Design Flow for Hybrid CMOS/Memristor Systems Part I: Modelling and Verification Steps

Sachin Maheshwari*, *Member, IEEE,* Spyros Stathopoulos*, Jiaqi Wang*, *Student Member, IEEE,* Alexander Serb*, *Senior Member, IEEE,* Yihan Pan*, *Student Member, IEEE,* Andrea Mifsud, Jiawei Shen, Lieuwe B. Leene, Christos Papavassiliou, *Senior Member, IEEE,* Timothy G. Constandinou, *Senior Member, IEEE,* Themistoklis Prodromakis, *Senior Member, IEEE,*

***Abstract*—Memristive technology has experienced explosive growth in the last decade, with multiple device structures being developed for a wide range of applications. However, transitioning the technology from the lab into the marketplace requires the development of an accessible and user-friendly design flow, supported by an industry-grade toolchain. In this work, we demonstrate the behaviour of our in-house fabricated custom memristor model and its integration into the Cadence Electronic Design Automation (EDA) tools for verification. Various input stimuli were given to record the memristive device characteristics both at the device level as well as the schematic level for verification of the memristor model. This design flow from device to industrial level EDA tools is the first step before the model can be used and integrated with Complementary Metal-Oxide Semiconductor (CMOS) in applications for hybrid memristor/CMOS system design.**

***Index Terms*—EDA tools, hybrid CMOS/memristor, modelling, verification**

I. Introduction

CMOS technology is facing numerous challenges due to the continuous decrease in the device dimension. It is now practically approaching its physical limits of miniaturization, however, due to its negligible static-power dissipation at higher technology nodes, it is still thought to be an important part of the future technology. This gradual end of Moore's law eventually commences a new era in research and development of emerging technologies for future intelligent computing systems. One such emerging nano electronic device is called memristor, postulated by Leon Chua in 1971 [1] after studying the relationships between charge and flux in inductors, capacitors and resistors.

S. Maheshwari, S. Stathopoulos, J. Wang, A. Serb, Y. Pan, T. Prodromakis, are with the Centre for Electronics Frontiers, Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO171BJ, UK. (E-mail: {s.maheshwari, s.stathopoulos, jw9y17, a.serb, y.pan, t.prodromakis}@soton.ac.uk)

A. Mifsud, C. Papavassiliou and †T. G. Constandinou are with the Department of Electrical and Electronic Engineering, Imperial College London, London, SW7 2AZ, U.K. †T. G. Constandinou is also with the Care Research & Technology Centre at Imperial College London, UK Dementia Research Institute (E-mail: {a.mifsud,c.papavas, t.constandinou}@imperial.ac.uk).

L. B. Leene is with Novelda AS, Norway, Oslo (E-mail: lieuwe.leene@novelda.no).

*These authors contributed equally in the paper

Memristor has a simple and nanoscale structure that consists of Metal-Insulator-Metal (MIM). As a result, the fabrication of these devices is similar to the processing of a via between two metal lines. Over the years, memristive technology have been vigorously researched and explored in a wide variety of applications including: non-volatile memory [2], programmable logic gates [3], reconfigurable computing [4], analog computing [5], [6], neuromorphic computing [7], image processing [8] and hardware security [9]. Moreover, the non-linear circuit and system theory plays an invaluable role in acquiring a comprehensive picture of the memristive device in the design of robust and systematic neuromorphic circuit design. [10], [11]. The use of memristors offers an improvement over the state-of-the-art CMOS technology in terms of integration density, energy consumption, multi-level programming capabilities and speed. However, designing memristor-based circuits requires a more reliable model of the memristor's behaviour at a physical level in order to attain similar system level design efficiency and accuracy of the much more matured CMOS technology. Thus, the plethora of memristor models have also been proposed in the literature [12]–[14].

Although a common memristive device still does not exist due to their different switching mechanisms and material properties, a design flow that is versatile, robust, user-friendly and can be integrated with any CMOS technology is the necessity in the current scenario. The purpose of this part is to give a holistic view of the memristor behaviour from experimental analysis to its verification steps using a cadence virtuoso design environment.

The work is split into two parts. Part I presents in-house fabricated $Pt/TiO_x/Pt$ VCM memristor model characteristics and thorough step-by-step guidelines for integrating the model into the Cadence design environment for verification. Part II is more a design and layout perspective showing by example the strategy to design CMOS circuits with memristor and a customised layout for the memristor cell. Also, instructions are provided for the standard cell to be integrated into the Calibre environment.

Part I of this work is organized as: Section II provides the behavioural model and the memristor device dynamics that has been developed, characterised and tested within the facilities at the University of Southampton. This section also demonstrates the switching behaviour of the device given an applied pulse/s of a particular width and amplitude. Section III, introduces the Verilog-A model followed by step-by-step construction and

integration of the model into the Cadence. Consistent with the previous section, this section also presents the switching behaviour using various stimuli for validation. Albeit separated from the main text, the memristor behaviour for the model utilizing exponential fitting [15] is provided in Appendix A for different resistive state ranges. Finally, section IV concludes the paper.

II. Modelling Memristive Behaviour

Before attempting to integrate memristors into higher level design we should establish a way of modelling the response of memristive devices for different input stimuli. This section presents a short description of several types of memristive devices from the point of view of a CMOS workflow and then it introduces the concept of a behavioural model for the specific class of devices that are used for the purposes of this work. This phenomenological data-driven model serves as the basis for the rest of the methodology hereof. Of course, given the breadth of available technologies, no model is infallible. Should a different model be more suitable for a given technology, information is presented in a modular manner so as to facilitate an easy transition to a different one.

A. Memristive Devices in a CMOS-oriented Workflow

Memristors come in different configurations and topologies and considerable work is available in the literature to describe the merits and disadvantages of each of those [16]–[18]. It is important to mention that the term *memristor* defines a behaviour rather than a specific form of device and materials. Although the memristor as originally postulated by Chua [19] connects flux, Φ, and charge, q, ($d\Phi = Mdq$) in practical terms it is realised through a variation of an internal state variable (typically resistance) based on the history of biasing that has been applied to the device. That behaviour is what gives rise to the *memory* characteristics typical of such structures [20].

Implementations of memristive devices has been demonstrated on a multitude of material systems over the past couple decades. These range from binary metal-oxides [21], halcogenides [22], perovskites [23] or even polymers [24] and other 2D materials [25], [26]. Out of these materials metal-oxides are probably the most common, with the original RRAM-based memristor using TiO_x as the active layer material [27]. Metal-oxide-based RRAM devices are grouped into two broad categories depending on the mechanism of operation. The first is *Valence Change Memory* (VCM) the operating principle of which is based on the modification of the stoichiometry within the active material when external voltage is applied. This change in stoichiometry is primarily driven by the movement of mobile oxygen vacancies [28]. The second is the *Electrochemical Metallisation Memory* (ECM) which is based on the formation of a metallic conducting filament due to partial diffusion of one of the electrodes into the metal-oxide [29]. Metal-oxides are also involved in the implementation of spin-torque transfer memories [30] that can also exhibit memristive behaviour [31] although in this particular case transitions between resistive states are based on magnetic coupling rather than ionic movements.

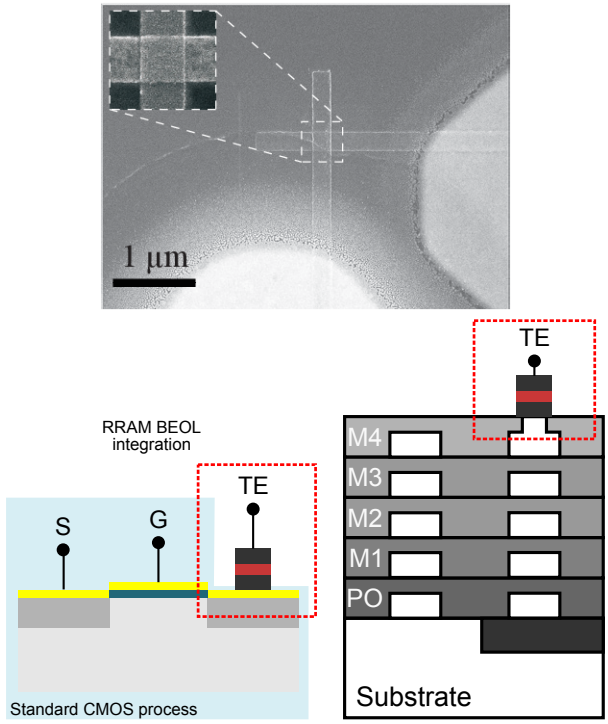


Fig. 1. Photo of a crosspoint sub-1μm^2 active area of a single device (top). Example of possible back-end-of-line (BEOL) integration on a standard CMOS process. The simplicity of the multi-layer structure allows for straightforward post-CMOS integration (bottom left and right). One of the terminals of the device is the biasing electrode used to program the device while the other is connected to the CMOS drain. Toggling the gates of the transistor effectively acts as a selector for that specified device.

For the purposes of this work we are going to focus on a simple VCM metal-insulator-metal structure based on TiO_x as an active layer and platinum as top and bottom electrodes. Devices based on the (bottom to top) Pt/TiO_x/Al_xO_y/Pt stack have given exemplary results in the past [2] and these will serve as the basis for this paper. Of course this approach is not limited to a specific type of device but can be generalised for a broad family of VCM memories. While this manuscript presents a methodology for designing circuits with emerging non-volatile memory technologies and uses an exemplar memristor technology and model, we note that the same principles can be utilized with alternative technologies [32] and models [33] from the literature, including volatile technologies [34].

In its simplest form a typical VCM memory is a stack of active material, usually a dielectric, between two metal electrodes (fig. 1). A variety of metal combinations can lead to different contact behaviours as discussed previously [35]. The relative simplicity of the structure is important for the mask-efficient integration process of such device in a CMOS standard process.

A CMOS integration effort featuring memristors require practical, fast and accurate models for the technology at hand. Depending on the underpinning physical mechanism, different models have emerged for the specific family of memristive materials and devices, be it resistive valence

change memories [36], spin-torque transfer memories [37] or electrochemical metallisation memories [12] to name a few. As it is probably not possible to have a model for every existing combination of materials from an integration point of view, what is more important is the overall *behavioural* characteristics of the device and how these are affected by volatility, noise and thermal effects. After all, Chua's definition of a generic voltage-driven memristor can be pinpointed to a set of two basic equations [19].

$$v = R(x)i \tag{1}$$

$$\frac{dx}{dt} = f(x, v) \tag{2}$$

where v, i and R the voltage, current and resistance of the device, respectively, which can be dependent upon an internal state variable, x. This is what Chua calls the differential algebraic form of a memristive response [38].

