A Study on the Programming Structures for RRAM-Based FPGA Architectures

Xifan Tang, *Student Member, IEEE*, Gain Kim, *Student Member, IEEE*, Pierre-Emmanuel Gaillardon, *Member, IEEE*, and Giovanni De Micheli, *Fellow, IEEE*

***Abstract*—*Field Programmable Gate Arrays* (FPGAs) can benefit non-volatility and high-performance by exploiting *Resistive Random Access Memories* (RRAMs). In RRAM-based FPGAs, the memories do not only replace the SRAMs and store configurations, but they can also replace the transmission gates and propagate datapath signals. The high-performance achievable by RRAM-based FPGAs comes from the fact that the on-resistance of the memory devices R_{LRS} is smaller than the equivalent resistance of a transmission gate. Efficient programming structures for RRAMs should provide high current density with a small area footprint, to obtain a low R_{LRS}. In this paper, we first examine the efficiency of the widely-used 2Transistor/1RRAM (2T1R) programming structure and identify four major limitations of the 2T1R structure. To overcome these limitations, we propose a 2*Transmission-Gates*/1RRAM (2TG1R) and a 4Transistor/1RRAM (4T1R) programming structures. We perform both theoretical analysis and electrical simulations on the evaluated programming structures. 4T1R programming structure is the best in terms of current density with 1.4 $\times$ and 1.1 $\times$ as compared to 2T1R and 2TG1R counterparts, respectively. We also investigate the effect of boosting the programming voltage V_{prog} of the programming structures. Experimental results show that boosting V_{prog} for all the programming structures improves driving current of the evaluated programming structures by 3 $\times$ and area efficiency by 1.7 $\times$ on average.**

***Index Terms*—FPGA, programming structure, resistive memory.**

I. INTRODUCTION

RESISTIVE random access memory (RRAM) technology [1]–[3] is one of the most promising candidates for next generation *non-volatile memory* (NVM) technology [4]. From a circuit-level perspective, RRAMs can be regarded as programmable resistors. By applying suitable programming voltages and currents, RRAMs can be set/reset into two stable resistance states: the *low-resistance state* (LRS) and the *high-resistance state* (HRS). Compatible with CMOS fabrication techniques, RRAMs can be integrated between metal layers over transistors during a *back-end-of-line* (BEoL) process, bringing opportunities to high-density co-integration. Device parameters of RRAMs, such as resistances and programming voltages, are dependent on the employed materials, the stack architecture and the fabrication techniques [3]. The large range of achievable device properties enable RRAMs to meet the different application needs, such as dense memory arrays [5] or non-volatile FPGAs [6].

RRAM technology has attracted intensive research interests in novel FPGA architectures [6]–[12]. These proposed RRAM-based FPGA architectures employ RRAMs not only to store configurations, i.e., replacing SRAMs, but also to propagate datapath signals, i.e., replacing transmission gates. RRAMs reduce the transistor area thanks to BEoL process and lead to a higher density of integration. In addition, the low-resistance state R_{LRS} of RRAMs introduces a equivalent resistance in datapaths lower than the transmission gates, improving the performance of routing elements. Programming structures allow individual access to the RRAMs and provide the voltage and current required during the resistance state switching. Efficient programming structures for RRAMs should employ a small area footprint while providing the high current density required to program a low R_{LRS}. Previous works [6]–[11] use 2Transistor/1RRAM (2T1R) programming structure to program the RRAMs. The 2T1R programming structure consists of two *n*-type transistors. However, the *n*-type transistors may suffer serious body effects when propagating high programming voltages, weakening their current density. Therefore, it is necessary to precisely study the properties of a 2T1R programming scheme.

In this paper, we first examine the efficiency of 2T1R programming structure and identify four major limitations: 1) it provides a rather low current density due to imbalanced voltage drops across transistors; 2) its bulk connections cause serious body effects; 3) it requires driving inverters that introduce voltage drops and reduce the current density; 4) the same transistors are used for the different programming phases requiring worst case condition designs and resulting in area inefficiencies. To avoid problems 1) and 2), we introduce *transmission-gates* (TGs) in 2T1R structure, called 2TG1R programming structure. To alleviate all the 2T1R limitations, we propose a more advanced 4T1R programming structure that uses two pairs of *p-type* and *n-type* transistors to set/reset RRAMs. We perform both theoretical analysis and electrical simulations on the introduced programming circuits. Simulation results show that 4T1R programming structure is the best in terms of current density with 1.4 $\times$ and 1.1 $\times$ as compared to 2T1R and 2TG1R counterparts, respectively. We investigate the effect of

Manuscript received August 19, 2015; revised December 15, 2015; accepted January 4, 2016. Date of publication March 10, 2016; date of current version April 15, 2016. This work was supported by the Swiss National Science Foundation under the project number 200021-146600. This paper was recommended by Associate Editor M. Hashimoto.

X. Tang, P.-E. Gaillardon, and G. De Micheli are with the Integrated Systems Laboratory, School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Vaud, Switzerland (e-mail: xifan.tang@epfl.ch).

G. Kim is with the Microelectronic System Laboratory, School of Electrical Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Vaud, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2016.2528079

1549-8328 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

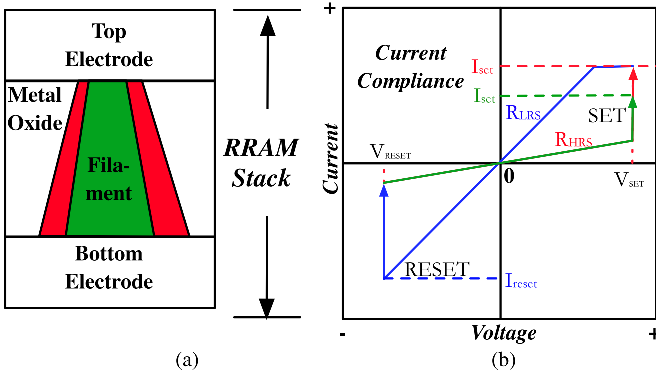


Fig. 1. (a) RRAM structure and illustration on conducting filaments; (b) bipolar RRAM I–V characterization.

boosting programming voltage $V_{\rm prog}$ on the three programming circuits. Experimental results show that boosting $V_{\rm prog}$ for all the programming structures can improve driving current by 3× and area efficiency by 1.7× on average.

The rest of this paper is organized as follows. Section II introduces background information about RRAM technology and RRAM-based FPGA architectures. Section III shows the experimental methodology used in the paper. Section IV analyzes in detail the 2T1R programming structure, while Section V and Section VI introduce the 2TG1R and 4T1R programming structures, respectively. Section VIII concludes this paper.

II. Background and Motivations

In this section, we give a brief introduction on RRAM technologies and RRAM-based FPGA architectures.

A. Overview of RRAM Technology

RRAM technologies have been heavily studied in recent years [1]–[3]. RRAMs can be considered as two-terminal programmable resistors, consisting of three layers: the top metal electrode, the switching metal oxide and the bottom metal electrode, as shown in Fig. 1(a). The conductivity of the metal oxide can be modified by applying a programming voltage between the electrodes, leading to a switching event between two stable resistance states: the *low resistance state* (LRS) and the *high resistance state* (HRS). According to its switching mechanism, RRAMs can be classified into categories, *unipolar resistive switching* (URS) and *bipolar resistive switching* (BRS) [3]. In this paper, we only focus on BRS RRAMs whose I–V characteristics are illustrated in Fig. 1(b). Before usage, RRAMs have to go through a forming process, during which the conductivity of the switching metal oxide is initialized. After forming, applying a positive programming voltage $V_{\rm set}$ induces a switching event from HRS to LRS, called the **set** process. Conversely, a negative programming voltage $V_{\rm reset}$ invokes a switching event from LRS to HRS, called the **reset** process. Note that there are threshold voltages for both $V_{\rm set}$ and $V_{\rm reset}$. Switching events in metal oxide happen when the applied programming voltage is above the threshold voltage. However, a current compliance $I_{\rm set}$ is often enforced during the **set** process to avoid a permanent breakdown of the device. Besides a threshold in programming voltage, conductivity switching in metal oxide also requires a minimum pulse width of programming voltage. The pulse width of programming voltage determines the writing speed of a RRAM device [3]. The conductivity of the switching metal oxide is determined by conducting filaments, which are formed during switching and whose widths depend on the programming current. The wider the filament is, the higher the conductivity is and the lower the $R_{\rm LRS}$ is. Therefore, a lower/higher resistance of RRAMs can be obtained by driving a higher/lower the programming current during the **set** process [16]. For example, the red filament in Fig. 1(a), which is achieved by the red set process in Fig. 1(b), leads to a lower $R_{\rm LRS}$ than the green filament in Fig. 1(a), which is achieved by the green set process in Fig. 1(b). The tunable $R_{\rm LRS}$ is a unique property of RRAM devices.

In addition, RRAMs should be able to afford a reasonably large number of writing operations, expressed by the endurance, and also should be able to maintain the resistance state for a long period without degradation, expressed by the data retention. *Back-End-Of-Line* (BEoL) technology allows RRAM to be fabricated on the top of, or between, metal layers, with a effective memory cell area as low as $4F^2$, where F is the feature size [17]. The BEoL integration leads to a low additional cost in fabrication, and the filament mechanism determines that RRAMs can be scaled down like transistors.

B. RRAM-Based FPGA Architecture

Fig. 2 describes the principles of the FPGA architecture, where an array of heterogeneous blocks is surrounded by global routing architecture. Heterogeneous blocks consist of *configurable logic blocks* (CLBs), memory banks and *digital signal processor* (DSP) blocks. The global routing architecture is built with *connection blocks* (CBs) highlighted in green, which connect heterogeneous blocks to routing tracks, and *switch boxes* (SBs) highlighted in red, which interconnect routing tracks together. Inside a CLB, there is a number of *basic logic elements* (BLEs) each of which consists of a *look up table* (LUT), a D *flip-flop* (FF), and an output selector (2:1 multiplexer). A group of multiplexers called local routing architecture interconnect the CLB input/output pins to BLE input/output pins. Modern FPGAs [18], [19] exploits more complex CLB architectures with additional circuitry such as hard carry chains or shift registers, in order to boost the performances of the architecture. Modern FPGA architectures exploit a high density of logic and routing resources, leading to $\sim 10^8$ programming bits [8].

In RRAM-based FPGAs [6]–[11], the *static random access memories* (SRAMs), used to store the configurations, are replaced with non-volatile SRAMs as shown in Fig. 2(a). In addition to non-volatility, RRAMs can also bring performance improvements to multiplexers as shown in Fig. 2(b). In the RRAM-based multiplexers, SRAMs and transmission gates are replaced by RRAMs and 2Transistors/1RRAM (2T1R) programming structures. RRAMs are employed not only for storing routing configurations but also to route signals. When programmed into LRS, the RRAMs propagate signals within the datapaths, having the same functionality as transmission

