

In the format provided by the authors and unedited.

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

Loai Danial¹, Evgeny Pikhay², Eric Herbelin¹, Nicolas Wainstein¹, Vasu Gupta ¹, Nimrod Wald¹, Yakov Roizin², Ramez Daniel³ and Shahar Kvatinsky ^{1*}

¹Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel. ²TowerJazz, Migdal HaEmek, Israel.

³Department of Biomedical Engineering, Technion – Israel Institute of Technology, Haifa, Israel. *e-mail: shahar@ee.technion.ac.il

Table of Contents

1. Y-flash Technology	2
2. Y-flash subthreshold small-signal model	5
a. Ideal subthreshold small-signal Y-flash analysis	5
b. Y-flash small-signal equivalent circuit	6
3. Modelling the Y-flash dynamics	10
a. Read mode	10
b. Programming mode	10
c. Erase mode	11
4. SPICE models of the Y-flash	13
a. Verilog-A code of the voltage-controlled Y-flash behavioural model	13
b. Layouts	17
5. Reliability characterization and analysis	18
a. Non-ideal effects of Y-flash array	18
b. Stochastics, Endurance, analogue data retention, and read disturb	22
c. Number of resistive levels	24
d. Adaptive Learning Rate	25
e. Hopfield CAM Energy	26
f. Readout Transistor	28
6. Comp. analysis b/w emerging memory tech. for neuromorphic computing	30
References	35

1. Y-flash Technology

In this section we describe non-volatile memory (NVM) cells, and more specifically, NVM cell arrays that are “embedded” in (*i.e.*, integrally formed with) complementary metal-oxide-semiconductor (CMOS) integrated circuits (ICs). Methods for fabricating the NVM cells in an inexpensive manner using standard CMOS process flows will also be discussed.

Most CMOS IC manufacturers (aka “fabs”) generate standardized process “flows” for generating CMOS ICs on monocrystalline silicon wafers. Each CMOS process flow includes a series of processing steps (*e.g.*, material deposition, photolithographic exposure, and etching) that are required to produce a desired CMOS IC product. Standard CMOS process flows are typically developed to produce “normal” CMOS IC devices (*i.e.*, CMOS IC devices that comprise mainly volatile n-type and p-type MOSFETS) using a minimum number of processing steps in order to minimize overall production costs. Significant effort is typically expended by each manufacturer to make their standard CMOS process flow as time and cost efficient as possible. Once a standard CMOS flow is optimized, it can typically be used to make a large number of CMOS IC designs by merely by providing a different set of photolithography masks for each IC design, and then repeating the standard CMOS process flow using the selected set of masks.

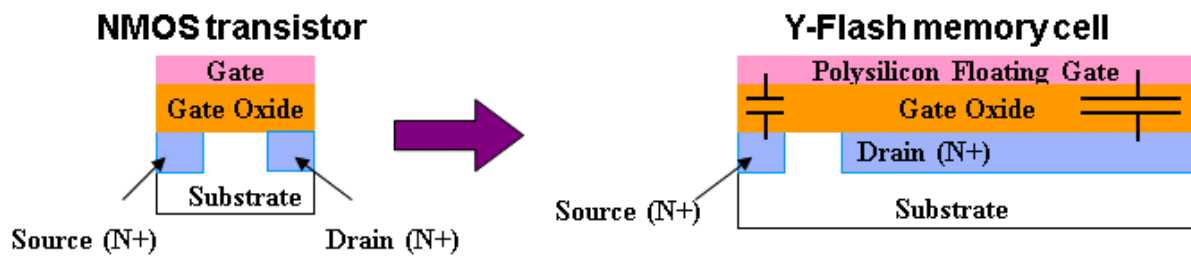
Although most standard CMOS process flows facilitate the inclusion of non-MOSFET circuit components into the CMOS IC products, a problem arises when a circuit design requires a circuit component that cannot be produced by the standard CMOS process flow. In this case, the CMOS process flow must be modified at a great expense to include additional steps in order to produce the needed circuit component. Non-volatile memory (NVM) or “floating gate” cells represent one type of non-standard circuit component that is often needed in large scale CMOS ICs. In contrast to volatile (aka primary storage) memory such as typical n-type and p-type MOSFETs, which require continuous power to retain stored information, NVM cells are able to retain a stored state even when power to an IC is turned off, thereby allowing the IC to “remember” important operating conditions and information upon restart.

It is strongly desirable to be able to fabricate a non-volatile memory (memristive devices) within the core CMOS flow without additional masks. Several solutions allow integrating reprogrammable NVM. Our Y-flash device features asymmetric NVM (logic) cells having record-small areas ($< 3\mu\text{m}^2/\text{cell}$) in a standard $0.18\mu\text{m}$ CMOS process flow. The memory is programmed with low voltages, and it exhibits high endurance and yield. In more advanced technologies, the Y-flash cell area can be significantly scaled down (*e.g.*, $< 1\mu\text{m}^2/\text{cell}$ in 65nm technology node).

Our Y-flash device is devised for use with medium-sized (*i.e.*, several Kbits to 1 Mbit), asymmetric NVM (logic) cells that are produced using standard single-poly CMOS process flows. Each NVM cell is asymmetric in that it includes a floating gate that has a first portion disposed over a channel region and a second (extension) portion that overlaps (*i.e.*, extends “over”) an enlarged drain (diffusion) region at a location away from the channel region, thereby providing a gate-to-drain capacitance that is higher than the gate-to-source capacitance. This asymmetric arrangement facilitates desirable hot electron programming of the floating gate by applying a positive programming voltage to the drain region and coupling the source region to ground, thereby causing channel hot electron (CHE) injection from the drain region into the floating gate. In one embodiment, the programming process is enhanced by applying a bias to the channel region. By programming in this way, the need for separate control gate diffusions may be avoided, thus minimizing the size of the asymmetric NVM cell. Erasing the floating gate involves coupling the drain diffusion to ground and applying the positive voltage to the source, thereby causing Band-to-Band Tunnelling (BBT) generation of holes and their injection into the floating gate.

Supplementary Table 1. Operational principles of Y-flash memory

Mode [V]	Terminal name			
	Drain	Source Read	Source Injection	Sub
Read	Sweep	0	V_Drain/ High-Z	0
Program	V_PRG	High-Z	0	0
Erase	High-Z	High-Z	V_ERS	0

**Supplementary Figure 1.** Single asymmetric Y-flash cross-sectional schematic compared to and derived from regular NMOS transistor using standard CMOS technology.

Various drain region and floating gate configurations are used to optimize cell function and minimize cell size. In one embodiment, the first (channel) and second (extension) portions of the floating gate form two sides of a J- or U-shaped polysilicon structure that is connected by a third portion extending there between. In another embodiment, the floating gate is formed in a digitated manner such that two extension portions extend from opposite ends of the first (channel) portion into the drain region. Optional select transistors are included in each NVM cell to provide selective access to the drain region for the purpose of supporting E2 (EEPROM) operations, and as a safeguard against possible large array effects. In yet another embodiment, an NVM cell without select transistors is provided with a minimal cell size using a contact disposed inside the U-shaped floating gate for increasing of capacitive coupling of the drain to the floating gate.

The asymmetric NVM cells are arranged in arrays that facilitate cell operations while minimizing occupied chip space. In one embodiment, each NVM cell is connected between one bit-line and one word-line, and program/erase operations are performed by applying programming voltages to one word-line and one bit-line at a time. In another embodiment, alternative schemes are utilized in conjunction with NVM cells having select transistors that include two word-lines or two bit-lines per cell, which provides additional immunity to over-erase, which can be important for NVM cells cycled many times. In yet another embodiment, the source regions of adjacent pairs of NVM cells are connected to a common word-line in order to decrease the size of the select transistors and thus save area.

A readout transistor can be configured in parallel using a transistor, thus avoiding read-disturb problems by means of its relatively long channel length, while an injection transistor that facilitates high-voltage EEPROM-type program/erase operations can be configured by means of its relatively short channel length and a relatively high drain-to-gate capacitance. The injection transistor includes a drain region (terminal) and source region (terminal) separated by a channel region, and the readout transistor also includes a drain region and a source region separated by a channel region. The readout transistor and injection transistor are parallel-connected by way of both drain regions being connected to a cathode terminal, and by way of both source regions being connected to anode terminal. The readout transistor and injection transistor have different threshold voltages (V_{th}), with the V_{th} of the readout transistor being relatively low to facilitate low-voltage readout operations (*i.e.*, by way of applying a

readout voltage below 1V across terminals), and the V_{th} of the injection transistor being relatively high to facilitate performing high-voltage program/erase operations (*i.e.*, by applying program/erase voltages above 5V across terminals). For example, the readout transistor is configured with a relatively long channel length (*e.g.*, 0.6 μm (microns)) and a low V_{th} of 0.3V that allow it to perform low-voltage readout operations, and the injection transistor is configured with a relatively short channel length (*e.g.*, 0.25 μm) and a relatively high V_{th} (*e.g.*, 1.5V) that allow it to perform high-voltage program/erase operations.

The readout and injection transistors utilize a shared (integral, single-piece) floating gate structure that stores a charge transferred to/from the injection transistor during program/erase operations, whereby the stored charge controls current flow through the readout transistor during subsequent readout operations. In addition, the floating gate structure forms the only polycrystalline silicon structure of the memristor, which facilitates fabrication using low-cost CMOS single-poly fabrication flows. This single-gate feature thereby provides a significant cost advantage over conventional EPROM NVM approaches that employ double-poly CMOS processes.

A diode can be connected to the drain terminals of the parallel asymmetrical (readout and injection) transistors such that the diode functions both as a current limiter that prevents over-erase during the high-voltage erase operations, and also functions to prevent sneak (leakage) currents during low-voltage readout operations. It could be implemented as a vertical p-n junction diode, a Schottky diode, or a lateral p-n or p-i-n diode, depending on desired fabrication cost and performance considerations. It is clear that external diode can be connected to drain of the Y-Flash cell with interconnected sources of read-out and injection transistors, when used for memristor investigations and feasibility studies of neuromorphic circuits. In principle, there is no need for a select transistor in the proposed NVM cells because the cells are asymmetric. That is, each proposed NVM cell is read-out in one direction only, similar to cross-wise memories where the memory element is a switching resistor connected in series diode. By combining these features, the proposed device reliably overcomes the problems associated with implementing conventional two-terminal EEPROM-type devices in cross-point arrays by way of utilizing the readout transistor to perform low-voltage readout operations, by utilizing the injection transistor to facilitate reliable high-voltage program/erase operations, and by utilizing the diode to prevent both sneak currents during the low-voltage readout operations and over-erase conditions during the program/erase operations.

2. Y-flash subthreshold small-signal model

The Y-flash cell structure at the device level is developed from two MOS transistors (Supplementary Fig. 3), *i.e.*, a read transistor and an injection transistor and another coupling three-dimensional MOS-transistor-based capacitor which is connected between the common drain and floating gate terminal of the device [1]. This configuration is used in our small-signal memristive model of the Y-flash. The following section demonstrates how the subthreshold region of the Y-flash can be used for small-signal voltage to current conversion.

a. Ideal subthreshold small-signal Y-flash analysis

In the subthreshold region, when a biasing voltage of V_{DS} is applied, I_{DS} , *i.e.*, the large-signal current from the drain to the source terminal of the READ transistor is

$$I_{DS} = I_{read} e^{\frac{CR \cdot V_{DS}}{mV_T}} e^{-\frac{V_{th}}{mV_T}}, \quad (2.1)$$

where CR stands for “coupling ratio.” The current through the injection transistor, in the subthreshold region, is negligible and can be ignored. The large signal voltage, which leads to a large-signal current as per Eq. (2.1), acts as a biasing point for small-signal AC voltages.

Now, if the bias point V_{DS} at the drain terminal is perturbed with a small signal v_{ds} , we observe that the current through the read transistor also gets disturbed by a small amount.

$$i_{DS} = I_{DS} + i_{ds} = I_{read} e^{\frac{CR \cdot (V_{DS} + v_{ds})}{mV_T}} e^{-\frac{V_{th}}{mV_T}}. \quad (2.2)$$

Here, the value of internal state variable is determined by fine-tuning the threshold voltage V_{th} using 4.5V/5V program and 8V erase pulses.

$$I_{DS} + i_{ds} = I_{read} e^{\frac{CR \cdot (V_{DS})}{mV_T}} e^{-\frac{V_{th}}{mV_T}} e^{\frac{CR v_{ds}}{mV_T}}. \quad (2.3a)$$

Using Taylor series expansion [2] across a fixed voltage V_{DS} ,

$$I_{DS} + i_{ds} = I_{read} e^{\frac{CR \cdot (V_{DS})}{mV_T}} e^{-\frac{V_{th}}{mV_T}} \left(1 + \frac{CR v_{ds}}{mV_T} + \left(\frac{CR v_{ds}}{mV_T} \right)^2 + \left(\frac{CR v_{ds}}{mV_T} \right)^3 + \dots \right). \quad (2.3b)$$

For small AC values of the drain-to-source voltage, that is, when $CR v_{ds} \ll mV_T$, we can ignore the second- and higher-order terms in the Taylor-series expansion and then the total subthreshold current can be given as the sum of small-signal current and the large-signal current, $I_{DS} + i_{ds}$. From large-signal fitting of the Y-flash cell model, $\frac{CR}{mV_T} = 144mV$, giving a maximum magnitude of $v_{ds} = 30mV$ for $\frac{CR v_{ds}}{mV_T}$ as 0.208 (0.347 for $v_{ds} = 50mV$), which can be safely ignored for the second- and higher-order terms. Linearizing the small-signal I-V characteristics:

$$I_{DS} + i_{ds} = I_{read} e^{\frac{CR \cdot (V_{DS})}{mV_T}} e^{-\frac{V_{th}}{mV_T}} \left(1 + \frac{CR v_{ds}}{mV_T} \right), \quad (2.4)$$

$$i_{ds} = \frac{I_{DS} CR}{mV_T} v_{ds}.$$

$$g_m = \frac{\delta I_{DS}}{\delta V_{DS}}, \quad (2.5)$$

$$g_m = \frac{I_{DS} CR}{mV_T}. \quad (2.6)$$

Therefore, the small signal current equation can be written as:

$$i_{ds} = g_m v_{ds}, \quad (2.7)$$

where

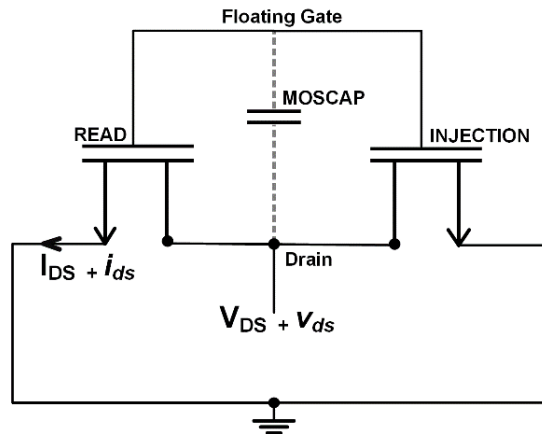
$$R_{mem} = \frac{1}{g_m}. \quad (2.8)$$



Supplementary Figure 2. Ideal Y-flash small-signal subthreshold model.

