can be retrieved, analogously to associative memory in the human brain. In this Article, a three-bit CAM Hopfield network is constructed (Fig. 4a) with 12 differential floating-gate memristive synapses inside an integrated array, and three decision-making neurons (see Methods).

Associative memory, a unique capability of the brain, allows us to retrieve a piece of information by associative recall of related information. The target memory that must be associatively recalled was set at ‘110’ (see Methods). Once the targeted incremental resistances are obtained, they remain unchanged during the network operation. The network can converge to ‘110’ automatically from any state in the range from ‘000’ to ‘111’. Figure 4b shows the

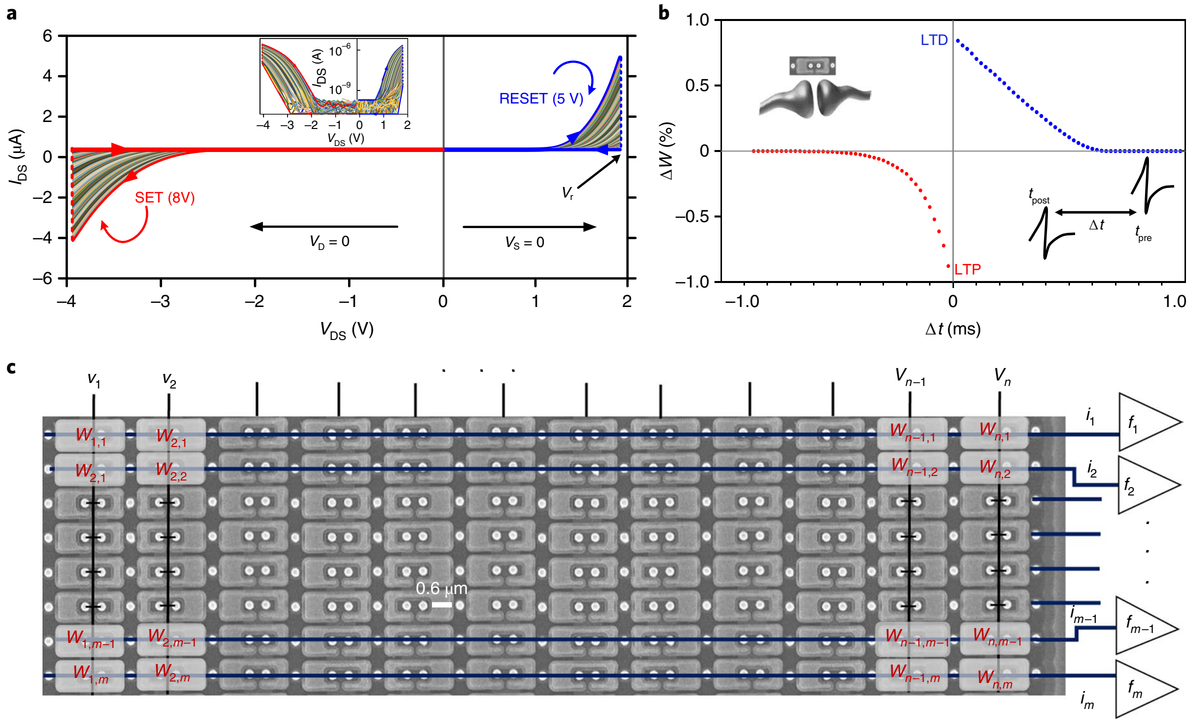

**Fig. 3 | Y-flash memristive synapses. a**, *I*–*V* curve of the asymmetric Y-flash memristive device exhibiting memory hysteresis (inset plot has a log scale). $V_{DS}$ is negative in erase when the drain is grounded. Gradual SET/RESET operations are performed by 8 V/5 V pulses. The solid lines correspond to high/low resistive states, while the dashed lines correspond to read voltages in subthreshold mode. **b**, STDP of the Y-flash operated in subthreshold mode (small-signal). The graph shows the weight (*W*) update percentage, normalized as a function of the temporal spike interval between pre- and post-synaptic neurons. Synaptic long-term potentiation (LTP) is equivalent to erase and long-term depression (LTD) is equivalent to programming. **c**, SEM image (flipped) of the fabricated 12 × 8 Y-flash crosswise array in a 180 nm process that implements a fully connected single-layer $n \times m$ neural network. Triangles denote neurons. All Y-flash cells (synapses) are operated in parallel by small-signal voltages ($v_1$…$v_n$) that naturally implement the VMM dot product, programmed separately in column granularity by program pulses and erased in full column segments by erase pulses. For illustration purposes, only some wires are shown.

waveforms of state vector $\mathbf{v}(t)$ (see equation (13) in the Methods) in the process of retrieving the pre-stored pattern '110' (in red) from eight different initial states. $\mathbf{v}(t)$ was updated recursively in three asynchronous cycles, with only one bit updated per cycle. A cycle update example is shown in Fig. 4b. The update starts from $\mathbf{v}(0) = (0\,0\,0)$. In the first cycle, the element $v_1$ is updated, yielding an intermediate state vector of $\mathbf{v}(1) = (1\,0\,0)$; $v_2$ is then updated in the second cycle, producing $\mathbf{v}(2) = (1\,1\,0)$. The network has now successfully recalled the pre-stored pattern '110'. In the following cycles, no updating occurs and the network stabilizes at '110'. Recalling the pre-stored '110' pattern by means of some intermediate states emulates weak memorization in the human brain: having to think hard to remember something. For different initial state vectors, the network may be updated with different intermediate state vectors before recalling the pre-stored pattern. Direct memorization emulates strong associative memory, where memories can be retrieved without experiencing associative states ({(0 1 0), (1 0 0), (1 1 0), (1 1 1)}).

In multi-associative memory, associative recall will lead us to recall different pieces of information, depending on the intermediate associative states we experience. In the hardware version of multi-associative memory, different pieces of data can be associatively recalled from more than one pre-stored pattern[43]. We implemented this in our network by minimally changing the resistance matrices and threshold vectors (see Methods) to reconfigure the weights to pre-store the pattern '101' in addition to the original pre-stored '110'. As shown in Fig. 4b (in blue), the network could retrieve the pattern '101' from the initial state vectors {(0 0 1), (0 1 1), (1 0 1), (1 1 1)}. The network continued to recall the previous pre-stored '110' with the rest of the initial states. Like the single CAM network, this network exhibited either strong or weak associative memory. For some initial state vectors, the network could directly recall '110' or '101', as they have good associability. Starting from some other initial state vectors, the network had to be updated with associative intermediate state(s) before successful retrieval (such as {(0 0 0), (0 1 1)}) due to weak associability. In Fig. 4c, we schematically summarize the retrieval of pre-stored '110' and '101' from different initial state vectors in a cube where each corner represents a state of the network. The single (red) and multiple (blue) CAMs are illustrated in the same schematic (the joint path is coloured black).