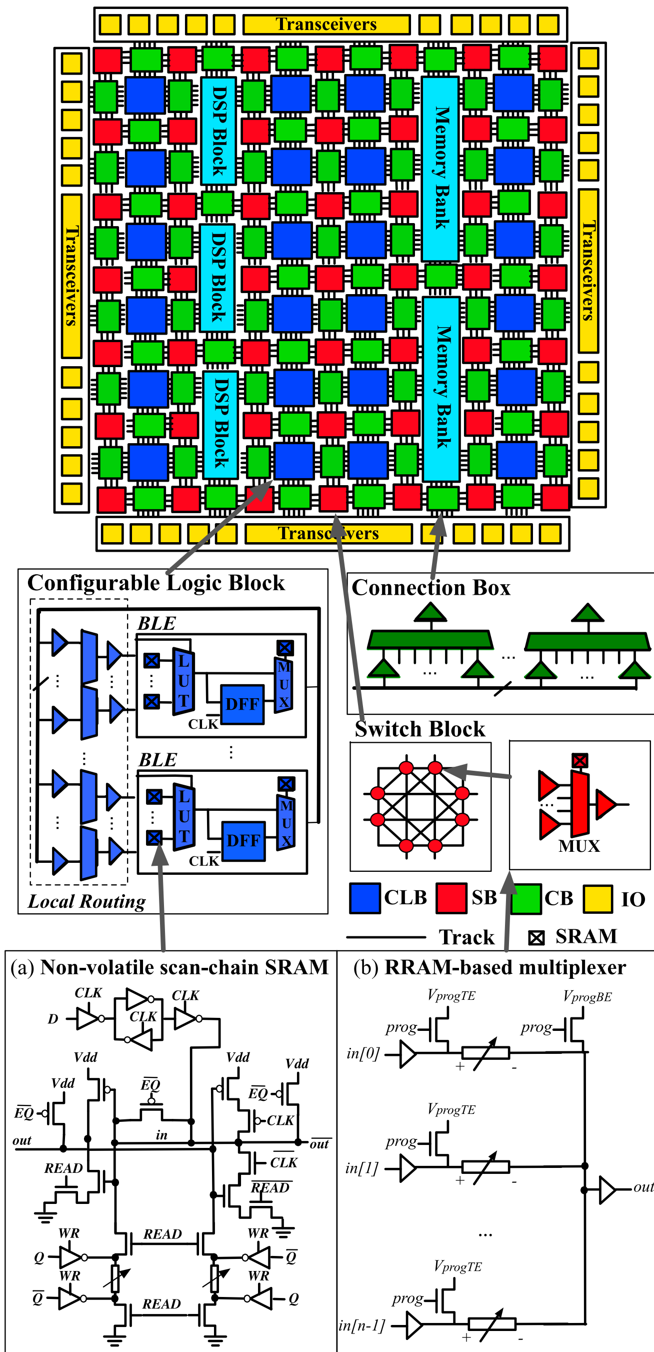


Fig. 2. RRAM-based FPGA architecture and circuit designs: (a) Non-volatile scan-chain SRAM; (b) RRAM-based multiplexer.

gates in *on* state. In contrast, when programmed into HRS, the RRAMs block signals in datapaths, corresponding to the transmission gates in *off* state. Compared to the transmission gates, the RRAMs can reduce the equivalent resistances of datapaths by up to 75%, significantly improving the performance of multiplexers.

When employed at nominal supply voltage, RRAM-based FPGAs can reduce the area by 7%–15%, increase the performance by 45%–58%, and reduce the power consumption by 20%–58%, compared to SRAM-based FPGAs [6]–[10]. Additionally, compared to transmission gates, the resistance values of RRAMs do not degrade when the operating voltage

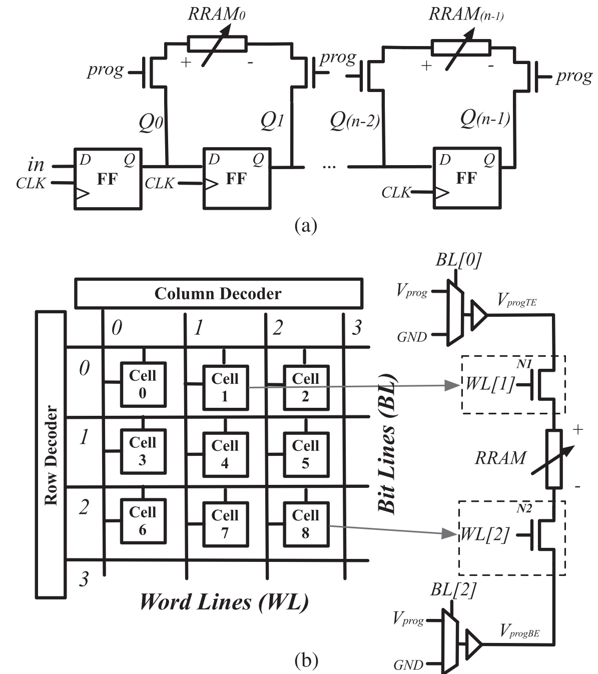


Fig. 3. System-level implementations exploiting the 2T1R programming structure: (a) scan chain [6]; (b) memory bank [8].

decreases [11]. Furthermore, the use of proper programming transistor sizing technique can further reduce the area, delay of RRAM-based circuits [11]. Operated in the near-V_t regime, RRAM-based FPGAs can achieve 20% area saving, 10% performance gain and 65% power reduction, compared to mainstream SRAM-based FPGAs [11].

C. Strategies for Individual RRAM Access

Previous works [6]–[11] mainly exploit two different strategies to access the individual 2T1R memory elements. A scan-chain organization, as shown in Fig. 3(a), has been proposed in [6] while a memory bank arrangement, as shown in Fig. 3(b), has been employed in [8]. With the scan-chain organization that is similar to modern FPGAs, RRAMs are programmed through *flip-flop* (FF) outputs when signal *prog* is set to *1*. For example, when $Q_0 = \mathtt{1}$, $Q_1 = \mathtt{0}$, a set process for RRAM_0 is started. In a memory bank arrangement, the RRAMs are programmed through *bit lines* (BLs) and *word lines* (WLs). For instance, when $\mathrm{WL}[1] = \mathtt{1}$, $\mathrm{WL}[2] = \mathtt{1}$, $\mathrm{BL}[0] = \mathtt{1}$, $\mathrm{BL}[2] = \mathtt{0}$, a set process for RRAM is initiated. Note that, with this strategy, only one RRAM is programmed at a given time—allowing to limit the programming current to be delivered to the chips.

Programming structures are of great importance for RRAM-based FPGAs, as they must provide high current to efficiently achieve low R_{LRS} while minimizing the area footprint. Previous works [6]–[11] propose that high current can be achieved by increasing the sizes of the transistors. Note that the principles in the circuit designs of programming structures are different from logic gates, because the programming structures are driving a resistive load instead of a capacitive one. To drive a resistive load like a RRAM, the source-to-drain voltages V_{DS} of transistors should be large enough in order to ensure

a high current. Moreover, when the $V_{\rm DS}$ voltage drops of the transistors take most of the supply range $V_{\rm prog}$ and the voltage difference between the RRAM electrodes goes below the programming threshold voltage, a correct programming cannot be guaranteed. To the best of our knowledge, previous literature treated RRAMs as standard CMOS loads, i.e., mostly capacitive, in the 2T1R structures [6]–[11] and the details of the associated programming circuit design were not investigated. In this work, we study the specificities and limitations of 2T1R programming structures through both theoretical analysis and electrical simulations, and propose solutions to the associated shortcomings, such as low current density and area inefficiency.

III. Experimental Methodology

In this paper, we consider the RRAM model in [13], whose $V_{\rm set}/V_{\rm reset}$ is 1.3 V/ −1.3 V respectively, $R_{\rm LRS}$ is 500 Ω, and $R_{\rm HRS}$ is 20 kΩ ($R_{\rm HRS}/R_{\rm LRS} = 40$). The current compliance $I_{\rm set}$ is set to 1 mA to avoid permanent device breakdown while the reset current $I_{\rm reset}$ is 1 mA. The minimum required pulse width for programming the RRAM element is 100 ns. The programming structures discussed in the paper are implemented with I/O transistors (W/L = 320 nm/270 nm) from a commercial 45 nm process technology. The associated transistor model is based on BSIM4. The standard $V_{\rm GS}$ and $V_{\rm DS}$ of transistors are 2.5 V. The transistors can be over-driven up to 3.0 V. The ratio between *p/n*-type transistors β is set to 3. In this paper, we also consider the area overhead of the P-well of *p*-type transistors for which a penalty factor $\gamma = 1.2$ is set.

Electrical simulations are run with HSPICE simulator [21]. The time step of electrical simulations is set to 0.1 ps. In each simulation, the RRAM is initialized to the HRS and then transistors are turned on to program the RRAM into LRS. At the end of programming period, we measure the voltage difference between the RRAM electrodes and the current passing through to calculate the LRS resistance $R_{\rm LRS}$.

We sweep two parameters: the width of transistors $W_{\rm prog}$ and the programming voltage $V_{\rm prog}$, to study their impact on the performance of programming structures. $W_{\rm prog}$ is defined as the width of the *n*-type transistors used in the structures expressed by the minimal size transistors. $W_{\rm prog}$ is swept in the range from 1 to 5 with a 0.1 step. $V_{\rm prog}$ is swept in the range from 2.5 V to 3.0 V with a 0.1 V step.

Note that, to achieve significant FPGA improvements, a $R_{\rm HRS}$ of at least 20 MΩ must be employed [23]. However, as the presented methodology and structures are general for any device parameters and for the sake of reproducibility, we present results using the base parameters of the RRAM model in [13].

IV. 2T1R Programming Structure

In this section, we first introduce the circuit design of the 2T1R programming structure and then discuss the limitations of the 2T1R structure with theoretical analysis and electrical simulations. In the rest of the paper, we focus on the set process when conducting theoretical analysis. Without loss of generality, our approach can be applied to the reset process as well.

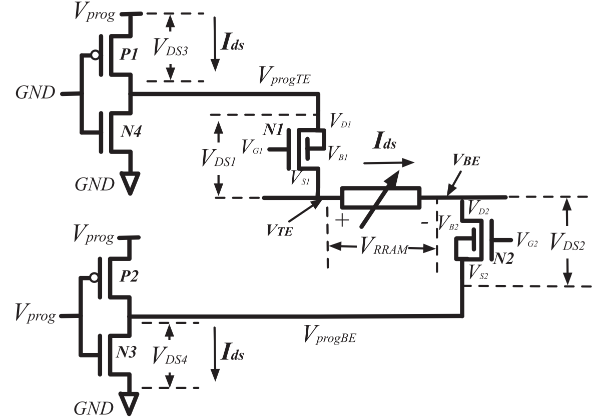


Fig. 4. A 2T1R programming structure extracted from system-level implementations in Fig. 3.

A. 2T1R Circuit Structure

In Fig. 4, we extract a 2T1R structure along with its driving inverters from the system-level implementation shown in Fig. 3. A 2T1R structure requires driving inverters to provide the voltage levels of $V_{\rm progTE}$ and $V_{\rm progBE}$ during a programming phase. In a **set** process, the terminals of 2T1R structure $V_{\rm progTE}$ and $V_{\rm progBE}$ are driven by a *p*-type transistor **P1** and a *n*-type transistor **N3**, respectively. As illustrated in Fig. 4, the driving inverters introduce two potential voltage drops caused by the drain-to-source voltage $V_{\rm DS3}$ and $V_{\rm DS4}$ of transistors **P1** and **N3**, while the 2T1R structure has two built-in voltage drops caused by $V_{\rm DS1}$ and $V_{\rm DS2}$ of transistors **N1** and **N2**. In a **reset** process, the terminals of 2T1R structure $V_{\rm progTE}$ and $V_{\rm progBE}$ are driven by a *n*-type transistor **N4** and a *p*-type transistor **P2**, respectively. Similar, another two drain-to-source voltage drops of transistors **P2** and **N4** are introduced. To avoid the effect of $V_{\rm DS3}$ and $V_{\rm DS4}$, the sizes of transistors **P1** and **N3** have to be far larger than **N1** and **N2**, so that $V_{\rm DS3}$ and $V_{\rm DS4}$ can be neglected compared to $V_{\rm DS1}$ and $V_{\rm DS2}$. We take this assumption in the rest of the analysis.

B. *I–V* Characteristics of 2T1R Structure

In this part, we consider the voltage drops $V_{\rm DS1}$ and $V_{\rm DS2}$ in Fig. 4 and discuss the *I–V* characteristics of a 2T1R structure. By considering Kirchhoff circuit laws

$$\begin{cases} I_{\rm ds} = f(V_{\rm GS1}, V_{\rm DS1}) = f(V_{\rm GS2}, V_{\rm DS2}) \\ V_{\rm RRAM} = I_{\rm ds} R_{\rm RRAM} \\ V_{\rm prog} = V_{\rm DS1} + V_{\rm DS2} + V_{\rm RRAM}. \end{cases} \quad (1)$$

where $I_{\rm ds}$ is the current passing through the transistors and RRAM. $R_{\rm RRAM}$ denotes resistance of RRAM. $f(V_{\rm GS1}, V_{\rm DS1})$ and $f(V_{\rm GS2}, V_{\rm DS2})$ represent the *I–V* relationships of transistors **N1** and **N2** in Fig. 4. To give an intuition on the operating points of transistors, we consider the following transistor model:

$$I_{\rm ds} = \begin{cases} k_n \frac{W}{L}\left[(V_{\rm GS} - V_T)V_{\rm DS} - \frac{1}{2}V_{\rm DS}^2\right], & V_{\rm DS} < V_{\rm GS} - V_T \\ \frac{1}{2}k_n \frac{W}{L}(V_{\rm GS} - V_T)^2, & V_{\rm DS} \geq V_{\rm GS} - V_T \end{cases} \quad (2)$$

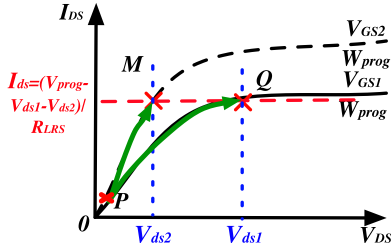


Fig. 5. I–V characteristics of the 2T1R structure.