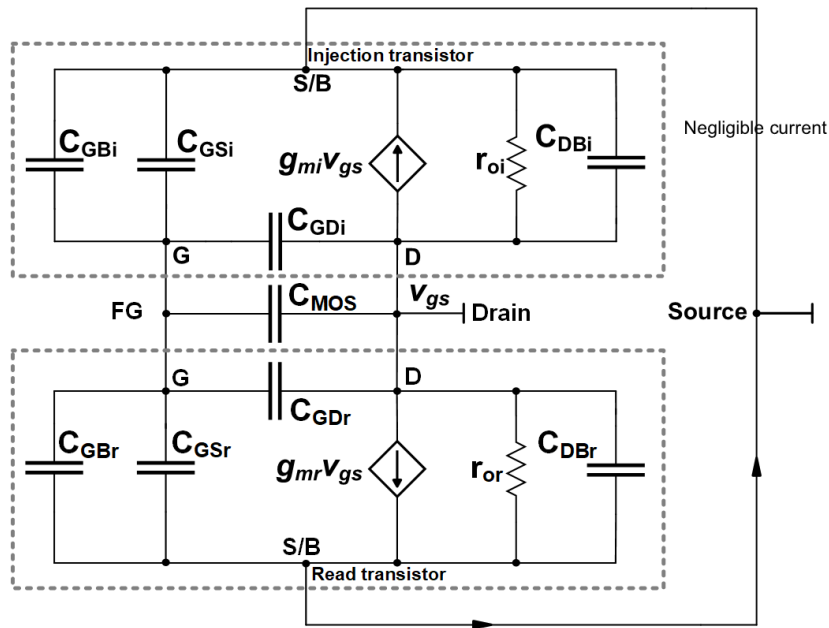
b. Y-flash small-signal equivalent circuit

In this approach, we calculate the unity-gain frequency of the Y-flash device, which allows the maximum switching speed to be estimated. The maximum frequency of the small-signal input at the drain terminal of Y-flash is limited by the coupling MOS capacitor and the parasitic capacitances of the MOS transistors.

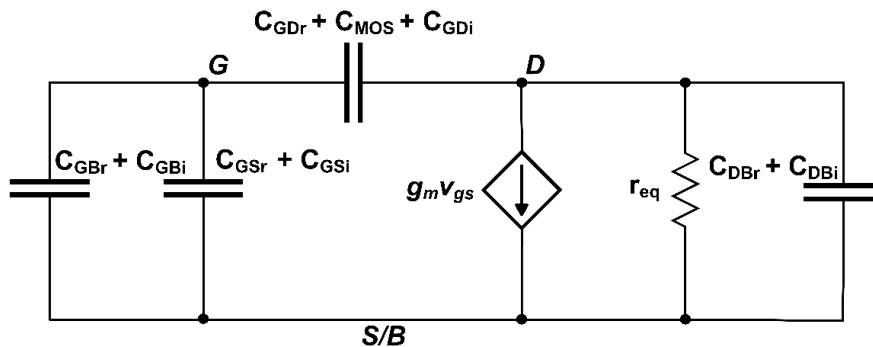


Supplementary Figure 3. A three-MOS schematic of the Y-flash cell.

To illustrate this in detail, we construct a subthreshold small-signal model of the device. Biasing the Y-flash with a voltage V_{DS} on the common drain, a small signal v_{ds} (dynamic range = 50mV) is applied. Illustrating the small-signal schematic of Supplementary Fig. 3 in subthreshold, we develop the circuits in Supplementary Fig. 4-5.

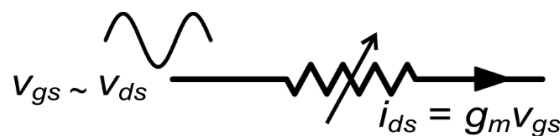


Supplementary Figure 4. Subthreshold small-signal schematic of the Y-flash device developed from a 2-NMOS topology. Since the two NMOS have the same terminal voltages, they can be folded to develop a single schematic in parallel. Note: C_{MOS} (or C_{GD}) $\gg C_{GS}$ (in subthreshold).



Supplementary Figure 5. Simplifying the circuit by folding it across the drain and source terminals

The v_{gs} , which is the drain-source voltage is, $v_{gs} = \frac{C_{GD}}{C_{GS}+C_{GD}} v_{ds}$. As specified, the total C_{GS} , which includes the parasitic capacitances, is quite small as compared to the C_{GD} , which is a combination of the parasitic and the additional MOS capacitance. Thus, the gate is floating and shorted to the drain (as a passive device). Therefore, the small-signal V-I conversion can be simplified and illustrated as in Supplementary Fig. 6, ignoring the bulk parasitic capacitances. Thus, the ideal current source is replaced with a resistor.



Supplementary Figure 6. An approximate small-signal schematic of the Y-flash (passive) device. As mentioned earlier, since, $C_{GS} \ll C_{GD}$, $v_{gs} = \frac{C_{GD}}{C_{GS}+C_{GD}} v_{ds} \sim v_{ds}$. And $g_m = \frac{1}{R_{mem}} = \frac{I_{DS}CR}{mV_T}$.

This circuit is essentially different than the diode connected mode of a regular floating-gate linearized in sub-threshold, which behaves as an active device (resistor instead of voltage-controlled current source). In addition, a regular three-terminal floating-gate in a diode connection cannot perform bipolar

resistive switching (gate voltage is very high while drain voltage is much lower, whereas the source is grounded, and the substrate is at high voltage during erasing).

We extracted the values of parasitic capacitances from the TowerJazz's TS18PM technology manual and performed the spectral analysis on the circuit depicted in Supplementary Fig. 5.

Supplementary Table 2. TowerJazz's 180nm process technology specifications

	n-ch	Units
T_{ox}	100-120	Angstrom
V_{t0}	0.7-1	V
μ	~600 at 10 ⁵ V/cm SiO ₂ field	cm ² /Vsec

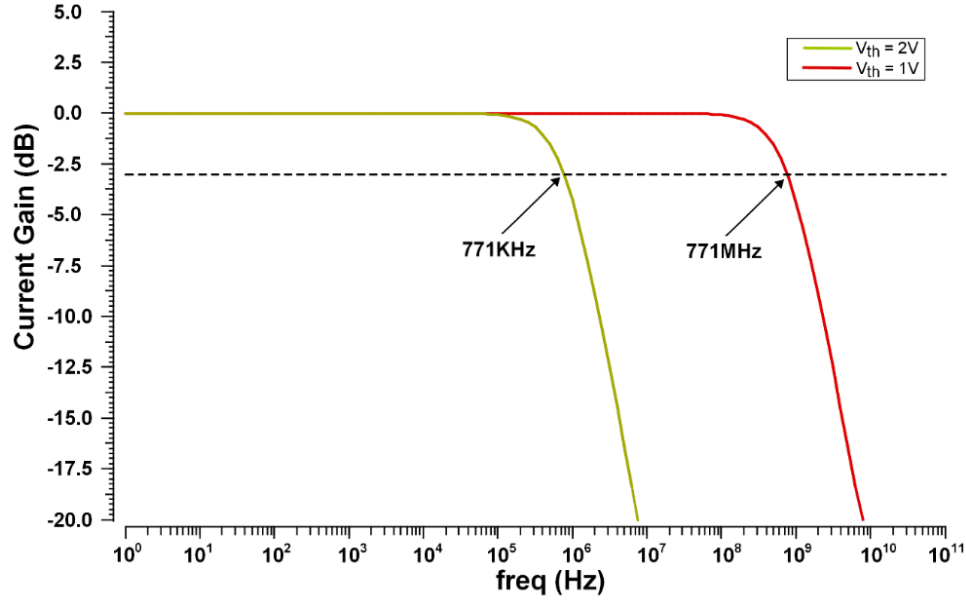
Supplementary Table 3. Dominant parasitic capacitances in the subthreshold region of Y-flash devices

Capacitance type	Injection transistor capacitance (aF)	Read transistor capacitance (aF)
C_{GB}	80.14	164.15
C_{GS}	48.51	49.29
C_{GD}	254.19	746
C_{DB}	320.079	320.737
Gate Capacitance	~2 – 3 fF/μm ²	

Supplementary Table 4. Y-flash transistor specifications

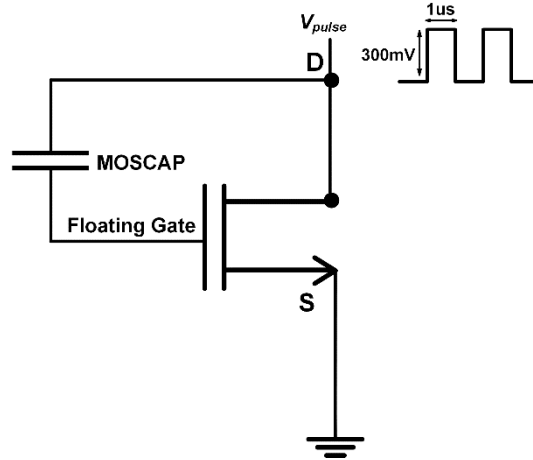
Dimensions	Length (μm)	Width (μm)
Read	0.6	0.22
Injection	0.3	0.22
Coupling transistor	1.75	0.18

This gives us a maximum operating range frequency of [770KHz: 770MHz] for a dynamic range of V_{th} from [1V: 2V]. At higher frequencies the Y-flash small-signal model behaves in different manner than in Supplementary Fig. 6 (passive device), where it is affected by the capacitors and limited by the cutoff frequency $f_{cutoff} = g_m/2\pi C_{gs}$, behaving as a capacitor rather than resistor (conductor).



Supplementary Figure 7. 3-dB frequency of current gain, which demonstrates the maximum frequency up to which the Y-flash device works as a linear resistor in response to small-signal changes. Specifications from the TS18PM (Power management) technology are used to define the parasitic capacitor values.

To understand the effect of coupling capacitance between the drain and gate terminals of the Y-flash in response to large-signal voltages, we use an NMOS based circuit as shown in Supplementary Fig. 8.



Supplementary Figure 8. Emulating the Y-flash characteristics using circuit elements, to understand the role of the coupling capacitor. Since we are using TS18SL (primary technology) for this simulation, $V_{th} = 450\text{mV}$. For subthreshold operation, we provide a voltage of 300mV at the drain terminal.

Functioning as a capacitive divider, $V_{GS} = \frac{C_{DG}}{C_{DG} + C_{GS}} V_{DS}$ in response to large-signals. Considering a high value of MOSCAP $\sim 2fF$, where the intrinsic parasitic capacitance C_{GS} is negligible, we obtain that the $V_{GS} \sim V_{DS}$, similar to diode mode connection.

3. Modelling the Y-flash dynamics

This section models the dynamics while programming, erasing and reading the internal state of the device.

a. Read mode

The internal state of the Y-flash device can be measured, in similar fashion to a resistor, in the subthreshold region. This is done by providing 2V across the drain and externally shorted source terminals and then measuring the current across the device given by Eq. (3.1).

$$I_{DS} = I_{read} e^{CR \frac{V_{DS}}{mV_T} e^{-\frac{V_{th}}{mV_T}}}. \quad (3.1)$$

b. Programming mode

The Y-flash characteristics, depending on the programming voltage, can be represented using two models: the long channel model for 4.5V programming voltage and the short-channel model for 5V as depicted in Supplementary Table 5. However, since the short channel model is simpler, requires fewer parameters, and is physically reasonable, we use it to model the device for both voltages. The equation for a short channel model is represented as:

$$\frac{dV_{th,RESET}}{dt} = \frac{I_{inj}}{C_{fr}} = \frac{K'}{C_{fr}} (CR \cdot V_{DS} - V_{th}) \cdot P(E_v). \quad (3.2)$$

Since K' and CR are constants and the variation in $P(E_v)$ is negligible w.r.t. the change in V_{DS} , for modelling purposes, we take their multiplication as a constant. Along with that, for a specific programming voltage, i.e., 4.5V or 5V, the parameter $CR \cdot V_{DS}$ remains constant. Therefore, the equation reduces to:

$$\frac{dV_{th,RESET}}{dt} = b(a - V_{th}), \quad (3.3)$$

where

$$a = CR \cdot V_{DS}, \quad b = \frac{K'}{C_{fr}} \cdot P(E_v).$$

To calculate the equation specific parameters, a and b , we use the MATLAB curve fitting tool to fit the measured Y-flash data to the fitted equations. The obtained expression is in the form of

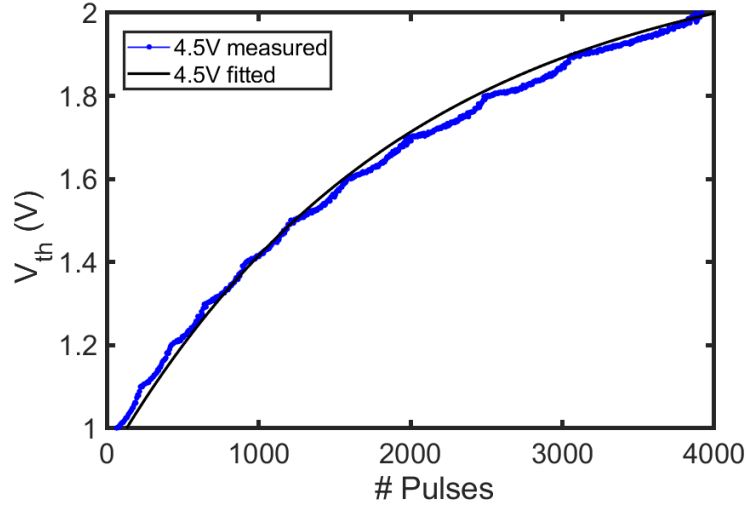
$$V_{th} = a - e^{c-bt}, \quad (3.4)$$

where c is the integration constant and t is the time period of each pulse input.

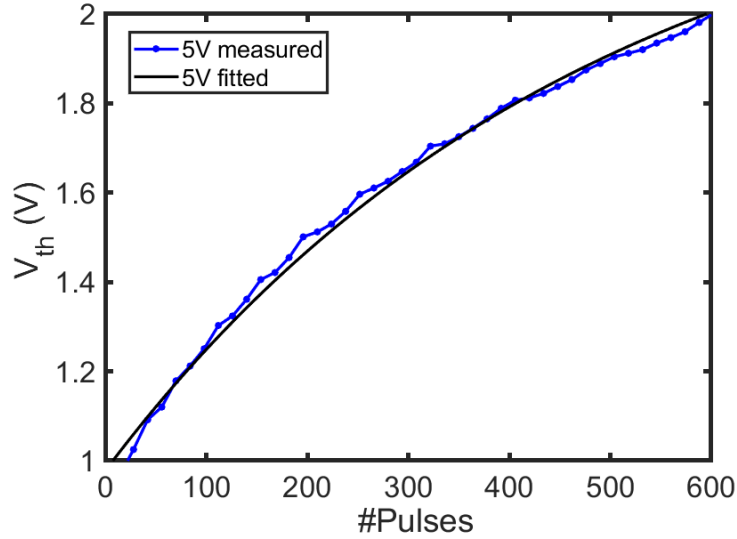
Since both programming voltages are modelled using the short channel model, the fitting parameters can be obtained for other voltage ranges as well, with measured programming data available.