B. Phenomenological Models and Device Dynamics

In a phenomenological model there are many approaches to model the temporal evolution of the function $R(x)$. Kvatinsky et al. [13] proposed a linear/exponential expression where others [39] have put forward a hyperbolic sinusoidal function which is appropriate for metal-insulator-metal devices. In the simplest approach, the function $f(x, v)$ is approximated by a product of decoupled equations in the form of $f(x, v) = s(v) \cdot g(x)$. Function $s(v)$ describes the voltage sensitivity of the device while $g(x)$ is the *window function* that delimits the operational boundaries of the device. This form of decoupling is necessary in order to address boundary issues with the original proposition of the memristor model as presented by Strukov et al. [27]. Choosing a window function that is generalised enough to be used across different devices and technologies has been a challenging effort and several have been presented that have been tailored to specific modelling efforts [13], [40].

For the purposes of this methodology we will be relying on the phenomenological model presented in [14] as it has been used extensively by the authors before for the devices described in this section. It is also generic enough for voltage-modulated devices and is capable of emulating the transient response during switching depending on the time step resolution. The state variable in this case is the resistive state itself, the sensitivity function is a voltage-dependent exponential function while the window function employed is a state-dependent quadratic. Of course the internal state variable itself will depend on a series of additional physical parameters, such as temperature. How this affects the resistive response of the device is being addressed elsewhere [41] as it goes beyond the scope of the present work. Being state and voltage dependent a *behaviour* of the device can be predicted for a given voltage stimulus at a specific resistive state.

$$i(R, v) = \begin{cases} a_p(1/R)\sinh(b_p v) & v \geq 0 \\ a_n(1/R)\sinh(b_n v) & v < 0 \end{cases} \tag{3}$$

$$\frac{dR}{dt} = g(R, v) = s(v) \cdot f(R, v) \tag{4}$$

with $s(v)$ being the switching sensitivity function, describing the switching rate changes with voltage amplitude.

$$s(v) = \begin{cases} A_p(-1 + \exp(t_p|v|)) & v > 0 \\ A_n(-1 + \exp(t_n|v|)) & v < 0 \\ 0 & v = 0 \end{cases} \tag{5}$$

and $f(R, v)$ the window function. Parameters $r_{p,n}$ depend on the voltage in a polynomial fashion and describe the boundaries of the state variable.

$$f(R, v) = \begin{cases} (r_p(v) - R)^2 & v > 0 \\ (R - r_n(v))^2 & v < 0 \\ 0 & v = 0 \end{cases} \tag{6}$$

All other variables are free fitting variables. Fig. 2 depicts an indicative surface as described by the equations 3, 5 and 6. The surface plots the switching rate from equation 3 as a function of the applied stimulus and current resistive state (ie. the state variable).

In the particular case illustrated in Fig. 2, there is a region around both R_0 (the initial resistance, typically in the middle of the operating range of the device) and $v = 0$ where voltage stimulus does not produce any appreciative change in the state variable. As the offset from the initial resistive state increases the effect of amplitude gets increasingly pronounced indicating a typical bipolar memristive behaviour.

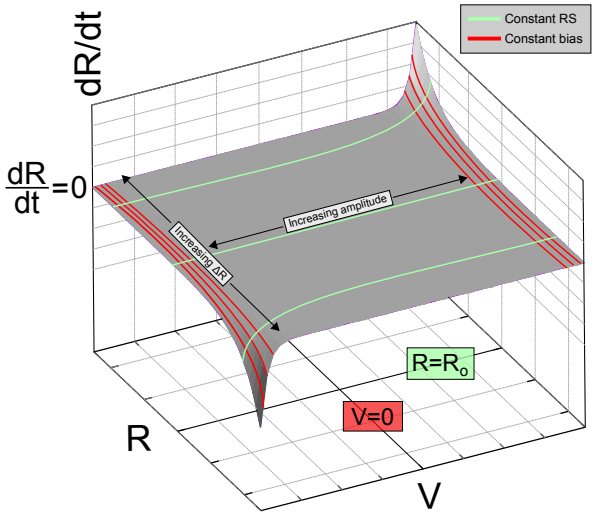


Fig. 2. Switching surface based on the model used for this work (reproduced from [15]). RS in this context stands for "resistive state".

A response of a bipolar device to an increasing amplitude voltage stimulus can be seen in Fig. 3. In this case positive pulses cause an increase in the resistance of the device whereas negative bias does the opposite. The device exhibits a gradual rate of increase (or decrease) to the given stimulus which is captured accurately from the model (solid coloured line on Fig. 3) as described by the equations above. Based on the captured model, fig. 4 displays simulated device responses based on fixed amplitude (top) and varied amplitude (bottom) stimulations for three different pulse widths (1–100 μs). Whilst a typical voltage source is assumed throughout these examples, we note that this methodology extends to include alternative

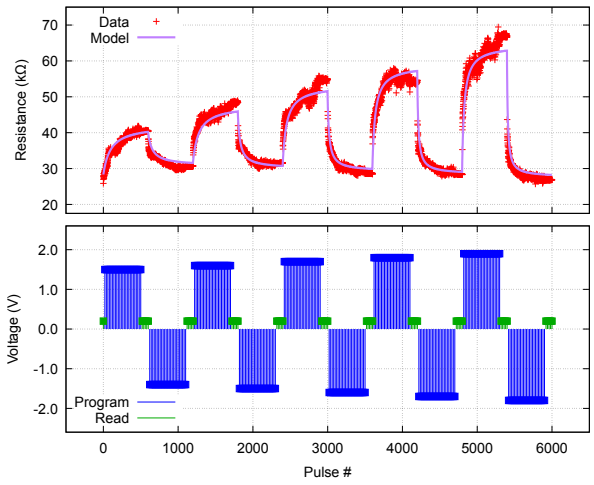


Fig. 3. Analogue switching (top) of a Pt/Al_2O_3/TiO_2/Pt RRAM device with respect to an applied stimulus (bottom). Read-outs (ie. samplings of the resistive state) are interspersed between programming phases. The modelled stimulus for the same input is noted with a solid coloured line (top). A number of 500 programming pulses were used to elicit this kind of response from the device. A set of 100 reading pulses was added between programming phases to assert stability of the current resistive state.

sources to reflect the need of distinct input waveforms as required by different applications.

Translation of the model into a more systems-specific Verilog-A code can be done directly as discussed in [15] using the domain integrator operator `idt()`. To avoid the discontinuities of the piecewise function these can be reshaped using the sigmoid function

$$\theta(x) = \left(1 + \exp\left(-\frac{x}{b}\right)\right)^{-1} \tag{7}$$

where b is a hardcoded parameter depending on the function that equation 7 is used to shape. For equation 5, $b = 10^{-6}$ whereas for equation 6, $b = 10^{-3}$. Finally to account for the steepness of the exponential in equation 7 the `limexp()` operator is used to bound numerical overflows.

Accurately capturing the transient behaviour of the device under a specific set of stimuli is definitely the primary characteristic of a device model. However there are also issues that need to be considered such as volatile effects [42], thermal static characteristics, noise at rest [43] or during programming [44] as well as variability issues. Above all, an important aspect of the process lies in assessing the parasitic elements involved, typically parasitic resistances and capacitances. Early in the design cycle, those can be estimated based on fundamental equations of conductance/capacitance and knowledge of device geometry at the device level (initial values to directly plug into the behavioural device model) [45], [46]. All of these add additional requirements to device models if they are to emulate all facets of the device response. Integration of the device model, in a TCAD workflow requires certain considerations in order to accommodate the discrete nature of a typical simulation scenario. Providing a device does not exhibit any appreciable drift resulting in unexpected volatility, an arbitrary programming pulse of duration T can be adequately approximated as a short pulse sequence corresponding to the

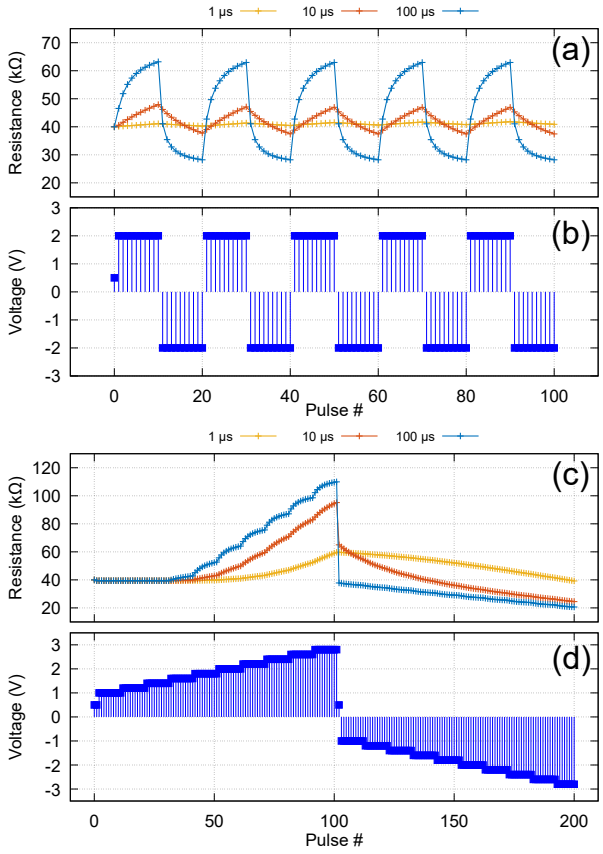


Fig. 4. Behavioural model response based on the response of the device shown in fig. 3. (top) Dependence of the modelled response on applied pulse width for constant amplitude pulses and (bottom) for programming voltage ramps. Batches of 10 programming and read-out pulses were used throughout. Traces (a, c) indicate the response of the modelled device for two different input waveforms (b, d) at three different pulse widths (1, 10 and 100 μs).

discrete time step of the simulator, t_s. In effect this process discretises the input waveform.

$$T = \sum_{i=0}^{k} t_s \tag{8}$$

This method ensures that the total amount of energy used to actuate the device remains the same regardless of the way the pulse is applied.

III. Integration of Memristor Model into Cadence

After stating the behaviour and the characteristics of the model in the previous section, a Verilog-A model representing the memristor behaviour will be explicated in this section. Then, this work will go through the detailed electrical design process under a 0.18μm CMOS technology that integrates the proposed Verilog-A model into an ASIC design workflow in Cadence Virtuoso using Spectre as a simulator. The simulation applies determined pulses to measure switching behaviours of the proposed model, followed by result analysis that contains calibration and validation.

