Robust and reliable ReRAM-based non-volatile sequential logic circuits in deeply-scaled CMOS tehnologies

Natalija Jovanovic

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0023

EDITE ED 130

PhD THESIS

delivered by

Télécom ParisTech

Electronics and Communication

Natalija JOVANOVIĆ

23 March 2016

Robust and reliable ReRAM-based non-volatile sequential logic circuits in deeply-scaled CMOS tehnologies

Thesis director : **Lirida ALVES DE BARROS NAVINER**
Advisors : **Olivier THOMAS, Borivoje NIKOLIĆ**

Jury

M. Jean-Michel PORTAL, IM2NP, Aix-Marseille Université,	President
M. Philippe CANDELIER, STMicroelectronics, Crolles	Examinator
M. Lionel TORRES, LIRMM, Montpellier	Reviewer
M. Jean-Didier LEGAT, UCL, Louvain	Reviewer
Mme. Lirida ALVES DE BARROS NAVINER, Telecom ParisTech, Paris	Thesis director
M. Olivier THOMAS, CEA-LETI, Grenoble	Advisor
M. Borivoje NIKOLIĆ, BWRC, Berkeley	Advisor

THESIS

Télécom ParisTech
école de l'Institut Mines Télécom – membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

Abstract

Non-volatile memories and flip-flops can improve the energy efficiency in battery-operated devices by eliminating the sleep-mode consumption, while maintaining the system state. Among emerging embedded NVM technologies, ReRAMs differentiate itself with a fast programming time, a simple CMOS-compatible structure and a good scalability. Previously proposed ReRAM-based non-volatile flip-flops (NVFF) have been implemented in 90nm or older CMOS nodes and suffer from CMOS reliability issues in scaled nodes due to high programming and forming voltages. This thesis makes the analysis of robust and reliable non-volatile design in 28nm CMOS node and below. It presents two novel thin-gate oxide CMOS design solutions for the programming of ReRAM devices. The programming circuits are applied in dual-voltage NVFF architecture which employs two ReRAM devices (2R). Alternative 1R NVFF architecture is also proposed in order to achieve higher density and lower consumption. With regard to the existing ReRAM technologies, given NVFF solutions are optimized for ReRAM programming conditions which improve endurance and minimize programming power. Statistical analysis of the FF core and its optimization was performed, to evaluate the best restore operation architectures which meet digital CMOS circuit design yield requirements. The NVFFs are implemented in 28nm CMOS FDSOI and benchmarked against a master slave flip-flop from a standard library and a data-retention flip-flop. Finally, to minimize the NVFF area overhead without impacting the robustness of non-volatile operations, multi-port non-volatile register file (NVRF) based on the 1R NVFF solution is proposed.

Key words: Flip-flop, ReRAM, Low-power, Non-volatile memories

Acknowledgments

First, I would like to thank Jean-Didier Legat, Philippe Candelier, Lionel Torres and Jean-Michel Portal for the evaluation of this thesis and their comments. It has been a real pleasure discussing my work with them. I'm grateful to Lirida Naviner for her support over the past years and her help in organizing my research. I thank Bora Nikolic for his constant feedback and advices, both technical and personal, which have been invaluable for my PhD adventure. Thanks to him I had the opportunity to spend time in BWRC where I expanded my knowledge in circuit design. Foremost, my gratitude goes to Olivier Thomas for sharing his expertise, his incredible enthusiasm, and for the patience he had with me. Without his guidance and energy this work would not be possible.

Many colleagues supported and assisted me throughout this thesis, provided helpful discussions and enriched my whole experience. I would like to thank Fabian Clermidy, Edith Beigne and Marc Belleville for their help in the crucial moments and valuable scientific and organizational inputs. I acknowledge the collaboration with Marina Reyboz and IM2NP memory team who provided ReRAM models. I really appreciate the help from my cubicle-mates, Santhosh and Ogun who were there for me at the beginning of the thesis, Guillaume for the layout work, Alain for test development, Alexandre V., Bilel, and Olivier D. for their collaboration on M0+ project, the test team for their assistance in the lab, and all the other DACLE members with whom I had the chance to work with. I would also like to thank DCOS members who shared their knowledge with me and allowed broadening this work in the technology aspect. Special thanks go to Elisa Vianello whose expertise helped me to better understand this topic and see a bigger picture. I also had a great pleasure collaborating with her on several publications. Finally, I particularly thank the memory design team: Adam, Kaya, Greg, Bastien, Alex, Yves, Navneet, and Jean-Philippe for their help, support and friendship.

On a non-professional side, I thank all my Grenoble friends for the great time we had together - they made my PhD years unforgettable. I am really happy to have by my side Dajana, Miki, and Bojan, thanks to whom I have started this journey, Thano and Marija who accompanied me in good and bad, and Quentin whose energy helped me at the finish line. I thank Luka for his understanding, encouragement and love (and proofreading my manuscript!). The biggest thanks goes to my parents and family, for their unconditional love and support during my thesis and beyond.

Contents

List of Acronyms

BEOL Back-End-Of-Line
BER Bit Error Rate
BET Break-Even Time
CBRAM Conductive Bridge Random Access Memory
CM Current Mirror
CMOS Complementary Metal-Oxide Semiconductor
CPU Central Processing Unit
DRAM Dynamic Random Access Memory
FDSOI Fully Depleted Silicon On Insulator
FeRAM Ferroelectric Random Access Memory
FF Flip-Flop
HL High-to-Low transition
IoT Internet Of Things
LH Low-to-High transition
LS Level-Shifter
MCU MicroController Unit
MIM Metal-Insulator-Metal
MLC Multi-Level Cell
MOS Metal-Oxide Semiconductor
MPU MicroProcessor Unit
MRAM Magnetic Random Access Memory
MSFF Master-Slave Flip-Flop
MTJ Magnetic Tunnel Junction
NV Non-Volatile
NVFF Non-Volatile Flip-Flop
NVL Non-Volatile Logic
NVM Non-Volatile Memory
NVP Non-Volatile Processor
NVRF Non-Volatile Register File
OxRAM Oxide-based Random Access Memory
PCM Phase Change Memory
PMC Programmable Metalization Cell

PMU Power Management Unit
ReRAM Resistive Random Access Memory
RF Register File
RTL Register-Transfer Level
SAFF Sense Amplifier Flip-Flop
SHE Spin Hall Effect
SRAM Static Random Access Memory
STT Spin-Transfer-Torque
ULP Ultra-Low Power
WIC Wake-up Interrupt Controller

List of Figures

List of Tables

Chapter 1

Introduction

Connected devices and their requirements

The evolution of technology, especially the miniaturization of electronic chips and sensors, the improvements in wireless communication, and their relatively low price, led to enormous expansion of small electronic devices. Indeed, connected (smart) devices, so-called Internet of Things (IoT), can be found in all aspects of modern human life, such as: smartphones, wearable fitness and medical devices, home automation and environmental control in smart buildings, transportation, smart grids, etc. Moreover, in the following years it is expected that the total number of such devices will drastically increase, not just in personalized applications but even more in business and manufacturing, e.g., for tracking inventory, managing machines, increasing efficiency, etc. (Figure 1.1). In fact, it is impossible to predict all potential applications in this area, as new creative solutions are emerging every day, fundamentally altering the way we interact with our physical environment [1].

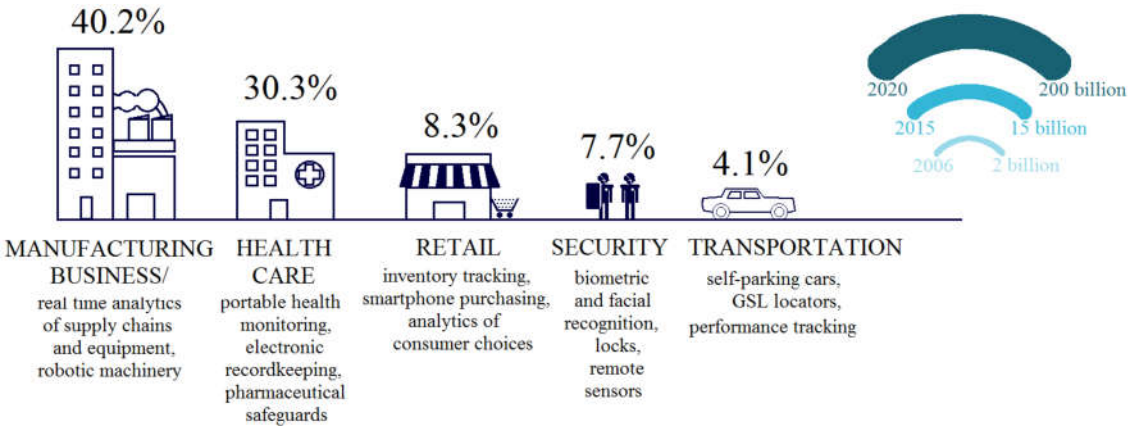
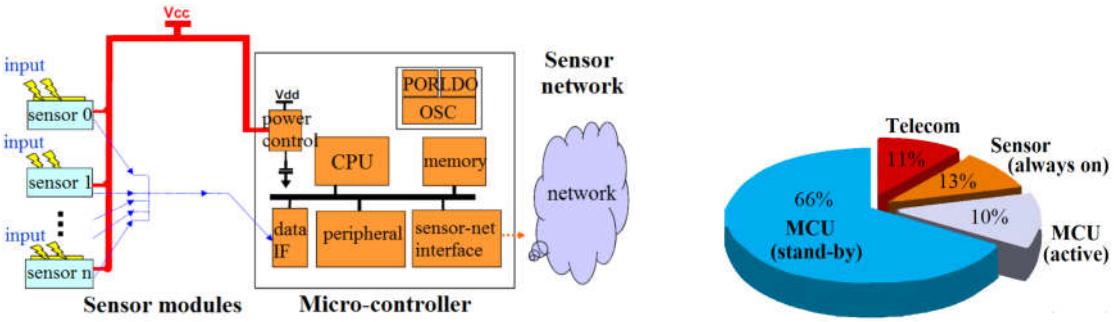


Figure 1.1: Predictions for IoT market for 2025, according to Intel [2].