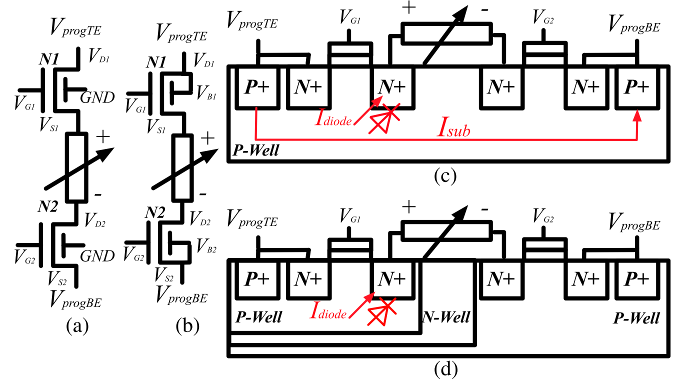


Fig. 6. (a) Asymmetric bulk management of the 2T1R structure. (b) Symmetric bulk management of the 2T1R structure. (c) Single well application of layout. (d) Triple well application of layout.

where k_n denotes the process transconductance parameter of a *n-type* transistor and V_T represents its threshold voltage. W and L are the width and length of channel, respectively. $V_{\rm GS}$ is the voltage difference between the gate and source terminals. $V_{\rm DS}$ is the voltage difference between the drain and source terminals. The intuitive results obtained with the model will be subsequently validated by SPICE simulations. In the theoretical analysis, we focus on studying how the current $I_{\rm ds}$ is changed with $V_{\rm GS1}$, $V_{\rm GS2}$, $V_{\rm DS1}$ and $V_{\rm DS2}$ during a set programming phase.

Fig. 5 illustrates the I–V curve of the transistors **N1** and **N2** during the programming phase. A programming phase starts when the transistors **N1** and **N2** are turned *on* and the RRAM is in HRS. At the start point **P**, $I_{\rm ds}$ is close to zero because the HRS resistance $R_{\rm HRS}$ of the RRAM typically is very high, leading to $V_{\rm DS1}$ and $V_{\rm DS2}$ approaching zero. $V_{\rm RRAM}$ is above the programming threshold voltage $V_{\rm set}$, and therefore a resistive transition occurs and the resistance decreases. Note that $V_{\rm GS2}$ equals to V_{G2} because the source voltage of transistors **N2** is GND, while $V_{\rm GS1} = V_{G1} - V_{\rm TE}$, is much smaller than $V_{\rm GS2}$. Then, the resistance of the RRAM is gradually decreasing from $R_{\rm HRS}$ to $R_{\rm LRS}$, leading to an increase in $I_{\rm ds}$. The growth in $I_{\rm ds}$ creates a positive feedback: $V_{\rm DS1}$ and $V_{\rm DS2}$ are increasing to provide a higher current which leads the voltage difference across the RRAM to decrease. The positive feedback continues until the $V_{\rm RRAM}$ reaches the $V_{\rm set}$ of the RRAM, i.e., the memory cannot switch anymore. At this point, $I_{\rm ds}$, $V_{\rm DS1}$ and $V_{\rm DS2}$ reach their peak values. Note that during the programming phase, $V_{\rm GS1}$ is increasing as the source voltage of transistors **N1**, $V_{\rm TE}$, is decreasing, but it is still smaller than $V_{\rm GS2}$. The difference in $V_{\rm GS}$ causes a $V_{\rm DS}$ gap because $V_{\rm DS1}$ has to be larger than $V_{\rm DS2}$ in order to drive the same current. Therefore, transistor **N1** may work in deep linear region or even saturation region while transistor **N2** has to work in linear region, causing the programming current to be much lower than what saturated transistors can offer.

Boosting $V_{\rm prog}$ can reduce the difference between $V_{\rm DS1}$ and $V_{\rm DS2}$, improving the driving strength of transistors. Its effort will be studied by electrical simulations.

C. Considerations About Bulk Connections

Typically, in digital circuit designs, the bulks of n-type transistors are connected to GND, as shown in Fig. 6(a). However, the regular bulk connections for the 2T1R structure causes serious body effects. In a set process where $V_{\rm progTE} \approx V_{\rm prog}$ and $V_{\rm progBE} \approx GND$, the $V_{\rm SB} = V_{S1}$ of transistor **N1** in Fig. 6(a) is larger than $V_{\rm set} = V_{S1} - V_{D2}$, which leads to a high threshold voltage of transistor **N1** and reduces its driving strength. Note that the $V_{\rm SB}$ of transistor **N2** is negligible due to the $V_{\rm DS3}$ and $V_{\rm DS4}$ and its driving strength is reduced as well. Similar conclusion can be drawn in a reset process where $V_{\rm progTE} \approx GND$ and $V_{\rm progBE} \approx V_{\rm prog}$.

To alleviate the serious body effect, a symmetric bulk connection can be envisaged as shown in Fig. 6(b). When $V_{\rm progTE} \approx V_{\rm prog}$ and $V_{\rm progBE} \approx GND$, the $V_{\rm SB}$ of transistor **N1** equals to $V_{\rm DS}$ which is smaller than in the previous case and improves the driving strength. The $V_{\rm SB}$ of transistor **N2** is strictly zero, totally eliminating the body effect. Similar conclusion can be drawn when $V_{\rm progTE} \approx GND$ and $V_{\rm progBE} \approx V_{\rm prog}$.

However, when a symmetric bulk is implemented with a single-well technology as shown in Fig. 6(c), the substrate is connected to two voltage sources $V_{\rm progTE} \approx V_{\rm prog}$ and $V_{\rm progBE} \approx GND$, resulting in a high leakage current $I_{\rm sub}$. Besides, the junction diode at the source of transistor **N1** is positively biased, introducing another high leakage current $I_{\rm diode}$. $I_{\rm sub}$ can be reduced to zero with a triple-well technology as shown in Fig. 6(d), but $I_{\rm diode}$ remains a concern. In short, there exist serious problems in connecting the bulks of 2T1R structure, limiting its feasibility from a physical design perspective.

D. Area Estimation

We estimate the area of the programming structures in terms of minimal size transistors. While we only considered the set process, it is worth noticing that in the 2T1R structure, the same transistors **N1** and **N2** are used in reset process as well. Typically, the reset current is not the same as the set current [3]. To be applicable in both set and reset, the size of transistors **N1** and **N2** should be determined by the largest of set/reset currents. Assume $W_{\rm prog,set}$ and $W_{\rm prog,reset}$ are the transistor sizes required for the set and reset operations, respectively. In the context of a memory bank, we assume that a driving inverter for a BL is shared by N 2T1R structures

$$\begin{cases} 2W_{\rm prog,set} + 2\cdot(1+\beta\gamma)W_{\rm inv}/N, & I_{\rm set} \geq I_{\rm reset} \\ 2W_{\rm prog,reset} + 2\cdot(1+\beta\gamma)W_{\rm inv}/N, & I_{\rm set} < I_{\rm reset} \end{cases} \quad (3)$$

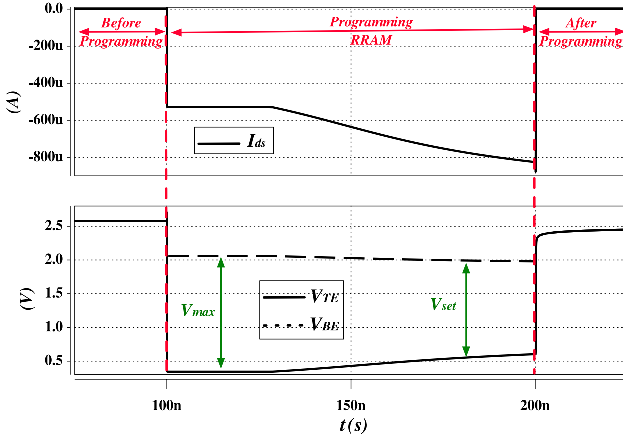


Fig. 7. Transient analysis on voltages and current in the 2T1R structure during a set process ($W_{\rm prog} = 5$, $V_{\rm prog} = 3.0$ V, $W_{\rm inv} = 20$, 1 $W_{\rm prog} = 320$ nm).

where β is the ratio of p-type and n-type transistors and γ is the penalty factor for the area overhead of the P-well of p-type transistors. $W_{\rm inv}$ is the size of driving inverters. When the set current is larger than the reset current, the area is determined by $W_{\rm prog,set}$. When the reset current is larger than the set current, the area is determined by $W_{\rm prog,reset}$. In this case, during the **set** process, transistor **N1** and **N2** should be under-drived by reducing V_{G1}, V_{G2} and $V_{\rm prog}$ to respect the current compliance. Unlike the $W_{\rm prog,set}$, a large $W_{\rm prog,reset}$ does not contribute to a high $R_{\rm HRS}$. In others words, a large $W_{\rm prog,reset}$ does not improve the performance as the $W_{\rm prog,set}$ does. Therefore, when $I_{\rm set} < I_{\rm reset}$, the area consumed by a large $W_{\rm prog,reset}$ is not directly contributing to a performance improvement.

E. Electrical Simulations

First, we validate our theoretical intuitions by presenting the SPICE transient analysis of the 2T1R structure. Then, we show the SPICE results of the $V_{\rm DS}$ and programming current $I_{\rm ds}$ of the 2T1R structure.

1) Transient Analysis: Fig. 7 illustrates current and voltage waveforms of the 2T1R structure during a set process. After the transistors are turned *on*, a voltage difference $V_{\rm MAX}$ between the RRAM electrodes is applied, initiating the set transition on the memory. The reduction on the resistance of the RRAM leads to an increase in $I_{\rm ds}$. To support the growing $I_{\rm ds}$, the $V_{\rm DS}$ of transistors have to increase, leading to $V_{\rm TE}$ is decreasing and $V_{\rm BE}$ is increasing. The RRAM stays in programming phase until $V_{\rm TE} - V_{\rm BE}$ reaches the threshold voltage $V_{\rm set}$.

2) $V_{\rm DS}$ of Transistors N1 and N2: Fig. 8(a) shows the trend of $V_{\rm DS}$ in a 2T1R structure by sweeping $W_{\rm prog}$ and $V_{\rm prog}$, where $W_{\rm inv}$ is 20 in order to keep $V_{\rm DS3}$ and $V_{\rm DS4}$ negligible. The $V_{\rm DS}$ difference reaches 0.65 V when $V_{\rm prog} =$ 2.5 V on average. Boosting $V_{\rm prog}$ can reduce the $V_{\rm DS}$ difference down to 0.5 V. A larger $V_{\rm prog}$ can increase the $V_{\rm DS2}$ by 2.8 $\times$. Fig. 8(b) depicts the trend of $V_{\rm DS}$ in 2T1R structure by sweeping $W_{\rm prog}$ and $W_{\rm inv}$, where $V_{\rm prog}$ is 3.0 V. Increasing $W_{\rm inv}$ can effectively reduce the $V_{\rm DS}$ gap by 15%.