A. Verilog-A Memristor Model

Verilog-A, a hardware description language, is widely used in semiconductor industries due to the simplicity and flexibility

in executing it with circuit simulators, including Spectre, HSPICE, Eldo, ADM and others [47]. It can be utilised to represent the behaviour of memristive devices through physical equations, laying the foundations to enable the inclusion of these devices into integrated circuits. This subsection focuses on the specific Verilog-A memristor model (in the range of $20k\Omega$ to $120k\Omega$) that uses quadratic fitting as proposed in [14]. We will go through the model in terms of the main concept of derivation, and the processing mode which makes it suitable for fast and large-scale simulations. The Verilog-A code, especially for the procedure and significant parameters, will be explained before providing users with approaches and restrictions when applying the proposed model.

As the Differential Algebraic Equation (DAE) set of the model (equations 3-6) has been explained in the previous section, the behaviour of the device can be obtained by applying a stimulus at a specific resistive state. Besides, the 'absolute threshold' function in equation 9 shows that the applied bias voltage sets the threshold/boundary of the RS. It inspired us to convert the DAE set to RS time-response equations analytically under constant bias voltage as shown in equation 10. This analytical solution for switching dynamic of memristor is also utilised in [48] Parameters in these equations are fully provided in [14].

The complex DAE set was analytically integrated to derive closed-form (implicit) formulas for the time evolution of the device resistance to the class of DC inputs of positive (negative) polarity

$$r(v) = \begin{cases} r_p(v) = r_{p,0} + r_{p,1}v, & v > 0 \\ r_n(v) = r_{n,0} + r_{n,1}v, & v \leq 0 \end{cases} \tag{9}$$

where $r_{p,0}$, $r_{p,1}$, $r_{n,0}$ and $r_{n,1}$ are fitting parameters extracted from physical device. This equation interprets the RS boundaries : under a positive bias voltage V_b and any RS below $r_p(V_b)$, the device can be pushed maximum to $r_p(V_b)$ and reach the saturation. Similarly, it is suitable for the negative bias voltage to reach $r_n(V_b)$ limit.

$$R(t)|_{V_b} = \begin{cases} \dfrac{R_0 + s_p(V_b)r_p(V_b)(r_p(V_b) - R_0)t}{1 + s_p(V_b)(r_p(V_b) - R_0)t} & for \quad V_b > 0 \& R < r_p(V_b) \\ \dfrac{R_0 + s_n(V_b)r_n(V_b)(r_n(V_b) - R_0)t}{1 + s_n(V_b)(r_n(V_b) - R_0)t} & for \quad V_b \leq 0 \& R > r_n(V_b) \\ R_0 & \text{else} \end{cases} \tag{10}$$

Equation 10 illustrates that the changes in initial RS (R_0) dependents on the bias voltage (V_b) in a fixed pulse duration (t), with the combination of switching sensitivity (equation 5) and window function (equation 6). Parameters $s_{p,n}$ refer to the $v > 0$ and $v < 0$ branch of equation 5 respectively.

The operation of the Verilog-A model presenting the $I - V$ characteristic will be divided into two steps which will be repeated multiple times: 1) program and keep tracking the last RS from initial RS under the applied voltage stimulus according to equation 10; 2) update the current flowing through the device based on the last RS through equation 3. The concept of breaking the DAE into two parts in Verilog-A model avoids integration of two variables: RS and bias voltage, which speeds up its execution in large-scale simulations. With the aid

Listing 1: the Verilog-A memristor model representing the in-house fabricated Pt/TiOx/Pt device using quadratic fitting in the range of 20-120kΩ

```
1.  module analytical (p, n);
2.      inout p, n;
3.  electrical p, n;
4.      parameter real Ap = 0.12340, An = -0.33000;
5.      parameter real tp = 2.74111, tn = 2.59685;
6.      parameter real rp0 = -40928.13784, rp1 = 55117.97865;
7.      parameter real rn0 = 41366.35820, rn1 = 7789.66771;
8.      parameter real Rinit = 40000;
9.      parameter real eta = 1;
10.     parameter real ap=0.225, bp=4.12;
11.     parameter real an=0.2801, bn=4.10;
12.     real Rmp, Rmn, svp, svn, vin, RS, IVp, IVn, IV;
13. real first_iteration, R0_last, dt, it;
14. analog function integer stp;
15.     real arg; input arg;
16.     stp = (arg >= 0 ? 1 : 0 );
17. endfunction
18. analog begin
19.     if (first_iteration==0) begin
20.     it=0;       R0_last=Rinit;
21.     end
22.     dt=$abstime-it;
23.     vin=V(p,n);
24.     Rmp=rp0+rp1*vin;    Rmn=rn0+rn1*vin;
25.     if (vin>0) begin
26.     svp=Ap*(-1+exp(abs(vin)/tp));
27.     RS=(R0_last+svp*Rmp*(Rmp-R0_last)*dt)/(1+svp*(Rmp-
        R0_last)*dt);
28.     end
29.     else begin
30.     svn=An*(-1+exp(abs(vin)/tn));
31.     RS=(R0_last+svn*Rmn*(Rmn-   R0_last)*dt)/(1+svn*(Rmn-
        R0_last)*dt);
32.     end
33.     if (RS>=Rmp && vin>0)                   RS=R0_last;
34.     if (RS<=Rmn && vin<0)                   RS=R0_last;
35.     if (abs(vin)<=0.5)                      RS=R0_last;
36.     IVp=ap*(1/RS)*sinh(bp*vin);
37.     IVn=an*(1/RS)*sinh(bn*vin);
38.     IV=IVp*stp(vin)+IVn*stp(-vin);
39.     I(p, n)<+ IV;
40.     R0_last=RS;
41.     first_iteration=1;
42.     it=$abstime;
43. end
44. endmodule
```

of the equations described in [14], our Verilog-A memristor model (for the in-house fabricated $Pt/TiOx/Pt$ device) was implemented as presented in Listing 1. The Verilog-A model is the simple and understandable combination of the equations in section II. The structure is organised as follows:

- Defining p, n as 'inout' ports, from where the bias voltage can be calculated and the current flowing through the ports, will be the output. The transient RS can be obtained (from lines 1-3, 23, 36-39), by dividing the applied voltage and the current across the device.
- Defining the fitting parameters (switching sensitivity, window function and I-V relationship parameters), switching direction parameter and initial RS (lines 4 to 11). Among these, fitting parameters are extracted from the experimental results whilst using our in-house fabricated device by applying multiple voltage levels. More details about the extraction can be found in [14]. The switching direction parameter, eta, can be defined as 1 or -1, depending on the desirable direction. The initial RS can be set within a proper RS range that is determined

by the absolute threshold function (line 24).

- In equation 3, two branches of current were derived depending on the polarity of the bias voltage (positive/negative), however, it will be calculated without recognising the polarity of the bias voltage (lines 36-37). Therefore, a 'step function' was defined as a multiplexer (lines 14 to 17) in order to choose the proper current branch as output in line 38. When 'Vin' is positive, the function 'stp(Vin)' is one and the current branch 'IVp' will be the output, whereas the negative bias voltage induces opposite results.
- Lines 19 to 21, assign the parameter initial RS to the latest RS that can be used for calculation in the first iteration. The '$abstime' represents the absolute time that has elapsed from the beginning of the simulation to the present time. The 'it' is defined as reference time that locates the '$abstime' of each iteration. The time step is obtained from line 22 by subtracting the absolute time from the reference time to help track each time-step duration. After one iteration, the reference time 'it' will be assigned by the previous absolute time '$abstime' in line 42.
- The boundaries of RS dependent on the bias voltage based on the chosen window function is calculated in line 24, according to equation 9. Line 35 sets the constraint, that below $0.5V$ the bias voltage will not induce switching which has been set as a read voltage (details in III-B2).
- Lines 25 to 35 calculates RS under the constant bias voltage condition, including situations when the operation is within or exceeds the boundaries of the window function.
- After deriving the RS at a specific voltage, the current will be updated (equation 3) and passed to ports (line 39). Finally, the initial RS will be updated for the next iteration (lines 40 to 42).

This is the data-driven model that we obtained by applying multiple voltages on the device based on the parameter extraction algorithm, of which more information can be found in [14]. At this stage, we present RS ranges that have been proved to fit our physical model with a low root mean square (RMS) error of 2.89%. Therefore, users are supposed to identify the operational RS range, which help them to set the initial RS (R_{init} in Listing 1, line 8). Besides the restriction of RS range, limits the applied bias voltage. For positive voltage, only when satisfying both $Vin > 0$ and $RS < Rmp$ can induce switching. Combining lines 6 and 24-28, the positive voltage that is above $0.5V$ at least can trigger switching in this model. Users are suggested to do a rough calculation before setting the initial RS and bias voltage. If users set these parameters beyond the 'window function' (equation 9), the simulation can still be completed but keep the RS as initial setting stage.

The simulation setup includes two stages:1) read RS through applying $0.5V$ triangular pulse, which prevent the device from switching; 2) write, or change RS, by applying pulses with defined duration/width, amplitude, polarity and numbers of pulse. This can be utilised to: 1) conduct static current-voltage measurement at specific RS within boundaries of the state variable. 2) gather transient switching characterisation by applying different bias voltages on defined initial RSs. Since the model only outputs the current responding to the applied voltage, the current can be obtained directly from the simulation results. Whilst, the RS needs to be calculated through the equation 3. For extension use, fitting parameters can be changed to represent other memristive devices. At this stage, the model does not incorporate AC analysis/small signal modelling, noise performance, parasitics and device variation.

B. Process Flow for Electrical Design

The analogue and mixed-signal system design flow is presented in Fig. 5 including both electrical and physical design steps. Before designing circuits with memristor, model integration into Cadence is an important task which will be described into detail in this section, whilst hybrid CMOS/memristor circuits and physical design is presented in part II.

1) Import verilog-A model and build the symbol: In order to utilise the device in schematic design as well as in simulation, a symbol was created that links to the Verilog-A model. This enables the device to be added in a schematic for a design, or for a testbench to be simulated. The instructions to enable this are as follows:

- Creating a library and refer to the chosen technology. Before integrating memristor with CMOS, users are supposed to be familiar with the behaviour of the memristor quantitatively such as the range of high/low resistive state, the allowed range of applied voltage/current, static I-V characteristic, etc. This allows the user to choose a more suitable memristor technology, as well as define circuit performance. A quantitative analysis is provided in section III-C to provide users with a template whilst using the operational range of our device, as well as measurement methodology.
- A cell view is created in a specific library with Verilog-A type, named 'memristor' in library 'DesignMethodology' (Fig. 6(a)) where Verilog-A code is scripted.
- A symbol is created from the Verilog-A memristor model (Fig. 6(c)).
- The position of ports are assigned at left and right sides which can be automatically detected from the Verilog-A code, followed by the symbol design (Fig. 6(b)).