Typically, connected devices contain sensors to gather environment information, local computing and storage unit, as well as transceivers, power optimization circuitry, etc., as depicted in Figure 1.2a. Programmable microcontroller units (MCU) are used as a main component, to manage hardware and processing requirements. They periodically sample and process the sensor data, and send it to the network.



(a) (b)

Figure 1.2: Conventional sensor node: (a) architecture, (b) energy consumption breakdown (figures taken from [3]).

An important aspect of such systems is their normally-off nature, meaning that they spend most of their lifetime in standby mode, interrupted only by short periods of activity. Depending on the application, standby intervals can range from several microseconds to several seconds, or even more. For example, sensors used in environment systems (temperature, brightness, motion sensors, etc.) perform data sampling with intervals ranging between 50ms and 10s [3]. Combined with the fact that with CMOS transistor shrinking static leakage increases [4], microcontrollers operating in long standby modes can contribute as a dominant component of total energy consumption of a connected device, as illustrated in Figure 1.2b.

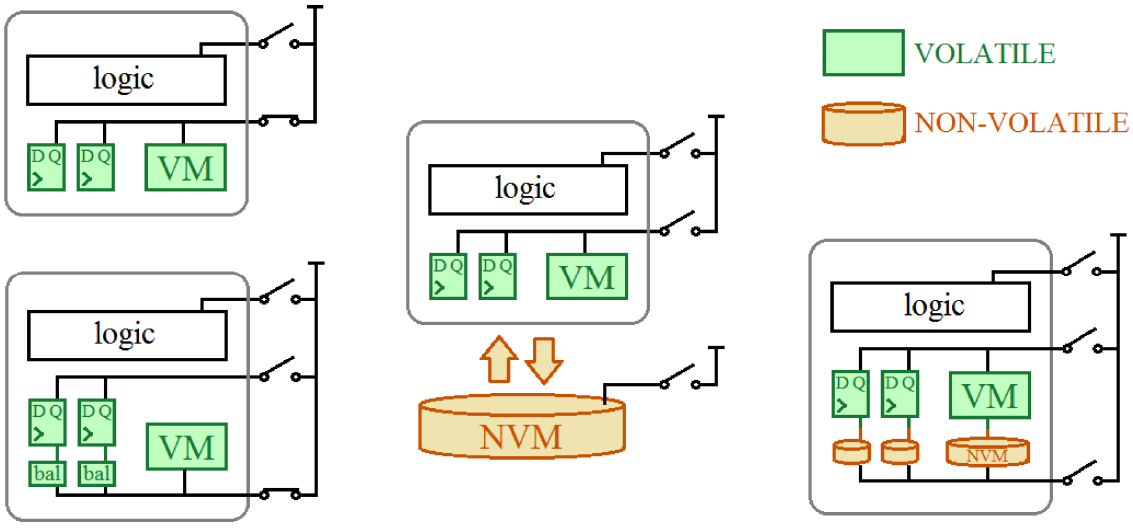
In most cases smart devices are battery-powered and, for practical reasons, they should be designed to support long operational lifetime without the need for battery replacement. Additionally, energy harvesting based on solar, wind, thermal and kinetic energy sources is often used to power the system, however, its availability is unstable and unpredictable. Therefore, one of the key challenges in IoT design is creating energy efficient systems which can operate at low power budgets. Particularly, there is a **need for ultra-low standby power hardware, without degrading the performance in the active mode**. Additionally, a design must be **able to tolerate sudden device failures**, which may be frequent due to unstable external power sources.

Standard design techniques for energy efficient IoT systems

Numerous solutions for achieving high energy efficiency in both standby and active modes exist, for instance: dynamic voltage and frequency scaling [5] and clock gating for the dynamic power reduction; power gating using multiple threshold transistors [6], stacking transistors, operating in near-threshold and subthreshold regime for static power reduction, etc. These methods are implemented in current commercial microcontrollers in the form of various operating modes with complex power management. Particularly interesting are multiple sleep modes characterized by different levels of power consumption, which can be entered depending on the application requirements. For example, in a "shallow" sleep mode clock can be gated and peripherals disabled, while in "deep" sleep mode almost whole MCU can be powered down. Thus, in the realization of efficient normally-off computing, an important role belongs to the power gating optimization and, consequently, to **the organization of registers and memory responsible for maintaining the system state** [7].

To disconnect the digital circuit from the power source, a high-threshold sleep transistor is inserted between the supply and the circuit, and turned off during standby. However, if the whole processor is power gated, the registers and the memory lose their state and the MCU needs reboot after each standby, which is a very slow and expensive process. Thus, in conventional architectures volatile registers and memory stay powered in standby, preferably at reduced voltage, while the rest of the circuit is disconnected (Figure 1.3a, top). For higher saving, registers can be upgraded to support data-retention, meaning that an always-on source can be applied only to the part where data is preserved – either the slave stage of a flip-flop [6], or an additional ("balloon") latch attached to the slave stage [8] (Figure 1.3a, bottom). Although this solution offers very fast active/sleep mode transitions, the power dissipation in a CMOS flip-flop is always present, and can accumulate to significant levels in long sleep modes.

Alternatively, during standby mode data can be preserved in external non-volatile memories, thus completely canceling power dissipation, as illustrated in Figure 1.3b. However, core-bus-memory and memory-bus-core data transfers are slow and costly in energy consumption, especially when the non-volatile module is Flash memory-based. Consequently, only a limited amount of data and processor state is maintained, in accordance with the go-to-sleep/wake-up time and energy requirements. In addition, this kind of system may often fail to handle sudden power loss. To avoid the effect of power failures in energy-



(a) (b) (c)

Figure 1.3: Saving the context during standby mode in normally-off computing systems: (a) in volatile flip-flops, (b) in external non-volatile memory, (c) in non-volatile flip-flops (figure updated from [7]).

harvesting systems, a checkpoint procedure is generally implemented. This means that a snapshot of the system state is periodically taken and, when the supply is stabilized again, the processor can recover and return to the last saved state. In this case, frequent checkpoints can bring significant energy and performance overhead.

Application of emerging non-volatile memories in IoT

Today, with the extensive research on new types of non-volatile memories (NVM), other opportunities have emerged. Initially envisioned as a Flash/DRAM replacement, they can fit in the previous concept as an external non-volatile memory, leading to better speed/consumption system characteristics. More importantly, they bring the possibility of easy integration with CMOS technology, allowing for incorporating non-volatile feature directly in the processor logic, as shown in Figure 1.3c. At the moment, emerging NVMs are too slow and power-hungry to completely replace SRAM, however, they can be combined with it (e.g., nv-SRAM [9]). Thus, fast volatile memories can satisfy performance-critical requirements, while NVMs can ensure zero-consumption standby storage. The essential components in this approach, used to save the whole context of microcontroller unit and enable fast transitions between modes, are **non-volatile flip-**

flops (NVFF). Moreover, using NVFFs improves system reliability, as it can efficiently support the checkpoint procedure.

Among various NVM technologies **resistive random access memories (ReRAMs)** are particularly attractive, as they rely on low operating voltages, fast programming and reading time, low power consumption and high scalability. In addition, they exhibit an adequate endurance for IoT applications. Due to their simple structure and compatibility with CMOS back-end-of-line process, ReRAMs offer the easiest integration with the CMOS digital logic for a low cost.

Numerous non-volatile flip-flop solutions based on several NVM technologies can be found in literature. Among them, reported ReRAM-based NVFFs have been implemented in 90nm or older CMOS nodes. However, in the scaled CMOS processes the maximum CMOS operating voltage becomes lower than the required ReRAM forming and programming voltages. Therefore, this voltage gap causes reliability issues in the designs which use thin-gate oxide transistors. Besides, the choice of NVFF topology is strongly affected by the characteristics of NVM devices, such as programming voltages, currents, memory window, etc. Thus, proposed NVFFs architectures which use NVM devices other than ReRAMs cannot be directly applied to ReRAM-based cells.