3) Programming Current $I_{\rm ds}$: The achievable programming currents $I_{\rm ds}$ are determined by $V_{\rm DS}$. A high $V_{\rm prog}$ can increase

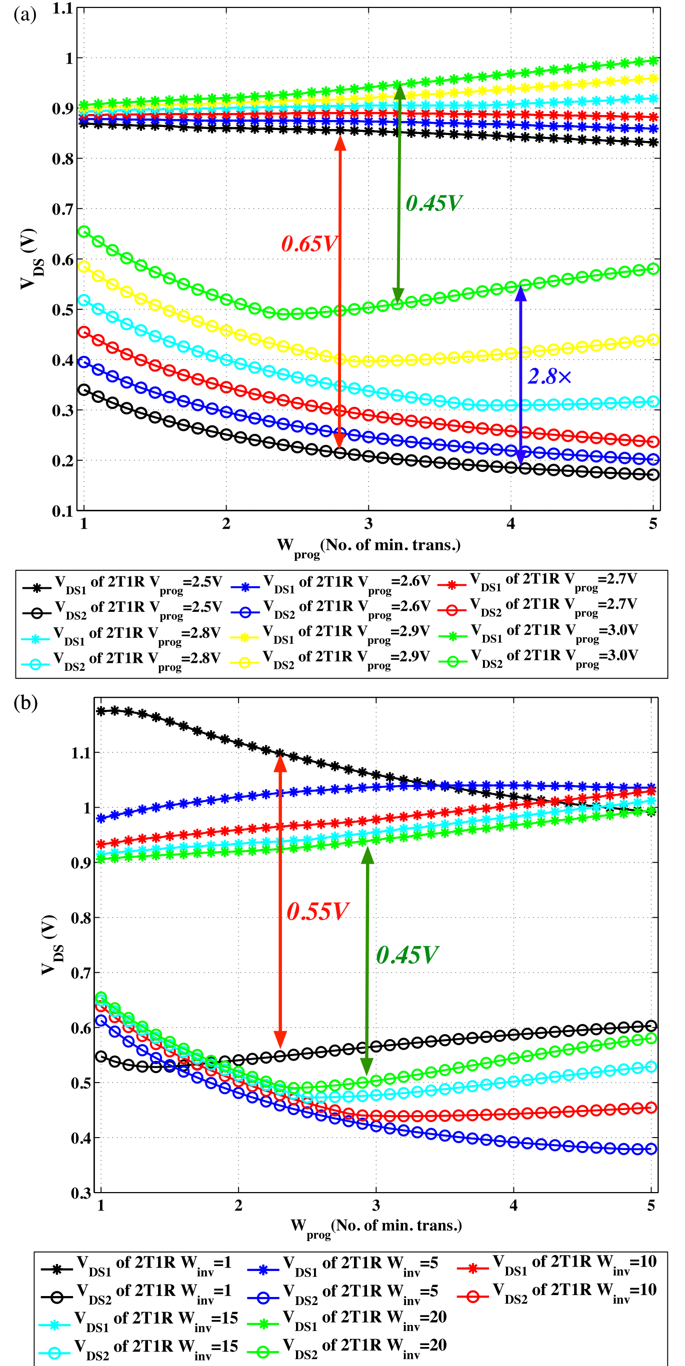


Fig. 8. (a) $V_{\rm DS1}$ and $V_{\rm DS2}$ in 2T1R structure under diverse $V_{\rm prog}$ ($W_{\rm inv} = 20$); (b) $V_{\rm DS1}$ and $V_{\rm DS2}$ in 2T1R structure under diverse $W_{\rm inv}$ ($V_{\rm prog} = 3.0$ V). (1 $W_{\rm prog} = 320$ nm).

the $V_{\rm DS}$, as explained in Section IV-B. Fig. 9(a) illustrates that for the same $W_{\rm inv}$, we can improve 3.4 $\times$ $I_{\rm ds}$ by boosting $V_{\rm prog}$ from 2.5 V to 3.0 V on average. $W_{\rm inv}$ is another important factor that influences the $I_{\rm ds}$. A large $W_{\rm inv}$ can reduce $V_{\rm DS3}$ and $V_{\rm DS4}$ while increase $V_{\rm DS1}$ and $V_{\rm DS2}$. As shown in Fig. 9(b), a large $W_{\rm inv}$, such as 20, leads to a 3.8 $\times$ higher $I_{\rm ds}$ than the smallest $W_{\rm inv} = 1$ on average. In short, boosting $V_{\rm prog}$ is an efficient method in improving $I_{\rm ds}$, which avoids the use of large transistors. A large $W_{\rm inv}$ (i.e., $= 20$) must be applied to avoid a serious degradation on $I_{\rm ds}$.

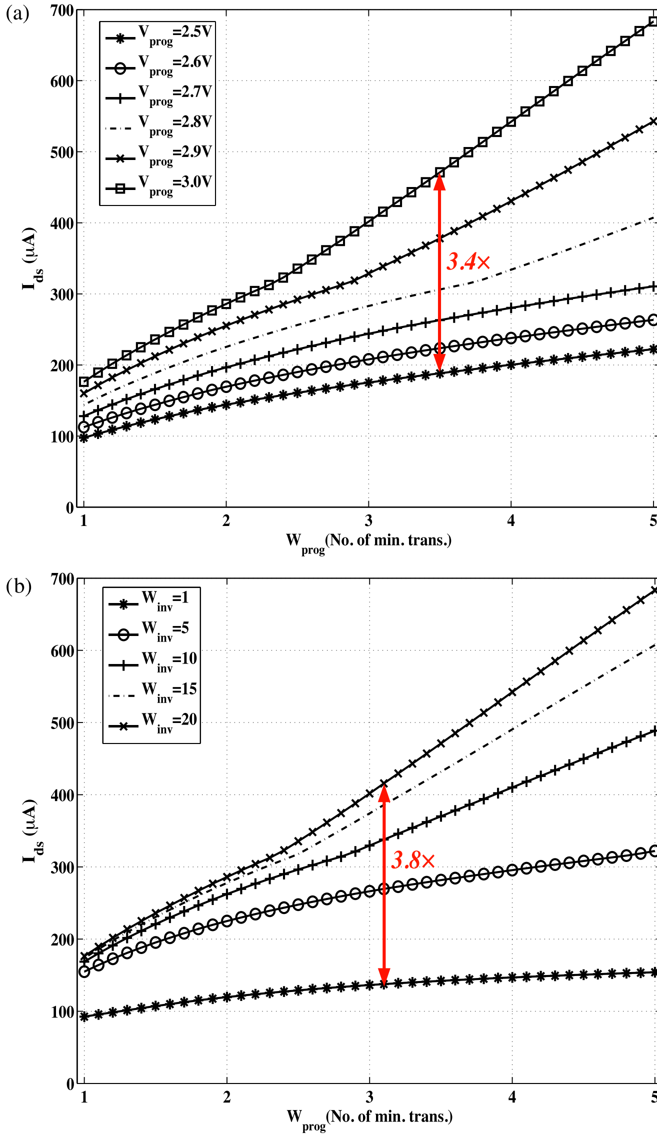


Fig. 9. (a) I_{ds} in 2T1R structure under diverse V_{prog} ($W_{\mathrm{inv}} = 20$); (b) I_{ds} in 2T1R structure under diverse W_{inv} ($V_{\mathrm{prog}} = 3.0$ V). (1 $W_{\mathrm{prog}} = 320$ nm).

F. Discussion About Limitations

From theoretical analysis and electrical simulations, we see four major limitations of 2T1R structure:

1) Its current density is low due to the intrinsic low V_{DS2};
2) Its bulk connections lead to a high leakage current;
3) Its current density is weakened by a small W_{inv};
4) Its area is bounded by the maximum of $W_{\mathrm{prog,set}}$ and $W_{\mathrm{prog,reset}}$, which is not efficient when I_{reset} is large.
5) It is not manufacturable due to the layout issues. Hence, in the rest of the paper, we only refer to it when comparing the current density.

V. 2TG1R Programming Structure

In this section, we improve the previous 2T1R circuit by replacing the n-type transistors and propose a 2TG1R programming structure. The 2TG1R circuit, comprising of four transistors, increases the current density significantly and overcomes the bulk management problem. The solution is validated using the electrical simulations.

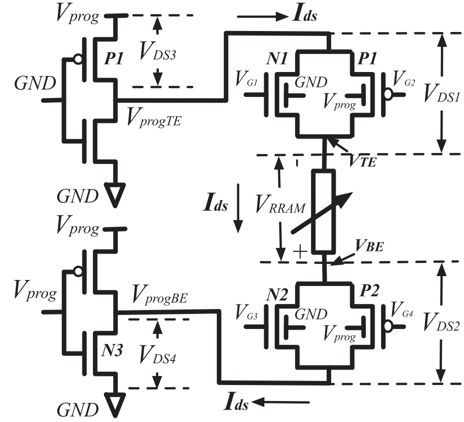


Fig. 10. A 2TG1R programming structure extracted from system-level implementations in Fig. 3.

A. 2TG1R Circuit Structure

Replacing the n-type transistors in 2T1R structure with transmission gates is a solution to the bulk management and driving strength. As shown in Fig. 10, the bulks of the n-type and p-type transistors (in total 4 transistors) are connected respectively to the highest and lowest potentials, similarly to common digital design practice, removing the bulk leakage and body effects. The driving inverters are still required to provide the voltage levels of V_{progTE} and V_{progBE} during the programming phases. Whatever in a set or reset process, there always exist a p-type transistor and a n-type transistor whose $V_{\mathrm{SB}} = 0$. Therefore, these two transistors whose $V_{\mathrm{SB}} = 0$ can provide higher current than 2T1R structure. Although the other two transistors (weak p-type and weak n-type) suffer serious body effects, they still contribute to the currents. Hence, the total current offered by 2TG1R structure is higher than 2T1R structure.

B. Area Estimation

We consider the area of a 2TG1R structure in the context of a memory bank as well. By considering the area of two p-type transistors, the area of a 2TG1R structure is:

$$\begin{cases} 2\cdot(1+\beta\gamma)W_{\mathrm{prog,set}}+2\cdot(1+\beta\gamma)W_{\mathrm{inv}}/N, & I_{\mathrm{set}}\geq I_{\mathrm{reset}}\\ 2\cdot(1+\beta\gamma)W_{\mathrm{prog,reset}}+2\cdot(1+\beta\gamma)W_{\mathrm{inv}}/N, & I_{\mathrm{set}}<I_{\mathrm{reset}}. \end{cases} \tag{4}$$

In summary, the area of 2TG1R circuit is still bounded to the largest of $W_{\mathrm{prog,set}}$ and $W_{\mathrm{prog,reset}}$. When $I_{\mathrm{set}} < I_{\mathrm{reset}}$, area investment on $W_{\mathrm{prog,reset}}$ does not bring any improvement on performance. This is extremely inefficient when $W_{\mathrm{prog,reset}}$ is large. A 2TG1R circuit leads to a even larger area overhead than 2T1R structure due to the use of p-type transistors.

C. Electrical Simulations

In this section, we show the electrical simulation results of 2TG1R structure. We focus on the improvements on V_{DS} and I_{ds} of 2TG1R structure, compared to the baseline 2T1R element.