2) Integrate model and setup simulation: After creating a symbol, the memritive device is ready to be applied into circuit visually, whose parameters can be changed to represent other models (in Fig. 6(d)). However, a read voltage below the threshold is required to measure the RS of the memristor indirectly. Thus, to trace the changing RS appropriately, the recommended testbench and stimulus is given below.

Considering that our Verilog-A memristor model calculates RS against time and keeps tracking the change of RS, the model run in transient simulation to process both write and read operation. In this case, we took a single memristor as a simulation example shown in Fig. 7. A chain of pulses was applied across it followed by a triangular wave as shown in Fig. 8 to conduct write and read of memristor respectively.

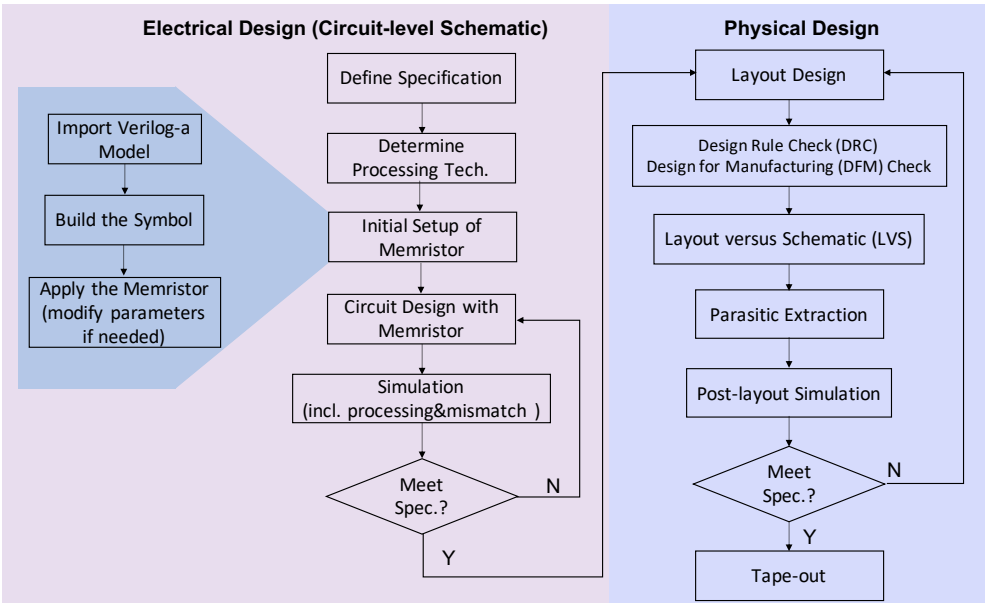


Fig. 5. Design Flow for analogue and mixed-signal systems. The design flow is divided into two parts: electrical and physical designs. The electrical design has a step-by-step methodology showing integration of model into Cadence environment and later designing hybrid CMOS/memristor circuits.

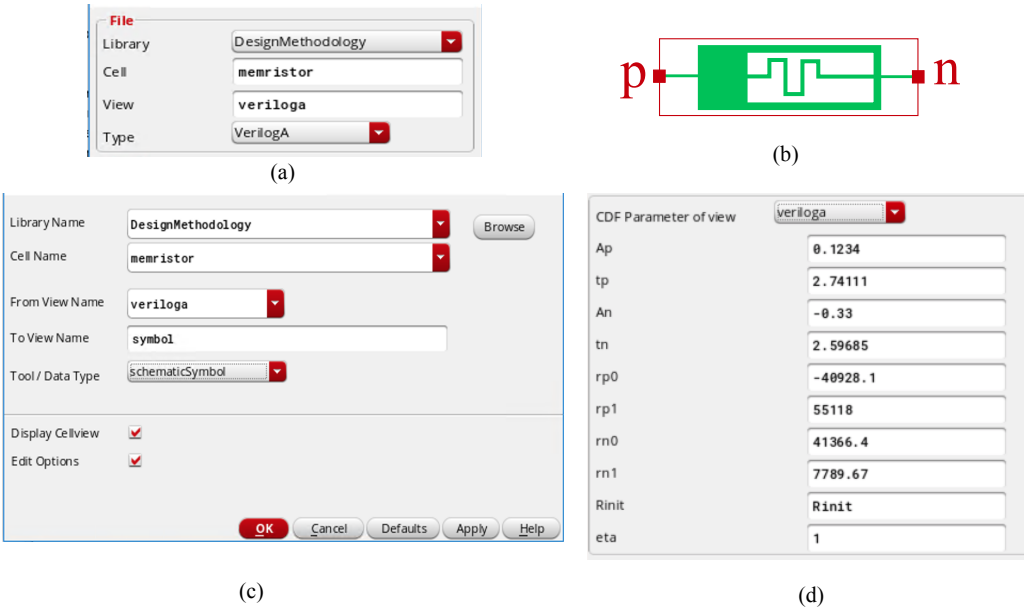


Fig. 6. Operation sequence of importing memristor into Cadence. (a) Building a new cell view in Verilog-A type for memristor. (b) Symbol of memristor. (c) Creating memristor symbol from Verilog-A cell view. (d) Parameters of memristor can be modified in the object properties window.

These are the basic setup before the calibration and validation analysis. The operation steps are as follows:

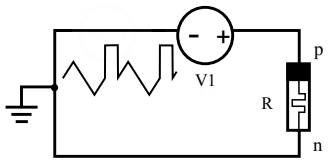


Fig. 7. Testbench of applying pulse and triangular wave to memristor to conduct write and read operations in transient simulation.

- The testbench consists of a schematic having one memristor, a single piece-wise linear voltage source (PWL) and a square pulse voltage source (Fig. 7). In the schematic, the direction that voltage sources are connected to the 'positive' (p) port of the memristor is defined. The positive bias voltage ($V_b > 0$) from 'p' excites memristor to a higher RS, while the RS decreases when positive voltage applies at 'n' port. In the testbench (Fig. 7), the proposed model is in OFF transition when stimulated with a positive voltage, while a negative voltage results in ON transition. The two voltage sources generate triangular waves and square pulses alternately to process the change and tracking in RS. The detail of the input

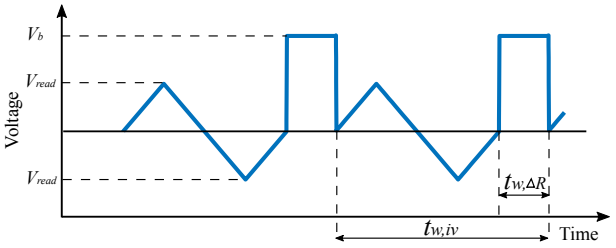


Fig. 8. Input pulse that programmes the memristor. The input signal sequence can be divided into two parts: triangular wave and pulse. For all the simulation of our device, the read voltage is defined as $V_{read} = 0.5V$ with $1ms$ duration. The pulses can be determined with specific duration/width ($t_{w,\Delta R}$), amplitude (V_b) and numbers of pulses to provoke the memristor. With the combination of two stages, the RS of memristor model can be tracked for each stimulus within $t_{w,iv}$, where $t_{w,iv} = 1$ ms $+ t_{w,\Delta R}$.

signals have been shown in Fig. 8. In this measurement for the proposed model, the identical triangular wave is defined as $0.5V$ peak amplitude (V_{read}) within $1ms$ duration. The square pulse mainly has three variables: width ($t_{w,\Delta R}$), amplitude (V_b) and the number of square pulses. The detailed simulation and evaluation of these effects on memristor will be given in section III-C with classification.

- After setting up the testbench and running the transient simulation, the change of RS is recorded. In this case, each current across the memristor at $V_{read} = 0.5V$ is recorded. With both read voltage and current, the RS can be obtained through division.

C. Calibration and Validation

The performance of the Verilog-A model is validated (see Fig. 9) with physical behaviour extraction done in section II, proving that the proposed model can represent the device finely. Then, we program the proposed model by modulating number of pulse, pulse width and amplitude in order to explore both the qualitative and quantitative impacts on model. To be consistent, all simulations will start from the same initial RS as baseline. Recommendation of programming the memristor will be given after evaluating above modulations.

1) Validation: The simulation data extracted from our device is presented in Fig. 4 that contains response based on pulses with different widths for constant and increasing/decreasing pulse amplitude. The results in Fig. 4 are obtained from python model, while Fig. 9 presents the Verilog-A simulation results. Both python and Verilog-A models use the same parameters and simulation processes (iteration), thus, simulation results from these two are consistent. They have different applications: python model is used on our instrument measurement interface and Verilog-A model can be used in circuit design. As shown in Listing 1, the RS is relative to timestep (dt) during simulation. In order to fit the Verilog-A model with the physical behaviour finely, the timestep in device extraction and Spectre simulation are taken same, i.e. $1\mu s$

2) Number of pulse modulation: The positive bias voltages: $1.5V$, $1.8V$ and $2.0V$ are applied to the Verilog-A memristor model to explore the behaviour when memristor is

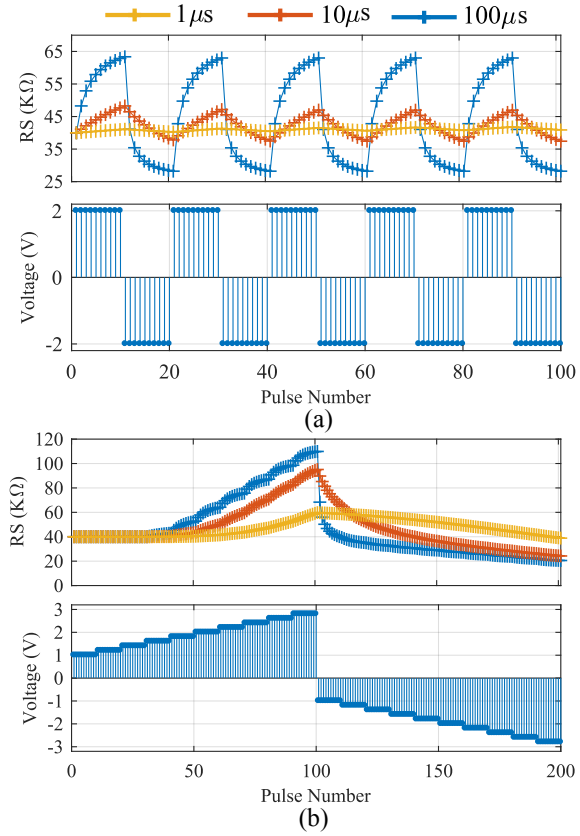


Fig. 9. Simulation results of Verilog-A memristor model in Cadence. For validation, the model in Cadence is programme by the same input signals with the one in Fig. 4 with three pulse widths. (a) shows the model response to constant amplitude pulses with $1, 10, 100\mu s$ pulse widths. (b) presents the model response to the voltage ramp.