If a memory of the system is represented by the location of a stable point in the state space, partial information about that memory will be contained in nearby states, from which a final stable state with the complete information can be reached[43]. In a Hopfield network, because the final state is reached by association and not by location, the memory can be considered genuinely content-addressable. The convergence flow to stable states is the crux of this CAM operation. The state hypercube can describe the network's energetic entropy. Alternatively, the Hopfield network operation can be described by an energy cost function[43] (see Methods). Correspondingly, we measured the cumulative dynamic small-signal power dissipation $\sum \Delta p_{v_i,v_j}$, which defines the cumulative differences between the total small-signal power consumption of the network in each of two subsequent states ($v_i$ and $v_j$) until convergence to the targeted state, as the invested power. Figure 4d shows the power dissipation difference from $v_i$

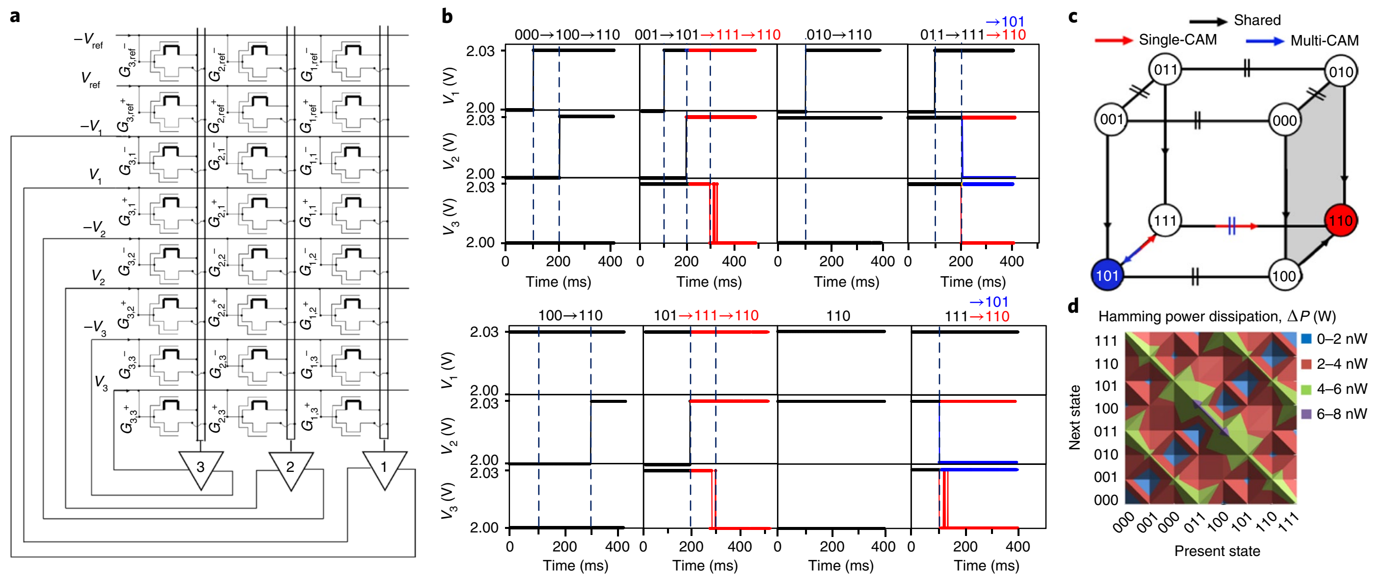

**Fig. 4 | Associative memory of a Y-flash memristive neural network. a**, Circuit schematic of the fabricated Hopfield neural network for three-bit CAM using a Y-flash memristive integrated array and differential synaptic weight structure with symmetric configuration weight matrix. $V_1$, $V_2$ and $V_3$ are the large-signal output bits of the CAM (also used as inputs) representing the output state vector **v** comprising $v_1$, $v_2$ and $v_3$, encoded in small-signal: '1' and '0' are represented by 30 mV and 0 V, respectively, around the 2 V large-signal. **b**, Experimental waveforms of the state vector recursive evolution from different initial states. The network had '110' pre-stored in it in the single CAM (in red), while in the multiple CAM it had an additional pre-stored pattern '101', to which the network converged in some of the initial states (in blue). Dashed lines have been added to some of the waveforms to highlight the transitions between different memory states, and purple and yellow colours have been added to distinguish between the convergence procedure in the single and multiple CAMs (the black line is the common path for both). **c**, Schematic of single ('110') and multiple ('110', '101') CAMs in a state hypercube representation with emphasis on the state vector dynamic evolution flow, where '||' symbols indicate a disconnected line where transistion is impossible between states. **d**, Power dissipation difference from $v_i$ to $v_j$ ($\forall i,j : 0 \leq i,j \leq 7$) as a function of the $i,j$ Hamming distance.

to $v_j$ ($\forall i,j : 0 \leq i,j \leq 7$) as a function of the $i,j$ Hamming distance. The smaller the Hamming distance, the lower the power dissipation. This proves that the network always evolves to the closest intermediate states until it converges to the targeted state. We show that the convergence dynamics of the network resulted in minimal power dissipation along the evolutionary path of the network, from each initial state to the targeted pattern '110' in the single CAM network (Supplementary Section 5e). Under certain conditions, the multiple CAM network also converged to '101' with minimal power dissipation. This widely studied phenomenon is called the local minima of the energy function[45,46]. Furthermore, we show that the synchronicity and firing order between neurons are crucial for minimal power dissipation along the evolutionary path (Supplementary Fig. 35). In our asynchronous winner-take-all configuration, minimal power dissipation was achieved. Other configurations yielded higher power dissipation.

## Training a floating-gate memristive neural network

The second neuromorphic application implements a floating-gate memristive integrated array for in situ training of a single-layer ANN to classify 3×3-pixel black/white images of stylized letters into three classes[22], as illustrated in Fig. 5a. This is a single-layer perceptron; its top-level (functional) scheme is shown in Fig. 5b, with ten inputs and three outputs, fully connected with 10×3 = 30 differential synaptic weights (see Methods), as shown in Fig. 5c. The perceptron's outputs $f_i$ ($i$ = 1, 2, 3) are determined as nonlinear activation functions

$$f_i = \tanh(\beta i_i) \tag{3}$$

of the VMM product components (small-signal currents) $i_i = \sum_{j=1}^{10} w_{ij} v_j$. Here, $v_j$ (with $j$ = 1,…,9) are the small-signal input signals, $v_{10}$ is a constant bias, $\beta = 6.3\times10^5\,\mathrm{A}^{-1}$ is a nonlinearity parameter chosen according to ref. [22], and $w_{ij}$ are trainable differential floating-gate memristive synaptic weights.