Supplementary Table 5. Fitting parameters for programming voltages 4.5V and 5V

Programming voltage	$CR \cdot V_{DS}$	$\frac{K'}{C_{fr}} \cdot P(E_v)$	R-squared
4.5 V	2.16	$5.1 * 10^{-5}$	0.9941
5 V	2.4	$2.1 * 10^{-4}$	0.9952



Supplementary Figure 9. Fitting V_{th} vs. number of pulses data for a programming voltage of 4.5V.



Supplementary Figure 10. Fitting V_{th} vs. number of pulses data for a programming voltage of 5V.

c. Erase mode

The simulated annealing method is used to determine optimized parameters for the data extracted from the measurements to fit the Eq. (3.5) for erasing. Supplementary Fig. 11 highlights the comparison between the measured data and the fitted equation for an erasing voltage $V_S = 8V$.

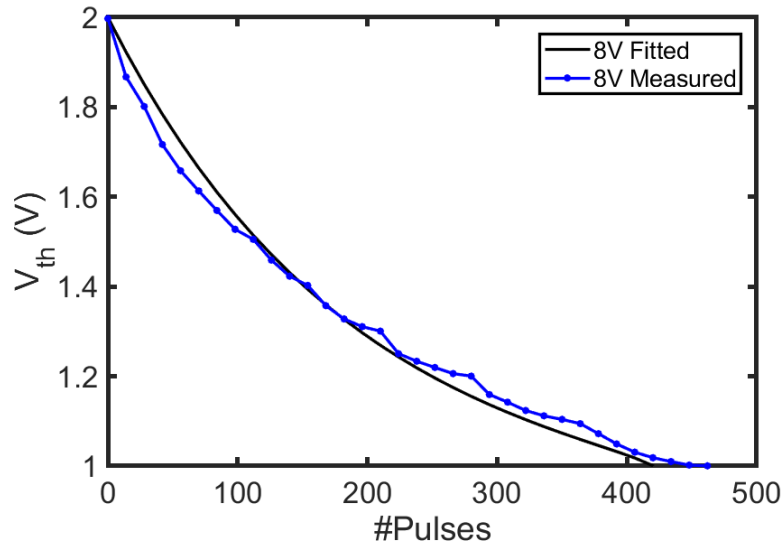
$$\frac{dV_{th,SET}}{dt} = \frac{I_{GIDL}}{C_{dep}} = A \frac{CR \cdot V_S + V_{ox} - V_{th}}{C_{dep} t_{ox}} e^{-\frac{Bt_{ox}}{CR \cdot V_{DS} + V_{ox} - V_{th}}}. \quad (3.5)$$

By considering $CR, V_{DS} + V_{ox}$ and $\frac{A}{C_{dep} t_{ox}}$ as constants, as in the programming mode, this equation can be reduced to:

$$\frac{dV_{th,SET}}{dt} = a \cdot (b - V_{th}) e^{-\frac{c}{b - V_{th}}}, \quad (3.6)$$

where

$$a = \frac{A}{C_{dep} t_{ox}} = 4.643 \cdot 10^{-4}, \quad b = CR \cdot V_S + V_{ox} = 0.9531, \quad c = Bt_{ox} = 0.07.$$



Supplementary Figure 11. Fitting V_{th} vs. number of pulses for an erasing voltage of 8V.

Supplementary Table 6. Parameters of the "lucky electron" CHE injection model for RESET, and of the GIDL current model in BTBT for SET.

Parameter	Model	
	Programming	
	Short Channel	Long Channel
$P(E_v)$	$e^{-\frac{E_v}{kT_e}}$	$e^{-\frac{E_v}{q\lambda E_m}}$
K'	$\frac{\mu C_{ox} W}{2}$	$\frac{\mu C_{ox} W}{2L^2 E_{sat}}$
n	1	2
A (fitted)	Erasing	
	$9 \cdot 10^{-15} \text{ [m}/\Omega]$	
B (constant)	21.3 [MV/cm]	

Supplementary Table 7. Parameter definitions and values for the Y-flash Verilog A model.

Parameter	Description	Default value
CRprog	Coupling ratio for the programming mode equations	0.48
Vth_init	User-defined value of the initial threshold voltage	1V
K_Cfr_45, K_Cfr_5	Constants ($\frac{K'}{C_{fr}}$) as defined in the programming mode equation for 4.5V and 5V	$5.1 \cdot 10^{-5}$, $2.1 \cdot 10^{-4}$
mVT	Technology constant m times V_T (0.026V)	144.76 mV
CR	Constant for the subthreshold region equation (pre-defined)	1
Iread	Read current when $V_{th} = 1V$	$1 \cdot 10^{-9}$
resolution	Regulates minimum step size	1

4. SPICE models of the Y-flash

We designed a Verilog-A based SPICE model of the Y-flash cell which models the deterministic characteristics of the device, as presented in the previous section.

a. Verilog-A code of the voltage-controlled Y-flash behavioural model

```
`include "constants.vams"
`include "disciplines.vams"
module yflash (src_read, src_inj, sub, dra);
inout src_read, src_inj, dra, sub;
electrical src_inj, dra, sub;
electrical src_read;
ground gnd;
//PARAMETERS
parameter real vth_init = 1;
parameter real CRprog = 0.48;
parameter real K_Cfr_45 = 5.1e-5;
parameter real K_Cfr_5 = 2.1e-4;
parameter real mvt = 0.144765;
parameter real CR = 1;
parameter real iread = 1e-9;
parameter real resolution = 1;

// LOCAL VARIABLES
real vds; //drain-source voltage
real vth; //state variable vth
real dvthdt;
real first_iter = 0; //initialised to zero
real i; //output current

////////// MAIN //////////
analog begin
    //initializing vth
    if(first_iter==0) begin
        vth = vth_init;
        first_iter = 1;
    end
    //device behaviour as a memristor
    //Includes programming(for 4.5V, 5V), erase(for 8V) and read for 2V
    //Read mode -- subthreshold mode conduction
    if(V(dra,src_read)<= 2.0)begin
        i = iread*exp((CR*(V(dra,src_read)))/(mvt))*exp(-vth/mvt);
        dvthdt = 0;
        I(dra) <+ i;
        I(src_read) <+ -i;
    end
    //Programming mode 4.5V short-channel
    if (V(dra,src_inj)== 4.5) begin
        dvthdt = K_Cfr_45*(CRprog*V(dra,src_inj) - vth);
        I(dra, src_inj) <+ 0;
    end
    //Programming mode 5V short-channel
    if (V(dra,src_inj) == 5.0) begin
        dvthdt = K_Cfr_5*(CRprog*V(dra,src_inj) - vth);
        I(dra, src_inj) <+ 0;
    end
    //Erase mode 8V
    if (V(src_inj,gnd) == 8) begin
        dvthdt = 4.6428e-4*(0.9531-vth)*exp((-0.07)/(0.9531-vth));
    end

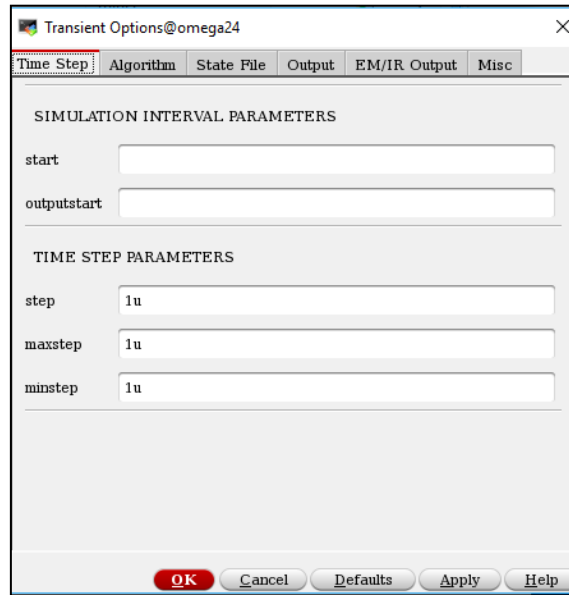
    //defining window function b/w vth = [1,2]V
    if(vth < 1.0 || vth > 2.0)begin
        dvthdt = 0;
    end
    vth = vth + resolution*dvthdt;
    //Updating output nodes
    I(sub) <+ 0;

    dvthdt = 0;
end
endmodule
```

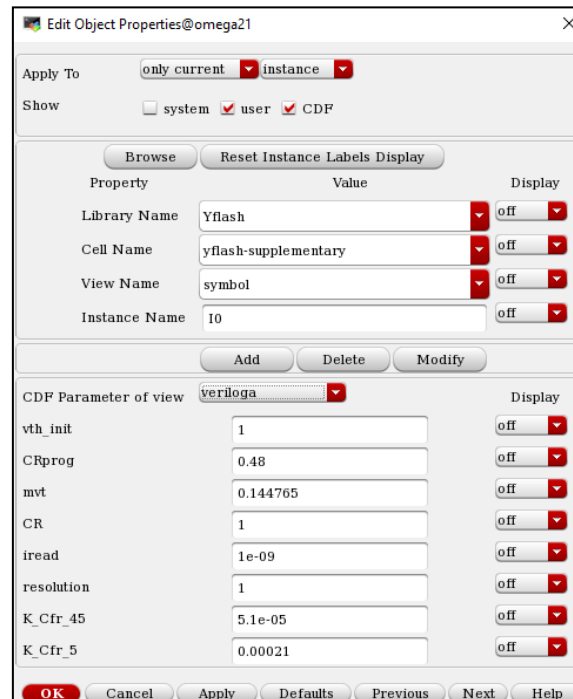
i. Simulation

The user may note a few important points while performing simulations using the model in the Cadence Virtuoso environment:

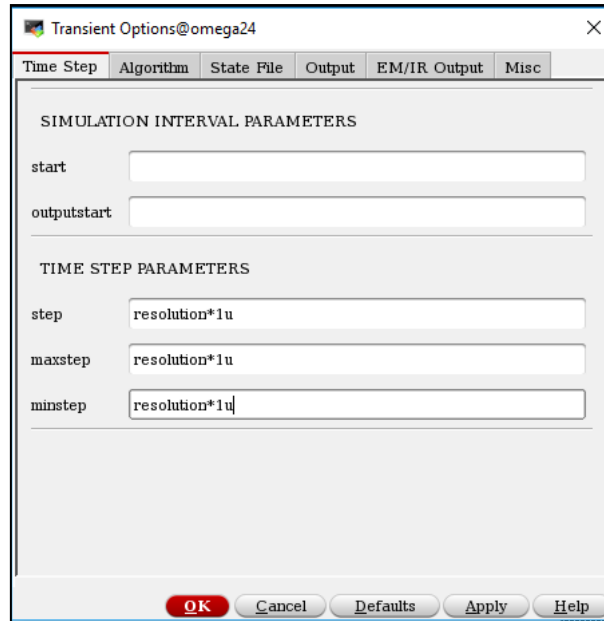
- The state variable V_{th} is bound between $[1V, 2V]$.
- The parameter **resolution** is given a default value of 1, which translates to a minimum step value of $1\mu s$ (resolution* $1\mu s$). To make the time-step of each iteration smaller, say $1ns$, the user can change the resolution to 0.001 and thus can vary the minimum time step as $1ns$ (resolution* $1\mu = 1n$).



Supplementary Figure 12. Providing a step size of $1\mu s$ for **resolution** = 1.



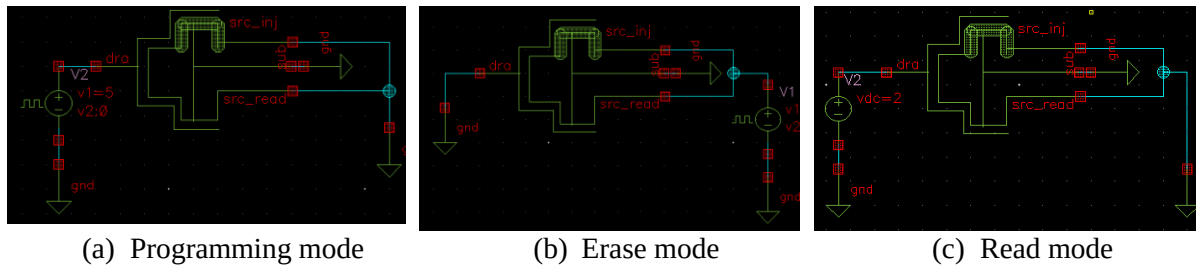
Supplementary Figure 13. Varying the resolution from ADEL window.