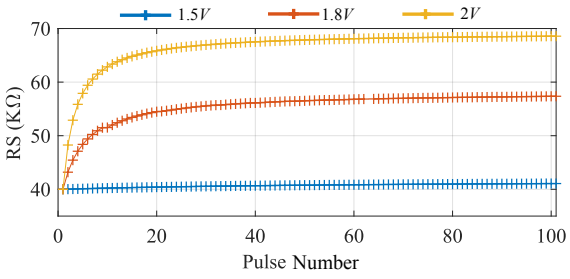


Fig. 10. Verilog-A memristor model response based on the pulse number. The device is programmed for 100 pulses with different voltages starting from initial $RS = 40k\Omega$, and it eventually saturates at different RSs. As the pulses keep increasing, the rate of change of RS slows down and it gradually saturate. Characterisation routine parameters based on the stimulus in Fig. 8: $t_{w,\Delta R} = 100\mu s$, $t_{w,iv} = 1.1ms$, $V_b = 1.5/1.8/2.0V$, and $V_{read} = 0.5V$

integrated into a CMOS circuit with a nominal supply voltage of $1.8V$. Meanwhile, $2V$ bias voltage is applied as a typical programming voltage. In our model, the upper limit boundary is dependent on the bias voltage for the same initial RS (refer to Listing 1, lines 6, 23 and 24). Thus, we can explore the phenomenon when sufficient number of continuous pulses for three bias voltages are applied.

Positive amplitude voltage-based pulses can push the RS to specific upper constant plateaus (see Fig. 10), meanwhile, negative bias voltages can programme the RS to corresponding

lower constant plateaus in Fig. 14(b). From Fig. 10, the rate of change of RS reduces gradually and RS eventually saturate. At the beginning of the transient, RS changes rapidly and the desired state might be programmed with less accuracy. While a specific RS can be achieved by increasing the number of pulses that helps to program the model in an accurate resistive state. As we are exploring the number of pulse modulation, the effect from amplitude will be omitted temporarily. The only discussion relative to amplitude in this subsection is to prove that a sufficient number of pulses can help discover the highest RS for specific bias voltage.

- By applying sufficient pulses to the Verilog-A model, it can provide users with access to measure the boundary of RS for a specific bias voltage. It helps users to evaluate whether the device can be applied under specific requirements and also helps to define operational voltage on the schematic.
- Within RS boundary, a sufficient number of pulse of different bias voltage can achieve the same desired RS. For instance, RS climbs from $49k\Omega$ to $53k\Omega$ by a pulse of $2V$, thus, users need to use $1.8V$ pulses to obtain $50k\Omega$. This induces a trade-off between amplitude and time.
- The number of pulses needs to be considered during memristor programming since more number of pulses helps to obtain a specific RS.

3) Pulse width modulation: Exploration of the pulse width effects on programming the memristive devices is demonstrated in this subsection, where we generate pulses at $|1.8V|$, for $1\mu s$, $10\mu s$ and $100\mu s$ pulse width. In Fig. 11, three pulse trains with 100 pulses are generated and its equivalent resistive states RS are recorded. Besides, negative voltage is also employed on the device to flush RS back to its initial state, proving that the modulation can be applied in both directions.

Fig. 12 presents pulse width modulation of positive voltage. The RS programmed by $100\mu s$ pulse train achieves to the constant plateau in higher speed compared with other stimulus in Fig. 12. The Fig. 12 indicates that the pulse with smaller

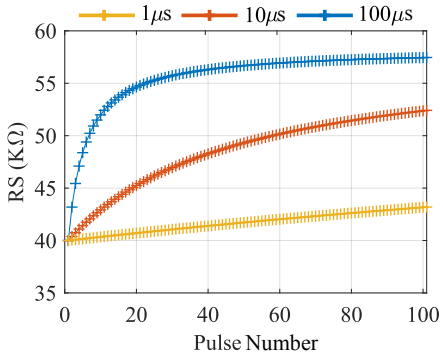


Fig. 12. Verilog-A memristor model response for different pulse width. Starting from initial $RS = 40k\Omega$, the rate of change of resistive states differ for different duration. The $100\mu s$ pulses induce that RS increases in a faster rate and generates a lot of states within 20 pulses which is less recognisable compared with $1\mu s$ pulses. However, the rate of change is slowed down as the number of pulse increases. It illustrates that the shorter duration pulse helps generate specific RS with higher resolution.

width can be applied to slow down the changing rate of RS, which contributes to programming the device to a specific resistive state more accurately in application. When programming the device, the approach that increases/decreases the RS is appreciate at a lower speed. It allows users to program the device slowly to the desired RS with higher accuracy.

4) Amplitude and polarity modulations: To evaluate the amplitude and polarity modulation, a stimulus containing bias voltages at $|1.5V|$, $|1.8V|$ and $|2.0V|$ in two polarities are generated. A few trains with fixed number of pulses, composed of 100 pulses each, were applied on the device. The negative pulses are applied to push the device to the initial state (see Fig. 13), which achieves the same RS change with less pulses.

The higher absolute bias voltage can programme the device to the saturated state/constant plateau in higher speed. If users intend to programme the device to the saturated level, the utilisation of appropriate high amplitude voltage is appreciated. But if users target the RS between the upper and lower

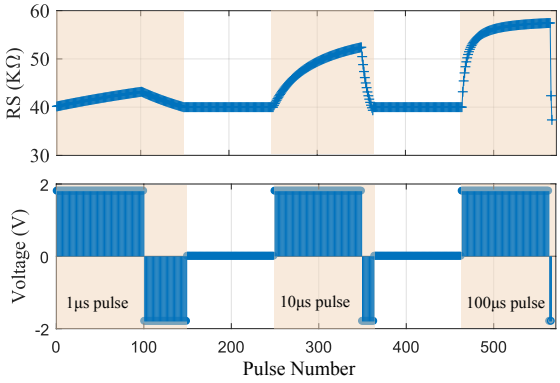


Fig. 11. Programming the device for varying pulse width. Bottom trace: 100 pulses each at $1.8V$ for varying pulse width are applied. In between the measurement, the inverse voltages are applied to flush the device to initial RS. Top trace: shows the modulation results corresponding to the bottom stimulus. The RSs modulated by positive bias voltage are presented in Fig. 12 for comparison. Characterisation routine parameters: $t_{w,\Delta R} = 1/10/100\mu s$, $t_{w,iv} = 1.001/1.01/1.1ms$, $V_b = |1.8V|$, and $V_{read} = 0.5V$.

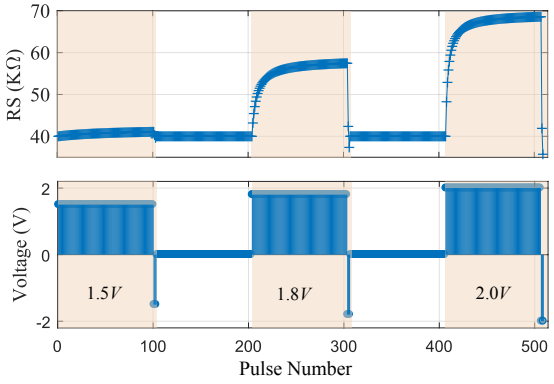


Fig. 13. Programming the device for pulses with different amplitudes. Bottom trace: Each number of 100 pulse (duration=$100\mu s$) under incremental bias voltage are employed to modulate device RS. In between the measurement, inverse voltages help flush the device back to initial state. The RS modulated by positive bias voltage has been highlighted and will be shown in Fig. 14 for comparison. Characterisation routine parameters: $t_{w,\Delta R} = 100\mu s$, $t_{w,iv} = 1.1ms$, $V_b = |1.5/1.8/2.0V|$, and $V_{read} = 0.5V$.

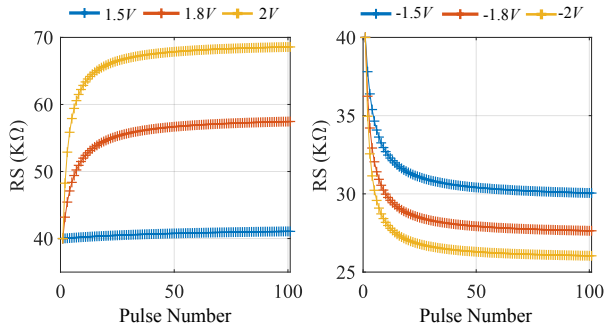


Fig. 14. Verilog-A memristor model response stimulated for different amplitude voltages, both for positive (left) and negative (right) bias voltages . It can be seen that the higher absolute voltage leads to faster changing rate of RS. Combining two sub-figures, positive voltages induces lower changing rate of RS, compared with negative voltages. The right figure also indicates that the saturated RS (boundary of lower limit) is dependent on the bias voltage, where a more negative voltage can push RS to a lower state.

constant plateau, they can select appropriate low voltage to program the memristor to a specific RS level with higher accuracy in lower speed. From Fig. 13, it can be found that to reach the same RS level, the number of pulses required for negative voltage is much less than positive one. In Fig. 14, negative voltages induce considerable RS drop (compared with positive voltage stimulus), where the RS cannot be discerned. It is because the device is in OFF transition under the positive voltage, while the negative bias voltage leads to ON transition that performs faster. It also indicates that negative bias voltage is not suitable to determine a specific RS between constant plateau due to the rapid change.