1) Transient Analysis: Basically, the waveforms of the transient analysis on a 2TG1R are the same as 2T1R structure. The only difference lies in the slope rate of V_{TE} and V_{BE} during the programming phase. In 2TG1R, V_{TE} decreases at the same rate

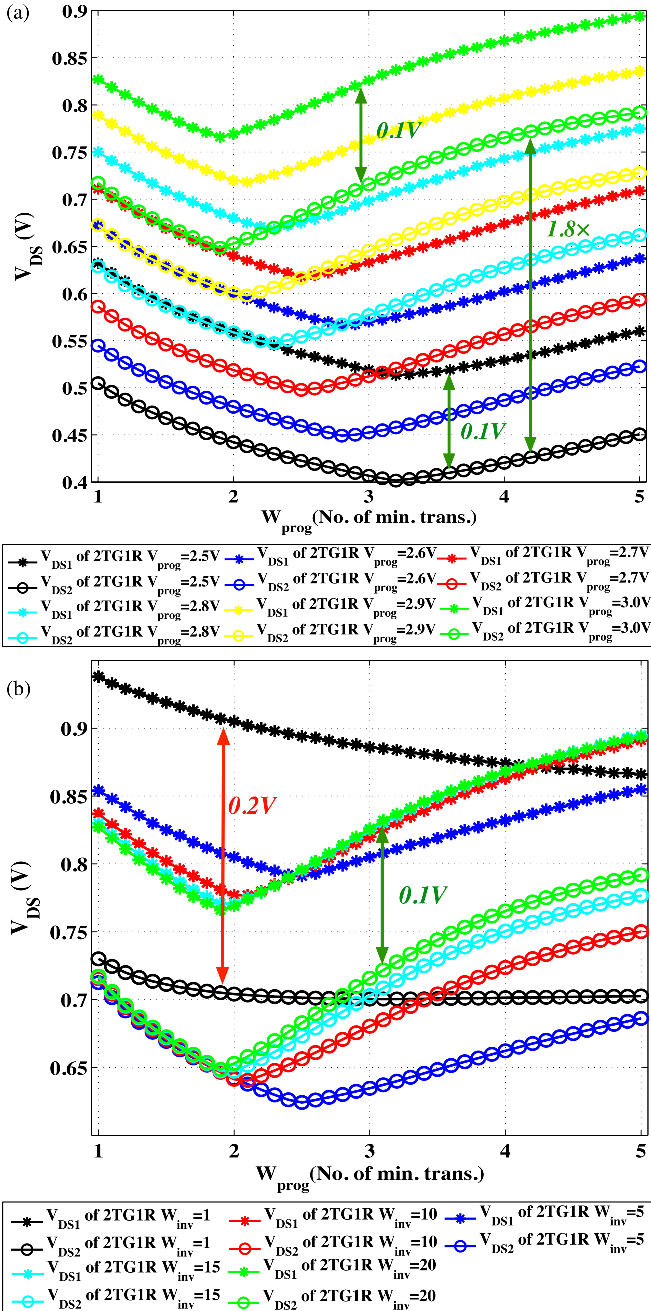


Fig. 11. (a) $V_{\rm DS1}$ and $V_{\rm DS2}$ in 2TG1R structure under diverse $V_{\rm prog}$ ($W_{\rm inv} = 20$); (b) $V_{\rm DS1}$ and $V_{\rm DS2}$ in 2TG1R structure under diverse $W_{\rm inv}$ ($V_{\rm prog} = 3.0$ V). (1 $W_{\rm prog} = 320$ nm).

as $V_{\rm BE}$ increases. In the other word, $V_{\rm DS1}$ and $V_{\rm DS2}$ in 2TG1R grow at the same rate.

2) $V_{\rm DS}$ Gap Improvement: As shown in Fig. 11(a) and (b), a 2TG1R structure reduces the $V_{\rm DS}$ gap by 5×, compared to a 2T1R structure. Like the 2T1R structure, boosting $V_{\rm prog}$ can improve $V_{\rm DS2}$ of 2TG1R by 1.8×. However, a 2TG1R still requires a large $W_{\rm inv} = 20$ to avoid the degradation on $V_{\rm DS}$ gap, coming from a non-negligible $V_{\rm DS3}$ and $V_{\rm DS4}$. When $W_{\rm inv} = 1$, the $V_{\rm DS}$ gap degrades by 2×.

3) Programming Current $I_{\rm ds}$: Boosting $V_{\rm prog}$ and $W_{\rm inv}$ achieves a similar effect on the $I_{\rm ds}$ than on the 2T1R structure. Boosting $V_{\rm prog}$ can improve $I_{\rm ds}$ of 2TG1R by 1.8×. Increasing $W_{\rm inv}$ from 1 to 20 can improve $I_{\rm ds}$ of 2TG1R by 4.3×. The $I_{\rm ds}$ of 2TG1R is 1.2× higher than 2T1R structure.

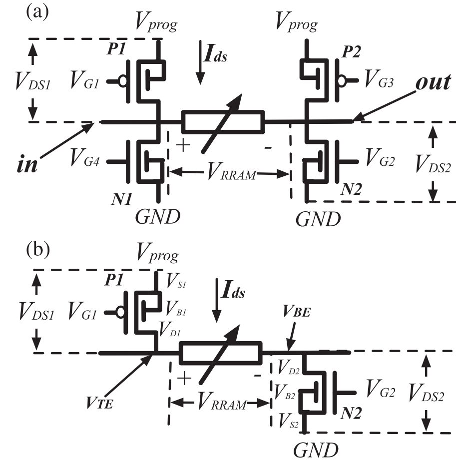


Fig. 12. (a) The proposed 4T1R structure. (b) Extracted 4T1R structure in a **set** process.

D. *Summary: Advantages and Limitations*

From theoretical analysis and electrical simulations, 2TG1R structures have the following advantages over 2T1R structure:

1) The $V_{\rm DS}$ gap is reduced by 5×, contributing to a 1.2× improvement in $I_{\rm ds}$;
2) Its bulk connections are regular, removing the bulk leakage and body effects.

However, the 2TG1R still shares two limitations with the 2T1R structure:

1) Large driving inverters are still needed to avoid current density degradation;
2) The area is still constrained by the worse case of $W_{\rm prog,set}$ and $W_{\rm prog,reset}$, which is inefficient when $I_{\rm set} < I_{\rm reset}$ and $W_{\rm prog,reset}$ is large.

VI. 4T1R Programming Structure

In this section, we propose a 4T1R programming structure able to alleviate the addressed limitations of 2T1R programming structures. We first introduce the circuit design and conduct theoretical analysis. Then, we compare the 4T1R structure with 2T1R and 2TG1R structures using electrical simulations.

A. *4T1R Circuit Structure*

Fig. 12(a) illustrates the schematic of the 4T1R structure which consists of two *p*-type transistors **P1** and **P2** and two *n*-type transistors **N1** and **N2**. The sources of the transistors in the 4T1R structure are directly connected to the voltage supplies, eliminating the driving inverters used with the 2T1R and 2TG1R solutions. The programming phase is launched by appropriately biasing the gates of the transistors. In a **set** process, the transistors **P1** and **N2** are turned *on* while the transistor **P2** and **N1** are turned *off*, applying a positive programming voltage between $V_{\rm TE}$ and $V_{\rm BE}$, as shown in Fig. 12(b). Conversely, when the transistors **P2** and **N1** are turned *on* and the transistors **P1** and **N2** are turned *off*, applying a negative voltage between $V_{\rm TE}$ and $V_{\rm BE}$, a **reset** process is operated. When the

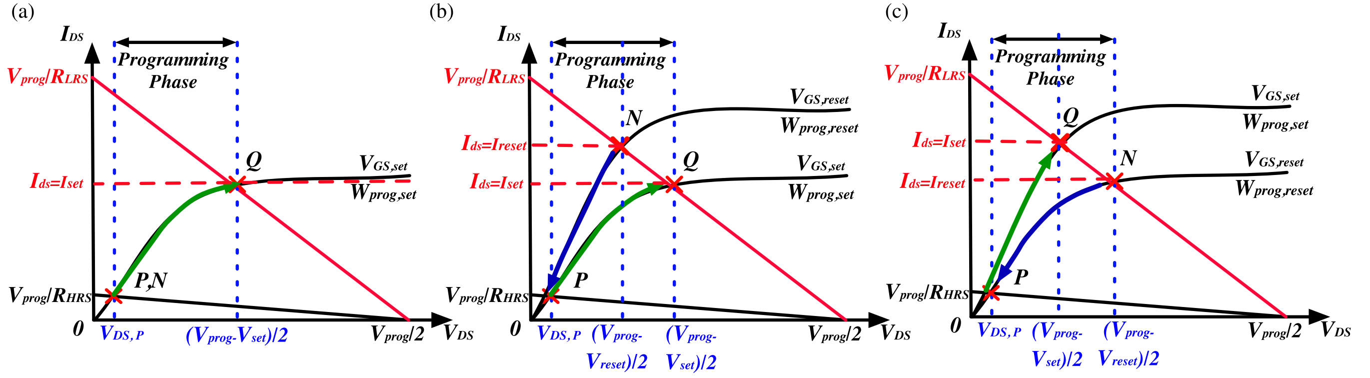


Fig. 13. I–V characteristics of the 4T1R structure: (a) $V_{\rm set} = V_{\rm reset}$; (b) $V_{\rm set} < V_{\rm reset}$ or $I_{\rm set} < I_{\rm reset}$; (c) $V_{\rm set} > V_{\rm reset}$ or $I_{\rm set} > I_{\rm reset}$.

programming segment is finished, all the transistors are turned *off*. The 4T1R structure is compatible to the system-level implementations in Fig. 3. In a scan-chain organization, V_{G1}, V_{G2}, V_{G3}, V_{G4} can be connected to $\overline{Q_0}$, Q_0, $\overline{Q_1}$, Q_1, respectively. In a memory bank organization, V_{G1}, V_{G2}, V_{G3}, V_{G4} can be connected to $\overline{BL[0]}$, $WL[2]$, $\overline{BL[2]}$, $WL[1]$, respectively.

B. Theoretical Analysis on I–V Characteristics

We first focus on the set process [Fig. 12(b)]. By applying Kirchhoff Circuit Laws, we can express the following relationships:

$$\begin{cases} I_{\rm ds} = f(V_{\rm GS1}, V_{\rm DS1}) = f(V_{\rm GS2}, V_{\rm DS2}) \\ V_{\rm RRAM} = I_{\rm ds} R_{\rm RRAM} \\ V_{\rm prog} = V_{\rm DS1} + V_{\rm DS2} + V_{\rm RRAM}. \end{cases} \quad (5)$$

$V_{\rm DS1}$ and $V_{\rm DS2}$ represent the drain-to-source voltages of transistors **P1** and **N2**, respectively. $V_{\rm GS1}$ and $V_{\rm GS2}$ represent the gate-to-source voltages of transistors **P1** and **N2**, respectively. Note that in the 4T1R structure, the sources of the transistors are connected to constant voltage supplies, giving stable $V_{\rm GS}$ during the programming phase. We can set $V_{\rm GS1} = V_{\rm GS2}$. According to the basic transistor model shown in (2), when $V_{\rm GS1} = V_{\rm GS2}$, we can find

$$V_{\rm DS} = V_{\rm DS1} = V_{\rm DS2}. \quad (6)$$

Combining (5) and (6), we can reach

$$I_{\rm ds} = \frac{V_{\rm prog}}{R_{\rm RRAM}} - \frac{2}{R_{\rm RRAM}} V_{\rm DS}. \quad (7)$$

We plot the I–V curves of (2) and (7) in Fig. 13(a). The crossing points **P** $(\sim 0, V_{\rm prog}/R_{\rm HRS})$ and **Q** $((V_{\rm prog} - V_{\rm set})/2, I_{\rm set})$ in Fig. 13(a) represent the starting and end points of a **set** procedure. From **P** to **Q**, $V_{\rm DS}$ gradually increases to provide a large $I_{\rm ds}$. On the other side, $R_{\rm RRAM}$ decreases as $I_{\rm ds}$ grows. The increment of $I_{\rm ds}$ further induces a increase in $V_{\rm DS}$ and a decrease in $R_{\rm RRAM}$. When $V_{\rm RRAM}$ reaches the threshold programming voltage $V_{\rm set}$ of the RRAM, the **set** process stops [point **Q** in Fig. 13(a)]. We can determine $V_{{\rm DS},Q} = (V_{\rm prog} - V_{\rm set})/2$ and $I_{{\rm ds},Q} = V_{\rm set}/R_{{\rm RRAM},Q}$ at the ending point **Q**. Note that $R_{{\rm RRAM},Q}$ is the programmed $R_{\rm LRS}$ of the RRAM while $R_{{\rm RRAM},P}$ is $R_{\rm HRS}$ of the RRAM.

In the reset process, let $V_{\rm reset}$ be the threshold programming voltage of the RRAM. The I–V curve of reset process could be different from set process because of the technological constraints ($V_{\rm reset}$ and $I_{\rm reset}$). Fig. 13 illustrates the three cases that could happen during a reset process. Similar to the analysis in set process, we define the operating point **P** $(\sim 0, V_{\rm prog}/R_{\rm HRS})$ as the ending point of a reset process and the operating point **N** $((V_{\rm prog} - V_{\rm reset})/2, I_{\rm reset})$ as the starting point of a reset process. Fig. 13(a) is applicable to all the conditions where $V_{\rm set} \geq V_{\rm reset}$, $I_{\rm set} \geq I_{\rm reset}$, where point **N** overlaps point **Q**. In this case, the reset process is an exact reverse trace of the set process. Fig. 13(b) covers the most difficult condition: $V_{\rm set} < V_{\rm reset}$ and $I_{\rm set} < I_{\rm reset}$. Compared to the set, the starting point **N** of the reset process is most stringent. As a result, a $W_{\rm prog,reset}/V_{\rm GS}$,reset larger than $W_{\rm prog,set}/V_{\rm GS}$,set will have to be used to reach point **N**. Note that Fig. 13(b) is applicable for other conditions where either $V_{\rm set} < V_{\rm reset}$ or $I_{\rm set} < I_{\rm reset}$ happens. Finally, Fig. 13(c) covers another case where $V_{\rm set} > V_{\rm reset}$ and $I_{\rm set} > I_{\rm reset}$, while the case shown in Fig. 13(a) still applies in the case, it would result in an oversizing for the reset process. In the case of Fig. 13(c), the starting point of reset process **N** leads to a smaller $W_{\rm prog,reset}/V_{\rm GS}$, reset than $W_{\rm prog,set}/V_{\rm GS,set}$.