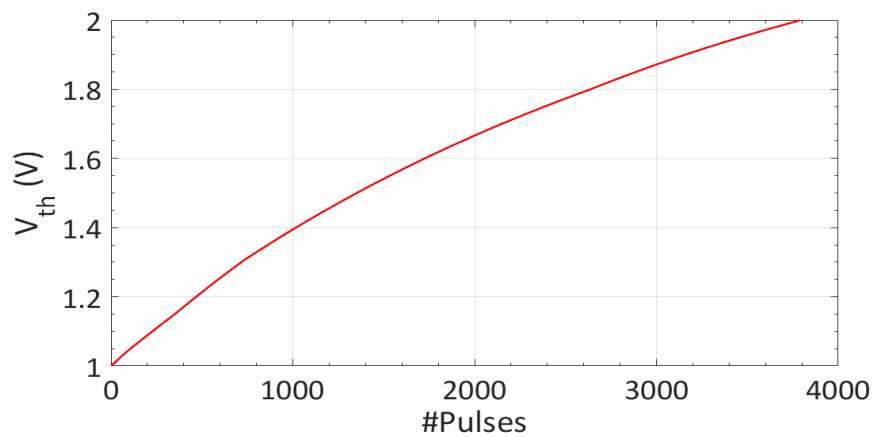
Supplementary Figure 14. Varying the time-step values accordingly.

ii. Schematic connections for different nodes

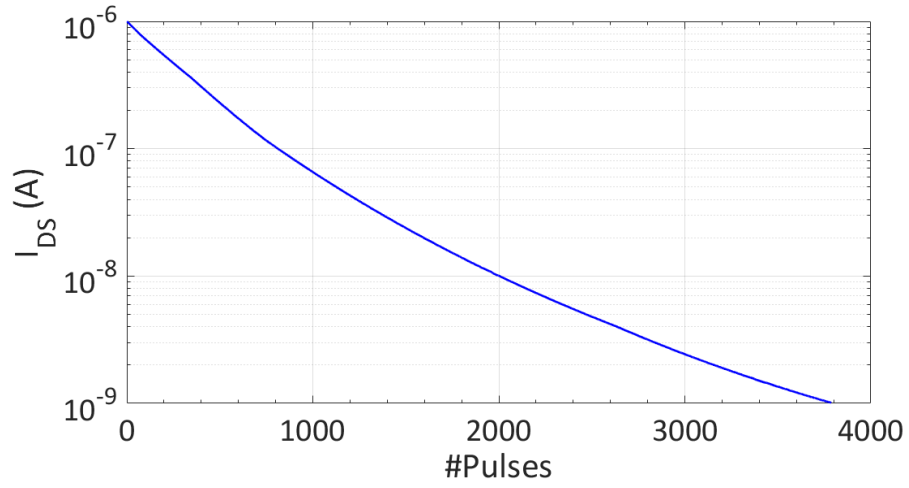
For a single **two-terminal Y-flash based memristor cell**, the user must make connections as follows:



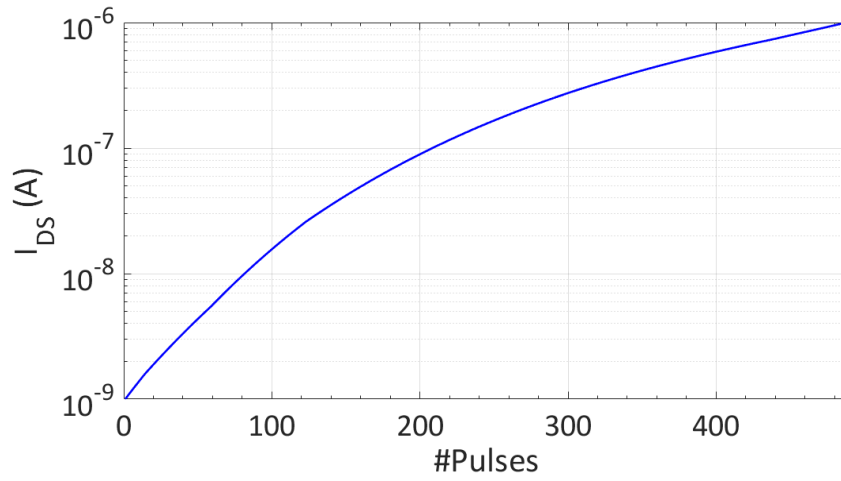
iii. Simulation results



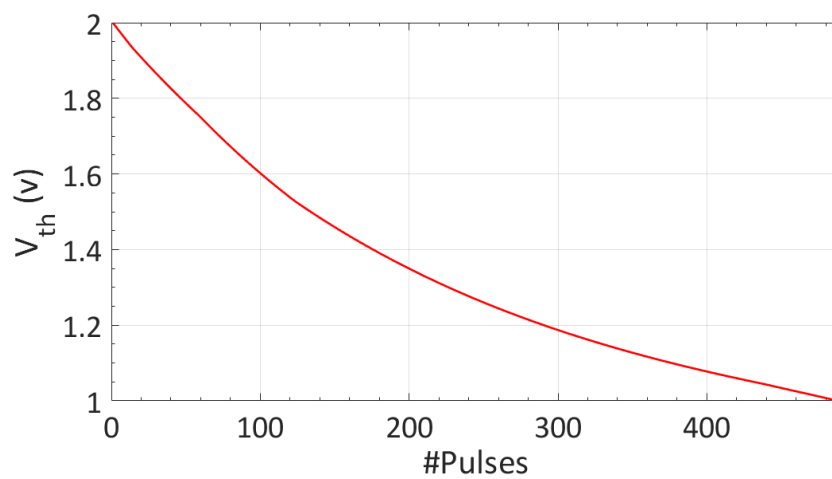
Supplementary Figure 15. Programming mode 5V, V_{th} vs. #Pulses.



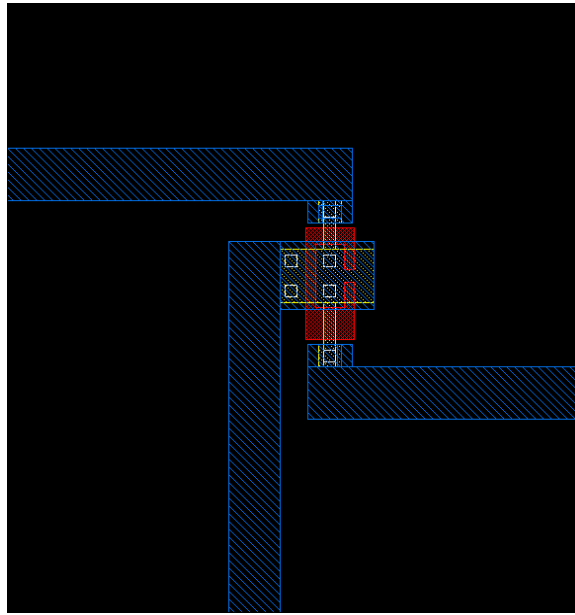
Supplementary Figure 16. Programming mode 5V, I_{DS} vs. #Pulses.



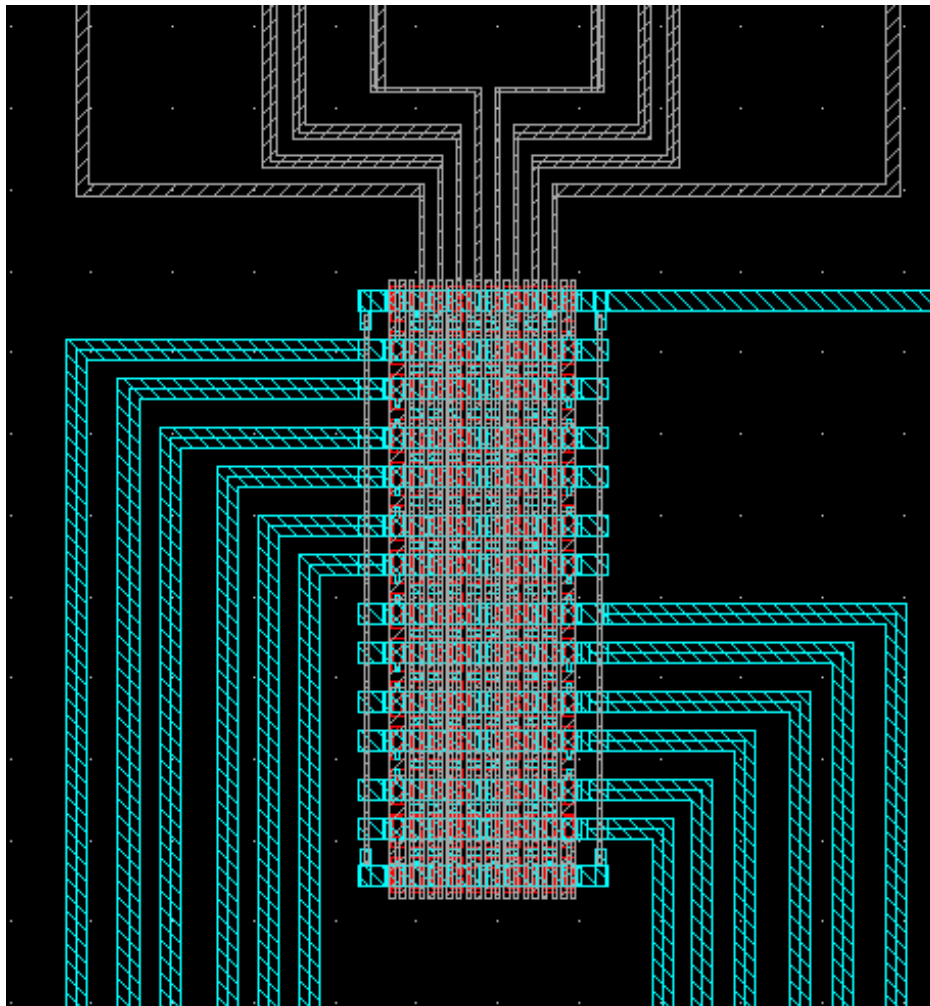
Supplementary Figure 17. Erase mode 8V, I_{DS} vs. #Pulses.



Supplementary Figure 18. Erase mode 8V, V_{th} vs. #Pulses.

b. Layouts:

Supplementary Figure 19. Layout of a single Y-flash memristive cell.

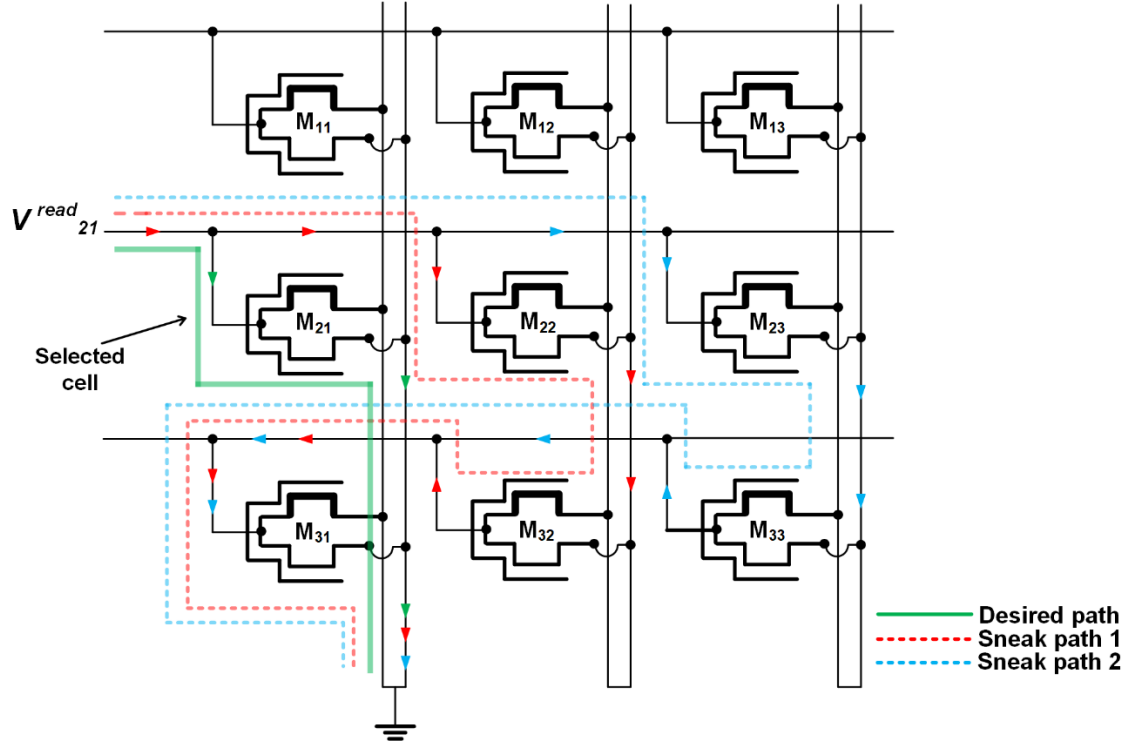


Supplementary Figure 20. Layout of a 12x8 Y-flash based crossbar array.

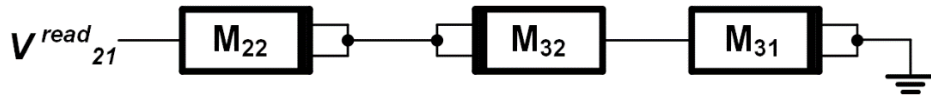
5. Reliability characterization and analysis

a. Non-ideal effects of Y-flash array:

i. Sneak path currents while reading

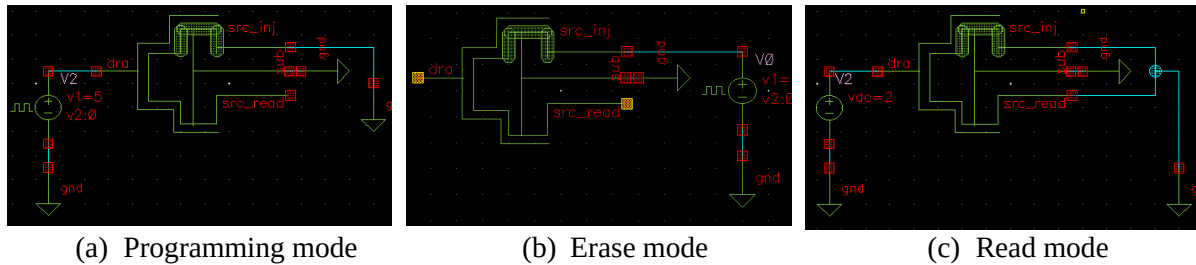


Supplementary Figure 21. Some sneak path currents depicted while reading the state of a selected device in a 3x3 array.



Supplementary Figure 22. Circuit representation of the smallest sneak path (sneak path 1) while reading.

First, note that the operational conditions of programming and erasing the Y-flash device inside an array are modified according to Supplementary Table 1. The user must make connections as per the following schemes.



Sneak path currents while reading the state of a DUT are not a major source of drift from the desired sensed current. This is mainly because the Y-flash devices connected in series do not have the same polarity and show asymmetry. As a result, device M_{32} in Supplementary Fig. 21–22, because of the Y-flash's asymmetric I-V characteristics, restricts the flow of sneak current. Along with that, M_{31} , because of multiple sneak currents, may have a higher voltage across its terminals. As we scale up the array,

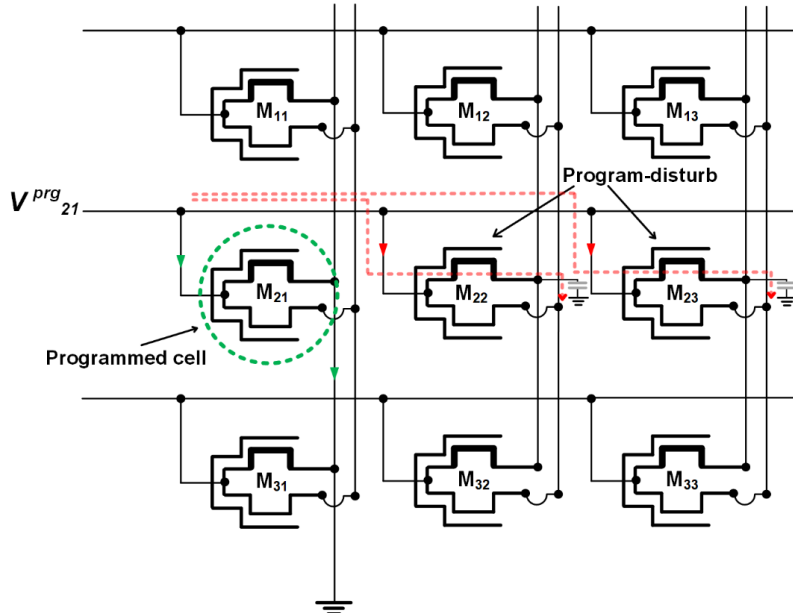
this effect may become more pronounced and program the device. However, this is checked by the exponential IV characteristics (Eq. 2.1) of the Y-flash device.