Goal and contributions of the thesis

This work explores co-integration of ReRAM with 28nm and below CMOS technologies in non-volatile flip-flop design. Process variability and static leakage in active mode, which are the weak points of advanced CMOS nodes, are minimized by using FDSOI technology [10], which also enables high-performance and low-voltage operations.

Two novel thin-gate oxide design solutions for the programming of ReRAM devices are proposed, and then applied in a dual-voltage two-ReRAM NVFF architecture. In order to improve the area overhead and reduce consumption, a new one-ReRAM architecture is also presented. NVFFs are designed as standard cells for compatibility with digital design flow. One of NVFF architectures is later used in an implementation of a non-volatile version of ARM Cortex-M0+ microprocessor.

Along with new design solutions, the electrical characteristics of available ReRAM technologies are explored, in order to choose the most appropriate architecture and optimize it accordingly. The main focus when defining optimum ReRAM resistive state values and

programming conditions lies in: (i) improving endurance and minimizing programming power, (ii) achieving reliable, low-voltage data recovery. The latter, which is compromised by the variability of both ReRAM and CMOS, is accomplished by statistical evaluation of different flip-flop core architectures.

The NVFFs are implemented in 28nm CMOS FDSOI and benchmarked against a master slave flip-flop from a standard library and a data-retention flip-flop, to evaluate the impact of adding non-volatility on regular flip-flop mode, and to estimate the break-even time for employing NVFF instead of standard, volatile solutions.

Finally, to achieve higher design density without affecting the robustness of non-volatile operations, novel multi-port non-volatile register file (NVRF) based on one of the NVFF solutions is proposed.

Thesis organization

Following this introductory part, Chapter 2 provides a brief description and comparison of emerging non-volatile memories – FeRAM, PCMs, MRAMs and ReRAM. It then focuses on ReRAM devices and explains in more details their behavior and electrical characterization. According to technology constrains, it highlights general challenges and guidelines for ReRAM-based design. Chapter 3 gives an overview of existing non-volatile flip-flops and non-volatile processors. It examines various aspects of their architectures (e.g., supplies, operating modes, etc.) while taking into account used NVM and CMOS technologies, and then completes the design guidelines for ReRAM-based NVFFs.

Chapter 4 covers four proposed NVFF solutions. First, it shows two design topologies related to the restore operation – 2R NVFF and 1R NVFF, analyzes them and determines their compatibility with available ReRAM devices. This is followed by the description of two ReRAM programming circuits (level-shifter-based and current programming-based). Chapter 5 is dedicated to evaluation of introduced non-volatile flip-flops and a comparison with their volatile counterparts. It gives the simulation results of the flip-flop performance and consumption in different operating modes, and shows the physical implementation of the cells. Design work is concluded in Chapter 6, which describes the architecture of non-volatile register file.

Finally, Chapter 7 summarizes presented work, proposes the application of NVFF and NVRF in complex systems, and suggests several ideas for future work.

Chapter 2

Non-volatile memories

Contents

2.1 Emerging non-volatile memories

Conventional memories

According to their ability to maintain the data when the power source is disconnected, semiconductor memories can be separated in two categories, as depicted in Figure 2.1. **Volatile memories**, which require power supply to preserve the stored state, are widely present in modern computing systems in the form of SRAM and DRAM. **Static random access memory (SRAM)** is a high-performance, CMOS-based memory which can be tightly integrated with the processor core. Consisting of six transistors (or more, in case of multi-port architectures), it is costly in area and power consumption. Therefore, it is usually reserved for registers, and instruction and data cache. On the other hand, **dynamic random access memory (DRAM)**, containing one transistor and one capacitor, features much higher density and lower power. Hovewer, it is slower than SRAM and requires periodical refresh as the capacitor discharges over time. Typically, DRAM is used as a primary working storage (main memory) as an off-chip module which is later cached in SRAM, or an embedded DRAM (eDRAM).

In the class of **non-volatile memories**, which do not lose the data when the power is

cut-off, the dominant today is **Flash memory**, based on a floating gate transistor – a MOS transistor structure with a modified gate stack. The floating gate, used as electron storage element, is placed between the standard, control gate and the MOS channel. The cell is written by forcing the electrical charge from the channel through the oxide layer to the floating gate. As the floating gate is isolated by the insulating layer, the electrons placed on it are trapped until they are removed by another application of electric field. An array of floating gate transistors can be arranged in different fashions, resulting in NOR and NAND Flash. Apart from non-volatility, Flash memory is characterized by low fabrication cost and higher density than DRAM. However, it has much lower write/read speed and endurance limited to around 10^6 cycles. Moreover, Flash requires programming voltages much higher than the operation voltage of CMOS devices, which results in high write consumption, and poses a strong limitation for integration with CMOS logic. Thus, in current systems it is used as a secondary storage for code (faster, NOR Flash) and data (denser, NAND Flash).

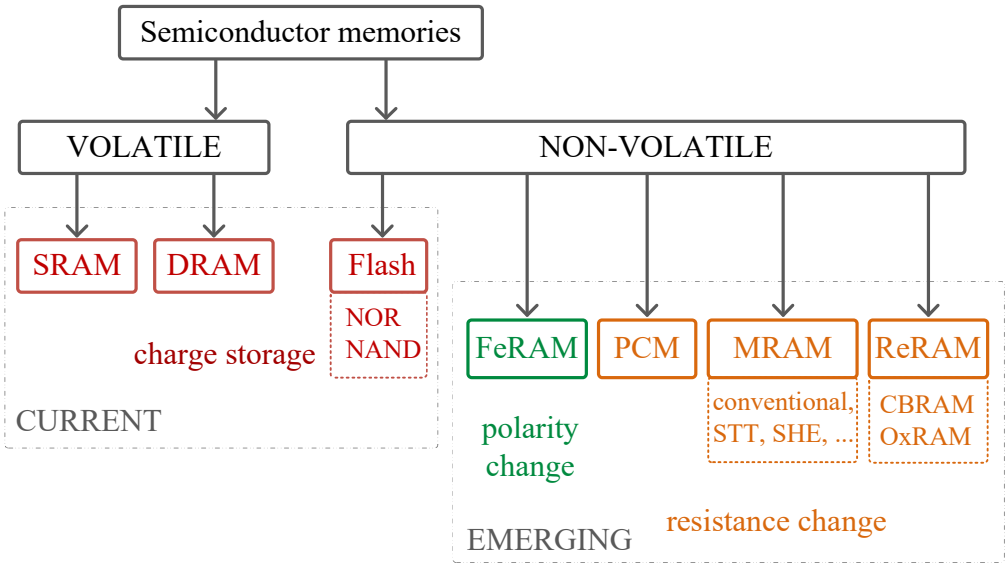


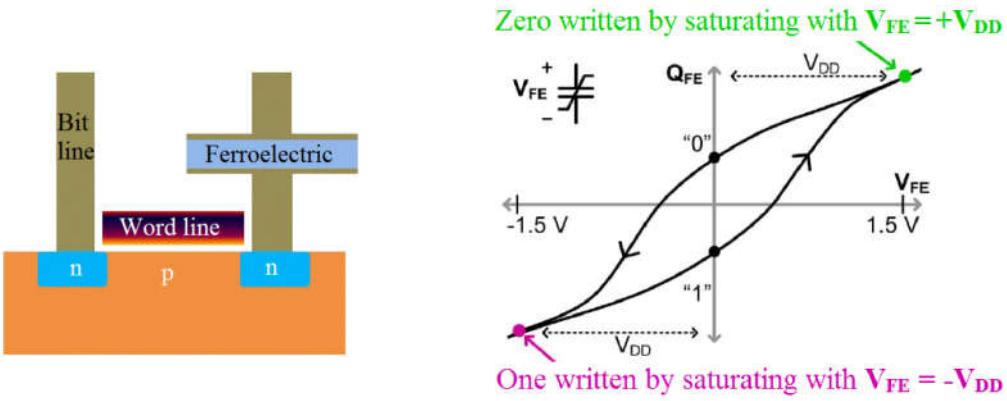
Figure 2.1: Semiconductor memories.

In order to improve memory integration and density, in the last years many memory candidates based on a different storage principles are being explored [11]. Aiming at replacing Flash and DRAM memories, research is driven towards the "universal" memory, characterized by low operating voltages, low energy programming, high operation speed, long retention time, high endurance and simple structure [12]. With the progress in emerging non-volatile memory technologies, it appears that their application can be further

extended, as some NVM promise good performances and potential for integration over CMOS processes. Hence, they can be used directly in the processor logic, as a non-volatile "upgrade" of SRAM. Besides, this may alter the memory hierarchy by introducing new levels of on-chip memory and offering significant reductions in off-chip memory transactions [13].