We used the Manhattan update rule[22] to train the network. Perfect classification accuracy was achieved after 29 training epochs (Fig. 5d,e), where the relevant neural activation function $f_i(\cdot)$ precisely classifies the letters encoded in one-hot vector representation in response to the corresponding input letter. Due to the exponential dynamics of the programming process, using a constant number of applied programming pulses will lead to a time-varying learning rate $\eta(t)$. Even if $\eta(t)$ is not compensated for (Supplementary Section 5d), its decay contributes to high training accuracy and robustness against variations, overfitting and imprecise quantization of the Manhattan update rule. Furthermore, applying the differential synapse model with the small-signal model immunizes the training against drastic environmental variations (such as temperature)[29]. Evaluating the small-signal power dissipation after training showed that the average small-signal current (in response to 30 mV small-signal voltage) of each Y-flash cell is 67.4 nA, which corresponds to a power consumption of 4 nW per synaptic weight and total power consumption of 3.6 μW of the whole network over a full testing dataset (epoch). The large-signal power consumption might decrease if we shift the $V_{th}$ dynamic range to smaller values and operate with lower $V_r$. Note that the read transistor could be removed from the Y-flash device structure, to potentially enable a denser 1T selector-free array of two-terminal floating-gate memristive devices, relative to the 1T1R configuration mostly used in memristive crossbars. However, this might entail increasing the large-signal power consumption during read operations due to a higher read voltage, while decreasing the effective number of resistive levels due to read and erase disturbs (see Supplementary Section 5f and Supplementary Table 9).

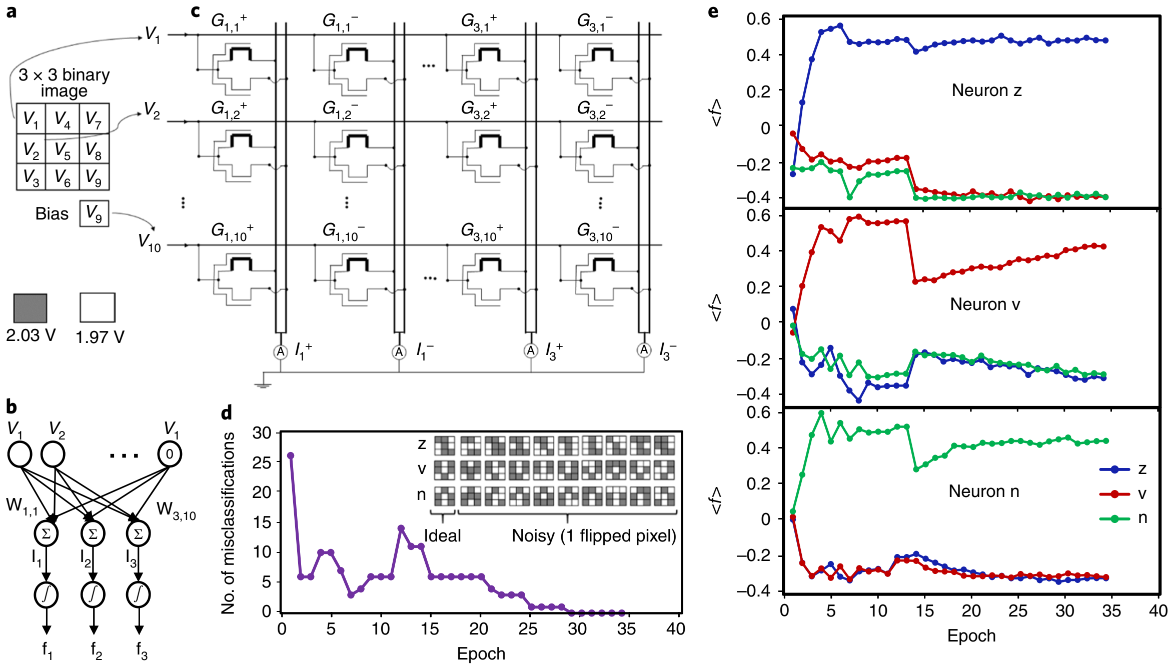

**Fig. 5 | Training and classification of a Y-flash memristive neural network. a**, Input image of 3 × 3 binary images that represent the letters z, v, n and a bias. Black and white pixels are encoded by 30 mV and −30 mV small-signal voltages, respectively. **b**, Top level abstraction of the fully connected single-layer perceptron for the input image classification. **c**, Physical level description of an implementation of a single-layer perceptron using a 10 × 6 fragment of the Y-flash array, where each synapse is differential. **d**, The number of misclassifications, corresponding to the training error, is optimized during the training process (as a function of number of epochs) to the perfect value (zero). The inset shows the used input pattern set. **e**, Experimental results of pattern classification and training, using the Manhattan update rule and mini-batch backpropagation algorithm. The evolution of the output signals is shown, averaged over all patterns of a specific class. The classification was considered successful when the output signal $f_i$ (unitless activation function) corresponding to the correct class of the applied pattern was larger than all other outputs and the corresponding error shown in **d** was zero. Such accurate classification was achieved, on average, after 29 epochs. The illustrated training was continued even after accurate classification was achieved at epoch 29, to verify that the difference between the output signals continued to increase.

## Conclusions

We have reported a power-efficient floating-gate memristive device that combines the technological maturity of floating-gate technology and the computational properties of a memristor. With the aim of closing the technological–functional gap between floating-gate and memristive devices as synapses in neuromorphic systems, we fabricated a two-terminal single-poly floating-gate non-volatile memory device in a standard 180 nm CMOS process (without additional masks), and connected it to a readout transistor. Our device, called a Y-flash cell, operates in an energy-efficient subthreshold memristive mode and is linearized for small-signal changes, allowing a resistance dynamic range of two orders of magnitude. It is precisely tuned using optimized switching voltages and times, and can achieve 65 discrete resistive levels, as well as long analogue data retention, high endurance and low noise margin. The experimental results also indicate that our device could be easily scaled down to advanced standard CMOS technology nodes. Basic learning rules for synaptic emulation were experimentally implemented, including STDP and gradient descent. We fabricated a floating-gate memristive integrated array without selectors to implement an ANN. A physical-level computation of the analogue VMM—the most computationally intensive operation in any neuromorphic network—was demonstrated, enabled by Ohm's law, Kirchhoff's current law and linearization. We employed a differential synapse model, comprising two Y-flash cells, to allow programming of positive and negative values. A three-bit Hopfield ANN was also constructed, and single and multiple associative memories were achieved. Moreover, a single-layer ANN for 3 × 3 image classification of z, v and n letters was trained in situ.