For example, in a 3×3 array, M_{32} allows the current from sneak path no. 1 and sneak path no. 2 to flow via its terminals. Because of this, the voltage across it must increase. To evaluate the maximum effect of sneak path current, all the cells except M_{21} are programmed to a minimum resistance R_{on} . We isolate sneak path no. 1 and sneak path no. 2 and then use the superposition principle to add the effects on M_{32} . We calculate that because of sneak path 1, the current via M_{32} is 26.1 pA and the voltage drop across it is 472 mV . Similarly, since sneak paths 1 and 2 are similar, the current due to sneak path 2 is again 26.1 pA and the voltage drop is 472 mV . Using superposition, as the current doubles (we are considering only these two paths; in an actual array the current is less than double because of additional negating sneak paths), the voltage drop due to the exponential I - V characteristics increases in a logarithmic fashion to $\sim 550 \text{ mV}$. This shows that the Y-flash based array can adjust to scaling without the additional issues caused by being programmed by the voltage increase that occurs as a result of sneak currents.

Furthermore, simulations using the device model were performed to analyse the effects of sneak paths in a Y-flash based 12×8 crossbar array. We program the device at position (1,1) to be at a maximum resistance with a threshold voltage value of 2 V and rest of the devices to be at minimum resistance, i.e., a threshold voltage of 1 V . The sensed value of the current for DUT at (1,1) is 1.7 nA , where the desired value should have been 1 nA . Thus, the maximum current offset, because of sneak paths, is 0.7 nA , which is acceptable.

Please note that a diode is not connected to the drain of Y-Flash memristive cells in the reported experiments. As reported above, the diodes are not obligatory for excluding the sneak path effects when operating Y-Flash memristive cells with the discussed characteristics. Nevertheless, special care has to be taken in order not to over-erase the memristive cells (erase by band to band tunnelling holes might lead to the avalanche multiplication hole generation mechanism), since the channel current in the erase mode is not limited.

ii. Program disturb

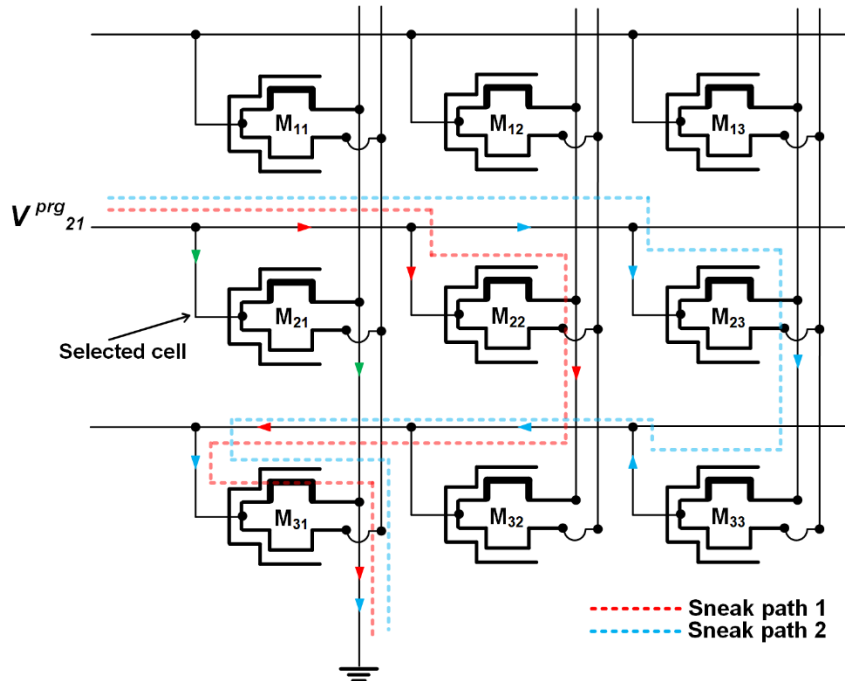


Supplementary Figure 23. The state of the adjacent devices in the same word-line may be disturbed because of substrate leakage current via the injection transistor.

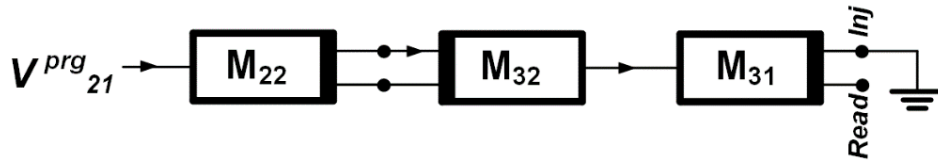
While programming, since the *source injection transistor* terminal of the unselected cells is floating, no current should be flowing through these cells. This is the ideal scenario. But because of the current

leakage and capacitance from the *source line* of the injection transistor to the *substrate* terminal, the unselected devices in the same row get programmed proportionally.

iii. Sneak paths in programming mode (Half-select)



Supplementary Figure 24. Sneak paths in the 3x3 Y-flash based crossbar in programming mode.



Supplementary Figure 25. Circuit representation of the smallest possible sneak path (sneak path 1) while programming a 3x3 array.

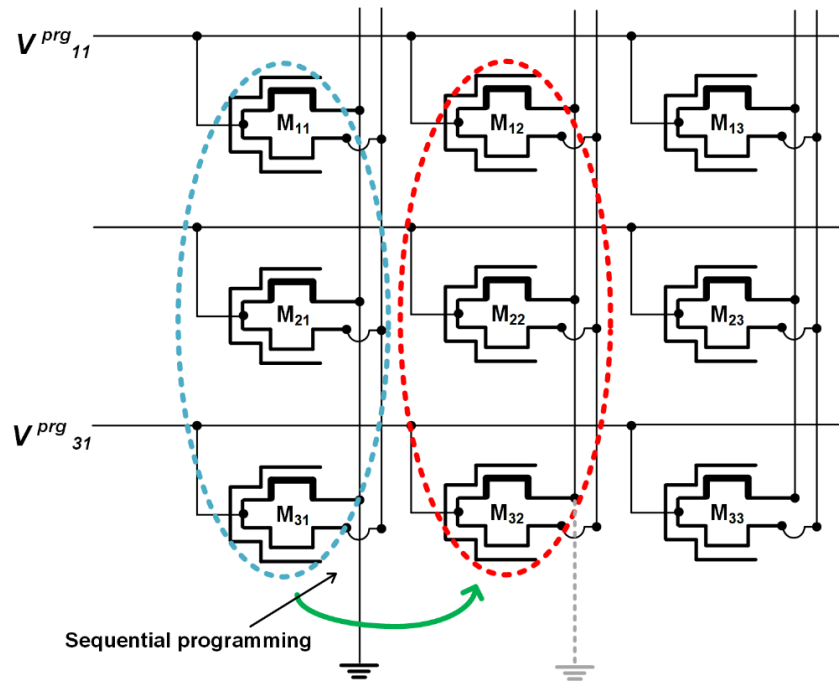
Similar to sneak currents while reading the state of a selected cell, there are sneak current paths while programming. For a single selected cell in a crossbar, this non-ideal behaviour is not an issue because of the asymmetric nature of the Y-flash. However, when multiple devices are selected, because of superposition, the voltage across M_{32} might lead to partial erasing the cell. When a diode is connected in series with the Y-Flash cell the effect is much smaller. Further circuit simulations need to be performed to analyse this problem. Until now, we have not encountered the influence of this effect in the fabricated Y-flash based arrays.

iv. Mitigating the effects of non-ideal effects in Y-flash based crossbar arrays

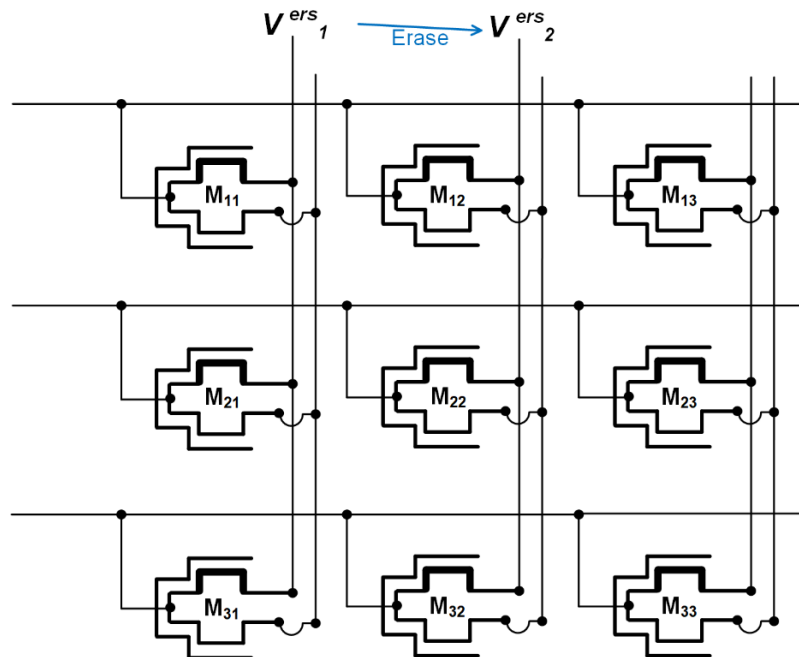
Many non-ideal phenomena such as the sneak-path problem and parasitic erase effects are non-significant issues and are avoidable in the Y-flash based crossbar arrays.

1. Sneak current problem – Sneak effects are automatically suppressed in the read mode due to exponential I-V characteristics and the asymmetric structure of the device itself.
2. Parasitic erase effects can be mitigated by using diodes in series of interconnected drains of readout and injection transistors. In Y-Flash memory applications these effects are negligible, but in case of memristors with 64 resistance levels, we take special care to exclude them.

Programming and Erasing schemes



Supplementary Figure 26. Sequential programming of columns in an array, where we program the required devices in the first column, followed by second column, and so forth.



Supplementary Figure 27. Erasing a column of Y-flash devices in one go. Note that the *drain* and *source read* terminals are floating in the schematic. We erase a complete column in an array in one go, at the beginning of implementing the ML task. This is mainly because erasing while programming leads to many unavoidable effects such as over-erase and erasing unselected cells, which are not encountered while working with a single Y-flash cell.

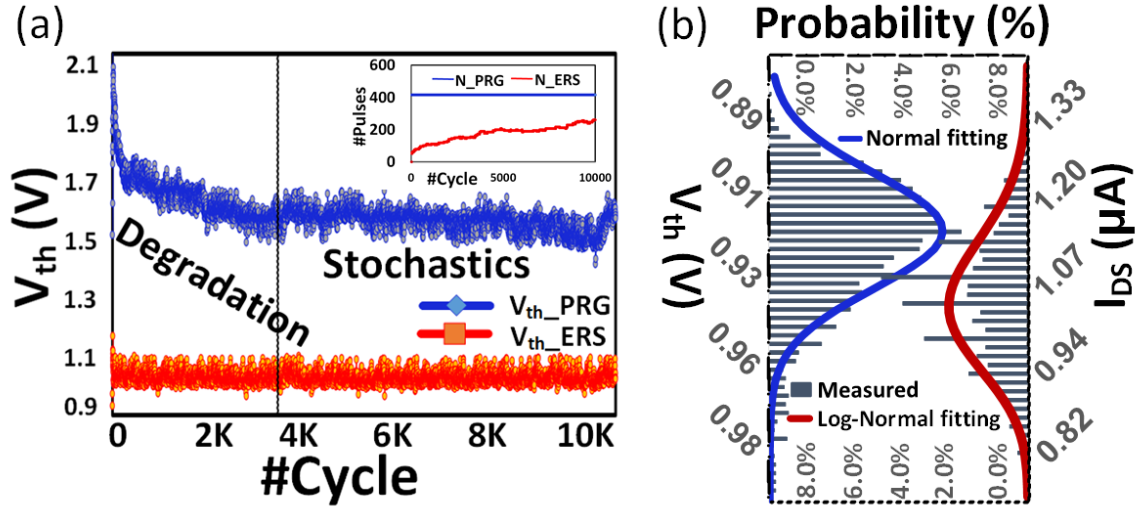
b. Stochastics, endurance, analogue data retention, and read disturb

Rather than enforcing deterministic dynamics, several neuromorphic applications take advantage of the non-deterministic memory for native stochastic computing, where the required randomness is intrinsic to the device and fundamentally changes its operating principle. Because of the stochastic nature of the injection process, the expected statistical spread of Y-Flash in $\Delta V_{th} = \frac{q\Delta n}{C_{fr}}$ (q is the electronic charge, and n is the number of injected electrons to obtain V_{th}):

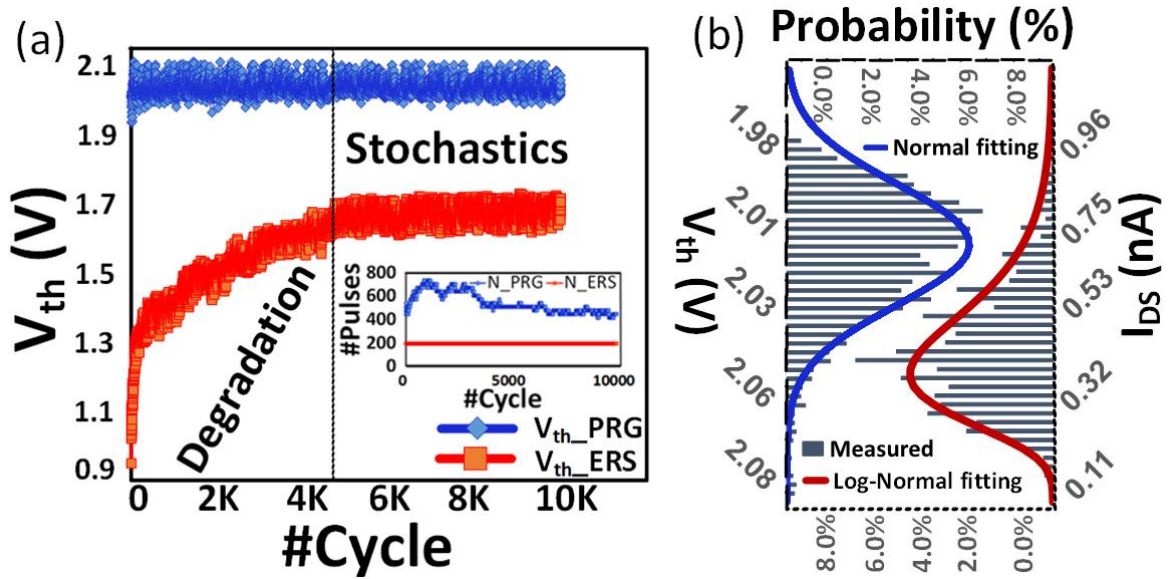
$$\sigma_{\Delta V_{th}} = \frac{q\sqrt{\Delta n}}{C_{fr}} = \sqrt{\frac{q}{C_{fr}} \Delta V_{th}}. \quad (5.1)$$

Assume that n is ruled by a Poisson process. Therefore, the V_{th} statistical spread in Y-Flash after programming/erasing is distributed by Poisson or normal distribution, as shown in Supplementary Fig. 28-29. Remarkably, the resistive state statistical spread after programming/erasing is log-normally distributed. This finding is consistent with the exponential I-V relationship in subthreshold mode. Oxide degradation in time affects the lifetime and performance of Y-Flash, as shown in Supplementary Fig. 28-29.