FeRAM

Ferroelectric random access memory (FeRAM) exploits the polarization properties of a ferroelectric material (e.g., lead zirconate titanate) placed between two electrodes, as illustrated in Figure 2.2a. When an external electric field is applied across the structure, electric dipoles formed in the crystal structure of the ferroelectric material align with the field direction. As the ferroelectric material has a nonlinear relationship (hysteresis) between the applied electric field and the stored charge (Figure 2.2b), the dipoles retain their polarization state after the external field is removed. During the read operation, the cell is forced into one state. If that state does not correspond to the previously stored value, the re-orientation of the dipoles will cause a current pulse at the output. Thus, reading the value stored in FeRAM device is a destructive process and requires the cell to be re-written if it was changed.



(a) (b)

Figure 2.2: (a) Basic structure of 1T1FeRAM cell [11]. (b) Charge vs. voltage hysteresis of the ferroelectric capacitor [14].

Compared to Flash, FeRAM has lower write delay, programming voltages and consumption, and offers much higher endurance (e.g., 10^{14}). Even though FeRAM performance is in theory expected to be comparable with DRAM, current FeRAM chips are slower,

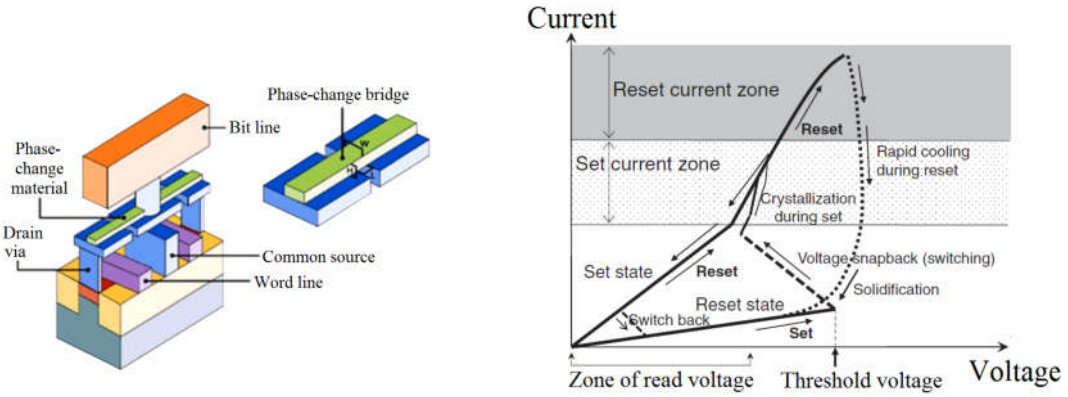
as they are produced in wider CMOS nodes and the access transistor limits the speed. Embedding FeRAM cells with conventional CMOS manufacture requires two additional masking steps, which is simpler than Flash (typically nine masks). Still, as FeRAM materials are not commonly used for CMOS, their co-integration may cause process compatibility and contamination issues. Apart from the destructive read disadvantage, FeRAM technology has low storage density (lower than Flash) and does not demonstrate multi-level cell (MLC) capability, leading to higher production cost. Additionally, to maintain its polarization capability the ferroelectric layer must be thick enough, **hampering the device scaling**.

At the moment, commercial FeRAM devices can be found in the market. For example, Texas Instruments has incorporated FeRAM memory into MSP430 microcontroller units in their new FeRAM series [15, 16].

PCM

Phase change memory (PCM) is based on chalcogenide glasses whose structure can change between two phase states which have different electrical resistivity – low-resistance crystalline state and high-resistance amorphous state. A PCM cell, shown in Figure 2.3a, consists of a phase change material layer (e.g., $Ge_2Sb_2Te_5$, commonly known as GST) sandwiched between metallic top and bottom electrode. Cell programming is achieved by applying electrical pulses through the heater, resulting in Joule heating of the active chalcogenide layer and, consequently, phase transition (Figure 2.3b) [17]. The phase change occurs in the active region of the cell, located at the interface between the phase change material and the heater, which is engineered to provide good thermal insulation from the metal layers.

Compared to Flash, PCM is characterized by several orders of magnitude faster switching time (a few tens of nanoseconds) and higher device endurance (around 10^{12}). However, **current PCM technologies require large programming current, increasing the power consumption and the selector size in 1T1PCM cell**. As the phase-change is a thermally driven process, the temperature sensitivity of PCM cells is high, bringing an additional challenge in the production process. This also impacts the cell retention in elevated temperature environments similar to thermal conditions that allow for fast crystallization. PCM cells have the ability to achieve a number of distinct intermediate states, thus storing multiple bits in a single cell (MLC). However, this property is limited,



(a) (b)

Figure 2.3: (a) PCM integration [11]. (b) Typical V-I characteristics of PCM [17].

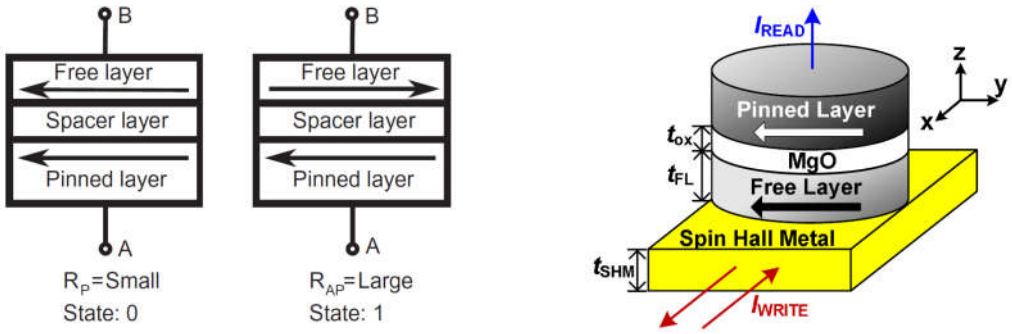
as the technology suffers from long-term low-resistance drift towards higher values. PCM technology offers high scalability and density. For example, Samsung recently proposed an 8Gb array integrated in 20nm node [18].

MRAM

Magnetic random access memory (MRAM) is based on a magnetic tunnel junction element (MTJ), consisting of a thin non-magnetic tunnel barrier layer between two ferromagnetic layers, as described in Figure 2.4a. The magnetization direction of the pinned layer is fixed in the fabrication, while the direction of the free layer can be changed between two states – parallel or anti-parallel to the direction of the pinned layer. This can be observed as resistance switching, since MTJ with parallel magnetization has lower resistance than MTJ with anti-parallel magnetization.

To program a **conventional MRAM** cell, the magnetization direction switching is induced by applying an external magnetic field. This write mechanism results in poor scalability of the device, as the current required to cause switching is inversely proportional to the cell size. The problem is resolved by introducing **spin-transfer-torque MRAM (STT-MTJ)**, for which the write operation is performed by direct injection of spin-polarized current to the device [19]. In this case, the required amplitude of write current decreases with the area of the junction. Later, various improvements of the technology have been developed, mostly to achieve lower switching currents: the solution with the magnetization perpendicular to the substrate surface (opposed to in-plane

magnetization of the free layer) [20], or more complex penta-layer MTJ [12].



(a) (b)

Figure 2.4: (a) Schematic illustration of the three-layer MTJ in low-resistance state (left) and high-resistance state (right) [12]. (b) Three-terminal spin Hall device structure [21].

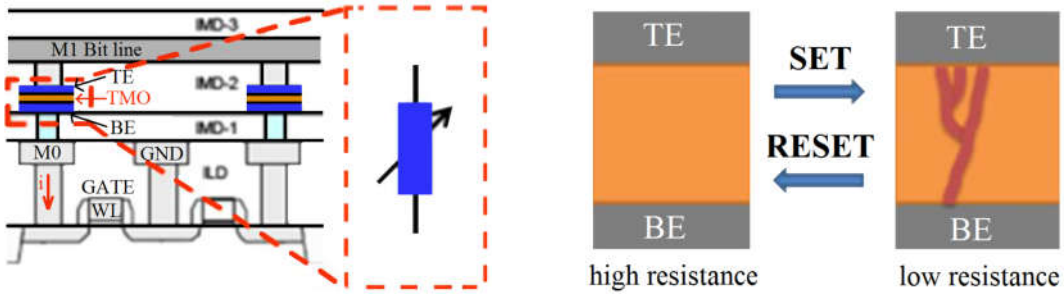
To read MRAM, a small current is flown through the device and its resistance is sensed. Therefore, as the sensing and writing current paths are the same, unintentional write may occur during the read operation. With the reduction of the programming current, which is beneficial for power consumption and the selector size, the device is even more prone to read disturbs, and this imposes a significant disadvantage of the technology. To solve this, **spin hall effect MRAM (SHE)** has been proposed [22], which is built with independent reading and writing paths. It is a three terminal device, with the spin hall metal attached to the free layer of MTJ, as depicted in Figure 2.4b.

Along with the high speed and low programming energy, the main advantage of MRAM which puts it in front of other NVMs is its very high, virtually infinite endurance. It is compatible with current CMOS technology, although its structure is not very simple, because many additional layers are added to improve the device characteristics. Also, the memory ratio between high and low-resistance state is relatively low. This leads to more complex design solutions, as high-sensitivity sense amplifiers are required for reliable read operation.

