FPAA/Memristor Hybrid Computing Infrastructure

Mika Laiho, Jennifer O. Hasler, Jiantao Zhou, Chao Du, Wei Lu, Eero Lehtonen, Jussi H. Poikonen

***Abstract*—This paper presents a circuit in which tungsten oxide -based analog memristors are post-processed on a CMOS-based Field-Programmable Analog Array Integrated Circuit (FPAA-IC). FPAAs are powerful tools for rapid analog experimentation, prototyping and power-efficient computing, and they allow custom analog circuits to be built and reconfigured. The primary motivation for this work is to introduce and demonstrate the operation of the FPAA/memristor hybrid circuit and the board-level infrastructure, and to form a basis for subsequent empirical work on analog memristive computing. The experiments shown in this paper demonstrate a successful fabrication of memristors on the FPAA substrate, and the usefulness of the hybrid computing infrastructure in terms of experimentation with memristors. The experiments suggest that a single state variable cannot capture the adaptation of a memristor. To this end, a SPICE compatible memristor model with two state variables is presented. Furthermore, a memristor-based adaptive coincidence detector is demonstrated on the FPAA/Memristor computing infrastructure.**

***Index Terms* — Memristor, field-programmable analog array (FPAA), hybrid integrated circuit, analog signal processing**

I. Introduction

Memristors are passive two-terminal circuit elements with a resistance that varies as a function of charge or flux passing through the device [1]; the theory of memristive devices can be used to describe the change of the resistance [2]. In fact, the theory of memristive devices can be used to capture the characteristics of two-terminal memory elements that rely on different mechanisms responsible for the actual change in resistance, such as change of phase, magneto-resistive effect or thin-film ionic transport [3].

In this paper we concentrate on memristors in which an electric field across the device redistributes the ion concentration in a thin film, and thus changes the resistance. Such devices based on various different technologies have been reported for example in [4]–[7]. Memristive devices can be either digital or analog depending on the resistance switching characteristics [8]; in this work we concentrate on the characterization and computing with the analog memristor originally reported in [9]. A natural way to fabricate memristors is to form the devices into crossing points of perpendicular wires, which facilitates the related lithography. Computation with passive devices such as memristors requires them to be interfaced to active components. If the memristor fabrication process is CMOS-compatible, for example in terms of temperature and materials, memristors can be fabricated on top of CMOS circuits. A CMOS-memristor hybrid circuit has benefits especially with parallel computing architectures that rely on local access to memory resources. Prior to this work, fabricated CMOS/memristor hybrid circuits have been presented for example in [10]–[12].

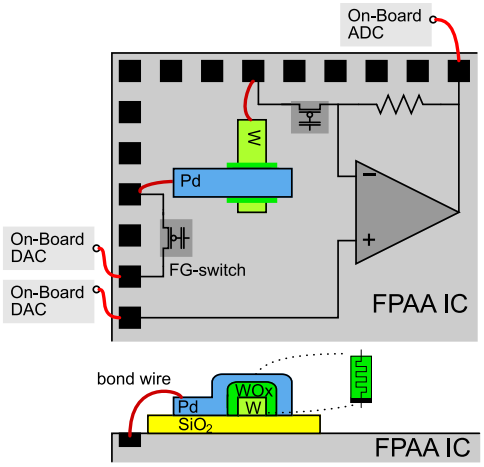


Fig. 1. Conceptual illustration of the FPAA/memristor hybrid computing infrastructure. Memristors are fabricated on top of the FPAA-IC. The memristor has a tungsten (W) bottom electrode, and a palladium (Pd) top electrode. The FPAA-IC is placed on a circuit board that houses digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) that are used to control and measure the circuit during the experiments. Memristors are connected to FPAA pads with bond wires, and internal FPAA wirings are routed using floating-gate (FG) switches.

Algorithms that can be mapped to such parallel architectures commonly have effective analog implementations. Examples are high complexity, low accuracy computing tasks such as analog filter banks and vector matrix multipliers. The primary advantage in this type of computing is the potential for higher energy efficiency as compared to digital signal processing [13]. The computing efficiency stems from mapping the algorithm topographically to the circuit, operation on unquantized values, and utilization of inherent device characteristics that originate from device physics. A major downside with typical integrated circuit realizations of analog computation is the amount of design time and effort; usually a fabricated circuit cannot be reconfigured, and changing the circuit requires a new fabrication round.

Analog equivalents of Field-Programmable Gate Arrays (FPGA), namely Field-Programmable Analog Arrays (FPAA), make it possible to design and reconfigure analog circuits repeatedly [14]. Circuit design is carried out by connecting

Mika Laiho (mika.laiho@utu.fi), Eero Lehtonen, and Jussi Poikonen are with Technology Research Center (TRC), University of Turku, Finland. Jennifer O. Hasler is with School of Electrical and Computer Engineering, Georgia Institute of Technology, United States. Jiantao Zhou, Chao Du and Wei Lu are with Electrical Engineering and Computer Science, The University of Michigan, United States. The work at TRC was funded by the Academy of Finland (131295, 140108, 258831, 253596, 264914, 277383), while the work at UM was supported in part by the Air Force Office of Scientific Research (AFOSR) through MURI grant FA9550-12-1-0038.

Computational Analog Blocks (CAB) via switches and routing fabrics. Here we consider integrating analog memristors on top of a CMOS FPAA. Such an FPAA-memristor hybrid circuit forms a powerful analog computing platform, combining accuracy and efficiency of Floating-Gate Transistor (FGT) -based analog elements with the adaptation properties of a memristor; using the FPAA architecture we can construct analog circuits containing active CMOS components and passive memristors, and reconfigure the circuits at will. It is significant to note that memristor-based designs, be they intended for characterisation or actual computation, immediately benefit from the computing infrastructure developed for FPAAs.

In the work presented in this paper we have integrated tungsten oxide -based analog memristors with FPAA Integrated Circuits (FPAA–IC). Fig. 1 presents a conceptual diagram of this hybrid circuit. The memristor has a tungsten (W) bottom electrode and a palladium (Pd) top electrode that are connected to the FPAA elements via floating-gate switches. The hybrid circuit interfaces to a board-level FPAA infrastructure (microcontroller, communication resources) via on-board digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). In the following we describe the FPAA/memristor hybrid circuit and computing infrastructure, and use it to experiment with analog memristors. An overview of the FPAA-memristor concept was given in [15] and this paper extends the explanations, analysis and experimental work. To the authors' knowledge, this work is the first in which an FPAA/memristor hybrid circuit is empirically demonstrated.

II. FPAA/Memristor hybrid circuits

A. Field-programmable analog arrays

Field-programmable analog arrays are essentially a collection programmable computational analog blocks and a network of reconfigurable interconnects. As a FPAA is a reconfigurable system, it facilitates testing, rapid prototyping, or final implementation of analog circuits in a wide variety of applications. FPAAs are capable of performing signal processing functions such as filtering, matrix multiplication, amplification, and current to voltage conversion entirely in the analog domain [16], [17].

A CAB in the FPAA consists of circuit elements and signal processing subcircuits of different levels of complexity, including transistors, amplifiers, and multipliers. Similarly to Flash memories, FPAAs are based on Floating-Gate (FG) MOSFET technology. This technology relies on the ability to control the amount of charge stored at the gate terminals of FG transistors; as altering the gate charge changes the threshold voltage of the floating-gate transistor, an FGT can act either as a programmable, nonvolatile current source, or as a switch.