Furthermore, analogue data retention is measured for ten years with less than one order of magnitude current loss in the high-resistance state, as shown in Supplementary Figure 30-31. This is thanks to the read transistor, which increases the analogue data retention and the device immunity to charge traps and mitigates accumulative read disturbs, and to the Y-flash simple geometry and thick GOX (110Å). While only two initial resistive levels were tested in our measurements, the results imply analogue data retention. The worst case of data retention is in low currents (HRS), while the best case of data retention is in the highest possible currents (LRS); data retention is dominated by erasing (decreasing V_{th} and R , or increasing currents) over time. As shown in Supplementary Fig. 30, a resistance/current drop of one order of magnitude (from 1nA to 10nA) is seen after one year, but then remains almost stable for up to 10 years. Thus, we assume that initial intermediate analogue levels will remain stable, with a worst-case drop of up to 10nA in their resistance/current. Therefore, in our experiments, we limited our resistive/current dynamic range to two orders of magnitude, with the low current level defined (HRS) as 10nA instead of 1nA. This 10nA level determines the noise margin of the analogue operation of the device and the 65 discrete resistive levels, accordingly. Thus, the reported ten-year data retention is analogue and is a distinguishing feature of Y-flash single poly technology.



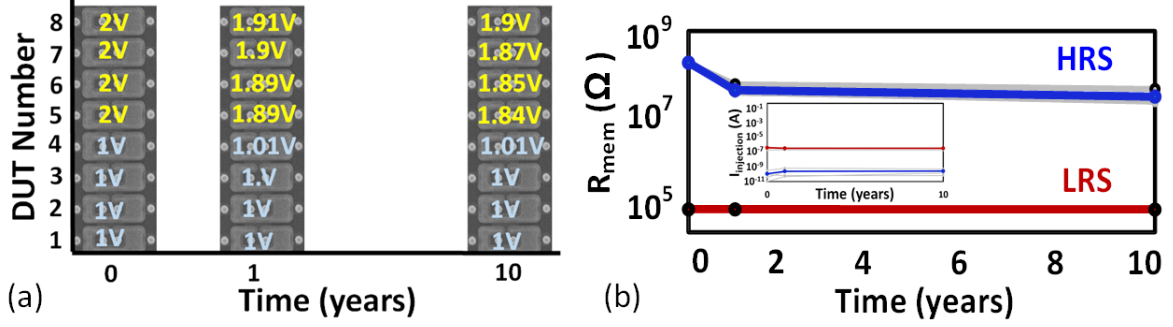
Supplementary Figure 28. (a) RESET degradation as a result of the injected electrons/holes into the floating gate. In the beginning, the injection is destructive, and the achieved V_{th} level is decreased under the same number of programming pulses, while (adaptively) varying the number of erase pulses to obtain a low resistive state (LRS), as shown in the inset. (b) After the degradation is almost over and steady state is reached, we characterize the switching stochastic dynamics by normal distribution of V_{th} and log-normal distribution of I_{DS} in LRS. Random telegraph noise is negligible in LRS.



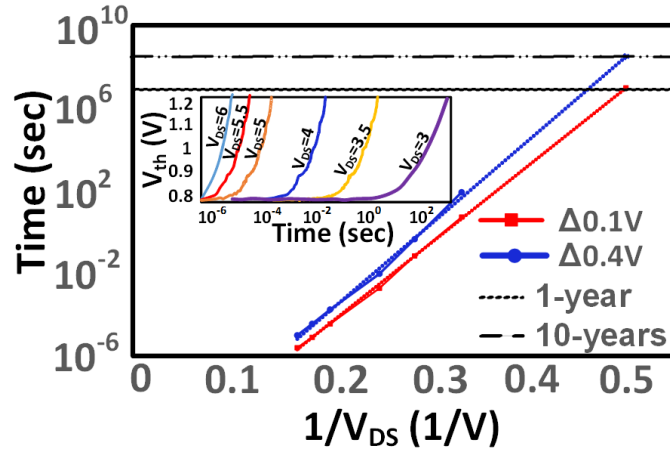
Supplementary Figure 29. (a) SET degradation as a result of the injected electrons/holes into the floating gate. In the beginning, the injection is destructive and the achieved V_{th} level is increased under the same number of erasing pulses, while (adaptively) varying the number of programming pulses to obtain a high resistive state (HRS), as shown in the inset. (b) After the degradation is almost over and steady state is reached, we characterize the switching stochastic dynamics by normal distribution of V_{th} and log-normal distribution of I_{DS} in HRS. Random telegraph noise is negligible in HRS due to the read transistor.

Supplementary Table 8. Statistical characterization of the Y-flash programming and erasing.

Statistics	V_PRG	V_ERS	I_PRG	I_ERS
variance	0.000329	0.00035	$2.1E - 20$	$9.5E - 15$
average	2.034088	0.934178	$4.76E - 10$	$1.1E - 06$
Stdev	0.01814	0.018709	$1.45E - 10$	$9.75E - 08$



Supplementary Figure 30. Analogue data retention emulation, performed using measurements at 250°C extrapolated to room temperature, with activation energy of 1.55 eV, for ten years. (a) V_{th} variation in 8 devices under test (DUT) in three different time stamps. (b) Analogue data retention in terms of resistance, one order of magnitude loss on average (solid line) in HRS. The inset shows the loss in the injection transistor current (two orders of magnitude on average). The read transistor increases the immunity to data loss and charge traps.



Supplementary Figure 31. Read disturb Takeda plot shows the time needed for voltage drop (0.1V to 0.4V) for different programming voltages. At V_r = 2V, read disturb will occur after ~10 years.

c. Number of resistive levels:

The theoretical number of resistive levels is a function of the number of injected electrons into the floating gate. The number of electrons in 180 nm Y-Flash cell is very high (compared to the number of ions in RRAM), where each injected electron leads to a change of the V_{th} and the corresponding memristance. Thus, the number of theoretical resistive levels is discretized by the number of the trapped electrons in the FG. The logarithmic nature of programming and erasing guarantees achieving of every level and high precision tuning for the continuous resistance spectrum. However, the physical number of resistive levels is a function of noise (thermal, environmental, peripheral circuit tolerance), variations (process, voltage), data retention, degradation, switching stochastics, leakage, and program disturb. Empirically we

found that the collective impact of these sources corresponds to 10nA variation of the read current in the array, which determined the noise margin as:

$$NM_{min} = \frac{10nA}{1\mu A} = 1\%$$

The corresponding number of resistive levels is determined accordingly:

$$\left(\frac{1 + NM}{1 - NM}\right)^N \leq \frac{HRS}{LRS} = 1000. \quad (5.2)$$

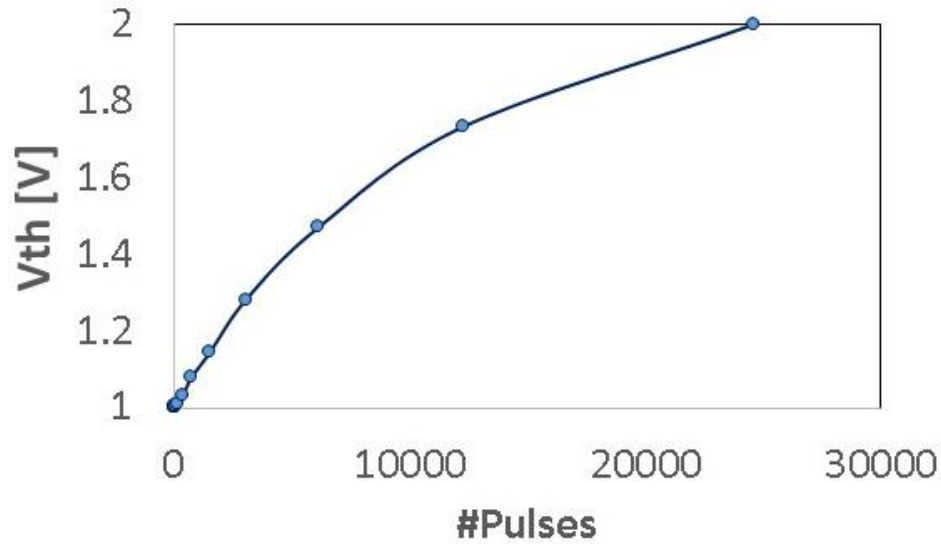
Solving this condition gives $N=65$. While the calculated number of resistive levels is not the maximal number that meets this condition, it is a sufficiently good N to meet this condition. Actually, the noise margin is not uniformly distributed. It depends on the resistive state (current) (e.g., 10nA/100nA = 10%) and thus should range between [1% - 10%]. An average NM of 5%, provides approximately 65 levels. In the same manner, if the maximum NM (at the numerator) is set as 10% and the minimal NM (at the dominator) as 1%, we will again obtain approximately 65 resistive levels. This is the number of resistive levels that peripheral circuits can reliably distinguish, which is roughly equivalent to 6-bit precision.

Alternatively, if the HRS/LRS is limited to 100 as a result of a 10nA variation (where the modified HRS corresponds to 10nA instead of 1nA), then the number of bits can be calculated as:

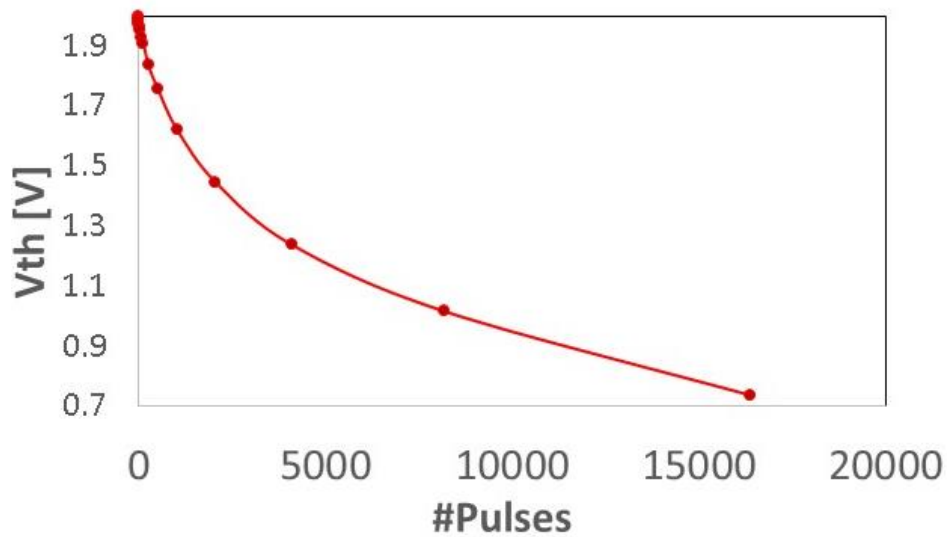
$$\#bits = \log\left(\frac{HRS}{LRS}\right) \approx 6$$

d. Adaptive Learning Rate:

Due to the logarithmic nature of the programming and erasing process, the change rate in terms of V_{th} or memristance varies according to the present state of V_{th} , pulse-time duration and programming/erasing voltage amplitude. This will lead to a time-varying learning rate $\eta(t)$, which changes the learning rate of the gradient descent. The non-linearity helps the algorithm to converge with higher accuracy and this intrinsic behaviour could outperform software-based implementation of ANN. One of the main benefits is that it eliminates the need to manually tune the learning rate. Thus, it could naturally implement the adaptive gradient descent algorithms (e.g., Adagrad), widely used techniques in deep neural networks. Nonetheless, one can compensate for the non-linearity in the learning rate and fix the rate to behave linearly-like by doubling the number of the applied pulses in each iteration, as shown in the following figures of programming and erasing:



(a)



(b)

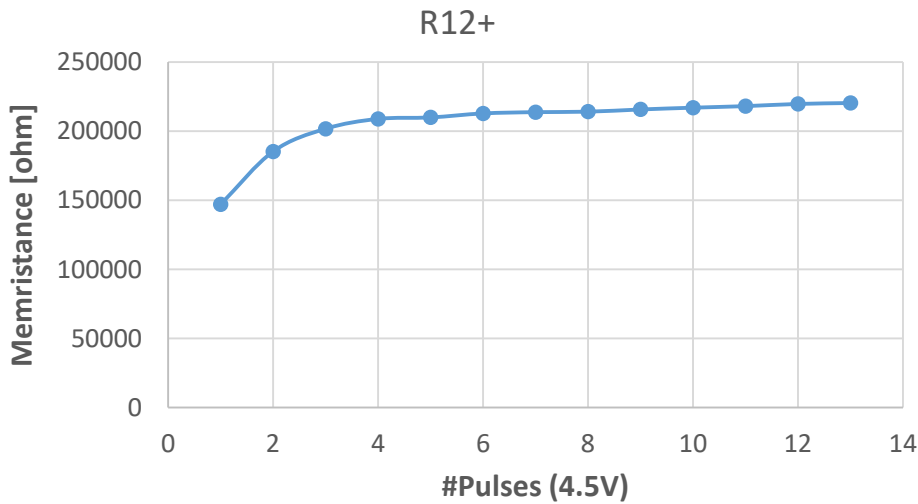
Supplementary Figure 32. Adaptive learning rate by doubling the (a) programming and (b) erasing time in each training iteration.

e. Hopfield CAM Energy:

As a result of the recurrency of the Hopfield network, it is complicated to train it to adapt to the desired functionality. Therefore, training the network is not necessary when the network deterministic functionality is known. We used the programming algorithm (we called it in this case offline training) implemented in LabView to precisely tune each memristor and configure it to the weight matrix proposed in the paper. However, the programming process has challenging side effects that might disturb the deterministic solution (program disturb). In order to achieve a solution that is close to the deterministic form, we employed the following heuristics:

- We used 4.5V programming voltage with longer programming time (rather than 5V).
- We used the differential synapse model to alleviate thermal and environmental variations.
- We performed programming for each synapse separately (two devices).