ReRAM

A **resistive random access memory (ReRAM)** element is a two-terminal, metal-insulator-metal structure (as in Figure 2.5a) which exhibits reversible resistance switching. Applying appropriate voltages/currents across the electrodes cause redox reactions which

form or dissolve the conductive filament(s) between two terminals, resulting in a resistance change. The existence of a conductive path in the insulator produces a low resistive state, while its absence results in a high resistive state, as illustrated in Figure 2.5b. Depending on the materials used in the stack, the mechanism for creation/disruption of the conductive path may be different [23]. Thus, two types of ReRAM are encountered – **conductive bridge random access memory (CBRAM)**, also known as programmable metalization cell (PMC), and **metal-oxide random access memory (OxRAM)**.



(a) (b)

Figure 2.5: (a) ReRAM BEOL integration. (b) ReRAM operations (figures from [24]).

Compared to PCM (see Table 2.1), ReRAM cells require lower operating voltages and programming currents, and exhibit faster switching times (few tens of ns, even less than 1 ns). Due to the filamentary nature of the conductive path, the technology is highly scalable, thus exceeding FeRAM possibilities. Furthermore, one of the major advantages is its compatibility with conventional semiconductor processes. Using only two additional masks at the back-end-of-line (BEOL), and materials and temperatures compatible with CMOS manufacture, it imposes as a simpler and cheaper solution than MRAM. As the technology is characterized by a large memory window (much higher than MRAM), the device can reach several current-controlled low-resistance or voltage-controlled high-resistance states, allowing for MLC. ReRAM has the potential of forming a cross-point structure without using access devices, achieving an ultra-high density. In addition, the usage of ReRAMs is not limited to memory arrays, as devices can be integrated directly on the top of a CMOS processor. The main drawback is the requirement for one-time programming of the fresh cell which must be performed with an elevated voltage (forming). Other aspects which must be considered in development of ReRAM technology are the memory variability and endurance.

Several demonstrators implementing Mb arrays with 1T1R bitcell (one transistor, one

Table 2.1: Comparison of key features of existing and emerging memories (updated from [25])

	SRAM	(e)DRAM	(e)Flash	FeRAM	PCM	STT-MTJ	ReRAM
endurance	unlimited	unlimited	10^6	10^{14}	10^{12}	10^{16}	CBRAM: $10^{5\text{-}6}$ OxRAM: $10^{6\text{-}11}$
write time	<<1ns	eDRAM: 1-2ns DRAM: 30ns	NOR: 1us NAND: 1ms	<100ns	<100ns	2-30ns	CBRAM: 100ns-1us OxRAM: 1-100ns
density	low (6T)	medium	NOR: medium NAND: high (MLC)	low (no MLC)	high (MLC)	medium	high (MLC)
write power	low	medium	high	medium	medium	medium	medium
standby power	medium	low	0	0	0	0	0
CMOS integration	x	off-chip/FEOL	off-chip/FEOL	BEOL	BEOL	BEOL	BEOL
voltages	CMOS	CMOS	~10V	~1.5V	1.5-3V	~1V	1-2.5V
scalability	CMOS	limited	limited	limited	good	good	good
issues		needs refresh	eFlash requires new CMOS process	destructive read	T° sensitive	read disturb low mem. window	forming voltage

ReRAM) have been developed [26–28]. Sandisk and Toshiba presented a 32Gb cross-point memory array designed as a replacement for Flash [29]. Recently, Micron and Sony introduced a 16Gb array that meets storage class memory specifications [30, 31].

2.2 ReRAM background

Figure 2.6 shows the typical quasi-static IV characteristics of a bipolar ReRAM stack, obtained by applying a slow voltage ramp. In the first programming cycle, called FORMING, the resistance of the fresh device changes from a very high initial value (R_{INIT}) to a low value (R_{ON}). The switching, i.e. the abrupt current increase, occurs when the threshold voltage is reached. Then, by applying a voltage of opposite polarity during the RESET operation, the device switches to the high resistance state (R_{OFF}). All successive changes from R_{OFF} to R_{ON} are performed as SET operations, which generally require lower threshold voltage than FORMING. During the SET and FORMING, it is important to limit the current through the device by defining the compliance current (or programming current, I_{COMP}) to prevent the degradation and failure of the device.

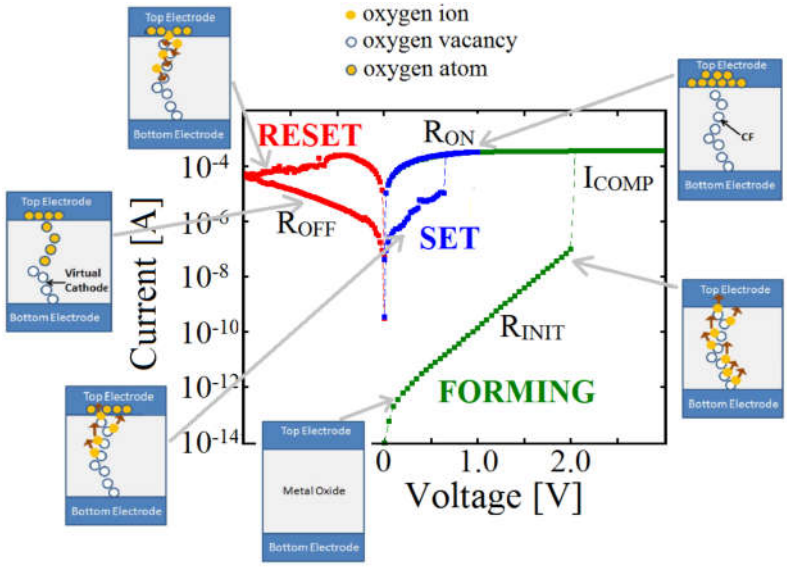


Figure 2.6: Quasi-static I-V characteristic obtained from experimental OxRAM data. The illustration of the switching process in the simple binary OxRAM is taken from [32].

This behavior corresponds to the asymmetric stack design (different materials of top and bottom electrode), leading to the bipolar switching which require different voltage polarity for SET and RESET operations. Alternatively, if same material is used for both electrodes, a ReRAM device can exhibit an unipolar nature, as depicted in Figure 2.7. In that case, the switching operations are performed under different voltage ranges, but with the same polarity. In the following, only bipolar devices will be discussed.

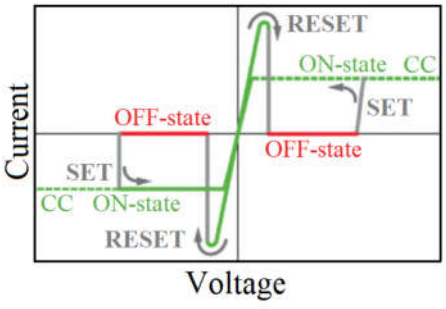


Figure 2.7: I-V curve of a unipolar switching device [23].

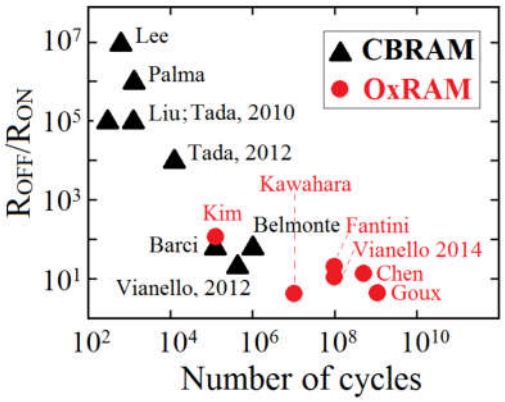


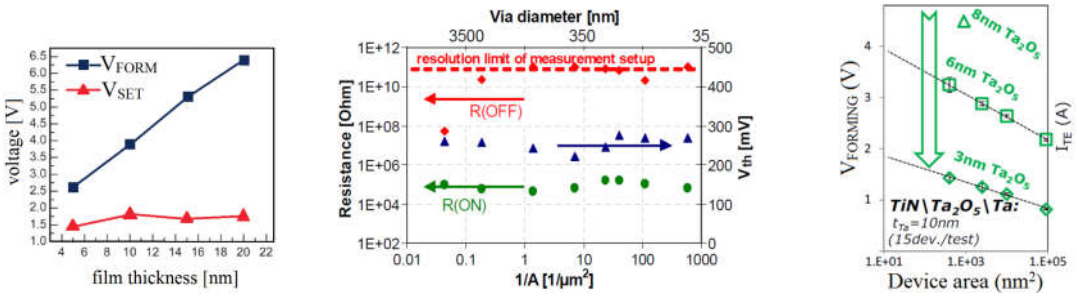
Figure 2.8: Memory window as a function of the endurance performances for CBRAM [33–40] and OxRAM [41–46] technologies.