Fig. 2 illustrates that the FGT used in this work is a transistor with two capacitors (C_{in} and C_{tun}) attached to the gate. In Fig. 2(a) electrons are removed from the floating gate using quantum tunneling by pulling V_{tun} to a high potential while connecting V_{in} to ground (GND). If $C_{\text{tun}} \ll C_{\text{in}}$, the floating gate voltage V_{fg} remains close to GND, resulting in a high voltage across capacitor C_{tun}. A sufficiently high voltage yields a usable tunneling current through C_{tun}, as this current depends exponentially on the voltage across the capacitor [18].

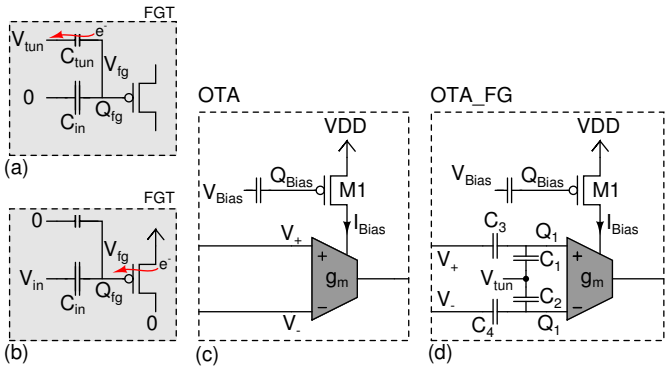


Fig. 2. (a) FGT during tunneling. Electrons are removed from the FG through C_{tun}. (b) FGT during hot electron injection, which is used to add electrons to the FG. (c) CAB element OTA: operational transconductance amplifier with programmable bias current. (d) CAB element OTA_FG: operational transconductance amplifier with programmable bias current and FGTs at inputs to eliminate input offset voltages.

Electrons are added to the floating gate by hot electron injection as shown in Fig. 2(b). In this method, the FGT is biased with a high source–drain voltage while controlling the current with V_{in}. The high electric field in the channel yields high energy holes that impact ionize electrons in the drain-to-channel depletion region. These electrons gain energy as they enter the channel region and, with sufficiently large source-drain voltages, they get injected through the gate insulator to the gate [18].

Figs. 2(c) and (d) show two CAB elements used in this paper. In the transconductance amplifier shown in Fig. 2(c), the bias current is controlled by a FGT and is thus programmable. The transconductance amplifier of Fig. 2(d) has FGTs also at the inputs so that nonvolatile electrical trimming can be used to eliminate the input offset voltage: an offset of opposite polarity is programmed to the input FGTs to counteract the initial offset [14]. The ability to mitigate mismatch effects facilitates scaling of analog designs to CMOS technologies of smaller line widths, and is useful in dealing with memristor mismatch.

An FPAA circuit board houses the FPAA along with a microcontroller, communication links and data converters. It should be noted that the low level programming mechanisms (tunneling and injection) are handled by the FPAA infrastructure so that accurate FG programming requires just defining a target value; the board, together with high level synthesis tools and software routines form a complete infrastructure for experimentation. For example, the tool set includes compilers from netlist or SPICE definitions to object code, routing structure viewers, and Matlab components, including a Simulink-based tool for block-level designs [19]. The interface between the PC and the FPAA is achieved with Matlab through the on-board micro controller, and all communication is performed in the digital domain. FPAA circuits, boards, and toolsets are discussed in detail for example in [14] and [20].

B. Memristive devices

The memristive device considered in this paper is a tungsten oxide -based analog memristor, originally presented in [9]. The memristive behavior of the device is attributed to the migration of oxygen vacancies at the oxide–electrode interface. A

structural diagram of this memristor is presented in Fig. 1. In contrast to digital memristors with discrete conductance states, the conductance of the analog memristor can be altered on a continuous range by applying sufficiently large voltages across it. A positive voltage increases the conductance of the memristor, while a negative voltage decreases it. To measure the conductance of the memristor, voltage pulses of short duration and low amplitude should be used in order to avoid unwanted programming. Ideally, a memristor is nonvolatile, which means that its conductance does not change while the device is unpowered. However, the considered tungsten-based memristor is volatile: after the device has been programmed, the memristor starts to return towards its equilibrium (relaxed) state. In certain applications the relaxation process that causes the volatility of a memristor may be useful, as discussed in [21] and in Section V-B.

C. FPAA/memristor integration

The starting point of fabricating the FPAA/memristor hybrid circuit was a wafer containing FPAA-ICs housing 84 CABs. The wafer had gone through a 350 nm CMOS process and had been cut to reticle-sized pieces before post-processing. It is notable that FPAAs have already been used as a substrate for fabricating devices on the wafers: integration of micro-electromechanical system (MEMS) devices — MEMS microphones and capacitive micromachined ultrasonic transducers (CMUT) — with FPAAs was demonstrated in [22]. Here we show that a similar approach can be used for memristors.

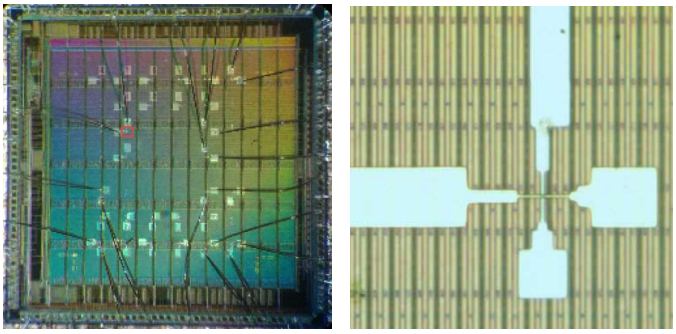
Chip planarization was performed prior to the fabrication of memristor devices. A 500 nm SiO_2 film was deposited as a passivation layer on the top of the FPAA CMOS chip by plasma enhanced chemical vapor deposition (PECVD). Spin-on-glass (SOG) was then deposited, baked, and etched back to create a planarized surface for memristor integration. Fig. 3(a) shows a die photograph of the FPAA chip with the post processed memristors.

The memristor device consists of a MIM structure with a 300 nm wide palladium (Pd) top electrode, a tungsten oxide (WOx) switching layer and a 360 nm wide tungsten (W) bottom electrode. Fig. 3(b) shows a microphotograph of one of the fabricated memristors. The memristor was formed into the cross section of the top and bottom electrodes. First, a 60 nm thick tungsten film was deposited by sputtering at room temperature on the platform. The bottom electrodes and contact pads were patterned by e-beam lithography and reactive-ion etching. Rapid thermal annealing in pure oxygen at 350°C for 1 minute was performed to form an approximately 30 nm thick tungsten oxide layer. The Pd/Au top electrodes were then patterned by e-beam lithography and lift-off to complete the memristor structure in a crossbar geometry. After contact hole opening, the electrodes of nine memristors were wire-bonded to the I/O pads of the chip for access to FPAA resources.

III. Experiments with the FPAA/memristor hybrid circuit

A. Transimpedance amplifier for measuring memristor current

Fig. 4(a) shows a transimpedance amplifier -based circuit that was used to measure memristors' currents. This circuit



(a) (b)

Fig. 3. (a) Die photo of the FPAA with postprocessed memristors. (b) Microphotograph of a memristor; the location of the memristor on the FPAA die is indicated by a red square in Fig. 3(a).

uses CAB elements OTA and OTA_FG. As shown in Fig. 2, the OTA uses an FGT for bias current, while the OTA_FG has FGTs also at the inputs for offset cancellation. The CAB elements are wired together via the FPAA routing fabric and FG switches. Transconductor OTA_FG acts as a resistive feedback element of the transimpedance amplifier as in [14].