Combining the simulation results above, some recommendations for application have been concluded:

- Applying specific bias voltages with sufficient pulses to explore the operational RS range in simulator at the beginning. The resulting data containing the range of bias voltage, boundary of RS and programming time, provides evidence on whether it can be utilised on the specific schematic and how to apply it.
- To program the RS to a higher level, the positive bias voltage can be applied directly. But there might be two approaches to program the memristor to a lower RS: 1) Programming the memristor with negative voltage directly. The negative amplitude voltage-based pulse train drops the resistance drastically to constant plateau (saturated level). It makes the programming procedure demanding to achieve desired RS above constant plateau. 2) It can be achieved by two steps: i) apply the negative pulses to drop the RS to constant plateau within a comparatively-short time frame, then ii) apply the positive voltage-based pulses to increase the resistance to the desired highly-resistive state. This is a more practical approach with multiple RS levels being achieved with higher resolution.
- To program the device with high resolution, it is recommended to employ a positive bias voltage with a low amplitude and a small pulse width to programme the device at a reduced rate.

IV. Conclusions

Over the past decade several implementations of memristor models and memristor based circuits have come across, yet, a design flow that is versatile and easily integrable in EDA toolchain is the necessity for the primary designers to validate circuits for applications that still have not been researched.

In summary, we have presented a design flow for integrating the behavioral model into the Cadence design environment. To begin with, this Part I demonstrated the in-house fabricated memristor model for different input stimuli. Then a Verilog-A for the physical model is written and a step-by-step integration flow for electrical simulation and verification for the same set of stimuli is shown in Cadence. Once the model integration and verification is done, the designers can integrate it with CMOS technology to build hybrid CMOS/memristor systems.

As memristive technologies mature the same will be true for device models that will be gradually accounting for device variability and nonidealities. This will add functionality to the proposed methodology, impacting our analysis to run montecarlo and effectively allow us to develop variation aware memristive circuit.

Acknowledgment

We acknowledge the support of EPSRC Programme Grant FORTE (EP/R024642/1), H2020-FETPROACT-2018-01 SYNCH and the RAEng Chair in Emerging Technologies (CiET1819/2/93). We also wish to thank Mr Jiawei Shen for his invaluable contribution to the design of integrated circuits, which employ and validate the design methodology presented here.

References

[1] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[2] S. Stathopoulos *et al.*, "Multibit memory operation of metal-oxide bi-layer memristors," *Scitific Report*, vol. 7, p. 17532, 2017.

[3] L. Gao, F. Alibart, and D. B. Strukov, "Programmable cmos/memristor threshold logic," *IEEE Transactions on Nanotechnology*, vol. 12, no. 2, pp. 115–119, 2013.

[4] A. Serb, A. Khiat, and T. Prodromakis, "Seamlessly fused digital-analogue reconfigurable computing using memristors," *Nature Communication*, vol. 9, no. 2170, 2018.

[5] A. Ascoli, I. Messaris, R. Tetzlaff and L. Chua, "Theoretical Foundations of Memristor Cellular Nonlinear Networks: Stability Analysis with Dynamic Memristors," *IEEE Transaction on Circuits and Systems–I : Regular Papers*, vol. 67, no. 4, pp. 1389–1401, 2020.

[6] M. A. Zidan, A. Chen, G. Indiveri, and W. D. Lu, "Memristive computing devices and applications," *Journal of Electroceramics*, vol. 39, pp. 4–20, 2017.

[7] M. Laiho, E. Lehtonen, A. Russell, and P. Dudek, "Memristive synapses are becoming reality," pp. 10–12, 2010.

[8] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *nature electronics*, vol. 1, pp. 52–59, 2017.

[9] J. Mathew *et al.*, "A novel memristor-based hardware security primitive," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 3, Apr. 2015.

[10] A. Ascoli *et al.*, "On Local Activity and Edge of Chaos in a NaMLab Memristor," *Frontiers in Neuroscience*, vol. 15, 2021.

[11] A. Ascoli, R. Tetzlaff, Sung-Mo Kang and L. Chua, "Theoretical Foundations of Memristor Cellular Nonlinear Networks: a DRM2-based Method to design Memcomputers with Dynamic Memristors," *IEEE Transaction on Circuits and Systems–I : Regular Papers*, vol. 67, no. 8, pp. 2753–2766, 2020.

[12] M. Stephan, T. Stefan, W. Rainer, and V. Ilia, "Switching kinetics of electrochemical metallization memory cells," *Physical Chemistry Chemical Physics*, vol. 15, no. 18, p. 6945, 2013.
[13] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM:A general model for voltage-controlled memristors," *IEEE Trans. on Circuits and Systems II*, vol. 62, no. 8, pp. 786–790, 2015.
[14] I. Messaris *et al.*, "A tio2 reram parameter extraction method," *IEEE International Symposium on Circuits and Systems*, pp. 1–4, 2017.
[15] I. Messaris *et al.*, "A data-driven verilog-a reram model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, p. 3151–3162, Dec. 2018.
[16] E. Ambrosi, A. Bricalli, M. Laudato, and D. Ielmini, "Impact of oxide and electrode materials on the switching characteristics of oxide reram devices," *Faraday Discussions*, vol. 213, p. 87–98, 2019.
[17] I. H. Im, S. J. Kim, and H. W. Jang, "Memristive devices for new computing paradigms," *Advanced Intelligent Systems*, vol. 2, no. 11, p. 2000105, Aug. 2020.
[18] F. Zahoor, T. Z. Azni Zulkifli, and F. A. Khanday, "Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications," *Nanoscale Research Letters*, vol. 15, no. 1, Apr. 2020.
[19] L. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, p. 209–223, 1976.
[20] L. Chua, "Resistance switching memories are memristors," *Applied Physics A*, vol. 102, no. 4, p. 765–783, Jan. 2011.
[21] S. Brivio *et al.*, "Experimental study of gradual/abrupt dynamics of hfo2-based memristive devices," *Applied Phys. Letters*, vol. 109, no. 13, 2016.
[22] Y. Li *et al.*, "Ultrafast synaptic events in a chalcogenide memristor," *Scientific Reports*, vol. 3, no. 1, apr 2013.
[23] J. Shi *et al.*, "A correlated nickelate synaptic transistor," *Nature Communications*, vol. 4, no. 1, Oct. 2013.
[24] Y. van de Burgt *et al.*, "A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing," *Nature Materials*, vol. 16, no. 4, p. 414–418, Feb. 2017.
[25] A. Younis *et al.*, "High-performance nanocomposite based memristor with controlled quantum dots as charge traps," *ACS Applied Materials & Interfaces*, vol. 5, no. 6, p. 2249–2254, Mar. 2013.
[26] H. Tian *et al.*, "Extremely low operating current resistive memory based on exfoliated 2d perovskite single crystals for neuromorphic computing," *ACS Nano*, vol. 11, no. 12, p. 12247–12256, Dec. 2017.
[27] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, p. 80–83, 2008.
[28] R. Dittmann and J. P. Strachan, "Redox-based memristive devices for new computing paradigm," *APL Materials*, vol. 7, no. 11, 2019.
[29] I. Valov *et al.*, "Electrochemical metallization memories—fundamentals, applications, prospects," *Nanotechnology*, vol. 22, no. 25, 2011.
[30] X. Fong *et al.*, "Spin-transfer torque memories: Devices, circuits, and systems," *Proceedings of the IEEE*, vol. 104, no. 7, p. 1449–1488, Jul 2016.
[31] S. Lequeux *et al.*, "A magnetic synapse: multilevel spin-torque memristor with perpendicular anisotropy," *Scientific Reports*, vol. 6, no. 1, Aug 2016.
[32] T. Chang, Sung-Hyun Jo and W. Lu, "Short-Term Memory to Long-Term Memory Transition in a Nanoscale Memristor," *ACS Nano*, vol. 5, no. 9, pp. 7669–7676, 2011.
[33] C. Giotis, A. Serb, S. Stathopoulos, and T. Prodromakis, "Bidirectional Volatile Signatures of Metal-Oxide Memristors—Part II: Modeling," *IEEE Transactions on Electron Devices*, vol. 67, no. 11, pp. 5166–5173, 2020.
[34] R. Berdan *et al.*, "Emulating short-term synaptic dynamics with memristive devices," *Sci Rep*, vol. 6, no. 1, p. 18639, 2016.
[35] L. Michalas, A. Khiat, S. Stathopoulos, and T. Prodromakis, "Electrical characteristics of interfacial barriers at metal—tio2 contacts," *Journal of Physics D: Applied Physics*, vol. 51, no. 42, p. 425101, Sep. 2018.
[36] F. Karsten, *et al.*, "Uniting gradual and abrupt set processes in resistive switching oxides," *Physical Review Applied*, vol. 6, no. 6, Dec. 2016.
[37] V. A. F. *et al.*, "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 2, p. 166–174, Apr. 2015.
[38] Chua, L., "Everything You Wish to Know About Memristors But Are Afraid to Ask," *Radioengineering*, vol. 24, no. 2, p. 319–368, Jun 2015.
[39] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device SPICE model and its application in circuit design," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 32, no. 8, pp. 1201–1214, 2013.
[40] P. Themistoklis, P. B. Pin, P. Christos, and T. Christofer, "A versatile memristor model with nonlinear dopant kinetics," *IEEE Transactions on Electron Devices*, vol. 58, no. 9, p. 3099–3105, Sep. 2011.
[41] D. Vaidya *et al.*, "Compact modeling of the switching dynamics and temperature dependencies in TiOx-based memristors: Part I - Behavioural Model," *IEEE Transactions on Electron Devices*, vol. , no. , p. , In Press.
[42] I. Gupta *et al.*, "Volatility characterization for rram devices," *IEEE Electron Device Letters*, vol. 38, no. 1, p. 28–31, Jan. 2017.
[43] F. M. Puglisi *et al.*, "Random telegraph noise in resistive random access memories: Compact modeling and advanced circuit design," *IEEE Transactions on Electron Devices*, vol. 65, no. 7, p. 2964–2972, 2018.
[44] S. Stathopoulos *et al.*, "A memristive switching uncertainty model," *IEEE Trans. on Electron Devices*, vol. 66, no. 7, p. 2946–2953, 2019.
[45] A. Serb, W. Redman-White, C. Papavassiliou and T. Prodromakis, " Practical Determination of Individual Element Resistive States in Selectorless RRAM Arrays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 6, pp. 827–835, 2016.
[46] A. Chen, "A Comprehensive Crossbar Array Model With Solutions for Line Resistance and Nonlinear Device Characteristics," *IEEE Transactions on Electron Devices*, vol. 60, no. 4, p. 1318–1326, 2013.
[47] G. J. Coram, "How to (and how not to) write a compact model in verilog-a," pp. 97–106, 2004.
[48] V. Ntinas, A. Ascoli, R. Tetzlaff, and G. Ch Sirakoulis, "A complete analytical solution for the on and off dynamic equations of a TaO memristor," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 4, pp. 682–686, 2018.