The presented I-V curve describes the electrical behavior of both, CBRAM and OxRAM devices. However, the physical phenomena which cause the resistive switching in these two device families are different. In OxRAM (illustrated in Figure 2.6), switching relies on migration of oxygen anions (i.e. creation of oxygen vacancies) in transition-metal-oxides towards the top electrode where they react with the oxidizable anode materials to form an interfacial oxide layer [32]. Thus, the electrode/oxide interface behaves like an oxygen reservoir. During RESET, oxygen ions migrate back to the bulk either to recombine with the oxygen vacancies or to oxidize the metal precipitates and return the memory cell to high resistive state. During FORMING, stress-induced defects are generated in the material. As in the following RESET process only a portion of the defects is recovered, the required SET voltage is smaller than the FORMING voltage, and the R_{OFF} is much smaller than R_{INIT}. On the other hand, CBRAM relies on the electrochemical metallization mechanism where during SET the highly mobile cations from the electrochemically active metal electrode drift in the ion conducting layer (an oxide or chalcogenide based electrolyte) and discharge at the inert counterelectrode, leading to a growth of highly conductive metal-rich filament(s) [47]. Then, RESET is an electrochemical dissolution of the metallic filaments.

The materials used in ReRAM stacks and their interfacial properties have a strong influence on the memory performances and reliability. Additionally, various material engineering methods have been investigated to improve their characteristics, mostly to reduce switching uniformity, increase high resistive state values and increase endurance. For instance, in CBRAMs the bottom electrode is usually tungsten, as manufacturing W vias is already the standard CMOS process, while the top electrode can be made from Ag, Cu, or their alloys. The ion conducting layer, typically includes germanium chalcogenides (Ge_xS_y and Ge_xSe_y), or different oxides (e.g., SiO_2, ZrO_2, Al_2O_3, Si_3N_4, TaSiO). Often, resistive switching is improved by introducing uniform and homogeneous impurities in the electrolyte, or by creating a bilayer electrolyte with an inserted buffer layer between the standard insulator and the electrodes. OxRAMs employ various oxydes, particularly binary-metal-oxides which are widely used in the current semiconductor technologies (HfOx, AlOx, NiOx, TiOx, TaOx), with Al or TiN for bottom electrode, and TiN, Ta or TaOx for the top electrode. Therefore, even within one ReRAM family, a wide range of electric characteristics is encountered. Figure 2.8 shows the memory window (R_{OFF}/R_{ON}) versus the cycling performance for different technologies, illustrating the trade-off between the resistance ratio and the number of successful programming cycles. Generally, CBRAMs can achieve a higher memory window at the expense of the

endurance.

One of the disadvantages for ReRAM-based design is the need for the initial electroforming cycle, which requires higher voltage than the subsequent switching cycles. Thus, significant efforts have been made to achieve forming-free devices. Results shown in Figure 2.9a indicate that the required forming voltage decreases linearly with a decrease of the insulator thickness, which on the other hand does not affect SET/RESET voltages. However, reducing the oxide thickness may increase the initial defect density and severely decrease the resistivity of the oxide layer, thus decreasing R_{OFF} resistance. Besides, ReRAM technology is particularly attractive due to its good scalability potential. As depicted in Figure 2.9b, R_{ON} and SET voltage are not impacted by the area reduction, while R_{OFF} increases, thus making the memory window even larger. However, lateral device scaling will increase the required voltage for FORMING operation, as illustrated in Figure 2.9c. Therefore, based on current research results, it can be assumed that a certain electroforming process is unavoidable, which must be taken into account during the design.



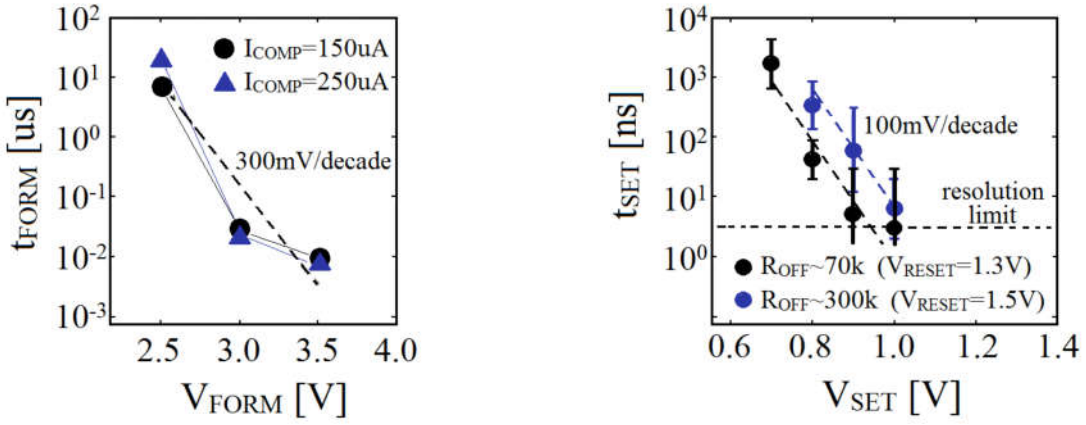
(a) (b) (c)

Figure 2.9: (a) FORMING/SET voltage dependence on the oxide film thickness [23]. (b) Dependence of R_{ON}, R_{OFF}, V_{SET} on device area [48]. (c) $V_{FORMING}$ as a function of device area [46].

2.3 ReRAM electrical characterization

Generally, for a given ReRAM technology, there is a trade-off between the programming voltages (V_{SET}, V_{RESET}, V_{FORM}), programming current (I_{COMP}), memory window (R_{OFF}/R_{ON} ratio), device endurance and power consumption. In this section, the relations between these parameters are demonstrated using experimental results on OxRAM TiN/Ti/HfO_2/TiN stack [41, 49].

One of the essential characteristics of ReRAM cells, highlighted in Figure 2.10, is the exponential dependence between the switching time and the applied voltage – higher voltage causes faster resistance switching. Thus, programming can be performed with wide range of supply levels. However, Figure 2.10b also shows that the higher R_{OFF} before the SET operation (thus larger memory window) demands higher V_{SET} for the same programming pulse.



(a) (b)

Figure 2.10: Exponential dependence between switching time and (a) FORMING, (b) SET voltages for the TiN/Ti/HfO_2/TiN OxRAM cells (figures taken from [41]).

Furthermore, Figure 2.11 demonstrates that higher programming voltages ($\sim$ 2V) are needed in order to improve the tails of the low resistance state distribution. These data have been measured on several tens of OxRAM cells through 500 SET/RESET cycles to capture both spatial (device-to-device) and temporal (cycle-to-cycle) variations.

Finally, simulation results in Figure 2.12 indicate that an increase of SET/RESET voltages also reduces the programming energy. This is the consequence of the exponential relation between switching time and programming voltage. The simulation is performed on a 1T1R cell, using 28nm FDSOI design kit and compact OxRAM model [50]. Various constant *VDDH* voltages are supplying the whole structure. Digital pulses (0/1V) of the *pw* widths, which correspond to the switching time, are applied at the gate. This graph also highlights the discrepancy between SET and RESET programming conditions in digital design. Using the same access path for both operations, especially with only one transistor type, may lead to significantly different switching times for the same programming voltage.

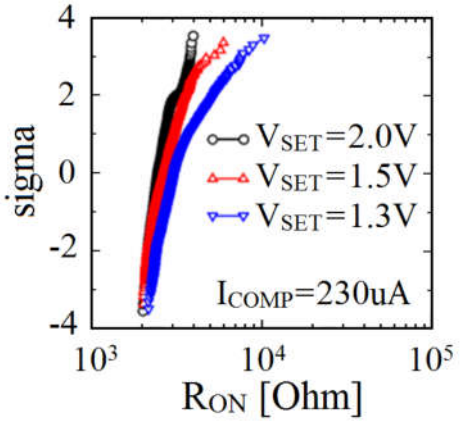


Figure 2.11: R_{ON} distribution for different SET voltages, using fixed pulse width (100ns) and programming current (230μA).

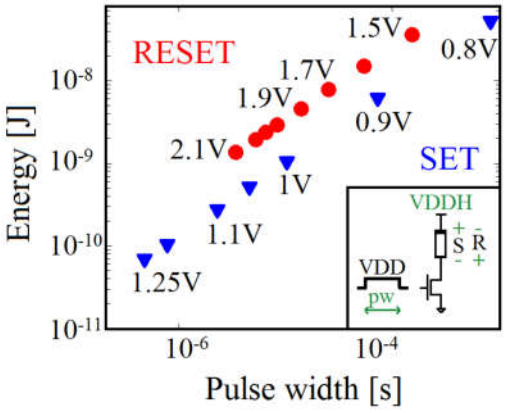


Figure 2.12: Simulated SET/RESET energy versus switching time for various voltage (VDDH) accross the cell shown in the inset.

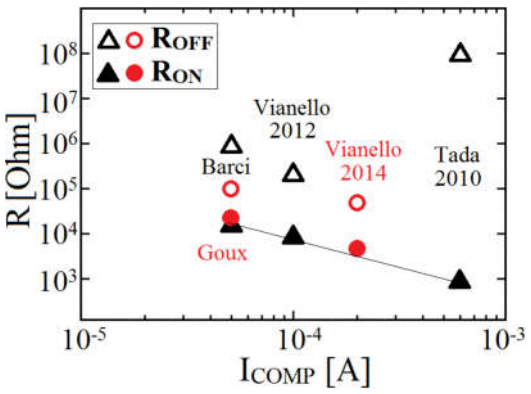


Figure 2.13: R_{ON} and R_{OFF} values versus the programming current (I_{COMP}), corresponding to some of the data presented in Figure 2.8.