The transimpedance amplifier keeps V_{mem1} at virtual ground level V_{in1}, assuming the voltage gain of the OTA is large enough. Voltage V_{in1} is driven by digital-to-analog converter DAC1. The other terminal of the memristor is driven to V_{in2} by DAC2. To measure the current flowing through the memristor, we use the transimpedance amplifier to transform the current into analog voltage V_{out}, which is further converted into digital domain using an analog-to-digital converter (ADC) on the FPAA board.

Note that in Fig. 4(a) no FG switches are depicted; usually the actual switch configuration does not affect the operation of the circuit, and is thus abstracted away with the simulink-based FPAA schematic editor. In reality the routing of the circuit requires multiple switches. This is illustrated in Fig. 4(b) that shows also the FG-switches. With the fabricated analog memristors the switches do matter because the memristor currents are in the order of 100 μA, while the ON-resistance of an FG-switch is in the kilo-ohm range, making the switches the dominant nonideality in the measurement. Also, a single OTA_FG cannot source or sink currents around 100 μA; multiple OTA_FGs are thus connected in parallel to reach sufficient current levels.

The Simulink-based design tool generates an FPAA configuration file, where the switches to be programmed are identified by row and column coordinates. Because of the key role of the switches, the number of switches in the current path was reduced by manual routing optimization. The resulting switch configuration is shown in Fig. 4(b), where the switch transistors shown in the darker background ($S5 - S11$) are in the current path. In order to further mitigate the influence of the high ON-resistances of the FG switches, several switches were added in parallel to switches $S5 - S11$.

We characterized the transimpedance amplifier using a 11.9 kΩ external resistor R_{ext}, and used this measurement to determine a relation between memristor current and V_{out}. In order to determine the voltage V_{mem} across the memristor,

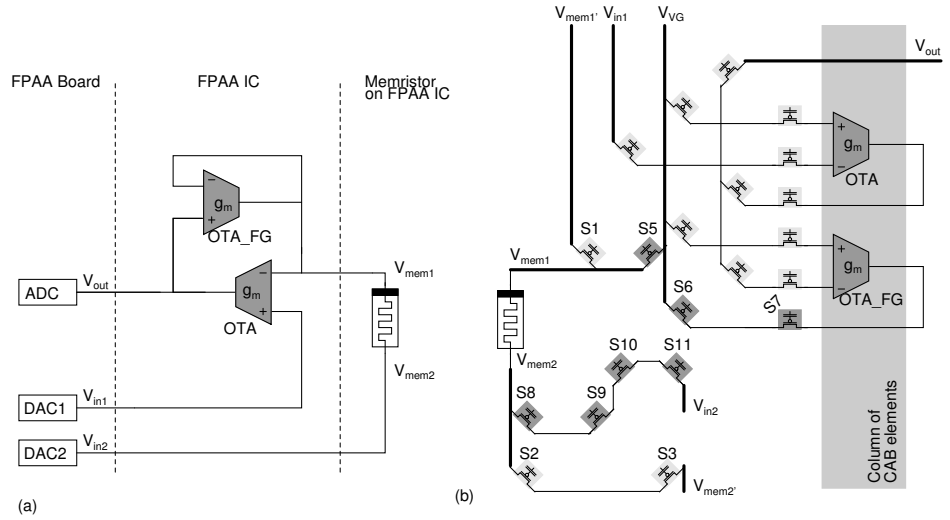


Fig. 4. (a) Transimpedance amplifier on FPAA for memristor characterisation. Transconductor OTA_FG acts as a resistive feedback element of the amplifier. The (b) The amplifier of Fig. 4(a) including routing switches. The bold lines depict wires that connect to I/O pads, whereas the switches either adjoin segmented wires, or perpendicular wire crossings. Desired CAB elements are selected for use from a column-wise arrangement of CAB elements.

the resistive voltage drop of FG switches $S5 - S11$ was characterised by using another set of ADCs, not shown in Fig. 4(b). Since memristor terminals V_{mem1} and V_{mem2} are not bonded to the circuit board, they were routed to $V_{\text{mem1'}}$ and $V_{\text{mem2'}}$ through switches $S1 - S3$. Note that these voltages are approximately equal to V_{mem1} and V_{mem2} since the ADCs have high-impedance inputs (no DC current goes through switches $S1 - S3$). Therefore, the current through R_{ext} is

$$I_{\text{Rext}} = (V_{\text{mem1'}} - V_{\text{mem2'}})/R_{\text{ext}} = V_{\text{out}}/R_{\text{fb}}, \qquad (1)$$

where R_{fb} is the equivalent feedback resistance of the transimpedance amplifier-based memristor characterization circuit. A characterisation measurement to relate V_{out} and I_{Rext} was carried out by sweeping V_{in2} and monitoring $V_{\text{mem1'}}$, $V_{\text{mem2'}}$ and V_{out}. Resistor R_{fb} was found to be $13.1k\Omega$, and I_{mem} can be computed as $I_{\text{mem}} = V_{\text{out}}/R_{\text{fb}}$.

It should be noted that using R_{fb} for memristor characterisation requires AD-conversion of three voltages ($V_{\text{mem1'}}$, $V_{\text{mem2'}}$ and V_{out}). However, the FPAA can perform measurements at a higher speed, if only one AD-conversion is required. In order to cope with one AD-conversion, voltages V_{in} and V_{out} need to be related to memristor voltage. Given that the input offset voltage of the transconductance amplifier can be trimmed to zero by using the OTA_FG, we define V_{drop} as a voltage that contains the resistive drop of the switches, as well as fluctuation of the virtual ground due to finite gain of the transimpedance amplifier,

$$V_{\text{drop}} = V_{\text{in1}} - V_{\text{in2}} - V_{\text{mem}}. \qquad (2)$$

V_{drop} is a nonlinear function of V_{out} since the PMOS-based FG-switches are more resistive when they connect between signals that are close to ground. V_{drop} was related to V_{out} using a fourth-order polynomial fit.

B. Characterisation of the fabricated memristors

Visual inspection of the nine memristors that were connected to the FPAA I/O pads revealed that two memristors had a broken wire in one of the terminals. Furthermore, one of the memristors could not be measured because it was connected to an I/O pad that was not accessible to the internal routing structure of the FPAA. Three of the six remaining memristors were open circuits, while three exhibited memristive behaviour; these are referred to as M1, M2 and M3. In the measurements of Sections III and IV, the virtual ground level V_{in1} was set to 1.9 V, while the operating voltage of the FPAA was 3.4 V. Values of voltage V_{in2} are given relative to the virtual ground level.

Fig. 5 shows voltage-current hysteresis loops of memristors M1 (a,b), M2 (c,d), and M3 (e,f) when programmed by sinusoidal pulses. In Fig. 5(a,c,e), a 1.35 V voltage pulse with a duration of 1 s was applied to V_{in2} in order to program the memristor into a high-conductance state. Then V_{in2} was driven with a full-wave rectified sinusoidal voltage of the form $V_{\text{in2}} = A_V|\sin(2\pi ft)|$ with amplitude $A_V = -1.35$ V and frequency $f = 15.625$ Hz for 0.16 s. The inset in Fig. 5(a) shows the V_{mem} and I_{mem} as a function of time; note that the amplitude of V_{mem} is lower than that of V_{in2} because of the resistive drop (see Eq. 2). The red curves in (a) show the initial I/V curves of memristor M1. During the measurements, the I/V curves of M1 changed unintendedly, and the negative currents (blue colour) scaled down by about a factor of three (see discussion in Section V-C).