## Methods

**Memristive device model.** In the classic representation, the conductance of a memristive device (or memductance $G$) depends on a state variable $w$, which is itself a function of the applied voltage $V$. Formally, a memristive device obeys the following[17]:

$$I(t) = G(w, V, t)V(t) \tag{4a}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = f(w(t), V(t)) \tag{4b}$$

where $f$ is a continuous function.

**Y-flash operation methods.** When a positive voltage is applied to the drain (while source and substrate are at ground potential), a fraction of the drain voltage (typically 60–80%) is transferred to the FG. If the transferred voltage exceeds the threshold voltage of NMOS ($V_{\mathrm{th}}$), the Y-flash cell conducts in saturation mode. When higher voltage is applied to the drain (such as 5 V in a 110 Å gate-oxide device), channel hot electrons are generated in the drain junction (CHEI mechanism). Some of these electrons are injected into the FG, increasing $V_{\mathrm{th}}$. This corresponds to the programming of a memory cell. To erase the Y-flash device, high positive voltage is applied to the source, while the drain and substrate are kept at ground potential. In this case, there is no current in the Y-flash device channel. Hot holes are generated by BTBT in the source junction. The hot holes are injected into the FG and reduce the $V_{\mathrm{th}}$ of the Y-flash device.

If the Y-flash cell has two NMOS transistors (dual-channel device), the program/erase part can be separated from the readout part. The injection transistor is optimized to enhance hot carrier injection (it has a shorter channel and a p-type component in the lightly doped drain junction). The read transistor is optimized to suppress hot carrier injection. During reading, the source of the injection transistor is floating (high-Z) or shorted to the drain. During programming, the source of the read transistor is floating or shorted to the drain. During erasing, the drain and source of the read transistor are floating or shorted to the substrate. The substrate is kept at zero potential in all operation modes. Similar operation has been obtained in RRAM devices by employing a buffer[47].

**Floating-gate memristive operation model.** The small-signal resistance of a floating-gate device operating in a memristive mode is determined as

$$R_{\mathrm{mem}}=\left(\frac{\mathrm{d}I_{\mathrm{DS}}}{\mathrm{d}V_{\mathrm{DS}}}\right)^{-1}=\frac{mV_{\mathrm{T}}}{\mathrm{CR}}\frac{1}{I_{\mathrm{DS}}}\bigg|_{V_{\mathrm{DS}}=V_{\mathrm{r}}} \tag{5a}$$

$$\frac{\mathrm{d}R_{\mathrm{mem}}}{\mathrm{d}V_{\mathrm{th}}}=\frac{R_{\mathrm{mem}}}{mV_{\mathrm{T}}}=\frac{1}{\mathrm{CR}\times \mathrm{I}_{\mathrm{read}}}\mathrm{e}^{\frac{-\mathrm{CR}\times V_{\mathrm{r}}}{mV_{\mathrm{T}}}}\mathrm{e}^{\frac{V_{\mathrm{th}}}{mV_{\mathrm{T}}}} \tag{5b}$$

Below 1 nA the current approaches the noise level of fast analogue amplifiers, and above 1 μA the device is in the above-threshold conduction mode. The state variable dynamic range is determined as $\Delta V_{\mathrm{th}}=\frac{mV_{\mathrm{T}}}{\mathrm{CR}}\ln\left(\frac{I_{\mathrm{on}}}{I_{\mathrm{off}}}\right)\approx 1\,\mathrm{V}$. Therefore, its dynamic range is $V_{\mathrm{th}}\in[V_{\mathrm{r}}-\Delta V_{\mathrm{th}}:V_{\mathrm{r}}]$.

The RESET/programming process is modelled by the 'lucky electron' model of CHE injection[48]:

$$\frac{\mathrm{d}V_{\mathrm{th,RESET}}}{\mathrm{d}t}=\frac{I_{\mathrm{inj}}}{C_{\mathrm{fr}}}\approx\overbrace{\frac{K'}{C_{\mathrm{fr}}}(\mathrm{CR}\times V_{\mathrm{DS}}-V_{\mathrm{th}})^{n}P(E_{\mathrm{V}})}^{f\left(V_{\mathrm{DS}},V_{\mathrm{th,RESET}}\right)} \tag{6}$$

where $C_{\mathrm{fr}}$ includes the floating gate oxide capacitance and fringing gate–drain capacitance, $I_{\mathrm{inj}}$ is the equivalent current of injected hot electrons to the shared floating gate via the injection transistor and equals $I_{\mathrm{DS}}\times P(E_{\mathrm{V}})$, and $P(E_{\mathrm{V}})$ is the probability of a hot electron travelling a sufficient distance to gain energy $E_{\mathrm{V}}$ without a collision. $I_{\mathrm{DS}}$ is the transistor channel current:

$$I_{\mathrm{DS}}=\frac{\mu C_{\mathrm{ox}}}{2}\frac{W}{L}(V_{\mathrm{GS}}-V_{\mathrm{th}})V_{\mathrm{DS,sat}} \tag{7}$$

where $V_{\mathrm{DS,sat}}$ is the saturation velocity as a result of high drain voltage and is equal to $V_{\mathrm{DS,sat}}(L)\approx(V_{\mathrm{GS}}-V_{\mathrm{th}})||(L\times E_{\mathrm{sat}})$. $K'$ is a technology parameter and is a function of the transistor width ($W$), carrier mobility ($\mu$), gate oxide capacitance ($C_{\mathrm{ox}}$), saturation velocity field ($E_{\mathrm{sat}}$) and channel length ($L$) (short/long channel modulation). Furthermore, the channel length modulation will determine $n$ and the probability $P(E_{\mathrm{V}})$[48] (Supplementary Section 3b).