Note that Fig. 13 reveals another shortcoming of 2T1R and 2TG1R structures, which use the same programming transistors for both the set and the reset processes. Due to this fact, they must be sized according to the worse case $\max\{W_{\rm prog,set}, W_{\rm prog,reset}\}$. Hence, for the conditions illustrated in Fig. 13(b) and (c), the 2T1R and 2TG1R structures have to use two different $V_{\rm GS}$ for the set and the reset processes ($V_{\rm GS,set} \neq V_{\rm GS,reset}$). When two different $V_{\rm GS}$ are needed, the system-level implementations in Fig. 3 will require additional circuitry for generating controlling signals, i.e., WL[1] and WL[2] should have three voltage levels: $V_{\rm GS,set}$, $V_{\rm GS,reset}$, and GND.

C. Current Density Boosting Methodologies

$V_{\rm prog}$ and $W_{\rm prog}$ are the two controllable parameters for circuit designers to boost $I_{{\rm ds},Q}$. In this part, depending on the working regions of the crossing point **Q**, we investigate the boosting methodologies for $I_{{\rm ds},Q}$ by tuning $V_{\rm prog}$ and $W_{\rm prog}$.

1) Linear Region: When the transistors work in the linear region at the crossing point **Q**, we can obtain the following equations:

$$\begin{cases} I_{\mathrm{ds},Q} = k_n \frac{W_{\mathrm{prog}}}{L} \left[(V_{\mathrm{GS}} - V_T) V_{\mathrm{DS},Q} - \frac{1}{2} V_{\mathrm{DS},Q}^2 \right] \\ V_{\mathrm{DS},Q} < V_{\mathrm{GS}} - V_T \\ I_{\mathrm{ds},Q} = \frac{(V_{\mathrm{prog}} - 2V_{\mathrm{DS},Q})}{R_{\mathrm{RRAM},Q}} \\ V_{\mathrm{DS},Q} = \frac{(V_{\mathrm{prog}} - V_{\mathrm{set}})}{2}. \end{cases} \tag{8}$$

From (8), we can determine $I_{\mathrm{ds},Q}$

$$\begin{cases} I_{\mathrm{ds},Q} = \frac{k_n W_{\mathrm{prog}} \left[(V_{\mathrm{GS}} - V_T)(V_{\mathrm{prog}} - V_{\mathrm{set}}) - \frac{1}{4}(V_{\mathrm{prog}} - V_{\mathrm{set}})^2 \right]}{L} \\ R_{\mathrm{RRAM},Q} = \frac{2L \cdot V_{\mathrm{set}} / W_{\mathrm{prog}}}{k_n \left[(V_{\mathrm{GS}} - V_T)(V_{\mathrm{prog}} - V_{\mathrm{set}}) - \frac{1}{4}(V_{\mathrm{prog}} - V_{\mathrm{set}})^2 \right]} \\ V_{\mathrm{prog}} < 2(V_{\mathrm{GS}} - V_T) + V_{\mathrm{set}}. \end{cases} \tag{9}$$

In this case, both W_{prog} and V_{prog} can influence $I_{\mathrm{ds},Q}$. By increasing W_{prog} and V_{prog}, $I_{\mathrm{ds},Q}$ can be magnified, leading to a higher current density.

2) Saturation Region: When the crossing point **Q** lies in the saturation region, we obtain the following equations:

$$\begin{cases} I_{\mathrm{ds},Q} = k_n \frac{W_{\mathrm{prog}}}{L} (V_{\mathrm{GS}} - V_T)^2 \\ V_{\mathrm{DS},Q} \geq V_{\mathrm{GS}} - V_T \\ I_{\mathrm{ds},Q} = \frac{(V_{\mathrm{prog}} - 2V_{\mathrm{DS},Q})}{R_{\mathrm{RRAM},Q}} \\ V_{\mathrm{DS},Q} = \frac{(V_{\mathrm{prog}} - V_{\mathrm{set}})}{2}. \end{cases} \tag{10}$$

From (10), we express $I_{\mathrm{ds},Q}$ as follows:

$$\begin{cases} I_{\mathrm{ds},Q} = \frac{k_n W_{\mathrm{prog}} (V_{\mathrm{GS}} - V_T)^2}{2L} \\ R_{\mathrm{RRAM},Q} = \frac{2L \cdot V_{\mathrm{set}} / W_{\mathrm{prog}}}{k_n (V_{\mathrm{GS}} - V_T)^2} \\ V_{\mathrm{prog}} > 2(V_{\mathrm{GS}} - V_T) + V_{\mathrm{set}}. \end{cases} \tag{11}$$

In the saturation region, only W_{prog} can boost $I_{\mathrm{ds},Q}$.

Equations (9) and (11) show that adjusting the W_{prog} and V_{prog} are the two methods in boosting $I_{\mathrm{ds},Q}$. The W_{prog} is linearly proportional to $I_{\mathrm{ds},Q}$ whatever the working region is. When V_{prog} is bound to the linear region, it has a quadratic impact on $I_{\mathrm{ds},Q}$. After V_{prog} meets the need of the saturation region, it has no impact on $I_{\mathrm{ds},Q}$. Therefore, to enhance the current density in the linear region, boosting W_{prog} is effective but requires a large transistor size, while boosting V_{prog} does not increase the transistor size and should be considered as a first choice. When V_{prog} increases, the transistors move from the linear region to the saturation region. In the saturation region, boosting W_{prog} is the only boosting method. Similar conclusions can be found for reset process.

D. Constraints From Breakdown Voltage

As addressed in Section VI-C, boosting V_{prog} can increase $I_{\mathrm{ds},Q}$. However, there exists a breakdown voltage V_{break} for the source-to-drain voltage V_{DS} of a transistor that provides an upper-bound. In this section, we discuss the range of V_{prog} that the 4T1R structure can safely afford.

The V_{DS} of all the transistors (**P1, P2, N1, N2**) in Fig. 12(a) should satisfy to

$$\begin{cases} (a): \max\{V_{\mathrm{prog}} - V_{\mathrm{TE}}\} = \max\{V_{\mathrm{DS1}}\} \leq V_{\mathrm{break}} \\ (b): \max\{V_{\mathrm{TE}}\} = V_{\mathrm{prog}} - \min\{V_{\mathrm{DS1}}\} \leq V_{\mathrm{break}} \\ (c): \max\{V_{\mathrm{prog}} - V_{\mathrm{DS2}}\} = V_{\mathrm{prog}} - \min\{V_{\mathrm{DS2}}\} \leq V_{\mathrm{break}} \\ (d): \max\{V_{\mathrm{BE}}\} = \max\{V_{\mathrm{DS2}}\} \leq V_{\mathrm{break}} \\ (e): \max\{V_{\mathrm{DS1}}\} = \max\{V_{\mathrm{DS2}}\} = \frac{(V_{\mathrm{prog}} - V_{\mathrm{set}})}{2} \\ (f): \min\{V_{\mathrm{DS1}}\} = \min\{V_{\mathrm{DS2}}\} = V_{\mathrm{DS},P}. \end{cases} \tag{12}$$

Equations (12) (a)–(d) consider the breakdown limitations of V_{DS} of the transistors **P1, N1, P2, N2**, respectively. Equations (12) (e)-(f) are derived from the range of V_{DS} of the transistors **P1, N2** in Fig. 13. As illustrated in Fig. 13, $\max\{V_{\mathrm{DS1}}\}$ and $\max\{V_{\mathrm{DS2}}\}$ happen when the RRAM is in LRS (point **Q**), while $\min\{V_{\mathrm{DS1}}\}$ and $\min\{V_{\mathrm{DS2}}\}$ happen when the RRAM is in HRS (point **P**). $V_{\mathrm{DS},P}$ can be calculated by applying the transistor model (2) to the crossing point **P** in Fig. 13:

$$\begin{cases} I'_{\mathrm{ds}} = k_n \frac{W_{\mathrm{prog}}}{L} \left[(V_{\mathrm{GS}} - V_T) V_{\mathrm{DS},P} - V_{\mathrm{DS},P}^2 / 2 \right] \\ I'_{\mathrm{ds}} = \frac{(V_{\mathrm{prog}} - 2V_{\mathrm{DS},P})}{R_{\mathrm{RRAM},P}}. \end{cases} \tag{13}$$

Note that here, we only consider the linear region because typically the $R_{\mathrm{RRAM},P}$ is large enough to let the V_{DS} of transistors **P1, N2** less than V_{GS}.

Solving (12) and (13), we find that the programming voltage V_{prog} constrained by

$$\begin{cases} P1\&N2: V_{\mathrm{prog}} \leq 2V_{\mathrm{break}} - V_{\mathrm{set}} \\ P2\&N1: V_{\mathrm{prog}} \leq V_{\mathrm{break}} + V_{\mathrm{DS},P} \\ V_{\mathrm{DS,min}} = \frac{2}{R_{\mathrm{RRAM},P}} k_n W_{\mathrm{prog}} / L + (V_{\mathrm{GS}} - V_T) - \sqrt{\Delta} \\ \Delta = \left[2 + k_n \frac{W_{\mathrm{prog}}}{L} (V_{\mathrm{GS}} - V_T) \right]^2 R_{\mathrm{RRAM},P} \\ \quad - 2V_{\mathrm{prog}} k_n \frac{W_{\mathrm{prog}}}{L} R_{\mathrm{RRAM},P} \end{cases} \tag{14}$$

Assume that $R_{\mathrm{RRAM},P}$ of RRAM is large, $V_{\mathrm{DS},P}$ is approximately zero. In such case, the upper-bound of V_{prog} is tied to $V_{\mathrm{prog}} \leq \min\{2V_{\mathrm{break}} - V_{\mathrm{set}}, V_{\mathrm{break}}\}$.

E. Area Estimation

In a 4T1R structure, V_{prog} and GND are directly connected to power supplies. Compared to the 2T1R and 2TG1R structures, no driving inverters are needed. The area of a 4T1R structure is the sum of the sizes of transistors used in **set** and **reset** process

$$2 \cdot (1 + \beta\gamma) W_{\mathrm{prog,set}} + 2 \cdot (1 + \beta\gamma) W_{\mathrm{prog,reset}}. \tag{15}$$

When $W_{\mathrm{prog,reset}}$ is much larger than $W_{\mathrm{prog,set}}$, all the transistors in the 2T1R and 2TG1R structures have to be as large as $W_{\mathrm{prog,reset}}$ while the 4T1R structure can use smaller transistor sizes for set process. Hence, the 4T1R structure brings more flexibilities in transistor sizes than the 2T1R and 2TG1R structures.

F. Benefits of 4T1R Structures

In this section, we compare the 2T1R, 2TG1R, and 4T1R structures in terms of three metrics: V_{DS} symmetry, I_{ds} current, area, delay, and power.