Figures 5(b,d,f) show an experiment in which a -1.35V voltage pulse with a duration of 1 s was applied to V_{in2} in order to program the memristor into a low-conductance state. After this, V_{in2} was driven with the same rectified sinusoidal voltage as above, but with amplitude $A_V = 1.35$ V. This positive sinusoidal input programs the memristor towards a more conductive state. The voltage drop in the FG switches limits the voltage across the memristor to approximately 1.2V.

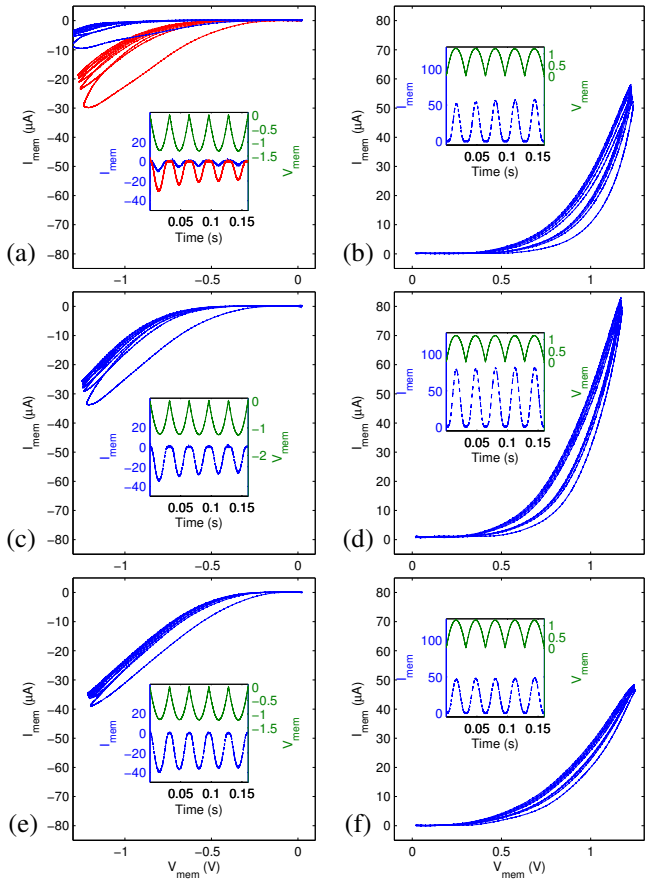


Fig. 5. Voltage–current hysteresis loops of fabricated memristors M1 (a,b), M2 (c,d) and M3 (e,f). In (a,c,e) the memristor is initialized first into a high-conductance state and then driven with a negative, full-wave rectified sinusoidal voltage. The corresponding decrease in the conductance of the memristor is visible in the hysteresis loops. The red curves in (a) show the initial I/V curves of memristor M1. At some point the I/V curves of M1 changed unintendedly, and the negative currents (blue colour) scaled down by about a factor of three (see Section V-C). In (b,d,f), the memristor is programmed to a low-conductance state and then driven with a positive, full-wave rectified sinusoidal voltage.

IV. Memristor Model

Our aim in the following is to create a SPICE compatible functional simulation model that captures memristor relaxation process (returning to an equilibrium state after stimulus) as well as transient response. Such a model provides important insight for a circuit designer, as the aim is to use the memristor as an analog computing element. Memristor M2 was chosen for additional characterization measurements in order to build the model.

Figure 6 shows the I/V response of the memristor with two sinusoidal voltages of different frequencies. The blue I/V curve is measured with a 94 Hz sine, whereas the black curve is measured with a 0.23 Hz sine. As expected, the black lower frequency I/V curve reaches a higher positive current as compared to the blue curve, since more adaptation takes place at lower frequencies. Similar conclusions can be made with negative voltages. It is worth noting that the amount of hysteresis in the I/V curves remains about the same, even if the frequency changes by a factor of 400. This is in contrast to a flux controlled memristor, where the amount of

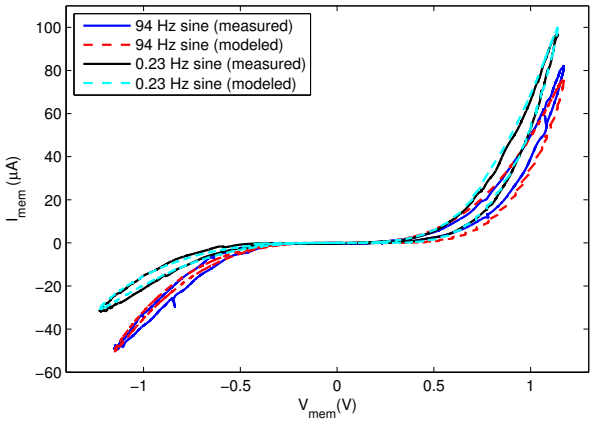


Fig. 6. I/V response of the memristor with two sinusoidal voltages of different frequencies. The blue I/V curve is measured with a 94 Hz sine, whereas the black curve is measured with a 0.23 Hz sine. Also shown in the figure are the modeled I/V curves corresponding to 94 Hz sine (red curve) and 0.23 Hz sine (light blue curve).

hysteresis increases inversely proportional to the frequency. Existing models like [9] work well on a particular timescale, but cannot capture the characteristics correctly with such a wide frequency range.

To overcome this limitation, we propose a model with two states W_1 and W_2, whose rates of change are defined as

$$\frac{dW_{\{1,2\}}}{dt} = \begin{cases} \alpha(V - V_{\text{th}})^{\beta} \frac{W_{\max}}{W_{\max} + W_{\{1,2\}}}, & \text{if } V - V_{\text{th}} > 0 \\ -\delta_{\{1,2\}}(V_{\text{th}} - V)^{\eta} W_{\{1,2\}}{}^{\gamma}, & \text{if } V - V_{\text{th}} \leq 0 \end{cases} \tag{3}$$

where α, β, δ_1, δ_2, η and γ are model parameters, $W_{\max}$ sets a soft limit for W_1 and W_2, whereas V_{th} sets the threshold voltage for conductance increase. When the voltage is below V_{th}, the conductance decreases. The only difference between W_1 and W_2 in Equation 3 is that $\delta_1 >> \delta_2$. Therefore, W_2 decreases much slower than W_1; W_1 is responsible for the rapid drop in the beginning of the relaxation process.

The threshold voltage V_{th} is not constant, but increases with state W_1 as

$$V_{\text{th}} = \rho + \xi \frac{\log W_1 - \log W_{\min}}{\log W_{\max} - \log W_{\min}} \tag{4}$$

so that the higher the state, the more voltage it takes to program the device. ρ and ξ are threshold voltage parameters. We define auxiliary state W as

$$W = \kappa(\log W_1 - \log W_{\min})^{\mu} + \lambda(\log W_2 - \log W_{\min})^{\mu} \tag{5}$$

where κ, μ and λ are model parameters. Since W_1 and W_2 are related to the flux, W relates to the logarithm of the flux. Note that $W_{\min}$ keeps the auxiliary state W positive. If states W_1 and W_2 fall below $W_{\min}$, the auxiliary state W becomes negative (the model stops working). However, no specific window function is used to limit the states, as the rate of change of states (3) decreases with the states; with small $W_{\min}$, it would take W_1 and W_2 a very long time to reach $W_{\min}$.