Analogously, the SET/erasing process is a BTBT hot holes injection[49] modelled using gate-induced drain leakage current (GIDL):

$$I_{\mathrm{GIDL}}=AE_{\mathrm{s}}\mathrm{e}^{-\frac{B}{E_{\mathrm{s}}}},E_{\mathrm{s}}=\frac{\mathrm{CR}\times V_{\mathrm{s}}-V_{\mathrm{th}}+V_{\mathrm{ox}}}{t_{\mathrm{ox}}} \tag{8a}$$

$$\frac{\mathrm{d}V_{\mathrm{th,SET}}}{\mathrm{d}t}=\frac{I_{\mathrm{GIDL}}}{C_{\mathrm{dep}}}=\underbrace{\frac{A\times(\mathrm{CR}\times V_{\mathrm{s}}+V_{\mathrm{ox}}-V_{\mathrm{th}})}{C_{\mathrm{dep}}t_{\mathrm{ox}}}\mathrm{e}^{-\frac{Bt_{\mathrm{ox}}}{\mathrm{CR}\times V_{\mathrm{s}}+V_{\mathrm{ox}}-V_{\mathrm{th}}}}}_{f\left(V_{\mathrm{DS}},V_{\mathrm{th,SET}}\right)} \tag{8b}$$

where $A$ and $B$ are constants for indirect phonon-assisted tunnelling (Supplementary Section 3c), $E_{\mathrm{s}}$ is the vertical electrical field at the silicon surface, $t_{\mathrm{ox}}$ is the oxide thickness in the overlap region, $V_{\mathrm{ox}}=-Q_{\mathrm{dep}}/C_{\mathrm{ox}}$ is the oxide voltage and $C_{\mathrm{dep}}$ is the depletion layer capacitance.

Notably, equation (1) emulates equation (4a); however, equation (2) is analogous to equation (4a) under certain conditions, and equation (6) and equation (8b) are analogous to equation (4b). Consequently, the proposed device operates in a memristive mode, exhibiting a linear $I$–$V$ in small-signal, as shown in the inset of Fig. 2c.

**STDP implementation.** A small-signal leaky-integrate-and-fire (LIF)[8] neuron is emulated in software, where its time event is determined proportionally to the sensed current magnitude of the device under test (DUT), after which the update rule is post-processed by means of a software driver that controls the equipment. This spike generator could be implemented by standard CMOS circuits and easily integrated with the Y-flash device: however, our methodology is sufficient for a proof of concept. For LTD, a reference small-signal current with maximal magnitude (−0.2 μA) is added, by Kirchhoff's current law (KCL), to the DUT current (a positive small-signal swing starting from 0.2 μA). This ensures that the post-synaptic neuron fires earlier than the pre-synaptic neuron in response to a negative current. Similarly for LTP, a reference small-signal current with minimal magnitude (−0.2 nA) is added (a positive small-signal input swing starting from −0.2 nA). The sign of the time difference between pre- and post-synaptic neural spikes $\Delta t$ determines whether potentiation or depression will be induced, and the magnitude determines the duration of LTP/LTD. The magnitude is then translated to the corresponding number of applied erasing/programming pulses.

Without loss of generality, from equations (6) and (8b), a parametric formula of $V_{\mathrm{th}}$ as a result of injection/tunnelling time can be obtained (Supplementary Section 3b,c):

$$V_{\mathrm{th}}(t)=\alpha\mathrm{e}^{-t/\tau}+\gamma \tag{9}$$

where $\alpha$, $\tau$ and $\gamma$ are fitting parameters. Therefore, we developed the synaptic weight update rule $\Delta W$ ($W=1/R_{\mathrm{mem}}$) of STDP using equations (5b) and (9), in response to $\Delta t$:

$$\frac{\Delta W}{\Delta t}=\frac{\mathrm{d}W}{\mathrm{d}V_{\mathrm{th}}}\times\frac{\mathrm{d}V_{\mathrm{th}}}{\mathrm{d}t}=\frac{\alpha W}{mV_{\mathrm{T}}\tau}\mathrm{e}^{-\frac{\Delta t}{\tau}} \tag{10}$$

**VMM array implementation.** A 12×8 Y-flash array was fabricated to implement the VMM. $W_{n,m}$ is initially determined after supplying a large-signal voltage $V_{\mathrm{r}}$ and measuring the current (equation (5a)), and $v_n$ is obtained by subtracting two voltages (large-signal and large-signal + small-signal) that are successively supplied to the same row ($n$ input). Correspondingly, $i_m$ is the small-signal readout current product of the $m$ column and is obtained by subtracting the currents in response to the two successive voltages. The sum (KCL) of the large-signal currents (operating point in response to $V_{\mathrm{r}}$) of all the devices in a certain column ($m$ output) is stored in software.

We measured, analysed and validated all the scenarios that complicate memristive crossbar operation. These included sneak path currents in parallel to the DUT, which decrease its resistance while reading[50], and other paths containing devices that might be disturbed while programming, proportionally to voltage divider values (half-select). There are also flash array issues, including erase in the granularity of columns (segmented erase), program disturb of adjacent cells while programming the DUT, read disturb, current leakage through the substrate, endurance (100,000 cycles) due to oxide degradation, charge traps that decrease analogue data retention and stochastics (Supplementary Section 5b). All of these are thoroughly addressed, modelled and illustrated using SPICE simulations (Supplementary Section 4). We propose simple techniques to overcome these issues, and modify the programming/erasing operational methods of a Y-flash device inside an array (Supplementary Section 5a). Our experiments showed that the maximum collective impact of these sources corresponds to a 10 nA variation of the read current within the array. The noise margin (NM) is determined as a ratio between the variation (10 nA) and the current magnitude (maximum 1 μA). The corresponding number of resistive levels is extracted with

$$\left(\frac{1+\mathrm{NM}}{1-\mathrm{NM}}\right)^{N}\leq\frac{R_{\mathrm{off}}}{R_{\mathrm{on}}}=1,000 \tag{11}$$

A sufficiently accurate solution to this condition is $N=65$. This is the number of effective distinct resistive levels, which is roughly equivalent to 6 bit precision (Supplementary Section 5c). The programming algorithm can program each cell with 1% accuracy.

Array issues can be handled by programming separate rows, using 4.5 V programming voltage instead of 5 V to decrease the crosstalk impact, starting from the farthest desired currents to be programmed, encoding the other floating devices in the same row to the highest resistive state (HRS) in order to reduce current leakage, and decreasing the practical current dynamic range to two orders of magnitude, with $R_{\mathrm{off}}\approx 14.5\,\mathrm{M\Omega}$, which corresponds to 10 nA (noise level). We also adopt a differential synapse model to overcome real-time variations and avoid the power-starved segmented erase (used only for initializing the devices to currents below 1 μA). In this model, each synapse comprises two Y-flash cells. In addition, this model facilitates hardware realization of zero, positive and negative weights, essential in ANNs. The effective synaptic weight at node $i,j$ is determined as

$$W_{ij}=G_{ij}^{+}-G_{ij}^{-} \tag{12}$$

In the first application, the two devices $\left(G_{ij}^{\pm}\right)$ are located in the same column and adjacent rows (vertical) and are driven by opposite voltages. In the second application, they are located in the same row and adjacent columns (horizontal) and their outputs are subtracted.