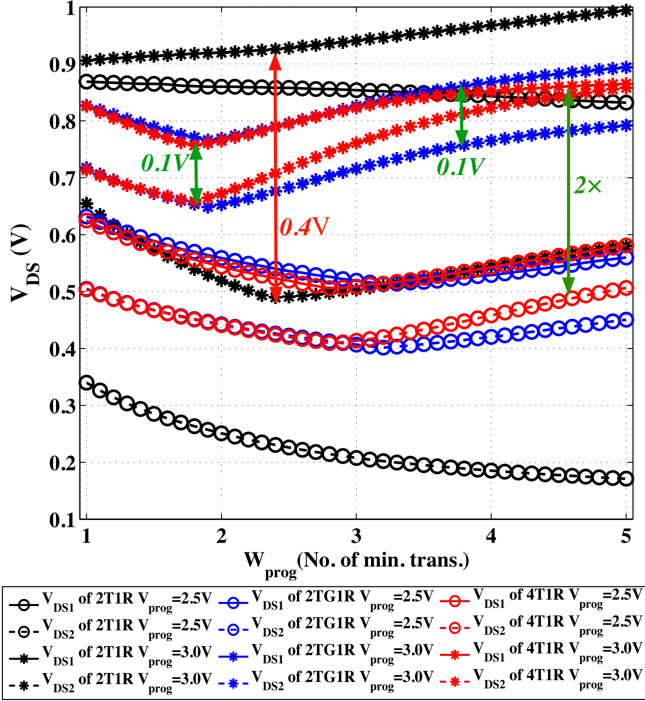


Fig. 14. Comparison on V_{DS} of programming transistors under diverse $W_{\rm prog}$ and $V_{\rm prog}$ in 2T1R, TG-based 2T1R and 4T1R structures ($W_{\rm inv} = 20$). (1 $W_{\rm prog} = 320$ nm).

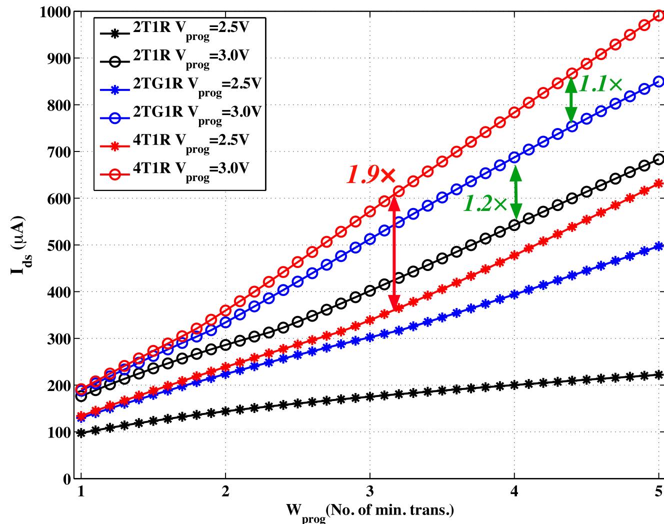


Fig. 15. Comparison on $I_{\rm ds}$ in 2T1R, 2TG1R, and 4T1R structures ($W_{\rm inv} = 20$). (1 $W_{\rm prog} = 320$ nm).

1) $V_{\rm DS}$ Gap Reduction: In Fig. 14, we compare the $V_{\rm DS}$ of 2T1R, 2TG1R, and 4T1R structures, where $W_{\rm inv} = 20$ is considered for the 2T1R and 2TG1R structures. The $V_{\rm DS}$ difference of 2TG1R and 4T1R structures are 75% smaller than 2T1R structure, because they employ *p*-type transistors to propagate $V_{\rm prog}$, as explained in Section VI-B. Note that if a small $W_{\rm inv}$, i.e., $W_{\rm inv} = 1$, rather than $W_{\rm inv} = 20$ is used, the $V_{\rm DS}$ gap of the 2TG1R structure would be larger than 4T1R.

2) Improvement on Programming Current $I_{\rm ds}$: As a result, the driving current shown in Fig. 15 of 4T1R structures is the best of the three solutions. $I_{\rm ds}$ of the 4T1R is 1.1× higher than 2TG1R structure, while 2TG1R improves $I_{\rm ds}$ by 1.3×, compared to 2T1R structure. Note that when $V_{\rm prog} = 2.5$ V, the improvement in driving current of 4T1R and 2TG1R structures are more significant than $V_{\rm prog} = 3.0$ V. When we investigate the driving current density of 2T1R, 2TG1R and 4T1R structures in Fig. 16, 4T1R structure is the best, which is 1.1× higher than 2TG1R structure and 1.4× higher than 2T1R structure. Note that the current density of 2T1R and 2TG1R are deceasing when $W_{\rm prog}$ increases, while the current density of 4T1R is increasing. When a larger $W_{\rm prog}$ is used, $W_{\rm inv}$ has to be increased to alleviate the impact of $V_{\rm DS3}$ and $V_{\rm DS4}$. If $W_{\rm inv}$ does not grow as $W_{\rm prog}$, $V_{\rm DS3}$ and $V_{\rm DS4}$ becomes non-negligible, resulting a degrading current density. Hence, without re-sizing $W_{\rm inv}$, when $W_{\rm prog}$ increases, 2T1R and 2TG1R provides a weaker $I_{\rm ds}$ than a 4T1R scheme. As a conclusion, 4T1R structure is more efficient in driving current than 2T1R and 2TG1R structures.

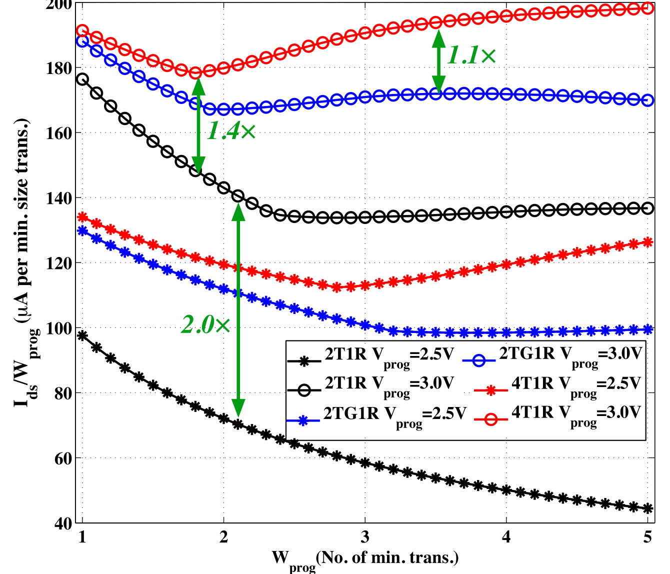


Fig. 16. Comparison on driving current per minimum transistor width under diverse $W_{\rm prog}$ and $V_{\rm prog}$ between 2T1R, TG-based 2T1R, and 4T1R structures ($W_{\rm inv} = 20$). (1 $W_{\rm prog} = 320$ nm).

3) Area, Delay, and Power: In this part, we evaluate the area, delay and power of SRAM-based multiplexers and 2TG1R, 4T1R RRAM-based multiplexers (see Fig. 2). The size of multiplexers is set to 60, which is the typical size of a large FPGA multiplexer (i.e., local routing). The SRAM-based multiplexer is built with a two-level structure [20]. The area of RRAM-based multiplexers is estimated with (4) and (15), where we assume $N = 32$, a typically size for a modern memory bank [22]. The area model in [20] is used to estimate the transistor area. We consider the propagation delay as the delay of the multiplexers, i.e., the signal delay from *in* to *out* in Fig. 12(a). To evaluate the switching energy, we assume that 50% of the inputs have switching activities, which is representative in FPGAs [20]. Because I/O transistors are used in 2TG1R and 4T1R structure while SRAM-based circuit use standard transistors, we consider that I/O transistors have twice area than standard transistors.

Table I compares the transistor area, delay and energy of 2TG1R, 4T1R, and SRAM-based circuit when different $R_{\rm LRS}$ and transistor sharing are considered. The area, delay, and

TABLE I
AREA, DELAY, AND ENERGY OF SRAM CIRCUIT, 2TG1R AND 4T1R STRUCTURES ($V_{\text{prog}} = 3.0$ V)

$R_{LRS} = 2k\Omega$	SRAM Circuit[1]	2TG1R	4T1R
Trans. Count	8	4	4
Trans. Area [2]	470.4	596.0	485.7
Delay (ps)	44.1	56.7	43.8
Energy (fJ)	171.7	151.9	110.6
$R_{LRS} = 6k\Omega$	SRAM Circuit	2TG1R	4T1R
Trans. Count	8	4	4
Trans. Area [2]	375.7	596.0	390.9
Delay (ps)	49.3	78.6	48.9
Energy (fJ)	113.9	151.9	77.5

[1]The area of a SRAM cell is considered to be 4. [20]
[2]Measured as a multiple of the area of a min. width n-type transistor.

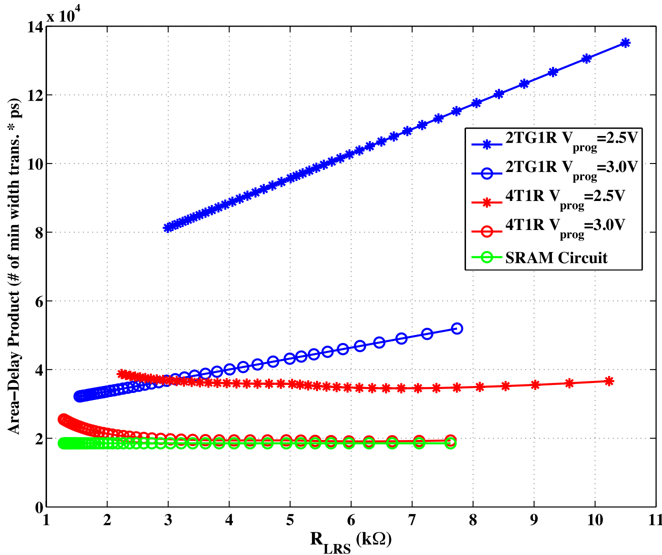


Fig. 17. Comparison on area-delay product of 2TG1R and 4T1R structures ($W_{\text{inv}} = 20$).

energy of RRAM-based circuits are significantly influenced by the choice of R_{LRS}. For a fair comparison, SRAM-based multiplexers have been sized to match the performance of 4T1R-based multiplexers. On average, the RRAM multiplexer reduces by 35% the energy consumption as compared to the SRAM-based multiplexer, with only a 3% area increase. The energy efficiency comes from the capacitance reduction at the output of the multiplexer as a RRAM-based multiplexer has only one pair of programming transistors while an SRAM-based multiplexer has a number of transmission gates in parallel When the target R_{LRS} is relaxed from 2 kΩ to 6 kΩ, the delay of RRAM multiplexer increases by 11% but the energy is reduced by 30%. The 4T1R circuits are more efficient in area, delay, and power when further compared to the 2TG1R solutions, thanks to the separated transistors for the set and the reset processes. Note that the area and power of 2TG1R circuit is constant when R_{LRS} varies from 2 kΩ to 6 kΩ because its $W_{\text{prog,reset}}$ dominates the area. Compared to the 2TG1R circuits, the 4T1R circuits can use smaller transistors for set process, leading to a reduction on parasitic capacitances. Therefore, we see that the 4T1R circuits degrade less in delay than the 2TG1R circuit, when R_{LRS} increases.

Figs. 17 and 18 illustrate the area-delay product and the power-delay product of 2TG1R and 4T1R structures respec-

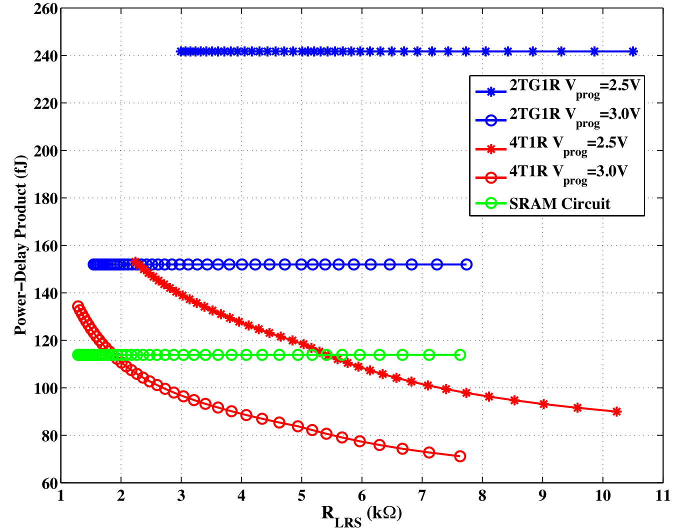


Fig. 18. Comparison on power-delay product of 2TG1R and 4T1R structures ($W_{\text{inv}} = 20$).

tively, when different target R_{LRS} and V_{prog} are considered. A low R_{LRS} requires large programming transistors, which introduces large capacitances to the circuit. When the reduction on R_{LRS} is not as significant as the increment on capacitances, the delay of a RRAM-based circuit increases. In addition, large programming transistors increase the area and large capacitances increase the power consumption. Therefore, a low R_{LRS} does not guarantee the best area-delay and power-delay products [11]. In Figs. 17 and 18, we see that the 4T1R RRAM-based multiplexers can be more area-delay/power-delay efficient than the SRAM-based multiplexers when $R_{\text{LRS}} > 2$ kΩ. Boosting V_{prog} is an efficient method to reduce the area-delay and power-delay products of programming structures. To fully exploit the area and delay of efficiency, it is better to apply the highest possible voltage within the breakdown limit of transistors, i.e., above the standard V_{DD} and close to the breakdown voltage of transistors. It is worth pointing out that the large V_{prog} is only raised during the programming phase, i.e., for a short period of time. As a result, the use of larger programming voltage does not introduce significant reliability hazards.