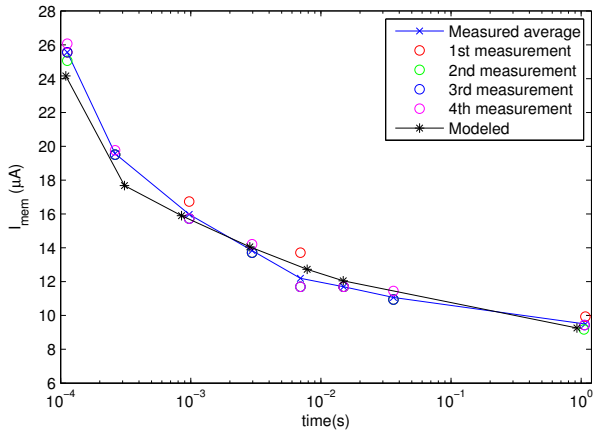


Fig. 7. Memristor relaxation curve measurement. The blue curve shows the average of four measurements in which the memristor is first programmed to high-conductance state, and then read with 200 μs, $V_{\text{in2}} = 0.7$ V pulses of exponentially increasing time intervals between read-pulses. The black curve shows a modeled relaxation curve.

The current through the memristor is defined as

$$I = \begin{cases} \nu W V^{\phi}, & \text{if } (V > 0) \\ -\psi(W+\zeta)|V|^{\phi}, & \text{if } (V \leq 0) \end{cases} \quad (6)$$

where ν, ϕ, ψ and ζ are model parameters.

The proposed logarithmic memristor model can reproduce the sinusoidal waveforms of Fig. 6. The red and light blue I/V curves in Fig. 6 corresponding to 94 Hz sine and 0.23 Hz sine are created with the memristor model. The model manages to capture the adaptation correctly at varying time-scales.

According to [21] the relaxation process causing the volatility of a WO_x -based memristor can be modeled by using a stretched-exponential, also known as a Kohlrausch function [23], which approximates an exponentially decaying process with multiple time constants. This can be observed from Fig. 7 that shows a measurement demonstrating memristor relaxation curve.

The measurement was performed by monitoring memristor current after initialization to a high conductance state using a 1 s, $V_{\text{in2}} = 1.35$ V pulse. After the initialization, 200 μs, 0.7 V read pulses were applied at exponentially increasing time-intervals. The 0.7 V amplitude in the read voltage was a compromise between unwanted change in memristor conductance and noise in the readout chain. Also, short pulse widths were used to minimise unwanted programming. Between pulses, the memristor voltage was set to zero. The measurement time spans from 10^{-4} s to 1 s. The measurement was repeated four times. The curve demonstrates a rapid decay in the beginning, followed by a gradual slow-down. The black curve in Fig. 7 shows that the proposed memristor model can successfully approximate the measured relaxation curve. The rapid drop in the current is created with state W_1, whereas W_2 produces a tail with a slower decay.

Fig. 8 illustrates the transient behaviour of the memristor in an experiment in which memristor M2 was programmed by pulses. The amplitudes of the pulses applied to V_{in2} were 0.7 V (V_{read}) and 1.35 V (V_{prog}). Between pulses, the voltage

TABLE I
MEMRISTOR MODEL PARAMETERS.

α	β	δ_1	δ_2	η	γ	ρ	ξ	κ
1.08e5	6	1.5e5	20	3	2.2	0.2	0.6	0.6

ψ	λ	W_{min}	W_{max}	ν	ϕ	ζ	μ
1.18e-6	0.35	1e-5	50	1.18e-6	3	5	1.5

was kept at zero volts. The memristor was first programmed to a low conductance state by applying $-V_{\text{prog}}$ to V_{in2} for one second. Next, a voltage waveform with V_{in2} alternating between V_{read} and V_{prog} as shown on top of Fig. 8 was applied. The first V_{read} pulse yields a current of 5.5 μA. Consequent V_{prog} pulses start to increase the conductance. It can be observed that between the V_{prog} pulses, the conductance decreases significantly. Even a 100 μs break in V_{prog} is enough to cause such a decrease, as visible after the fourth V_{prog} pulse.

The fast volatility of the measured memristor is also apparent after the sixth V_{prog} pulse which is immediately followed by a V_{read} pulse. During the read pulse the current is 16 μA, whereas the next V_{read} pulse, which does not immediately follow the seventh V_{prog} pulse, results in a 11.5 μA current. Also shown in Fig. 8 are memristor states W_1, W_2, auxiliary state W, modeled memristor current, and threshold voltage V_{th} (black curve in the top subfigure).

Fig. 9 shows an experiment similar to that of Fig. 8, except that the time-scale was stretched from 16 ms to 640 ms. In the beginning of the first, sixth and seventh read pulses, the measured currents were 6 μA, 20 μA and 13 μA, respectively.

In Fig. 10 a related experiment was carried out with an initialization to high-conductance state by applying a one-second V_{prog} pulse. This was followed by a pulse train similar to the one used in Fig. 8, but with the voltage polarities reversed as shown in Fig. 10. The aim was to show that the model manages to capture the transient behaviour with negative pulsing. In the beginning of the first $-V_{\text{read}}$ pulse, the current is -14 μA, whereas during the last $-V_{\text{read}}$ pulse, the current is -6 μA.

Overall, characterising and modeling memristors with two states as in the model introduced in Section IV is challenging, since a single measurement of memristor current with a particular read voltage reveals only the auxiliary state W; getting insight in the values of states W_1 and W_2 requires recording a longer I/V history. In the transient measurements of Figs. 8, 9 and 10, the model manages to reproduce the transient currents quite similarly to the measurements.

Table IV shows the memristor model parameters that were determined based on the measurements of memristor M2 shown in Figures 6-10. The same set of parameters was used in all the simulations of this paper. Initially, optimization methods such as binary search algorithm with threshold acceptance were used to find the parameters. However, in the end, optimization by manual iteration provided a parameter set that gave the best balance between I/V characteristics, relaxation curve and transient behaviour. Matlab was used to perform the simulations of this paper since it allows more options for parameter optimization as compared to SPICE. After the optimization, it was verified that Matlab and SPICE give

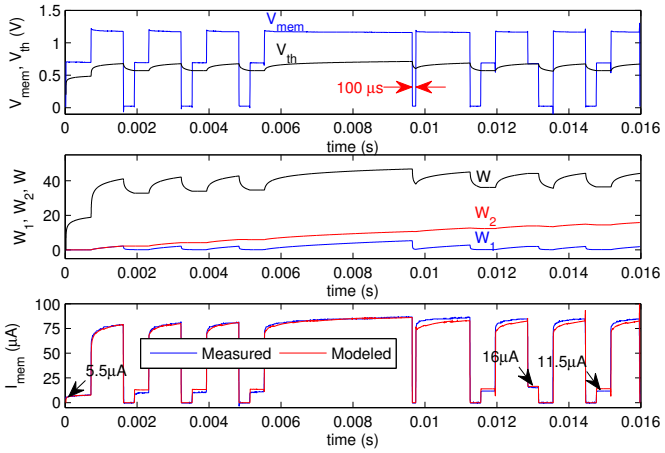


Fig. 8. 16 ms pulse programming experiment. Memristor M2 is initialized to a low-conductance state with a one-second negative programming pulse, followed by positive programming and read pulses that increase the conductance. In addition to measured current, memristor voltage V_{mem}, memristor states W_1, W_2, auxiliary state W, modeled memristor current and threshold voltage V_{th} (black curve in the top subfigure) are shown.