**CAM implementation.** The synaptic weights of the entire network are described by a symmetric matrix with zero diagonal elements, $W=\begin{pmatrix} w_{11} & w_{12} & w_{13}\\ w_{21} & w_{22} & w_{23}\\ w_{31} & w_{32} & w_{33}\end{pmatrix}$, where $\forall i\neq j$, $w_{ij}=w_{ji}$ and $w_{ii}=0$. The threshold of the artificial neurons is represented by the small-signal current threshold vector $\mathbf{T}=(\theta_1\,\theta_2\,\theta_3)$, where $\theta_i=\left(G_{i,\mathrm{ref}}^{+}-G_{i,\mathrm{ref}}^{-}\right)\cdot v_{\mathrm{ref}}$, and the small-signal reference voltage (its state always is '−1') is $v_{\mathrm{ref}}=-30\,\mathrm{mV}$; the states of the three neurons are represented by the state vector $\mathbf{v}=(v_1\,v_2\,v_3)$, where $v_1$, $v_2$ and $v_3$ are the corresponding small-signal voltages of the states of neurons 1, 2 and 3, respectively. The synapses and thresholds were implemented in hardware using a 12×8 integrated Y-flash array. Neurons were emulated in software first using the sign function on the total small-signal current sensed from a column. '1' and '0' are represented by 30 mV and 0 V small-signal, respectively (around 2 V large-signal) and determined by the software. New states represented by the small-signal product (determined by two successive steps) of the neurons and supplied by software are recursively updated according to

$$v(t+1)=\mathrm{sign}(v(t)\cdot W-T) \tag{13}$$

where $t$ represents the number of updating cycles and $t=0$ corresponds to the initial vector $\mathbf{v}(0)$. The transient neuron function of $v_1$, $v_2$, $v_3$ occurs in asynchronous order proportionally to the input strength of each neuron (for example, the column with the highest sensed current drives the first neuron to fire), using an asynchronous LIF small-signal neuron emulated in software.

In the neuromorphic context, the pre-coded patterns are stored into the network by a recursive fine-tuning of the matrix $W$ (the incremental resistance matrix of a pair of devices that implement the differential synapse). To store the binary pattern '110' into the network, the positive and negative resistance matrices and threshold resistance vectors were set as

$$R^+ = \begin{pmatrix} 2.16 & 0.22 & 0.19 \\ 0.18 & 3.11 & 4.93 \\ 0.41 & 0.36 & 2.6 \end{pmatrix} \mathrm{M\Omega},\ \mathbf{R}^+_{\mathrm{ref}} = \begin{pmatrix} 0.27 \\ 0.21 \\ 0.19 \end{pmatrix} \mathrm{M\Omega}$$

$$R^- = \begin{pmatrix} 3.01 & 2 & 0.22 \\ 0.83 & 4 & 0.4 \\ 0.64 & 0.2 & 3 \end{pmatrix} \mathrm{M\Omega},\ \mathbf{R}^-_{\mathrm{ref}} = \begin{pmatrix} 0.43 \\ 0.3 \\ 0.25 \end{pmatrix} \mathrm{M\Omega}$$

The corresponding weight matrix and threshold weights vector were approximated as

$$W \approx 0.7\mu \begin{pmatrix} 0 & 6.17 & 1.13 \\ 6.17 & 0 & -3.31 \\ 1.13 & -3.31 & 0 \end{pmatrix},\ \mathbf{G}_{\mathrm{ref}} \approx 0.7\mu \begin{pmatrix} 1.95 \\ 1.95 \\ 1.95 \end{pmatrix} \quad (14)$$

To achieve the targeted matrices and vectors in equation (14), the incremental resistances of the relevant Y-flash cells, determined by equation (5a), were tuned by applying fine-tuned 4.5 V programming pulses using the developed techniques (Supplementary Section 5e) and the offline training algorithm[43]. The positive and negative resistance matrices are not symmetric, but the division on their difference yields a symmetric weight matrix. This property enhances network flexibility. The corresponding weight matrix and threshold weights vector for the multi-associative memory were approximately reconfigured to

$$W \approx 0.7\mu \begin{pmatrix} 0 & 1.23 & 1.13 \\ 1.23 & 0 & -3.31 \\ 1.13 & -3.31 & 0 \end{pmatrix},\ \mathbf{G}_{\mathrm{ref}} \approx 0.7\mu \begin{pmatrix} 1.95 \\ 1.95 \\ 1.95 \end{pmatrix} \quad (15)$$

We observed that multi-associative memory could be implemented with minimal changes to the original matrix and vector in equation (14); we therefore reconfigured only $w_{12}$ and $w_{21}$ using the same tuning methodology. Consequently, we programmed $R^+_{12}$ and $R^+_{21}$ to decrease the synaptic values to 0.74 MΩ and 0.49 MΩ, respectively. This is visualized in Fig. 4c: the schematic is divided into two different squares, connected by an arrow from '111' to '110' in the single CAM case. For multiple CAM, this arrow is disconnected, where the convergence in the left square is towards '101'. We thus reconfigured the network to pre-store the pattern '101'. The arrow from '111' to '110' is disconnected as a result, and the squares of the cube are divided into two parts, neither of which is accessed by the initial states of the other.

The energy cost function used to evaluate the optimal power dissipation is given by[43]

$$E = -\frac{1}{2}\sum\sum\nolimits_{j\neq i} w_{ij}v_iv_j - \sum\nolimits_j v_ji_j - \sum\nolimits_j \left(G_{j,\mathrm{ref}} + \sum\nolimits_i w_{ij}\right)\int_0^{v_j} f^{-1}(V)\mathrm{d}v \quad (16)$$

fitted to the proposed system (small-signal energy), where $v_i$, $v_j$ are the pre- and post-synaptic neurons, respectively, and $f(\cdot)$ is the neural activation function (sign in our case). The last term can be ignored for very steep transfer functions[43]. When the state of the neuron $j$ is updated by $\Delta v_j$

$$\Delta E = -\left[\sum\nolimits_{i\neq j} w_{ij}v_i - i_j\right]\Delta v_j = -\left[\sum\nolimits_{i\neq j} w_{ij}v_i - \left(G^+_{j,\mathrm{ref}} - G^-_{j,\mathrm{ref}}\right)\cdot v_{\mathrm{ref}}\right]\Delta v_j \quad (17)$$

where the product in brackets is equal to the total current of neuron $j$ by KCL (small-signal). $v_j$ is positive (after applying the sign function on the product) when this product is positive, and zero otherwise. Thus, any change in $E$ is negative. The network will settle with minimal energy changes $\Delta E$ until the convergence to the targeted pattern in steady state. This energy function is used as a training cost function by the STDP rule, when calibrating the weights until the targeted pre-stored pattern is robustly recalled.