APPENDIX

A. Verilog-A Memristor Model using Exponential Fitting

The Verilog-A memristor model proposed in [15] utilises exponential fitting. Compared with the quadratic model in Listing 1, the model in Listing 2 uses difference DAE set which have been shown in Eq. 11-16.. We utilised the same methodology to obtain the quadratic and exponential models and both of them can be utilised to simulate our device. Since the main derivation concept is the same as the one in section III, we only go through some differences in terms of equations, parameter and processing step of this model.

The applied DAE set and derived analytically equation shows in the following:

$$i(R,v) = \begin{cases} a_p(1/R)\sinh(b_p v) & v \geq 0 \\ a_n(1/R)\sinh(b_n v) & v < 0 \end{cases} \tag{11}$$

$$\frac{dR}{dt} = g(R,v) = s(v)\cdot f(R,v) \tag{12}$$

$$s(v) = \begin{cases} A_p(-1+\exp(t_p|v|)) & v > 0 \\ A_n(-1+\exp(t_n|v|)) & v < 0 \\ 0 & v = 0 \end{cases} \tag{13}$$

$$f(R,v) = \begin{cases} -1+\exp\left[\eta k_p(r_p(v)-R)\right] & R < \eta r_p(v) \quad v > 0 \\ -1+\exp\left[\eta k_n(R-r_n(v))\right] & R > \eta r_p(v) \quad v < 0 \\ 0 & v = 0 \end{cases} \tag{14}$$

with $s(v)$ being the switching sensitivity function, $f(R,v)$ the window function and convert the DAE set to RS time-response equations analytically under constant bias voltage.

$$R(t)|_{V_b} = \frac{ln(e^{\eta k_p r_p(V_b)} + e^{-\eta k_p s_p(V_b)t} \times (e^{\eta k_p R_0} - e^{\eta k_p r_p(V_b)}))}{k_p} \tag{15}$$
$$for \quad V_b > 0 \quad \& \quad R < \eta r_p(V_b)$$

Listing 2: the Verilog-A memristor model representing the the in-house fabricated Pt/TiOx/Pt device using exponential fitting in the range of 10-17kΩ

```
1.  module analytical (p, n);
2.      inout p, n;
3.  electrical p, n;
4.      parameter real Ap = 743.47, An = -68012.28374;
5.      parameter real tp = 6.51, tn = 0.31645
6.      parameter real kp = 5.11e-4, kn = 1.17e-3;
7.      parameter real rp0 = 16719, rp1 = 0;
8.      parameter real rn0 = 29304.82557, rn1 = 23692.77225;
9.      parameter real Rinit = 16250;
10.     parameter real eta = 1;
11.     parameter real ap=0.24, bp=3;
12.     parameter real an=0.24, bn=3;
13. real Rmp, Rmn, vin, RS, IVp, IVn, IV;
14. real first_iteration, R0_last, dt, it;
15. analog function integer stp;
16.     real arg; input arg;
17.     stp = (arg >= 0 ? 1 : 0 );
18. endfunction
19. analog begin
20.     if (first_iteration==0) begin
21.     it=0;
22.     R0_last=Rinit;
23.     end
24.     dt=$abstime-it;
25.     vin=V(p,n);
26.     Rmp=rp0+rp1*vin;          Rmn=rn0+rn1*vin;
27.     if (vin>0)
28.     RS=(1/kp)*ln(exp(eta*kp*Rmp)+exp(-eta*kp*(Ap*(-
        1+exp(eta*tp*abs(vin))))*dt)*(exp(eta*kp*R0_last)-
        exp(eta*kp*Rmp)));
29.     else
30.     RS=-(1/kn)*ln(exp(-eta*kn*R0_last+eta*kn*(An*(-
        1+exp(tn*abs(vin))))*dt)-exp(-eta*kn*Rmn)*(-1+exp
        (eta*kn*(An*(-1+exp(tn*abs(vin))))*dt)));
31.     if (RS>=Rmp && vin>0)      RS=R0_last;
32.     if (RS<=Rmn && vin<0)      RS=R0_last;
33.     IVp=ap*(1/RS)*sinh(bp*vin);
34.     IVn=an*(1/RS)*sinh(bn*vin);
35.     IV=IVp*stp(vin)+IVn*stp(-vin);
36.     I(p, n)<+ IV;
37.     R0_last=RS;
38.     first_iteration=1;
39.     it=$abstime;
40. end
41. endmodule
```

$$R(t)|_{V_b} = \frac{ln(e^{-\eta k_n R_0 + \eta k_n s_n(V_b)t}) - e^{-\eta k_n r_n(V_b)} \times (-1 + e^{\eta k_n s_n(V_b)t})}{k_n}$$
$$for \quad V_b < 0 \quad \& \quad R > \eta r_n(V_b) \tag{16}$$

else R_0

With the aid of equations described in [15], the Verilog-A memristor model (the in-house fabricated $Pt/TiOx/Pt$ device under $10-17k\Omega$ RS range) was implemented as presented in Listing 2. Note that the parameter rp1 = 0, which means that the positive boundary is fixed to $16.719k\Omega$ (lines 7 and 26), while the lower boundary is still dependent on initial RS as well as the bias voltage (lines 8 and 26). Table I presents the fitting parameters of the exponential models with the RS ranges of $[4.5k\Omega,\ 6.0k\Omega]$ and $[10k\Omega,\ 17k\Omega]$ respectively. And for demonstration, we use the model in Listing 2 to exploit impacts from different types of stimulus, including pulse number, pulse width and amplitude.

TABLE I
PARAMETER VALUES THAT FIT THE $Pt/TiOx/Pt$ MEMRISTOR IN TWO RS RANGES.

Parameters	Range 1: $4.5-6.0k\Omega$	Range 2: $10-17k\Omega$
A_p	0.12	743.47
A_n	-79.03	-6.8×10^4
t_p	0.59	6.51
t_n	1.12	0.31
k_p	8.10×10^{-3}	5.11×10^{-4}
k_n	9.43×10^{-3}	1.17×10^{-3}
r_{p0}	3085	16.71×10^3
r_{p1}	1862	0
r_{p2}	0	0
r_{n0}	5193	29.30×10^3
r_{n1}	378	23.69×10^3
r_{n2}	0	0
$a_{p,n}$	0.24	0.24
$b_{p,n}$	2.81	2.81

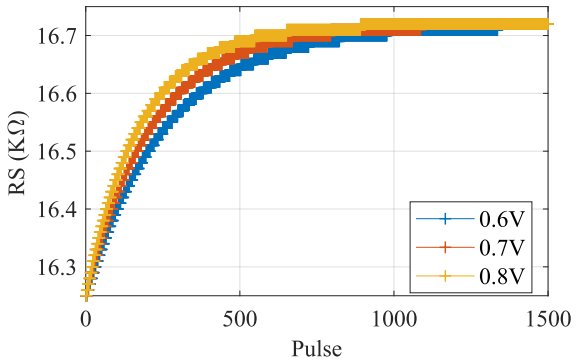


Fig. 15. The device is programmed by 1500 pulses with different voltages starting from initial $RS = 16.25k\Omega$, and it eventually saturate at $RS = 16.71k\Omega$. As the pulses keep increasing, the changing rate of RS slows down and it gradually saturate. Characterisation routine parameters based on the stimulus in Fig. 8: $t_{w,\Delta R} = 100\mu s$, $t_{w,iv} = 1.1ms$, $V_b = 0.6/0.7/0.8V$, and $V_{read} = 0.5V$

B. Calibration of Exponential verilog-A Model

We programme the proposed model by modulating number of pulse, pulse width and amplitude in order to explore the both qualitative and quantitative impacts on model. To keep the consistency, all simulations will start from the same initial RS as baseline. Recommendations are given after evaluating all modulations. The initial RS (Rinit) is set to $16.25k\Omega$ in order to be consistent with [15].

1) Number of pulse modulation: The positive bias voltages from $0.6V$ to $0.8V$ are employed on the Verilog-A memristor model (see Fig. 15). The reason we choose the positive voltage is that in our model the boundary of upper limit is fixed and will not be affected by the bias voltage (refer to Listing 2, lines 7 and 26). Fig. 15 shows that three levels of amplitude voltages can programme the device to the boundary of model limit in different speeds. The higher bias voltage induces faster RS changing rate. As for the number of pulse modulation, a similar observation can be found in Fig. 19(b).

As for the constant plateau of models in Listings 1 and 2, the main difference is induced by the parameter rp1. In Listing

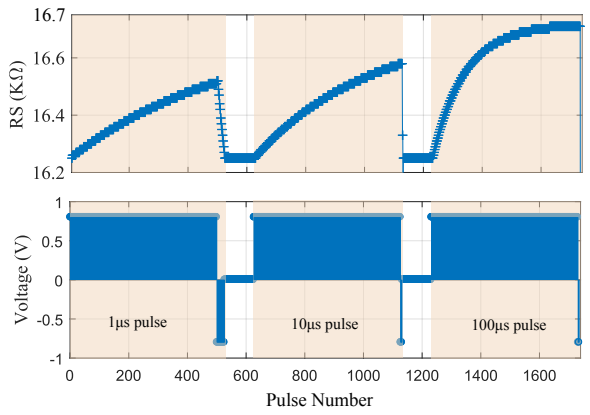


Fig. 16. Programming the device for different pulse duration. Bottom trace: Each number of 500 pulses at $0.8V$ for three different pulse duration. In between the measurement, the inverse voltages are applied to flush the device to its initial RS. Top trace: shows the modulation results corresponding to the bottom stimulus with both positive and negative bias voltage. Results with highlight is shown in Fig. 17 with a clear view of programming memristor to a specific RS. Characterisation routine parameters: $t_{w,\Delta R} = 1/10/100\mu s$, $t_{w,iv} = 1.001/1.01/1.1ms$, $V_b = |0.8V|$, and $V_{read} = 0.5V$.

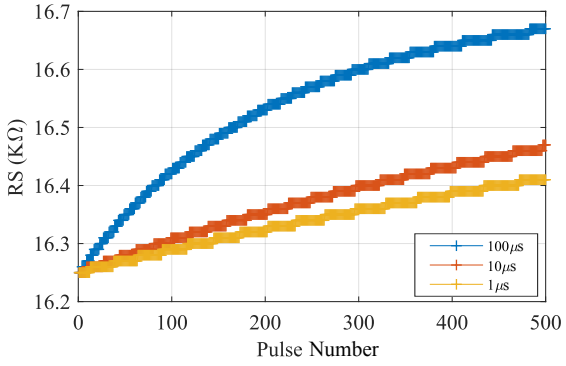


Fig. 17. Verilog-A exponential memristor model response for varying pulse width. Starting from initial $RS = 16.25k\Omega$, resistive states climb to different states for different pulse width. The $100\mu s$ pulse have higher changing rate and generates more states within 200 pulses which is less recognisable compared with $1\mu s$ pulse. However, the longer pulse width induce faster RS change that lead to lower RS resolution. It illustrates that the shorter duration pulse helps generate specific RS with higher resolution.