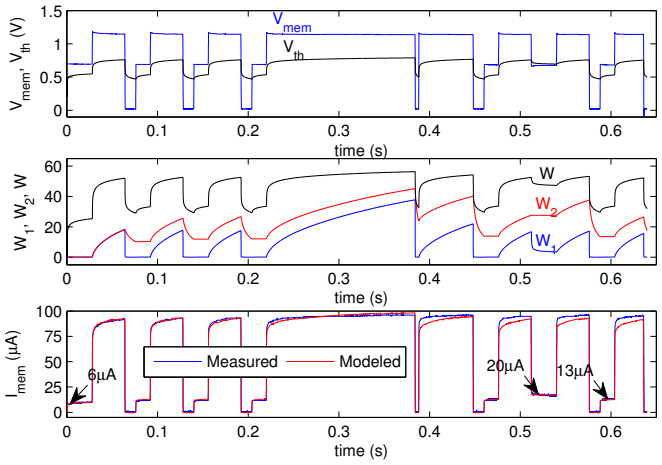


Fig. 9. 640 ms pulse programming experiment. Memristor M2 is initialized to a low-conductance state with a one-second negative programming pulse, followed by positive programming and read pulses that increase the conductance. In addition to measured current, memristor voltage V_{mem}, memristor states W_1, W_2, auxiliary state W, modeled memristor current and threshold voltage V_{th} (black curve in the top subfigure) are shown.

similar results.

The SPICE code of the memristor model is listed below; note that parameters `W1INIT` and `W2INIT` are used to set the initial values of state variables W_1 and W_2.

```
.SUBCKT logristor P M W1 W2 W Vth PARAMS:

+alpha=1.08e5 beta=6 rho=0.2 xi=0.6
delta1=1.5e5 delta2=20 eta=3 gamma=2.2
kappa=0.6 mu=1.5 lambda=0.35  nu=1.18e-6
phi=3 zeta=5 psi=1.18e-6
W1INIT=1e-3 W2INIT=1e-3
WMIN=1e-5 WMAX=50
```

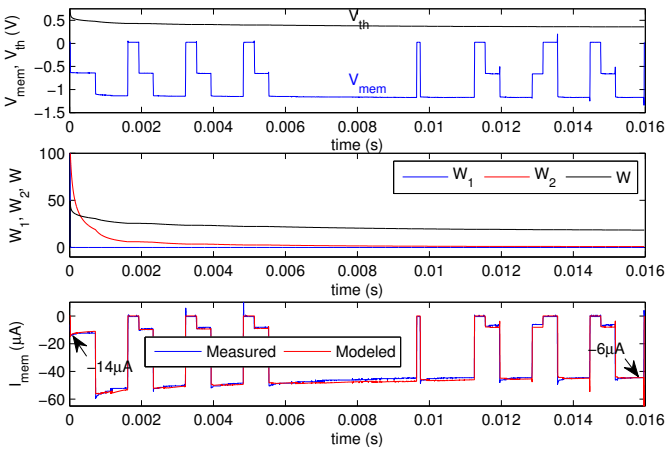


Fig. 10. 16 ms memristor pulse programming experiment with negative pulses. Memristor M2 is initialized to a high-conductanse state with a one-second positive programming pulse, followed by negative programming and read pulses. In addition to measured current, memristor voltage V_{mem}, memristor states W_1, W_2, auxiliary state W, modeled memristor current and threshold voltage V_{th} (black curve in the top subfigure) are shown.

```
* Threshold voltage Vth
Eth Vth 0 value = {rho+xi*(log(V(W1))-
log(WMIN))/(log(WMAX)-log(WMIN))}

*State variable W1
Gw1 0 W1 value= {alpha*
sign2(V(P,M)-V(Vth))*
pwr(V(P,M)-V(Vth),beta)*WMAX/(WMAX+V(W1))-
sign2(V(Vth)-V(P,M))* delta1*
pwr(V(Vth)-V(P,M),eta)* pwr(V(W1),gamma)}
Cw1 W1 0 1
.IC V(W1) {W1INIT}

*State variable W2
Gw2 0 W2 value= {alpha*
sign2(V(P,M)-V(Vth))*
pwr(V(P,M)-V(Vth),beta)*WMAX/(WMAX+V(W2))-
sign2(V(Vth)-V(P,M))* delta2*
pwr(V(Vth)-V(P,M),eta)* pwr(V(W2),gamma)}
Cw2 W2 0 1
.IC V(W2) {W2INIT}

* Auxiliary state W
Esv W 0 value = {
kappa*pwr(log(V(W1))-log(WMIN),mu)+
lambda*pwr(log(V(W2))-log(WMIN),mu)}

*Output
Gmem P M value= {sign2(V(P,M))*nu*
pwr(V(P,M),phi)*V(W) - sign2(V(M,P))*
psi*pwr(abs(V(M,P)),phi)*(V(W)+zeta)}

*Auxiliary function:
.func sign2(var) = {(sgn(var)+1)/2}

.ENDS logristor
```

V. Computing with FPAA-Memristor Hybrid Circuits

A. Dividing Computing Tasks Between CABs and Memristors

A few remarks are in order with respect to mapping computing tasks between the FPAA computational analog blocks and memristors. A FGT has three terminals (four including bulk) that can be used to alter the state (FG charge). The state is available to other transistors simply by connecting the gate to the FG, without having to perform a specific read cycle. The programming methods (tunneling and injection) and retention time are well defined; ignoring temperature, only the current state and instantaneous voltages at the terminals affect the programming sensitivity and retention time.

On the other hand, the state of a memristor has to be altered with the same terminals that are used to read it, and the state cannot be distributed to other devices without performing a read cycle. Furthermore, the programming sensitivity depends on recent history of stimuli in addition to instantaneous terminal voltages, and the relaxation time (or retention time if the memristor is nonvolatile) depends on the duration of the programming [21]. The two-state memristor model introduced in this paper is an effort to capture the temporal adaptation properties of the fabricated analog memristor.

These differences have a great impact on the preferred use of the devices. It is beneficial to use FGTs in tasks that require accurate programming such as trimming in addition to signal conditioning, amplification and driving within a hybrid circuit. Although the use of a memristor as an analog memory with separate read and write cycles is possible, a natural role of memristors is to act as continuously adaptive elements in the current path. Analog memristors have interesting temporal dynamics in their I/V characteristics and relaxation processes of the state(s). These qualities are obtained inherently with the physics of the devices. Provided that computing tasks can be mapped to utilise the physics, highly energy efficient computing is possible.