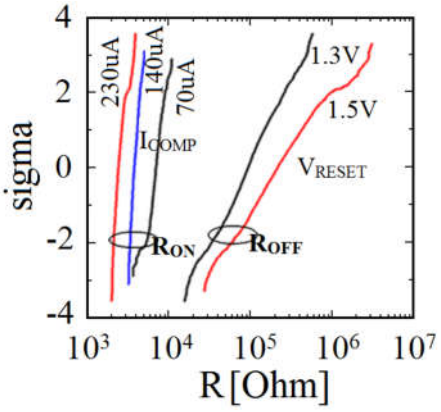


Figure 2.14: Resistance value distribution versus programming conditions (R_{ON} vs. I_{COMP}, R_{OFF} vs V_{RESET}), for the TiN/Ti/HfO_2/TiN OxRAM cells.

Another important ReRAM characteristic is that the low resistive state is defined by the SET compliance current. As illustrated in Figure 2.13, which shows the R_{ON} and R_{OFF} values as a function of the programming current, higher I_{COMP} leads to lower R_{ON}, regardless of the ReRAM stack. In fact, the MLC capability of ReRAM is based on this property, and different resistive levels are achieved by controlling the programming current. On the other hand, R_{OFF} depends on the technology, and generally is not related to I_{COMP}, but rather to the RESET voltage. Figure 2.14 demonstrates how R_{ON} and R_{OFF} values depend on I_{COMP} and V_{RESET} in TiN/Ti/HfO_2/TiN OxRAM devices, respectively.

Resistance distributions on this graph include both spatial and temporal variation.

To reduce the programming energy consumption, it is desirable to use low compliance currents. Moreover, low I_{COMP} proves to have better endurance, as high I_{COMP} gradually leads to degradation of memory window by decreasing R_{OFF} (Figure 2.15).

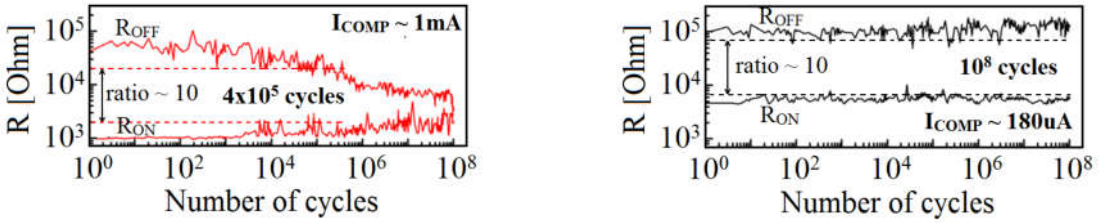


Figure 2.15: Pulse cycling endurance test for a SET compliance current of: (a) 1mA, (b) 180μA. Pulse width = 10μs, V_{SET}=1V, V_{RESET}=1.3V [49].

2.4 Summary and design guidelines

ReRAM stands out as a promising emerging non-volatile memory technology, due to its low power consumption, fast switching, high memory window and excellent scalability. Its simple structure and low cost makes it attractive for usage as high-capacity storage memories. Moreover, ReRAM technology is highly compatible with CMOS, therefore it is not limited to memory arrays, and can be integrated directly into the logic.

In ReRAM-based design, it is important to take into account the impact of ReRAM programming conditions on other device properties, such as resistance values, speed, programming energy and endurance. Generally, R_{ON} and R_{OFF} values can be adjusted by tuning the SET compliance current and the RESET voltage, respectively. Lowering I_{COMP} improves device endurance and reduces programming consumption, at the expense of a higher R_{ON}. The lower memory window can be compensated with a higher V_{RESET} that allows R_{OFF} to be increased. Maximizing programming voltages also favorably lowers the programming energy and overcomes the device variability in large memory arrays. Also, devices need the FORMING operation, which is performed under a voltage higher than V_{SET} and V_{RESET}. Thus, ReRAM cells require programming voltages higher than the typical operation voltages of advanced CMOS technologies.

Chapter 3

Related work

Contents

3.1 Design of non-volatile circuits

This chapter summarizes various NVFF designs found in the literature by explaining different aspects of their architecture – power supply organization, circuitry for different operating modes, etc. The solutions are analyzed with regard to used NVMs and CMOS nodes, as the choice of technology can greatly influence NVFF design decisions. The extracted conclusions provided in this chapter can be used as guidelines for the future design of non-volatile flip-flops.

Finally, an overview of existing non-volatile processors is given, in order to investigate the application of NVFFs in complex systems.

3.2 Non-volatile flip-flops

3.2.1 State of the art solutions in different NVM technologies

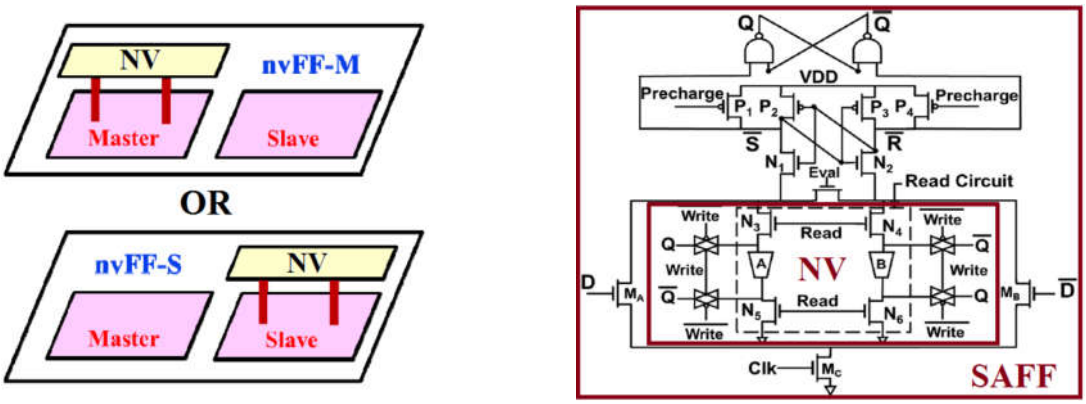
With the CMOS technology scaling, leakage power consumed during the inactive mode of the system may take the significant portion of the total consumption [4]. As one of methods for static power reduction, **data-retention flip-flop** [8] has been developed to replace the standard flops in the processing unit. Small retention latch attached to the flip-flop can take its state (store operation), preserve it during the sleep mode, and return it back to flop (restore operation). Therefore, the supply of flip-flop core can be turned off during sleep, while the retention latch supply is maintained at reduced level.

To achieve even lower consumption, **non-volatile flip-flop** appears to be good substitution for data-retention flip-flop, as it provides zero-consumption standby mode, while enabling relatively fast store/restore transitions. Recently, various NVFF solutions built by adding NVMs to flip-flops have been proposed. As the most mature of emerging non-volatile memories, FeRAM is used in several silicon-proven NVFFs [14,51,52]. Then, MRAM-based designs have been introduced: NVFF using a conventional MTJ is fabricated [53], numerous NVFFs employ STT-MTJs [54–62], while some rely on SHE MRAM [21,63]. Finally, ReRAM-based NVFFs have been described [64–69].

3.2.2 Design architecture

Top level

A majority of reported NVFFs are obtained using the **master-slave flip-flop**, by adding **1-bit** non-volatile part (which may include several NVM devices and their store/restore circuitry) to either master or the slave stage, as illustrated in Figure 3.1a. In the similar way, NV part can be attached to the **sense-amplifier flip-flop** [62,69], as in Figure 3.1b, or other pulse-triggered latches. Generally, volatile master-slave flip-flops have better race immunity and lower consumption, but they are slower than volatile sense-amplifier flop. However, the performance and consumption of non-volatile flip-flop in the regular operating mode depends on both, the chosen flip-flop core and NV part, as well as their connection.



(a) (b)

Figure 3.1: 1-bit NVM added to: (a) master-slave flip-flop (figure taken from [66]), (b) sense amplifier flip-flop (figure taken from [69]).

Additionally, the architectures which store **multiple bits** in one NVFF have also been introduced. For example, in [61] four MTJs are tied to one flip-flop and one store/restore circuitry to form a 4-bit NVFF, as shown in Figure 3.2. For even higher number of bits, authors in [68] propose attaching the small cross-bas memory to the flip-flop (Figure 3.3). These solutions are intended for implementation in multithreaded processors, for saving several contexts without the replication of registers.