1 line 6, rp1 is non-zero that means the upper plateau depends on both initial RS and bias voltage (line 24). In Fig. 10, three amplitudes of pulse trains programme the device to different constant plateaus (from same initial RS). Whilst rp1 is equal to zero in Listing 2, which means that the constant plateau of this model is fixed to the value of rp0, $16.719k\Omega$ (see lines 7 and 26). It can be found that in Fig. 15, RSs excited by different amplitude voltage pulses eventually reach the same upper constant plateau.

2) Pulse width modulation: We now explore the effect on programming the memristive device at a bias voltage of $|0.8V|$ for $1\mu s$, $10\mu s$ and $100\mu s$ different pulse width. In Fig. 16, three trains with fixed number of pulses (500) are generated to capture the changing rate of RS. Besides, negative voltage are employed on the device to flush the RS to its initial value. It also proves that the modulation in positive voltage can be applied on negative one.

Fig. 17 presents pulse width modulation. It is difficult to discern the RS that programmed by $100\mu s$ pulses due to the fast changing rate, especially in the first 200 pulses. It indicates that pulse with smaller width can be applied to slow down the changing rate, which contributes to programming the device to a specific resistive state accurately.

3) Amplitude and polarity modulations: To evaluate the amplitude and polarity modulation, stimulus contains bias voltage from $|0.6V|$ to $|0.8V|$ in two polarities are generated. A number of 500 pulse trains were applied on the device in order to have comparison with Fig. 17 with fixed pulse number. Then, negative pulses are also applied on the device to push the device to its initial RS (in Fig. 18). Combining Figs. 15 and 19, we can observe that the model treats each polarity independently. Two polarities induce to different RS constant plateaus as well as RS changing rates.

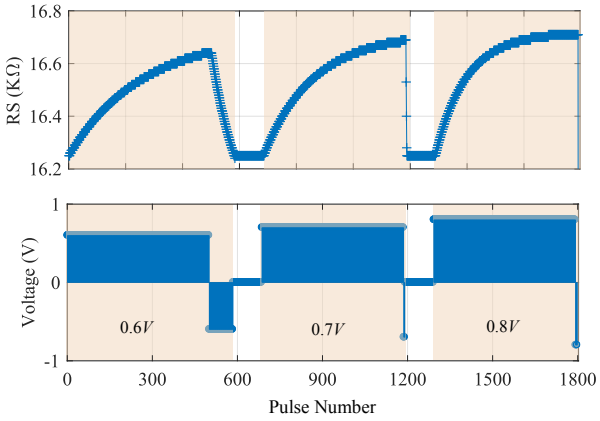


Fig. 18. Programming the device with three amplitude pulses. Bottom trace: Each number of 500 pulse (duration=$100\mu s$) under incremental bias voltage are employed to modulate device RS. In between the measurement, inverse voltages help flush the device back to initial state. The RS modulated by positive bias voltage has been highlighted and will be shown in Figure 19 for comparison. Characterisation routine parameters: $t_{w,\Delta R} = 100\mu s$, $t_{w,iv} = 1.1ms$, $V_b = |0.6/0.7/0.8V|$, and $V_{read} = 0.5V$.

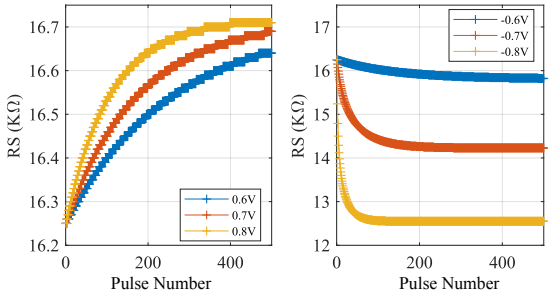


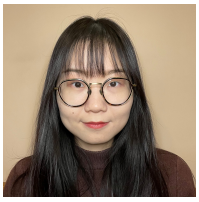
Fig. 19. Verilog-A memristor model response based on the amplitude of applied pulses. Being stimulated by different amplitude voltage, simulation results from both positive and negative bias voltages present in left and right figure respectively. It can be seen that the higher absolute voltage leads to faster changing rate of RS. Combining two sub-figures, positive voltage induces lower changing rate of RS, compared with negative voltage. The right figure also indicates that the saturated RS (boundary of lower limit) is dependent on the bias voltage, where a more negative voltage can push RS to a lower state.

Sachin Maheshwari (S'12–M'21) received the Bachelor's degree in Electrical and Electronic Engineering from the ICFAI University, India and Master's in Microelectronics from Birla Institute of Technology and Science, Pilani, India. He then obtained his PhD degree in Electronics Engineering from the University of Westminster, London, U.K. Currently, he is a Research Fellow at the Centre of Electronics Frontiers, University of Southampton, U.K. His research interests include energy recovery logic and neural network design with memristive crossbar arrays.

Spyros Stathopoulos received his Ph.D. degree in applied physics researching on shallow junction engineering in silicon and germanium from the National Technical University of Athens, Greece. He is currently with the Centre for Electronics Frontiers, University of Southampton, working on the fabrication, characterization and CMOS integration of memristive devices.

Jiaqi Wang (S'21) received the bachelor degree in Microelectronic Science and Engineering from Shenzhen University, China, in 2017 and her M.Sc. degree in Microelectronics Systems Design from University of Southampton, UK, in 2018. And she is currently pursuing her PhD studies in Zepler Institute, University of Southampton, working towards memristor-based hardware design, analogue and mixed-signal integrated circuit design for biosignal processing.

Alexander Serb received his degree in Biomedical Engineering from Imperial College in 2009 and his PhD in Electrical and Electronics Engineering from Imperial College in 2013. Currently he is a research fellow at the Zepler Institute (ZI) dept., University of Southampton, UK. His research interests are: cognitive computing, neuro-inspired engineering, algorithms and applications using RRAM, RRAM device modelling and instrumentation design.



Yihan Pan (S'21) received the B.Eng. degree in Electronic Engineering at the University of Manchester, Manchester, U.K. and the M.Sc. degree in Analogue and Digital Integrated Circuit Design at Imperial College London, London, U.K. She is currently moving toward the Ph.D. degree at the University of Southampton, Southampton, U.K. Her research interests include hardware topologies for symbolic processing.

Andrea Mifsud received his B.Eng. degree in 2016 at the University of Malta, and the M.Sc. specializing in full-custom integrated circuit design, from Imperial College London in 2017. Andrea then joined Science and Technology Facilities Council as a CMOS Sensor Design Engineer within the CMOS Sensor Design group. His work focused on the design and testing of CMOS image sensors including both pixel and periphery circuit design. He joined the Next Generation Neural Interfaces (NGNI) lab and the Centre for Bio-Inspired Technology (CBIT) at Imperial College London, in 2020, where he is currently a Research Associate.

Lieuwe B. Leene received the B.Eng. degree in electronic engineering from the Hong Kong University of Science and Technology in 2011, and the M.Sc. and Ph.D. degrees in electronic engineering from Imperial College London, in 2012 and 2016, respectively. He is currently a Senior IC Design Engineer at Novelda AS. Previously he was a Research Associate with the Centre for Bio-inspired Technology, Department of Electrical and Electronic Engineering, Imperial College London. His current research interests include low-noise instrumentation systems, brain machine interfaces, data converters, and mixed signal circuits for biomedical applications.

Jiawei Shen received the B.Eng. degree in electronic engineering from the University of Nottingham, and the M.Sc. in Analogue and Digital Integrated Cir-cuit Design at Imperial College London, London. He is currently pursing the Ph.D. degree at Impe-rial College London. His current research interests include developing CMOS-memristor technologies, periphery circuits for on-chip memristor array and analogue applications of memristor.

Christos Papavassiliou received the B.Sc. degree in physics from the Massachusetts Institute of Technology, and the Ph.D. degree in applied physics from Yale University.He is with the Electrical Engineering Department at Imperial College London. He currently works on memristor applications, sensor devices, and systems and antenna array technology. He has contributed to over 70 publications on weak localization, GaAs MMICs, and RFIC.

Timothy G. Constandinou (A'98–M'01–SM'10) received his B.Eng. and Ph.D. degrees in electronic engineering in 2001 and 2005, respectively, from Imperial College London. He is a reader of neural microsystems within the Circuits and Systems Group, Department of Electrical and Electronic Engineering, and also the deputy director of the Centre for Bio-Inspired Technology at Imperial College London. His research interests include neural microsystems, neural prosthetics, brain–machine interfaces, implantable devices, and low-power microelectronics. He serves on the IEEE Circuits and Systems (CAS) Society BioCAS and sensory systems technical committees and the IEEE Brain Initiative steering committee. He was the technical program cochair of the 2010, 2011, and 2018 IEEE BioCAS conferences and the general chair of the BrainCAS 2016 and NeuroCAS 2018 workshops. He is a Fellow of the Institution of Engineering and Technology, a chartered engineer, and a member of the Institute of Physics. He is a Senior Member of the IEEE. He is the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.

Themistoklis Prodromakis (SM'08) received his Bachelor in Electrical and Electronic Engineering from the University of Lincoln (UK), his MSc degree in Microelectronics and Telecommunications from the University of Liverpool (UK) and a PhD in Electrical and Electronic Engineering from Imperial College London (UK). He then held a Corrigan Fellowship in Nanoscale Technology and Science with the Centre for Bio-inspired Technology at Imperial College London (UK) and a Lindemann Trust Visiting Fellowship with EECS UC Berkeley (US). He is a Professor of Nanotechnology Director of the Centre for Electronics Frontiers at the University of Southampton, UK. He currently holds a Royal Academy of Engineering Chair in Emerging Technologies and a Royal Society Industry Fellowship. His background is in electron devices and nanofabrication techniques. His current research interests include memristive technologies for advanced computing architectures and biomedical applications. Prof Prodromakis is a Fellow of the Royal Society of Chemistry, the British Computer Society, the IET and the Institute of Physics, and is also Senior Member of the IEEE.