B. Example: Adaptive Coincidence Detector

Fig. 11 shows a transimpedance amplifier circuit for detecting coincidences of input pulses. The amplifier is adaptive so that the gain increases with frequent coincident input pulses, and decreases at periods of low coincident pulse activity. A memristor is placed on the feedback path to make the gain of the amplifier adapt with the inputs; input voltages V_{in2}, V_{in3} and V_{in4} connect to the virtual ground (sum) node through a set of FG switches. The series combinations of the switches act as resistances (denoted in Fig. 11 by $R1$, $R2$ and $R3$) that are used to convert input voltages to currents. Let us first assume that the OTA has a large enough voltage gain, and the memristor be bypassed with a short circuit. In this case, the circuit acts as a conventional transimpedance amplifier, where the output voltage V_{out} relates to the input current I_{in} as

$$V_{\text{out}} = -\frac{I_{\text{in}}}{g_{\text{m,OTA\_FG}}} \tag{7}$$

When a memristor is added to the feedback path, it can only affect the output current of the OTA_FG through the transistors at the output stage of the OTA_FG. As long as these transistors are in saturation (there is a small voltage across the memristor), the output resistance of the OTA_FG is high and the memristor has little effect on the amplifier. When the memristor has reached a low enough conductance state, and I_{in} is large enough, the output transistors of the OTA_FG fall out of saturation, causing an abrupt reduction in $g_{\text{m,OTA\_FG}}$ and, thus, increase in V_{out} as indicated by Equation (7).

Fig. 12 shows a measurement in which the circuit of Fig. 11 is used to detect coincidences of V_{in2}, V_{in3} and V_{in4} (three topmost curves). The offset of the OTA_FG is tuned so that without input pulses, the memristor is at an intermediate conductance level and the rest voltage of V_{mem} is about 0.4 V. A positive voltage pulse relative to a 1.5 V virtual ground at one of the inputs causes a negative voltage across the memristor. Note that memristor adaptation is a nonlinear function of voltage (η in (3) is 3). With one or two coincident input pulses, the magnitude of the memristor voltage stays below 1 V. On the other hand, three simultaneous pulses increase $|V_{\text{mem}}|$ above 1 V, that is able to cause more significant adaptation. The red horizontal line in the plot shows a possible threshold for detected coincident pulses. As explained above, the OTA_FG is biased so that once the memristor has reached a low enough conductance level, the transistors in the output stage of the OTA_FG fall out of saturation during coincident pulses; any adaptation after that increases V_{out} significantly, thus making coincident pulses easier to detect.

The first set of three coincident pulses is not enough to cause V_{out} to reach the threshold level, neither do single pulses or two coincident pulses. On the other hand, consecutive occurrences of three coincident pulses increase the gain of the transimpedance amplifier enough, as memristor conductance gets reduced by the pulsing. Without pulsing activity, the memristor voltage is at 0.4 V and the conductance increases slowly until V_{th} reaches 0.4 V; a single occurrence of three coincident pulses at the end of the experiment is not enough to reach the threshold level at V_{out}.

Therefore, the sensitivity improves with the summed duration of recently occurred three-pulse coincidences. On the other hand, the sensitivity reduces with the time from previous three-pulse coincidence. Thus, the detector filters out isolated coincident activity.

C. Discussion

The fabricated memristor draws tens of microamps of current at voltages above 1 V. For FPGA-based low power, continuously adaptive circuits, it would be desirable to downscale the current. This way the finite on-resistance of the FPAA routing switches would not degrade circuit performance, and the FPAA could be biased to its inherent current regime. One way to decrease memristor currents would be to make the area and thus the number of parallel conducting filaments smaller. However, the effect of device area on the current is not exactly linear since the filament growth is affected by many factors.

The devices were measured over a period of several months and they kept working consistently. As can be observed from Fig. 5, there is significant device-to-device deviation in the I/V

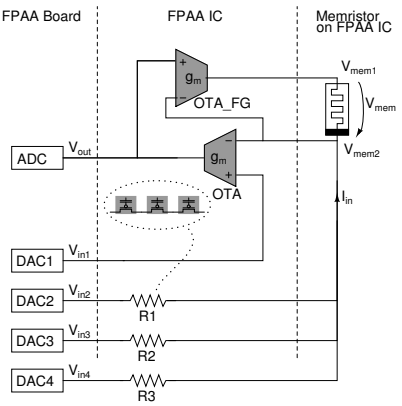


Fig. 11. Adaptive transimpedance amplifier. A memristor in the feedback path makes the gain of the amplifier adapt with incoming currents.

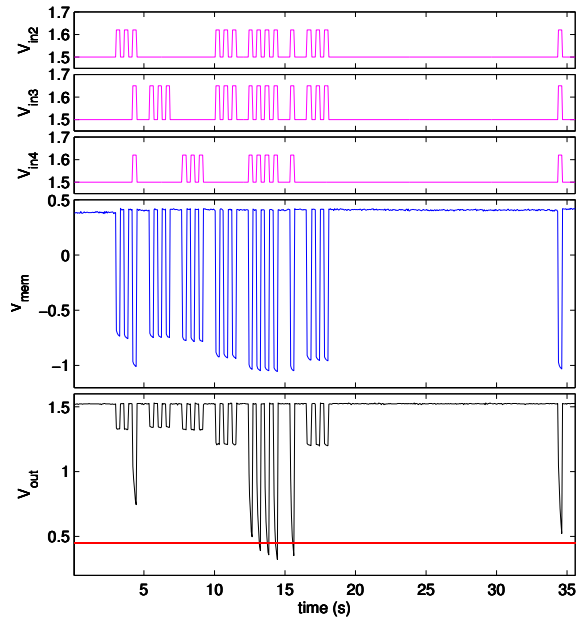


Fig. 12. Coincidence detection measurement with the adaptive transimpedance amplifier.

curves of the three measured memristors. It is possible that post-processing the devices on a CMOS substrate could have increased the deviation. This could be due to many factors such as insufficient planarization.

Consecutive measurements of a memristor gave consistently quite similar curves, but with days and weeks between the measurements there was some drift in the current levels. The drift was in the order of ten percent with one exception: the I/V curve of memristor M1 changed significantly; with negative memristor voltages the currents reduced to one third. The change in the characteristics occurred as a large negative voltage was unintendedly applied across the memristor for several minutes.

The proposed memristor model can rather faithfully capture the I/V characteristics of the measured memristor. However, this is not to say that the model is a complete description of the memristor. Furthermore, it is not a model where the different terms would have a direct link to device physics. Rather, it is a tool for a circuit designer to get an insight on the characteristics of an analog memristor for purposes of circuit design.

In upcoming FPAA-memristor fabrication experiments, the emphasis is on improving device homogenity, aiming at lower memristor currents, and considering memristor crossbars in addition to single devices.

VI. Conclusions

We presented a hybrid circuit which interfaces a field-programmable analog array with tungsten-based memristors. We described the architecture of this hybrid circuit, the integration of memristors to the FPAA, and a method of measuring memristors using the FPAA. Furthermore, current-voltage characteristics and relaxation curve measurements of the memristor were presented. Based on the measurements, a SPICE-compatible two-state circuit model that can capture the transient and relaxation curve of the memristor was developed. Finally, an adaptive coincidence detector using a memristor as a feedback element was demonstrated. The presented considerations form a basis for future work on analog computing using FPAA/memristor hybrid circuits.

References

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

[2] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE,*, vol. 64, no. 2, pp. 209–223, 1976.

[3] L. O. Chua, "Resistance switching memories are memristors,," *Applied Physics A*, vol. 102, no. 4, pp. 765–783, 2011.

[4] S. H. Jo and W. Lu, "CMOS compatible nanoscale nonvolatile resistance switching memory," *Nano Letters*, vol. 8, no. 2, pp. 392–397, 2008.