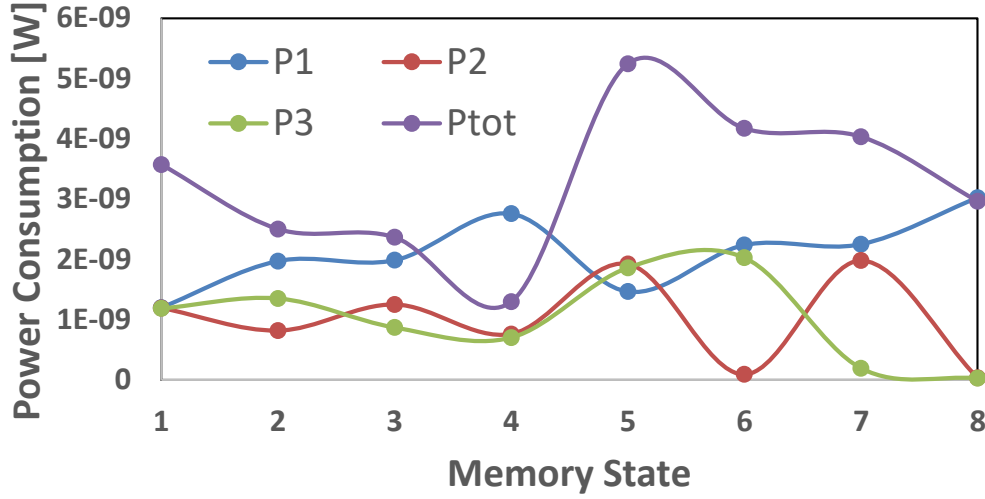
- We started programming from the largest synapse weight and down to smaller values. The l value for a differential synapse corresponds to the largest distance between the values of the two cells inside the synapse. Programming it ensures minimal disturbance from adjacent cells. Considering the logarithmic nature of the programming process, the relevant DUT will be minimally disturbed when adjacent cells are programmed. Thus, our training algorithm can decide upon the degree of programming accuracy prior to the programming. Then, when we program adjacent cells, the disturbance ensures sufficient precision of the DUT.
- We decreased the current dynamic range to two orders of magnitude [10nA:1uA] as a result of the noise margin and program disturb inside an array. This can increase the input dynamic range (IDR) to 100mV.
- We performed this protocol iteratively for each synapse. In every iteration we validated the impact on other synapses. After we finished, we performed two recursive iterations to calibrate the synapses and eliminate the accumulated disturbance from adjacent cells. We tested the symmetry of the network, and we saw that even if there are still some offsets the CAM still behaves as required. It is similar to the process variations and their impact on the CAM analysed in [3]. Our synapse model demonstrated robust results, and much flexibility in the resistance values of each cell where “symmetrical” values are derived (as a subtraction product between two cells in the synapse) from completely asymmetrical values.



Supplementary Figure 33. Fine-tuning protocol for configuring the Y-flash synapse in an array, taking into account non-ideal effects.

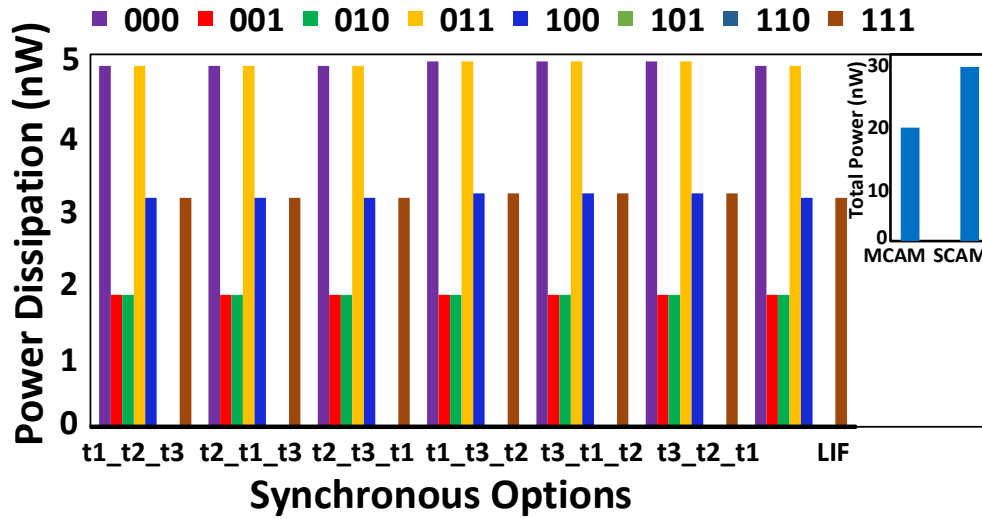
We implemented the single CAM Hopfield network under different operational conditions where the small-signal voltage was 100 mV (IDR=200 mV) with the same synaptic weights used in the original network. However, we failed to show convergence for specific initial states because of the nonlinearity of the Y-flash devices.

We analysed the power consumption of the network until convergence to the desired memory pattern in steady state. The AC power consumption of each neuron P1, P2, P3 (and the total network P_{tot}) for each memory state updated in the multiple CAM (MCAM) network is shown in the following figure:



Supplementary Figure 34. The AC power consumption of each neuron in each memory state at the steady state.

We found that the order of the firing is critical for the power consumption. In our experiments, we performed all the possible firing order combinations (7 different orders). The AC power consumption is calculated in each order set, and show that minimal power consumption (from any state i to any state j , with Hamming distance of 1) is achieved when the order of firing meets the leaky-integrate-and-fire neuron principle, where the neuron firing event is asynchronous and occurs proportionally to its current input strength, as shown in Supplementary Fig. 35. Furthermore, MCAM consumes less power than SCAM due to having two possible final states that could be reached from different initial states (local-minima). Based on that, we used the STDP learning rule for fine calibration of the synaptic weights to achieve an accurate and robust network that is less prone to fluctuations due to variations.



Supplementary Figure 35. Total power dissipation for transient updates from initial state to the final state by using different orders of updates.

f. Readout Transistor:

The readout transistor was added to eliminate read-disturb issues for the digital memory applications of the Y-flash, where its current is sensed in the saturation conduction mode (above threshold) and those voltages can inject channel hot electrons. This readout transistor has larger channel length to prevent hot electron injection, and lower threshold value. In the subthreshold domain, we do not necessarily need the readout transistor since we are reading using low voltage values (below threshold) without the concern for channel hot electron injection while

reading. I_{read} is a function of the transistor dimensions W/L. However, the subthreshold current is dominated by the threshold value since it exists in the exponent. Thus, the total current will be determined by the read transistor as depicted in Eq. 5.3:

$$I_{tot} = I_{read_r} e^{\frac{CR \cdot V_{DS}}{mV_{Tr}}} e^{\frac{V_{th_r}}{mV_T}} + I_{read_{inj}} e^{\frac{CR \cdot V_{DS}}{mV_{T_{inj}}}} e^{\frac{V_{th_{inj}}}{mV_T}} \approx I_{read_r} e^{\frac{CR \cdot V_{DS}}{mV_{Tr}}} e^{\frac{V_{th_r}}{mV_T}}, \quad (5.3a)$$

$$V_{th_{tot}} = V_{th_r} || V_{th_{inj}} \approx V_{th_r}. \quad (5.3b)$$

This drives higher currents using the desired voltages of ~2V (which guarantees no hot electron injection). Elimination of the read transistor will potentially decrease the area and increase the density, but will also require an increase in the read voltages applied on the injection transistor to drive the desired currents (1nA: 1μA), which will again cause hot electron injection and read disturb in low band currents (10pA: 100nA). Programming the injection transistor to the desired current dynamic range (1nA: 1μA), in the absence of the readout transistor, will improve the device characteristics, as shown in Supplementary Table 9. Furthermore, charge traps in the channel, and temperature will have less impact when adding the read transistor.

Supplementary Table 9. Y-flash performance evaluation with and without the read transistor.

	With readout	Without readout	Programmed without readout
Current	1nA: 1μA	10pA: 100nA	1nA: 1μA
V_{th}	1V: 2V	1.4V: 2.75V	2.05V: 3.1V
Applied Voltage	2V	2.75V	3.1V
#levels	65	< 16	< 32
Programming Time	10ms	10ms	~20ms
Endurance	100 Kcycles	<< 100 Kcycles	< 100 Kcycles
Analogue data retention	10 years	<< 10 years	< 10 years
area	3 μm ²	~1 μm ²	~1 μm ²

6. Comparative analysis between emerging memory technologies for neuromorphic computing:

In the early synapse-related modelling, attempts were made to emulate memristor-type performance using floating gate technology, where EEPROM-type memory devices (cells) were connected in cross-point arrays to implement neuro-inspired circuits that model learning processes in biological systems. As used herein, the phrase “floating gate technology” refers to integrated circuit fabrication techniques utilized, for example, to produce EEPROMs and other floating gate MOSFET (FGMOS) field-effect transistors. Such floating gate transistors are similar to conventional MOSFETs, but drain-to-source currents are controlled by a gate structure made up of an electrically isolated (*i.e.*, completely surrounded by oxide or other highly resistive material) block of doped polycrystalline silicon (polysilicon, or simply “poly”) capable of storing an applied charge for long periods of time. Floating gate technology employs the various programming mechanisms (*e.g.*, Fowler-Nordheim tunnelling and hot carrier injection) and associated circuit features to adjust (*i.e.*, program or erase) the charge stored on floating gates.

Most attempts to implement synapse-related modelling based on floating gate technology utilize three-terminal EEPROM-type devices. Note that, although these floating gate devices were able to simulate memristor behaviour, they are technically not memristors (*e.g.*, three terminals). Moreover, attempts to utilize three-terminal floating gate devices were very complicated due to the need to control the third terminal (*i.e.*, the control gate in non-volatile memory), which made the synapse-related modelling circuit schematically difficult and costly. Such three-terminal devices were used by Bernard Widrow and Ted Hoffin in the first neuromorphic simulators named Adaline and Madaline. Examples of such devices can be found in [4] and recently used in [5].

Some three-terminal EEPROM-type devices utilized in neural networks were fabricated using double-poly CMOS processes, as shown in [6]. The suggested device includes source and drain terminals, and a third (control gate) terminal implemented using a second polysilicon (POLY2) structure disposed over the floating gate, which is implemented using a first polysilicon (POLY1) structure. These devices are programmed by channel hot electrons and erased by Fowler-Nordheim tunnelling. The authors of [6] introduced a two-dimensional synaptic array in which each three-terminal EEPROM-type device formed one synapse of the array. In addition to the devices not being memristors and in addition to their fabrication using a complicated double-poly fabrication technology, the three-terminal EEPROM-type devices encountered array limitations in that row synapses shared a common drain line, so tunneling at one synapse caused undesired tunnelling and hot carrier injection at other row synapses.

A two-terminal EEPROM-type floating gate cell is disclosed in [7]. Based on practical experience, we understand that cross-point arrays generated using conventional two-terminal EEPROM-type devices encounter over-erase problems during erase and require the use of high-voltage readout operations, where both of these problems lead to read disturb issues that prevent reliable synapse-related modelling. An uncontrollable erase mode may occur that can result in an over-erase condition, which can lead to degradation or even complete failure of the device. That is, the uncontrollable erase mode occurs when the element's floating gate reaches a fully erased state during an erase operation, and channel electrons generate secondary holes that enter the floating gate, producing an undesirable further reduction of the stored charge that produces a threshold voltage below the desirable level. This uncontrollable erase mode cannot be prevented unless the drain electrode is disconnected or otherwise isolated during the erase operation. However, because the drain terminal is directly connected to an associated bit line in order to facilitate programming operations, and because a third terminal is not available to selectively isolate the drain terminal during erase operations, these elements are subject to over-erase.

Several types of NVM cells have been developed that can be produced with minimal changes to a standard CMOS process flow, for example, serial connection of two PMOS transistors to compose a programmable read only memory. However, they suffer from several drawbacks:

- i. Impractical structure for performing erase operations due to the large voltage on the control gate that is needed to stimulate electron back-tunnelling from or hole injection to the floating gate—even for voltages approaching the breakdown of the N-well junction, the potential difference between the floating gate and N-well is smaller than several volts.
- i. The symmetric arrangement requires a select transistor to address the cell.
- ii. Along with the two-PMOS arrangement, the special N-well requires additional space, thus making the overall cell size undesirably large.

In the last decade, another revolution has taken place in the field of nanoelectronic memory devices. Of particular importance, is the development of the technology for programmable, nonvolatile two-terminal devices based on resistive switching called RRAM. Novel passive devices with new computational capabilities and applications including: memory, logic, signal processing, and sensing. These devices have low chip footprint, which is determined only by the overlap area of the metallic electrodes. The theoretical concept of a memristor was applied to physically measured devices in 2008 at HP Labs (D. Strukov *et. al*) and since then there has been rapid progress in the development of such devices, leading to a series of recent demonstrations of memristor-based neuromorphic hardware systems. Many kinds of memristors have been reported as candidates for synaptic devices, since they are easy to construct into a dense matrix-type neural network. These include RRAM phase change, ferroelectric, and magnetic memories, among others—for a review see [8-9].

Much of the previous impressive demonstrations of neuromorphic networks based on resistive switching memory devices, including pioneering work by IBM [10], were based on the so-called 1T1R technology, in which every memory cell is coupled to a select transistor. The reports of neuromorphic functionality based on passive 0T1R or 1D1R circuits (in which acronyms stand for 0 Transistor or 1 Diode +1 Resistive switching device per memory cell, respectively) have been so far very limited, in part due to much stricter requirements for memristor's $I-V$ uniformity for successful operation. In particular, these emerging devices have already been used to demonstrate small neuromorphic networks, by the UCSB research group led by D. Strukov [11] in 2015. The focus on 0T1R memristive crossbar circuits is specifically due to their better density, performance and energy-efficiency prospects. Recently (2018), the same UCSB group reported a much more complex network with one hidden layer comprised of almost an order of magnitude higher number of passive devices as compared to the previously reported neuromorphic classifiers [12]. The successful demonstration was facilitated by improvements in fabrication technology of memristors, specifically by lowering variations in their $I-V$ characteristics. However, their fabrication technology is still in much need of improvement and not ready yet for the large-scale integration necessary for practically valuable neuromorphic networks.

Fortunately, during the last decade the nonvolatile floating-gate memory cells have not only been highly optimized and scaled down, but also may now be imbedded in CMOS integrated circuits. Recently, a prototype of a three-layer mixed-signal neuromorphic network was implemented, by the same UCSB research group, using digital-input-analogue-weight arrays of highly optimized embedded nonvolatile floating-gate cells redesigned from a commercial 180nm NOR flash memory [5]. Their main advantage is the mature fabrication technology. The memory arrays have been redesigned to allow for individual, precise adjustment of the memory state of each device. Their training experiments achieved a sub-1 μ s time and sub-20nJ energy—both numbers much better than those in the best reported digital implementations. Custom design has recently been demonstrated for the industrial-grade 55 nm NOR flash memories, and 28nm memories, which are already commercially available [5].

Stability of numerous V_{th} levels over a wide dynamic range is a key requirement when EEPROM operates in an analogue mode. The SuperFlash memory device is erased from the edge of the floating-gate to the control-gate, in the presence of a strong electrical field. This leads to generation of a strong charge trap (and then electron trapping) in the control-gate oxide. Fortunately, the excellent endurance of SuperFlash may compensate for the V_{th} shifts with sufficient data retention [13-14].