**Neural network training implementation.** The network receives nine inputs corresponding to the pixel values. We tested the network on a set of $N=30$ patterns, including three stylized letters (z, v and n) and three sets of nine noisy versions of each letter, formed by flipping one of the pixels of the original image (inset, Fig. 5d). Because the set size was very small, it was used for both training and testing[22]. Each input signal was represented by a small-signal voltage $v_j$ equal to either +30 mV or −30 mV, corresponding, respectively, to the black or white pixel, while the bias input $v_{10}$ was equal to −30 mV. Such coding balances the benchmark input set: the sum of all input signals across all patterns of a particular class is guaranteed to be close to zero, speeding up the convergence process. To realize the neural activation function, we measured the currents in each column (in two successive steps), after which the small-signal value, differential subtraction and the tanh function defined by equation (3) were post-processed in software.

At each iteration ('epoch') of this procedure, patterns from the training set were applied, one by one, to the network's input, and its outputs $f_i(n)$, where $n$ is the pattern number, were used to calculate the delta-rule weight increments based on the derivative of the mean square error cost function and backpropagation algorithm:

$$E = \frac{1}{2}\sum_{p=1}^{N_p}\left[f_i^{(g)}(n) - f_i(n)\right]^2 \quad (18a)$$

$$\Delta_{ij} = \frac{\mathrm{d}E}{\mathrm{d}w_{ij}} = \frac{\mathrm{d}E}{\mathrm{d}f_i(n)} \times \frac{\mathrm{d}f_i(n)}{\mathrm{d}w_{ij}} = \delta_i(n)v_j(n) \quad (18b)$$

$$\delta_i(n) = \left[f_i^{(g)}(n) - f_i(n)\right]\frac{\mathrm{d}f}{\mathrm{d}i}\bigg|_{i=i_i(n)} = \left[f_i^{(g)}(n) - f_i(n)\right]\cdot\left(1 - f_i(n)^2\right) \quad (18c)$$

Here $f_i^{(g)}(n)$ is the target value of the $i$th output for the $n$th input pattern. (In our system these values were chosen to be +0.85 for the output corresponding to the correct pattern class and −0.85 for the output corresponding to the wrong class, as suggested in ref. [22].) Once all $N$ patterns of the training set were applied and all $\Delta_{ij}(n)$ calculated in software, the synaptic weights were modified using the following Manhattan update rule:

$$\Delta w_{ij} = \eta\,\mathrm{sign}\sum_{n=1}^{N}\Delta_{ij}(n) \quad (19)$$

where $\eta$ is a constant that scales the training rate. The Manhattan update rule and the batch-mode delta rule of supervised training differ only in the binary quantization, expressed in equation (19) by the 'sign' function, which simplifies the hardware implementation of the delta rule as recommended in ref. [30]. The training is executed in hardware after calculating equation (19) in software. We randomly initialized all the positive and negative incremental resistances of the Y-flash cells ($R^+$, $R^-$) around the lowest resistive state (LRS). Then, during training and according to the average delta rule (equation (19)) along the entire epoch, we updated the differential synapse using only 5 V programming pulses (applied on the positive resistance to decrease the weight, and on the negative resistance to increase the weight). We floated the whole bias row ($v_{10}$) at training epoch number 14 and validated convergence to the desired classification function with sufficient accuracy.

**Measurements and characterization set-up.** The d.c. conduction and switching characteristics of the Y-flash were collected by an HP4156A Precision Semiconductor Parameter Analyzer, which was connected to the devices under test using a Cascade Microtech 12000 probe station. An eight-channel arbitrary waveform generator (NI PCI-6733) was a part of the set-up. During training, the drain node of the Y-flash was connected to the pulse generator, while the other nodes were connected to the source measure units (SMUs) to probe the total current. The waveform generator and fast current measurement units were connected to the inputs of a switching matrix. A Keithley 707 Switching Matrix (64→8) was adopted to independently access each device during the read, programming and erasing phases. The switching matrix was connected to the array of Y-flash devices using a 32 pin non-wired probe card (Wentworth). Drain inputs with the same voltage value shared the same channel of the waveform generator. All equipment in the set-up was controlled by the NI Labview environment using in-house customized software. This software enabled the different accurate read/programming/erasing tuning algorithms and cycling/training/multi-tasking protocols discussed in this Article.

## Data availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

## Code availability

The computer codes used in this study are available within this paper and its Supplementary Information files.

Received: 21 February 2019; Accepted: 24 October 2019;
Published online: 09 December 2019