[5] M.-J. Lee, C. B. Lee, D. Lee, S. R. Lee, M. Chang, J. H. Hur, Y.-B. Kim, C.-J. Kim, D. H. Seo, S. Seo, U.-I. Chung, I.-K. Yoo, and K. Kim, "A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures," *Nature materials*, vol. 10, pp. 625–630, 2011.

[6] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, 2012.

[7] K. Gorshkov and T. Berzina, "On the hysteresis loop of organic memristive device," *BioNanoScience*, vol. 1, no. 4, pp. 198–201, 2011.

[8] W. Lu, K.-H. Kim, T. Chang, and S. Gaba, "Two-terminal resistive switches (memristors) for memory and logic applications," in *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, pp. 217–223, Jan 2011.

[9] T. Chang, S.-H. Jo, K.-H. Kim, P. Sheridan, S. Gaba, and W. Lu, "Synaptic behaviors and modeling of a metal oxide memristive device," *Applied Physics A*, vol. 102, pp. 857–863, 2011.

[10] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov, G. S. Snider, G. Medeiros-Ribeiro, and R. S. Williams, "Memristor – CMOS hybrid integrated circuits for reconfigurable logic," *Nano Letters*, vol. Vol. 9, no. 10, pp. 3640–3645, 2009.

[11] K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Letters*, vol. 12, pp. 389–395, 2012.

[12] P. Lin, S. Pi, and Q. Xia, "3D integration of planar crossbar memristive devices with CMOS substrate," *Nanotechnology*, no. 25, 2014.

[13] C. Twigg and P. Hasler, "Configurable analog signal processing," *Elsevier Journal on Signal Processing*, 2008.

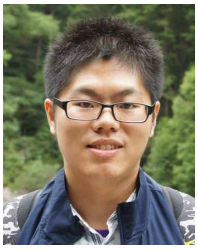
[14] A. Basu, S. Brink, C, Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, "A floating-gate based field programmable analog array," *IEEE Journal of Solid State Circuits*, vol. 45, no. 9, pp. 1781 – 1794, 2010.

[15] M. Laiho, J. O. Hasler, J. Zhou, C. Du, W. Lu, E. Lehtonen, and J. Poikonen, "Analog signal processing on a fpaa/memristor hybrid circuit," in *International Symposium on Circuits and Systems*, pp. 2265–2268, 2014.
[16] C. Schlottmann and P. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 1, no. 3, pp. 403–411, 2011.
[17] S. George, J. Hasler, S. Koziol, S. Nease, and S. Ramakrishnan, "Low power dendritic computation for wordspotting," *Journal of Low Power Electronics and Applications*, vol. 3, no. 2, pp. 73–98, 2013.
[18] P. Hasler and J. Dugger, "An analog floating-gate node for supervised learning," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 5, pp. 834–845, 2005.
[19] C. R. Schlottmann, C. Petre, and P. E. Hasler, "A simulink framework for design to and automated conversion on large-scale FPAA devices," *IEEE Transactions on VLSI*, 2010.
[20] S. Koziol, C. Schlottmann, A. Basu, S. Brink, C. Petre, S, Ramakrishnan, and P. Hasler, "Hardware and software infrastructure for a family of floating-gate FPAAs," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2010.
[21] T. Chang, S.-H. 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.
[22] A. Sisman, J. Zahorian, G. Gurun, M. Karaman, M. Balantekin, F. L. Degertekin, and P. Hasler, "Evaluation of CMUT annular arrays for side-looking IVUS," in *Proceedings of the IEEE International Ultrasonics Symposium*, 2009.
[23] M. N. Berberan-Santos, E. N. Bodunov, and B. Valeur, "Mathematical functions for the analysis of luminescence decays with underlying distributions 1. kohlrausch decay function (stretched exponential).," *Chemical Physics*, vol. 315, no. 1-2, pp. 171–182, 2005.

Mika Laiho (M'04) received the M.Sc., Lic.Sc., and D.Sc. degrees in Electrical engineering from Aalto University, Espoo, Finland, in 1999, 2001, and 2003, respectively. In November 2003 he started as a Postdoctoral Researcher at University of Turku, Finland. Since 2008 he has acted as an Adjunct Professor at University of Turku, where he currently holds an Academy of Finland research fellowship. He has published more than 100 papers in the areas of analog/mixed-mode processor arrays and massively parallel sensing/computing. His current research interests are in harnessing emerging memory technologies to computing, especially using locally connected architectures and associative memory circuits for cognitive tasks.

Jennifer O. Hasler (SM04) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University, Tempe, in 1991, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, in 1997. She is an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. Her current research interests include low power electronics, mixed-signal system ICs, floating-gate MOS transistors, adaptive information processing systems, smart interfaces for sensors, cooperative analog-digital signal processing, device physics related to submicron devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology. Dr. Hasler was the recipient of the National Science Foundation CAREER Award in 2001 and the Office of Naval Research YIP Award in 2002. She was also the recipient of the Paul Raphorst Best Paper Award from the IEEE Elec- tron Devices Society in 1997, the CICC Best Student Paper Award in 2006, the ISCAS Best Sensors Paper Award in 2005, and the Best Paper Award at SCI in 2001.

Jiantao Zhou (S11) received the B.S. degree in electrical engineering from Peking University, Beijing, China, in 2011. He is currently working toward the Ph.D. degree in the University of Michigan, Ann Arbor, MI. His research interests include the fabrication and characterization of novel resistive random-access memory and select device.



Chao Du received B.S. in microelectronics from Tsinghua University, Beijing, China, in 2011 and is now working towards his Ph.D. degree in the University of Michigan, Ann Arbor, MI. His main research area includes analog memristor behavior investigation and memristor-based neuromorphic network applications.

Wei Lu (M05) received B.S. in physics from Tsinghua University, Beijing, China, in 1996, and Ph.D. in physics from Rice University, Houston, TX in 2003. From 2003 to 2005, he was a postdoctoral research fellow at Harvard University, Cambridge, MA. In 2005, he joined the faculty of the EECS Department at the University of Michigan is currently an Associate Professor. His research interest includes high-density memory based on two-terminal resistive switches (RRAM), memristor-based logic circuits, aggressively scaled transistor devices, and electrical transport in low-dimensional systems. Prof. Lu is a recipient of the NSF CAREER Award, co-Editor-in-Chief for Nanoscale, a member of the IEEE, APS, MRS, and an active member of several IEEE technical committees and program committees.

Eero Lehtonen received the M.Sc. degree in Mathematics, and the D.Sc. degree in Electrical engineering from University of Turku, Finland, in 2006 and 2013, respectively. His doctoral thesis, which is comprised of 14 published papers, considers the use of memristors for computing. His research has been focused on the application of memristors and memcapacitors in ultra-low power massively parallel mixed-mode processing architectures, ranging from parallel memristive logic circuits to bio-inspired event-based computing architectures.

Jussi H. Poikonen received the M.Sc. and D.Sc. (Tech.) degrees in telecommunications from University of Turku, Turku, Finland, in 2005 and 2009, respectively. Since 2009 he has worked as Special Researcher and Lecturer at University of Turku, and in 2011 – 2014 worked as an Academy of Finland post-doctoral Researcher at the Department of Communications and Networking at Aalto university, Espoo, Finland. His research interests include computing in memristive circuits, and wireless communication systems and algorithms.