The SuperFlash is principally a three-terminal (or more) device comprising 1.5T. Peripheral circuits that control the terminals have been developed to handle half-select disturb issues in programming/erasing, and avoid disturbing sneak paths [5][13]. However, the overhead of control

circuitry supplying standard CMOS voltages is typically negligible compared to the muxes supplying high voltages, which is similar, in that respect, to other EEPROM technologies. Note that read/inference in SuperFlash can be performed at standard CMOS voltages [14].

Furthermore, the floating-gate cells were operated at the energy-efficient subthreshold domain with exponential I-V characteristics, where its current is dominated by V_{gs} , and could be represented as a multiplication of two separable functions (input and weight). The inputs can be analogue, as was shown for gate-coupled and source-coupled floating-gate VMM implementations [5][15]. Note that such gate- and source-coupled VMMs allow for a large dynamic range of input/output currents, which would be important for reducing peripheral overhead. Moreover, time-domain approaches, in which analogue inputs are encoded in digital input pulse durations, simplify peripheral circuitry used in gate- and source-coupled designs. Furthermore, small-signal linearization can be used to encode analogue inputs. Therefore, the desirable synaptic weights calculated in an external computer running a similar software implemented network, using standard machine learning algorithm, were transferred into the network by analogue tuning of the memory state of each floating-gate cell, with peripheral analogue demultiplexer circuitry (ADCs and voltage shifters). The digital encoders and shift register circuits and their layouts were synthesized from Verilog in a standard 1.8V digital CMOS process.

A growing crowd of companies are hoping to deliver chips that accelerate otherwise onerous deep learning applications [16], using massively parallel embedded flash memory (operating in above threshold domain) to greatly reduce the amount of power needed to perform deep-learning inference computations. However, the large footprint, high switching voltages, number of terminals, endurance, and the exponential I-V characteristics of the device still constrain an upper bound on the overall performance that could be achieved. Thus, further optimizations of these cells should be investigated.

Previous attempts to optimize floating-gate devices and to take further steps towards memristive devices, proposed using the MemFlash cell [17-18] to emulate the resistive switching mechanism of memristive devices. The MemFlash cell is an EEPROM transistor with a particular wiring scheme, in which the gate terminal is connected to the source terminal. By also applying a negative bulk potential, the interconnections lead to a two-terminal device with memristive operation mode. Originally, the MemFlash concept was demonstrated on single-cell devices and not within an array. Thereafter, the MemFlash configuration has been implemented using industrial SONOS cells, which do not require a bulk voltage supply [19]. The two-terminal diode wiring scheme of MemFlash cells has been extended by a resistively shunted pn-diode. This fact demonstrates memristive device compatibility with industry standards and scaling possibilities [19]. The additional electrical devices in the diode-like MemFlash wiring scheme are necessary to create an electric field distribution that allows for charging and discharging of all the distinct traps in the nitride. This circuitry differs from the original MemFlash concept, where an accumulation-type transistor was used [17]. Compared to previous EEPROM MemFlash cells [17], which require an additional external potential for operation, such SONOS configuration is a true two-terminal device [19]. An external resistor and a pn-Si diode were placed in parallel between the source terminal and ground on a breadboard. The polarity-dependent asymmetry in the resistance of the memristive device can easily be adjusted via the circuit parameters and might prove useful in applications. In particular, in crossbar arrays, the diode behaviour of the device helps address problems by eliminating sneak paths [19].

In addition, the charge trap densities in the nitride layer may be increased. Compared with SONOS solutions, the original MemFlash has thin tunnelling oxide, which might lead to a V_{th} drift and shorter data retention. The effect that tunnelling oxide thinning has on the retention and the corresponding hysteretic current-voltage curves of MemFlash cells, was later investigated through measurements as well as simulations [20]. Furthermore, the variation of charging and discharging voltages along with different cycle frequencies have been explored with respect to the change in hysteretic behaviour. In addition, the influences of changing device parameters on its behaviour as artificial synapse, have been investigated by emulating LTP for different MemFlash cells.

The MemFlash operates in different manners than the floating-gate synapse transistor. It conducts above threshold in the linear (ohmic) mode, where the device current is a function of the input voltage applied on the drain. Yet, unlike the floating-gate synapse transistor where its low current is exponentially

dominated by the gate-source voltage, the two-terminal MemFlash had $V_{gs} = 0$ and did not operate in the subthreshold regime. Finally, the MemFlash was used in spiking neural network using STDP, as was shown in [18], implemented by high-voltage and long-time programming pulses.

We utilize the technological maturity of floating-gate devices and the conceptual novelty of memristive devices, including their efficient computational capabilities, to propose a novel memristive floating gate device. Aiming to close the gap between floating-gate and memristive devices as synapses in large-scale neuromorphic systems, we fabricated a two-terminal single poly floating gate non-volatile memory device in a purely standard CMOS process (without additional manufacturing masks), similar to a SONOS MemFlash [19]. Our device is gradually switched as a floating gate memory device with optimized voltages and times (5V/8V voltages for RESET/SET with 10 μ s pulses), and operates in the subthreshold memristive mode similarly to a “synaptic transistor”, but linearized for small-signal voltages as a resistor (linear I-V). We show the analogies between the operation of our device, including its switching kinetics and the theoretical memristive device, based on a physical model fitted to experimental results we performed on devices using 180nm CMOS process. We strongly believe that our device could be easily scaled down to advanced standard CMOS technology nodes.

Analogously, we employed novel memristive design techniques that were used in recent neuromorphic implementations. We added a readout transistor connected in parallel to the proposed device to enable lower operational voltages, and to enhance reliability performance of the device. This decreases the power consumption, but at an expense of some area penalty. Overall, the equivalent cell comprising both transistors operates in a memristive mode, corresponding to the MemFlash cell [17]. Besides the beneficial inherent non-linearity of the device, we designed the dimensions and capacitances of the device in asymmetric style where it mostly enables current flow in one direction at a certain voltage. This will prevent disturbing issues (e.g., sneak path, half-select, and so forth), thus obviating the need for selector devices in large arrays, potentially leading to higher density and lower cell area than 1T1R memristive crossbars. Therefore, we fabricated several integrated arrays prototypes based on our device (12x8 until 1M bit arrays) with >96% yield.

Our measurements showed a combination of the benefits of both floating-gate and memristive devices: two orders of magnitude current dynamic range (high-to-low-resistive-state ratio), 65 discrete resistive levels (with 10nA noise margin), 100 mV input dynamic range (linear I-V), 100K cycles for endurance, and ten-year analogue data retention (Supplementary Table 10). Furthermore, the device demonstrated the STDP learning rule, widely used for spiking neural networks, which updates the device conductance according to the exponent of the time difference between pre- and post-synaptic neurons. We utilized the exponential nature of the switching dynamics of our device, which has been previously shown in floating gate devices, combined with our memristive properties in small-signal, to show efficient STDP without the need for digital time division multiplexing circuits used in RRAM devices. Later, we used the differential synapse model to enable negative weights, similar to a previous report [11]. Then, we demonstrated linear VMM on the 12x8 fabricated arrays using small-signal analysis.

Using VMM and the gradual switching dynamics of the device, we investigated two proof-of-concept neuromorphic applications, recently shown on small-scale RRAM networks [3][11]. First, we show associative memory using our array connected in a Hopfield neural network topology. The Hopfield network has proved useful in content-addressable memories (CAM). It is a dynamic, asynchronous, and recursive system consisting of a set of interconnected neurons. 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 work, a three-bit CAM Hopfield network is constructed. This network has the potential for emulating memorization and for deep biological study using large scale neuromorphic networks based on our mature device.

The second neuromorphic application implements an integrated array for *in situ* training in small-signal of an analogue-inputs-analogue-weights single-layer ANN to classify 3x3-pixel black/white images of stylized letters into three classes.

Supplementary Table 10. Summary of specific previous RRAM devices, floating-gate devices and the Y-flash, reported for neuromorphic applications studies.

Device	RRAM [11] memristor	RRAM [3] memristor	Synapse transistor [6] EEPROM floating-gate	SuperFlash [5][13][14] NOR flash floating-gate	MemFlash EEPROM floating-gate [17][18][20]*	Y-Flash [1] floating-gate memristive device
Structure	0T1R	1T1R	1T	1.5T	1T	2T (potential 1T)
#terminals	2	3	3	5	3/2	3 (potential 2)
Materials	Al ₂ O ₃ /TiO _{2-x}	Au/Ni/HfO ₂ /SiO ₂	Double poly MOS	Double poly MOS	Double poly MOS/SONOS	Single poly MOS
Technology	30nm CMOS/RRAM	RRAM	2μm BiCMOS	180nm ESF	350nm	180nm CMOS
RESET Pulse voltage	$V_{TE} = -1.3V$ $V_{BE} = 0V$	$V_{TE} = -1V$ $V_{BE} = 0V$	$V_{G_1} = 5V$ $V_D = 25V$ $V_S = 0V$	$V_S = 6V$ $V_D = 0V$ $V_G = 1.6V$	$V_S = 0V$ $V_G = 0V$ $V_D = -7V$ $V_B = V_{D,min} $	$V_D = 4.5V$ $V_S = 0V$
SET Pulse voltage	$V_{TE} = 1.3V$ $V_{BE} = 0V$	$V_{TE} = 1V$ $V_{BE} = 0V$	$V_{G_2} = 4.5V$ $V_D = 35V$ $V_S = 2V$	$V_S = 0V$ $V_D = 0V$ $V_G = 9V$	$V_S = 0V$ $V_G = 0V$ $V_D = 10V$ $V_B = V_{D,min} $	$V_D = 0V$ $V_S = 8V$
RESET pulse time	500 μs	5 ms	1 s	5 μs	100 ms	10 μs
SET pulse time	500 μs	5 ms	1 s	50 μs	100ms	10 μs
Reading voltages	$V_{TE} = \pm 0.1V$ $V_{BE} = 0V$	$V_{TE} = 0.1V$ $V_{BE} = 0V$	$V_{G_1} = 5V$ $V_D = 5V$ $V_S = 0V$	$V_S = 0V$ $V_D = 1V$ $V_G = 2.5V$	$V_S = 0V$ $V_G = 0V$ $V_D = 2.5V$ $V_B = V_{D,min} $	$V_D = 2V$ $V_S = 0V$ $v_{ds} = \pm 50mV$
#Resistive levels	>32	>32	~10	>64	N/A	65
HRS/LRS	1000	>1000	67	30000	600/2000	100
Low current	100nA	~1μA	2μA	10pA	-0.66/ 0.225μA	10nA
High current	100μA	1mA	30nA	300 nA	-0.4/0.45mA	1 uA
Endurance	5 Kcycles	N/A	Low	500 Kcycles	N/A	100 Kcycles
Analogue data retention	10 years	N/A	<10 s	>7 months	16 hr	10 years
Area	200nm x 200nm	N/A	11μm x 8μm	~1.56 μm x 1.92 μm	84.98 μm ²	3 μm ² (or 1 μm ²)
Array size	12x12	Single cell	2x2	785x128	Single cell	12x8
Yield	74%	N/A	Medium	High	High	>96%
I-V	Linear	Linear	Exponential	Exponential**	Quadratic	Linearized

*The MemFlash can be implemented using a two-terminal SONOS 180nm transistor [19] without a bulk voltage supply, achieving: high HRS/LRS, high endurance, high data retention and yield, as well as small area. In addition, it uses a diode selector to eliminate sneak paths in large-scale arrays.

**Linearized using time-domain modulation techniques or gate- and source-coupled VMMs.

References

- [1] Roizin, Y., & Pikhay, E. Memristor using parallel asymmetrical transistors having shared floating gate and diode. *U.S. Pat.* 9,514,818 (2016).
- [2] Sharroush, S.M., Abdalla, Y.S., Dessouki, A.A. and El-Badawy, E.-S.A. Subthreshold MOSFET transistor amplifier operation. In *4th Int. Design Test Workshop (IDT)* 1–6 (IEEE, 2009).
- [3] Hu, S.G. *et al.* Associative memory realized by a reconfigurable memristive Hopfield neural network. *Nat. Commun.* **6**, 7522 (2015).
- [4] Durfee, D.A., and Shoucair F.S. Comparison of floating gate neural network memory cells in standard VLSI CMOS technology. *IEEE Trans. on Neural Netw.* **3**, 347-353 (1992).
- [5] 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).
- [6] Diorio, C., Hasler, P., Minch, A., and Mead, C.A. A single-transistor silicon synapse. *IEEE Trans. Electron. Dev.* **43**, 1972–1980 (1996).
- [7] Kohlstedt, H., *et al.* EEPROM memory cell as a memristive component. *Int. App. No.* WO2013178730 (*German App. No.* DE102012209336) (2013).
- [8] Upadhyay, K.N. *et al.* Emerging memory devices for neuromorphic computing. *Adv. Mater. Techno.* (2019)
- [9] Kim, C. *et al.* Emerging memory technologies for neuromorphic computing. *Nanotechnology* (2019)
- [10] Burr, G.W. *et al.* Experimental demonstration and tolerancing of a large-scale neural network (165000 synapses) using phase-change memory as the synaptic weight element. *IEEE Trans. Electron Dev.* **62**, 3498–3507 (2015)
- [11] Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
- [12] Bayat, F. Merrikh, *et al.* Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* **9**, 2331 (2018).
- [13] Guo, X. *et al.* Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology. In *Proc. IEEE Int. Electron Dev. Meet. (IEDM)* 6.5.1-6.5.4 (IEEE, 2017).
- [14] Guo, X. *et al.* Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm NOR flash memory cells. In *Proc. IEEE Cust. Integr. Circ. Conf. (CICC)* 1-4 (IEEE, 2017).
- [15] Merrikh Bayat, F., Guo, X., and Strukov, D. Exponential-weight multilayer perceptron. In *Proc. IEEE Int. Joint Conf. on Neural Netw. (IJCNN)* 3709–3715 (IEEE, 2017).
- [16] Moore, Samuel K. Two startups use processing in flash memory for AI at the edge. (2018) Available at: <https://spectrum.ieee.org/tech-talk/computing/embedded-systems/two-startups-use-processing-in-flash-memory-for-ai-at-the-edge>. (Accessed: 1st January 2019)
- [17] Ziegler, M. *et al.* Memristive operation mode of floating gate transistors: A two-terminal MemFlash-cell. *Appl. Phys. Lett.* **101**, 263504 (2012).
- [18] Ziegler, M., and Kohlstedt, H. Mimic synaptic behavior with a single floating gate transistor: A MemFlash synapse. *J. Appl. Phys.* **114**, 194506 (2013).
- [19] Himmel, N. *et al.* Memristive device based on a depletion-type SONOS field effect transistor. *Semiconductor Science and Technology* **32.6**, 06LT01 (2017).
- [20] Winterfeld, H. *et al.* Technology and electrical characterization of MemFlash cells for neuromorphic applications. *J. Appl. Phys.* **51**, 324003 (2018).