G. Summary on the 4T1R Programming Structures

In summary, the 4T1R programming structures have the following advantages over the 2T1R and 2TG1R structures:

1) The small V_{DS} gap improves the driving strength of transistors;
2) Since the **set** and **reset** processes use separated transistors, transistor sizes in 4T1R can be more flexible than 2T1R and 2TG1R, leading to a better area efficiency.
3) Drain/source of transistors are directly connected to voltage supplies, eliminating the driving inverters;
4) The bulk connections of 4T1R structure follow the common digital design practice, and avoid the hazards in 2T1R structure.

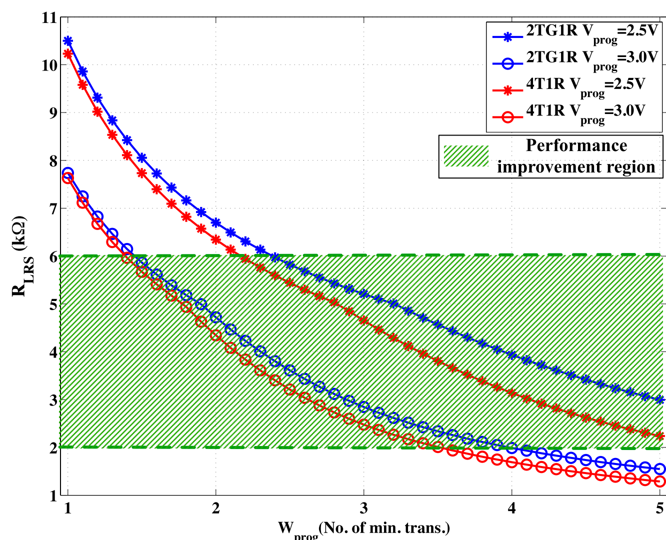


Fig. 19. Comparison on $R_{\rm LRS}$ in 2TG1R and 4T1R structures ($W_{\rm inv} = 20$). (1 $W_{\rm prog}$ = 320 nm)

VII. Discussion

RRAM-based FPGAs use a low $R_{\rm LRS}$ to improve the performance of routing elements. Previous studies [11], [23] predict that a proper $R_{\rm LRS}$ target for FPGA architectures is between 2 kΩ and 6 kΩ depending on the design context, while $R_{\rm HRS}$ should be at least 20 MΩ to mitigate a leakage power increase. The mentioned ranges of $R_{\rm LRS}$ and $R_{\rm HRS}$, achievable as worst case target in current RRAM technologies, show that, beyond the performance gain, FPGA architectures can tolerate a wide distribution of $R_{\rm LRS}$ and $R_{\rm HRS}$ without delay and power increase [11], [23]. The performance of RRAM-based routing elements are not only determined by the $R_{\rm LRS}$ but also the parasitic capacitances of programming transistors. As a result, programming structures offering a high current density are preferred. Fig. 19 shows the $R_{\rm LRS}$ values that can be driven by 2TG1R and 4T1R structures as a function of $W_{\rm prog}$. To obtain a proper $R_{\rm LRS}$ in FPGA, the applicable $W_{\rm prog}$ of transistors are between 1.5 and 4. Boosting $V_{\rm prog}$ can significantly reduce the $R_{\rm LRS}$, which brings opportunities in further area and delay improvement on RRAM-based FPGAs. When considering more advanced technology nodes, such as 28 nm, 14 nm, and beyond, it is expected that lower $V_{\rm reset}$ and $V_{\rm set}$ voltages can be employed as a consequence of the $V_{\rm DD}$ reduction. As a result, the effect of boosting $V_{\rm prog}$ is expected to gain further in efficiency.

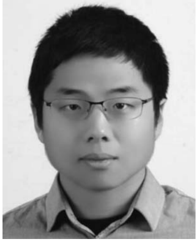
VIII. Conclusion

In this paper, we investigated the programming circuits for RRAMs. We pointed out four limitations of the commonly considered 2T1R programming structure: the low current density, the serious body effect, the need for large driving inverters and the area inefficiency. We introduced a 2TG1R structure to alleviate the first two limitations and a 4T1R programming structure which solves all the limitations. We conducted theoretical analysis and electrical simulations on 2T1R, 2TG1R and 4T1R programming structures. Simulation results showed that the 4T1R programming structure can increase 1.3× current compared to 2T1R structure on average. Electrical simulations demonstrate that boosting $V_{\rm prog}$ improves the current density of programming structures by 3× and area efficiency by 1.7× on average, respectively.

References

[1] R. Waser *et al.*, "Nanoionics-based resistive switching memories," *Nature Mater.*, vol. 6, pp. 833–840, 2007.
[2] H. Akinaga *et al.*, "Resistive random access memory (ReRAM) based on metal oxides," *Proc. IEEE*, vol. 98, no. 12, pp. 2237–2251, 2010.
[3] H.-S. P. Wong *et al.*, "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.
[4] G. W. Burr *et al.*, "Overview of candidate device technologies for storage-class-memory," *IBM J. RD*, vol. 52, no. 4/5, Jul./Sep. 2008.
[5] Y. S. Chen *et al.*, "Highly scalable Hafnium Oxide memory with improvements of resistive distribution and read disturb immunity," in *Proc. IEEE IEDM*, 2009, pp. 1–4.
[6] P.-E. Gaillardon *et al.*, "Design and architectural assessment of 3-D resistive memory technologies in FPGAs," *IEEE Trans. Nanotechnol.*, vol. 12, no. 1, pp. 40–50, 2013.
[7] S. Tanachutiwat *et al.*, "FPGA based on integration of CMOS and RRAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 11, pp. 2023–2032, 2010.
[8] J. Cong and B. Xiao, "FPGA-RPI: A novel FPGA architecture with RRAM-based programmable interconnects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 4, pp. 864–877, 2014.
[9] P.-E. Gaillardon *et al.*, "Emerging memory technologies for reconfigurable routing in FPGA architecture," in *Proc. ICECS*, 2010, pp. 62–65.
[10] P.-E. Gaillardon *et al.*, "GMS: Generic memristive structure for non-volatile FPGAs," in *Proc. IEEE/IFIP VLSI-SoC*, 2012, pp. 94–98.
[11] X. Tang *et al.*, "A high-performance low-power near-Vt RRAM-based FPGA," in *IEEE ICFPT*, 2014, pp. 207–215.
[12] Y. Yang-Liaum *et al.*, "Non-volatile 3D-FPGA with monolithically stacked RRAM based configuration memory," in *Proc. IEEE ISSCC*, 2012, pp. 406–408.
[13] J. Jiang *et al.*, "Verilog-A compact model for oxide-based resistive random access memory," in *Proc. IEEE SISPAD*, 2014, pp. 41–44.
[14] S. Yu *et al.*, "3D vertical RRAM-scaling limit analysis and demonstration of 3D array operation," in *Proc. Symp. VLSI Tech.*, 2013, pp. 158–159.
[15] Z. Zhang *et al.*, "Nanometer-scale HfO_x RRAM," *IEEE Electron Device Lett.*, vol. 34, no. 8, pp. 1005–1007, Aug. 2013.
[16] W. Kim *et al.*, "Forming-free nitrogen-doped AlO_x RRAM with sub-μA programming current," in *Proc. Symp. VLSI*, 2011, pp. 22–23.
[17] M. J. Lee *et al.*, "2-stack 1D-1R cross-point structure with oxide diodes as switch elements for high density resistance RAM applications," in *Proc. IEEE IEDM*, 2007, pp. 771–774.
[18] Altera Corporation, "Stratix 10 advance information brief," Jul. 2015.
[19] Xilinx, "Virtex-7 user guide DS180" (v1.17), May 2015.
[20] C. Chiasson *et al.*, "Should FPGAs abandon the pass-gate?" in *Proc. FPL*, 2013, pp. 1–8.
[21] Synopsys, "HSPICE user guide: Simulation and analysis," Version I-2013.12, Dec. 2013.
[22] J. M. Rabaey *et al.*, *Digital Integrated Circuits*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[23] X. Tang *et al.*, "Accurate power analysis for near-V_t RRAM-based FPGA," in *Proc. IEEE FPL*, 2015, pp. 174–177.

Xifan Tang (S'13) received the B.Sc. degree in microelectronics from Fudan University, Shanghai, China, in 2011, and the M.Sc. degree in electrical engineering from the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2013, where he is currently working toward the Ph.D. degree. His current research interests include computer-aided design for programmable architecture and emerging technologies.



Gain Kim (S'13) received his B.Sc. and M.Sc. degrees in Electrical Engineering from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2013 and 2015, respectively, where he is currently working toward the Ph.D. degree. His research interests include reconfigurable gate array architecture and software-defined transceiver.

Pierre-Emmanuel Gaillardon (S'10–M'11) received the Electrical Engineer degree from CPE-Lyon, France, in 2008, the M.Sc. degree in electrical engineering from INSA Lyon, France, in 2008, and the Ph.D. degree in electrical engineering from CEA-LETI, Grenoble, France and the University of Lyon, France, in 2011. He was with EPFL, Lausanne, Switzerland, as a Research Associate at the Laboratory of Integrated Systems (LSI). Starting January 2016, he assumed an Assistant Professor position within the Electrical and Computer Engineering (ECE) department at University of Utah, Salt Lake City, UT, USA. Previously, he was Research Assistant at CEA-LETI, Grenoble, France, and Visiting Research Associate at Stanford University, Palo Alto, CA, USA. Dr. Gaillardon is recipient of the C-Innov 2011 best thesis award and the Nanoarch 2012 best paper award. He is an Associate Editor of the IEEE TRANSACTIONS ON NANOTECHNOLOGY. He has been serving as TPC member for many conferences, including DATE'15–16, VLSI-SoC'15, CMOS-ETR'13–15, Nanoarch'12–15, ISVLSI'14–15 conferences, and is reviewer for several journals and funding agencies. The research activities and interests of Dr. Gaillardon are currently focused on the development of reconfigurable logic architectures and circuits exploiting emerging device technologies and novel EDA techniques.

Giovanni De Micheli (F'94) received the Nuclear Engineer degree from Politecnico di Milano, Italy, in 1979 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, CA, USA, in 1980 and 1983, respectively. He is Professor and Director of the Institute of Electrical Engineering and of the Integrated Systems Centre at EPF Lausanne, Switzerland. He is program leader of the Nano-Tera.ch program. Previously, he was Professor of Electrical Engineering at Stanford University.

Prof. De Micheli is a Fellow of ACM and a member of the Academia Europaea. His research interests include several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips and 3D integration. He is also interested in heterogeneous platform design including electrical components and biosensors, as well as in data processing of biomedical information. He is author of: Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994, co-author and/or co-editor of eight other books and of over 600 technical articles. His citation h-index is 85 according to Google Scholar. He is member of the Scientific Advisory Board of IMEC (Leuven, B), CfAED (Dresden, D) and STMicroelectronics.

Prof. De Micheli is the recipient of the 2012 IEEE/CAS Mac Van Valkenburg award for contributions to theory, practice and experimentation in design methods and tools and of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He received also the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000, the D. Pederson Award for the best paper on the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS in 1987, and several Best Paper Awards, including DAC (1983 and 1993), DATE (2005) and Nanoarch (2010 and 2012). He has been serving IEEE in several capacities, namely: Division 1 Director (2008–9), co-founder and President Elect of the IEEE Council on EDA (2005–7), President of the IEEE CAS Society (2003), Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (1997–2001). He has been Chair of several conferences, including Memocode (2014) DATE (2010), pHealth (2006), VLSI SOC (2006), DAC (2000), and ICCD (1989).