## References

1. Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
2. Hasler, J. & Marr, B. Finding a roadmap to achieve large neuromorphic hardware systems. *Front. Neurosci.* **7**, 118 (2013).
3. Benjamin, B. V. et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* **102**, 699–716 (2014).
4. Furber, S. B., Galluppi, F., Temple, S. & Plana, S. The SpiNNaker project. *Proc. IEEE* **102**, 652–665 (2014).
5. Indiveri, G. et al. Neuromorphic silicon neuron circuits. *Front. Neurosci.* **5**, 73 (2011).
6. Likharev, K. K. CrossNets: neuromorphic hybrid CMOS/nanoelectronic networks. *Sci. Adv. Mater.* **3**, 322–331 (2011).
7. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
8. Diorio, C., Hasler, P., Minch, A. & Mead, C. *Neuromorphic Systems Engineering: Neural Networks in Silicon* Ch. 14 (Springer, 1998).
9. Diorio, C., Hasler, P., Minch, A. & Mead, C. A single-transistor silicon synapse. *IEEE Trans. Electron. Dev.* **43**, 1972–1980 (1996).
10. Hasler, P., Minch, B. A. & Diorio, C. Adaptive circuits using pFET floating-gate devices. In *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI (ARVLSI)* 215–229 (IEEE, 1999).
11. Hasler, P., Diorio, C., Minch, B. A. & Mead, C. Single transistor learning synapses. In *Proceedings of the 7th International Conference on Neural Information Processing Systems (NIPS)* 817–824 (ACM, 1994).
12. Hasler, P., Minch, B. A. & Diorio, C. An autozeroing floating-gate amplifier. *IEEE Trans. Circ. Syst. II* **48**, 74–82 (2001).
13. Ramakrishnan, S., Hasler, P. & Gordon, C. Floating gate synapses with spike time dependent plasticity. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)* 369–372 (IEEE, 2010).
14. Wong, H. S. P. & Salahuddin, S. Memory leads the way to better computing. *Nat. Nanotechnol.* **10**, 191–194 (2015).
15. Waser, R. & Aono, M. Nanoionics-based resistive switching memories. *Nat. Mater.* **6**, 833–840 (2007).
16. Chua, L. O. Memristor—the missing circuit element. *IEEE Trans. Circuit Theory* **18**, 507–519 (1971).
17. Chua, L. O. & Kang, S. M. Memristive devices and systems. *Proc. IEEE* **64**, 209–223 (1976).
18. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453**, 80–83 (2008).
19. Jo, S. H. et al. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **10**, 1297–1301 (2010).
20. Wang, Z. et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* **1**, 137–145 (2018).
21. Xia, Q. & Yang, J. J. Memristive crossbar arrays for brain-inspired computing. *Nat. Mater.* **18**, 309–323 (2019).
22. Prezioso, M. et al. Training and operation of an integrated neuromorphic network based on metal–oxide memristors. *Nature* **521**, 61–64 (2015).
23. Merrikh Bayat, F. et al. Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* **9**, 2331 (2018).
24. Adam, G. C., Khiat, A. & Prodromakis, T. Challenges hindering memristive neuromorphic hardware from going mainstream. *Nat. Commun.* **9**, 5267 (2018).
25. Niu, D., Chen, Y., Xu, C. & Xie, Y. Impact of process variations on emerging memristor. In *Proceedings of the 47th Design Automation Conference (DAC)* 877–882 (IEEE, 2010).
26. Waser, R., Dittmann, R., Staikov, G. & Szot, K. Redox-based resistive switching memories—nanoionic mechanisms, prospects and challenges. *Adv. Mater.* **21**, 2632–2663 (2009).
27. Pouyan, P., Amat, E. & Rubio, A. Reliability challenges in design of memristive memories. In *Proceedings of the 5th European Workshop on CMOS Variability (VARI)* 1–6 (IEEE, 2014).
28. Indiveri, G. et al. Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* **24**, 384010 (2013).
29. Merrikh Bayat, F. et al. High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 4782–4790 (2018).
30. Merrikh Bayat, F. et al. Redesigning commercial floating-gate memory for analog computing applications. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)* 1921–1924 (IEEE, 2015).
31. Merrikh Bayat, F. et al. Model-based high-precision tuning of NOR flash memory cells for analog computing applications. In *Proceedings of the Device Research Conference (DRC)* 1–2 (IEEE, 2016).
32. Guo, X. et al. Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)* 1–4 (IEEE, 2017).
33. Guo, X. et al. Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology. In *Proceedings of the International Electron Devices Meeting (IEDM)* 6.5.1–6.5.4 (IEEE, 2017).
34. Ziegler, M. et al. Memristive operation mode of floating gate transistors: a two-terminal MemFlash-cell. *Appl. Phys. Lett.* **101**, 263504 (2012).
35. Ziegler, M. & Kohlstedt, H. Mimic synaptic behavior with a single floating gate transistor: a MemFlash synapse. *J. Appl. Phys.* **114**, 194506 (2013).
36. Himmel, N. et al. Memristive device based on a depletion-type SONOS field effect transistor. *Semicond. Sci. Technol.* **32.6**, 06LT01 (2017).
37. Winterfeld, H. et al. Technology and electrical characterization of MemFlash cells for neuromorphic applications. *J. Appl. Phys.* **51**, 324003 (2018).
38. Roizin, Y. & Pikhay, E. Memristor using parallel asymmetrical transistors having shared floating gate and diode. US patent 9,514,818 (2016).
39. Sharroush, S. M., Abdalla, Y. S., Dessouki, A. A. & El-Badawy, E. S. A. Subthreshold MOSFET transistor amplifier operation. In *Proceedings of the 4th International Design Test Workshop (IDT)* 1–6 (IEEE, 2009).
40. Chang, T., Jo, S. H. & Lu, W. Short-term memory to long-term memory transition in a nanoscale memristor. *ACS Nano* **5**, 7669–7676 (2011).
41. Caporale, N. & Dan, Y. Spike timing-dependent plasticity: a Hebbian learning rule. *Annu. Rev. Neurosci.* **31**, 25–46 (2008).
42. Brivio, S. et al. Extended memory lifetime in spiking neural networks employing memristive synapses with nonlinear conductance dynamics. *Nanotechnology* **30**, 015102 (2018).
43. Hu, S. G. et al. Associative memory realized by a reconfigurable memristive Hopfield neural network. *Nat. Commun.* **6**, 7522 (2015).
44. Verleysen, M., Sirletti, B., Vandemeulebroecke, A. & Jespers, P. G. A. A high-storage capacity content-addressable memory and its learning algorithm. *IEEE Trans. Circ. Syst.* **36**, 762–766 (1989).
45. Tank, D. & Hopfield, J. J. Simple 'neural' optimization networks: an A/D converter, signal decision circuit and a linear programming circuit. *IEEE Trans. Circ. Syst.* **33**, 533–541 (1986).
46. Hopfield, J. J Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl Acad. Sci. USA* **81**, 3088–3092 (1984).
47. Sandrini, J. et al. Effect of metal buffer layer and thermal annealing on HfOx-based ReRAMs. In *Proceedings of the IEEE International Conference on the Science of Electrical Engineering (ICSEE)* 1–5 (IEEE, 2016).
48. Ko, P. K., Hu, C. & Tam, S. Lucky-electron model of channel hot-electron injection in MOSFET's. *IEEE Trans. Electron Dev.* **31**, 1116–1125 (1984).
49. Chan, T. Y., Chen, J., Ko, P. K. & Hu, C. The impact of gate-induced drain leakage current on MOSFET scaling. In *Proceedings of the International Electron Devices Meeting (IEDM)* 718–721 (IEEE, 1987).
50. Zidan, M. A., Fahmy, H. A. H., Hussain, M. M. & Salama, K. N. Memristor-based memory: the sneak paths problem and solutions. *Microelectron. J.* **44**, 176–183 (2013).

## Acknowledgements

This research was partially supported by the Israeli Planning and Budgeting Committee Fellowship, by the Israel Ministry of Economics KAMIN project no. 57681, by the Andrew and Erna Finci Viterbi Graduate Fellowship and by the European Research Council under the European Union's Horizon 2020 Research and Innovation Programme under agreement no. 757259.

## Author contributions

L.D., Y.R., R.D. and S.K. designed the study. L.D., E.H. and E.P. performed experiments and collected data. E.P. and Y.R. invented the Y-flash memristive structure and L.D., R.D. and S.K. invented the subthreshold small-signal memristive operation mode. L.D developed models and executed simulations. All authors analysed the data, discussed the results and wrote the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at https://doi.org/10.1038/s41928-019-0331-1.

**Correspondence and requests for materials** should be addressed to S.K.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.