Power supply

Most of the reported NVFFs use **single-rail** for the programming of NVM devices and operating the flip-flop core, as this approach has the simplest power grid routing and minimizes the signal routing congestion. Then, depending on the required programming conditions and CMOS operating voltage, the supply may need dynamic change in different modes. For example, MRAM programming voltage is generally fully compatible with the CMOS range, so the nominal operating voltage can be used during the regular flip-flop and store operations. However, due to narrow memory window of the technology, it may impose the increase of supply during the restore operation, as suggested in [53,62]. On the other hand, in ReRAM-based solutions, NVFF supply is increased during programming of NVM. In this case, the CMOS-NVM voltage gap of cells designed in wider CMOS nodes (130nm and above) is tolerable. But, with scaled CMOS processes, the maximum CMOS operating voltage becomes lower than the required ReRAM forming and programming levels, causing the reliability issues in the whole single-rail NVFF.

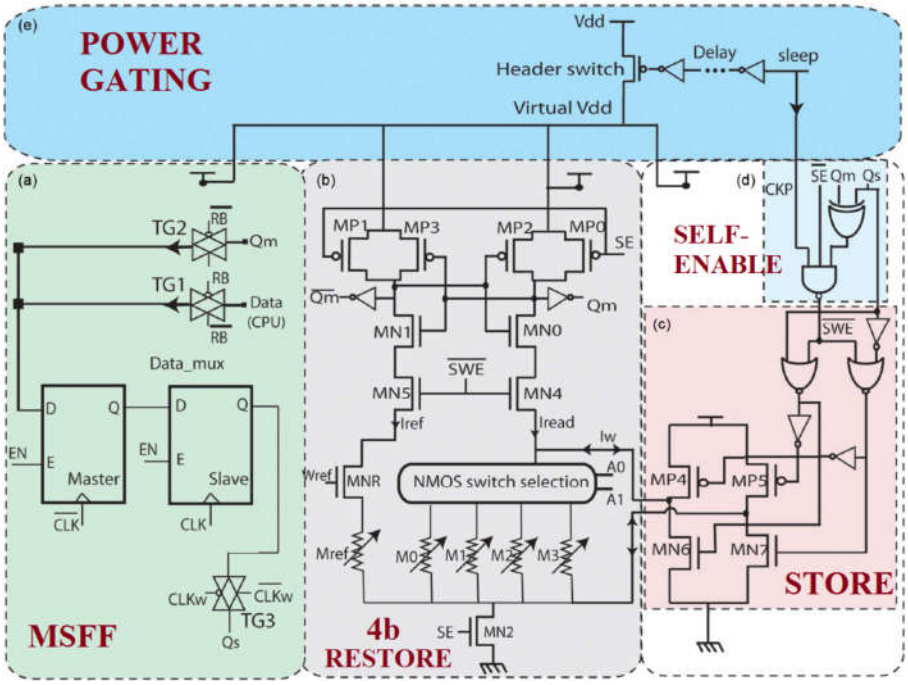


Figure 3.2: 4-bit NVFF with self-enable circuit. Store and restore circuits are separated from master-slave flip-flop [61].

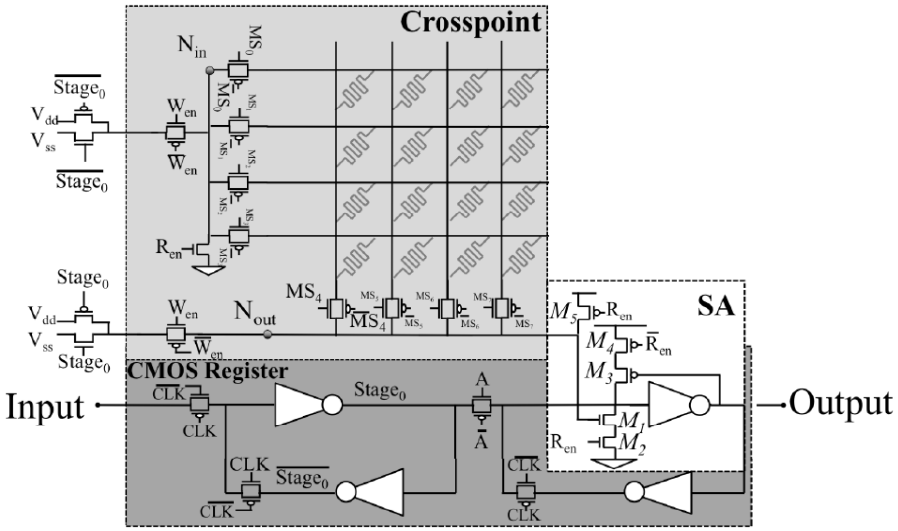


Figure 3.3: Multistate register with cross-bar NV part [68].

To reduce the impact on power consumption in regular flip-flop operation, authors in [14] apply the **dual-rail** configuration in which dedicated power rails for NV, implemented at the same voltage level as the main supply, allow the power gating of the programming

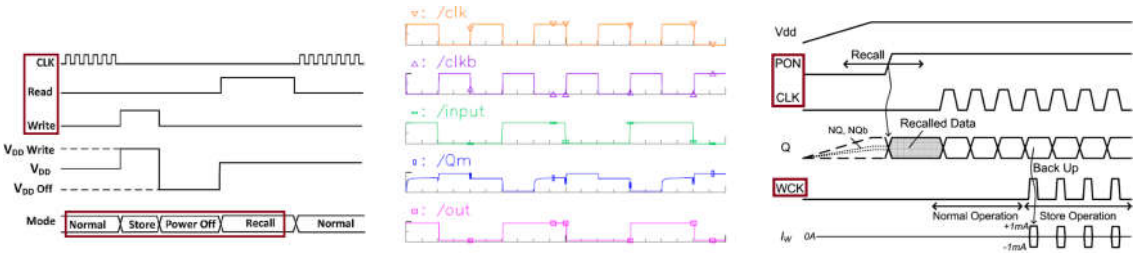
part when it is not needed.

Finally, NVFF presented in [55] uses the alternative approach using a **dual-voltage** architecture. As the solution does not use level-shifting, the significant current leakage is present during store operation. Moreover, the transistors supplied at high voltage are not protected.

Operating modes

Typically, proposed NVFFs require clear distinction between operating modes – store and restore procedures are controlled by dedicated signals and can not be performed during the regular flip-flop activity, as depicted in Figure 3.4a. On the other hand, NVFF presented in [54] stores the data to NV in each clock cycle (Figure 3.4b). Finally, solution introduced in [53] is the combination of both designs – the back-up requires dedicated signal, yet it is possible to perform it simultaneously with the regular write operation, as shown in Figure 3.4c.

Even though the "fully non-volatile" option which stores in each cycle has simpler circuit control, its performance and dynamic consumption are degraded as the speed of NVM technology is lower than of CMOS, and the write energy is higher. Additionally, the endurance of such system is limited by NVM endurance. Thus, ReRAM-based NVFFs support only "the back-up when needed" option (e.g., before going to sleep mode, or at the checkpoints).



(a) (b) (c)

Figure 3.4: Operating modes: (a) store/restore/active are separated (figure taken from [70]), (b) store to NV in each cycle (figure taken from [54]), (c) store can be performed in parallel with write (figure taken from [53]).

Restore operation

The most common approach for data restore (in literature also referred to as: sensing, read, recall) is **differential** restore using **2 NVM**. It relies on sensing the current difference between two branches containing non-volatile devices which are programmed to different states. Depending on the sense amplifier which is used, several possibilities illustrated in Figure 3.5 are found in the literature. Numerous NVFFs have NVM branches "outside the latch", i.e. tied to the gate nodes of cross coupled inverters, as in Figure 3.5a. On the other hand, many MRAM-based NVFFs connect NVMs to the sources of the latch transistors ("inside the latch", Figure 3.5b), as this solution offers higher yield of restore operation. However, in case the sense amplifier is actually the slave stage of flip-flop, this configuration may increase the flip-flop propagation delay. The impact on performance is reduced by adding restore-access and ground transistors, as depicted in (Figure 3.5c). Still, resistance added in series has negative influence on the restore yield, which can be significant for low-memory window technologies as MRAM. Finally, NVMs can be tied to the gates of voltage sense amplifier, as in Figure 3.5d.

The variation of this architecture is **differential** restore using **1 NVM**, as implemented in [61] (Figure 3.2). One programmable NVM device is compared to the reference device. However, referent NVM device also requires one-time programming and is susceptible to the read disturb. Additionally, this halves the effective NVM window and may cause insufficient restore yield. To compensate this, the authors suggest NVFF area increase.

Furthermore, the differential restore circuit can be integrated in NVFF in two ways. Usually, already existing latch (e.g., slave stage of flip-flop) is used as a restore sense amplifier – restore circuit is **merged with flip-flop**, as illustrated in Figure 3.5c. Alternatively, is can be **separated from flip-flop**, as implemented in solution in Figure 3.2. At the cost of increased area, the latter allows separate optimization of regular mode and restore mode circuits. This is introduced in MRAM-based NVFFs where the low sensing current is imposed as a requirement in order to prevent the read disturb [71]. Thus, in case the small sense amplifier is designed and used as slave stage, the flip-flop performance is degraded. Typically, ReRAM is not disturbed at CMOS nominal voltage, so the first option can be employed for saving the area. Note that in case the latch is not re-optimized for sensing, merged restore may actually have lower impact on flip-flop performance than the separated one.

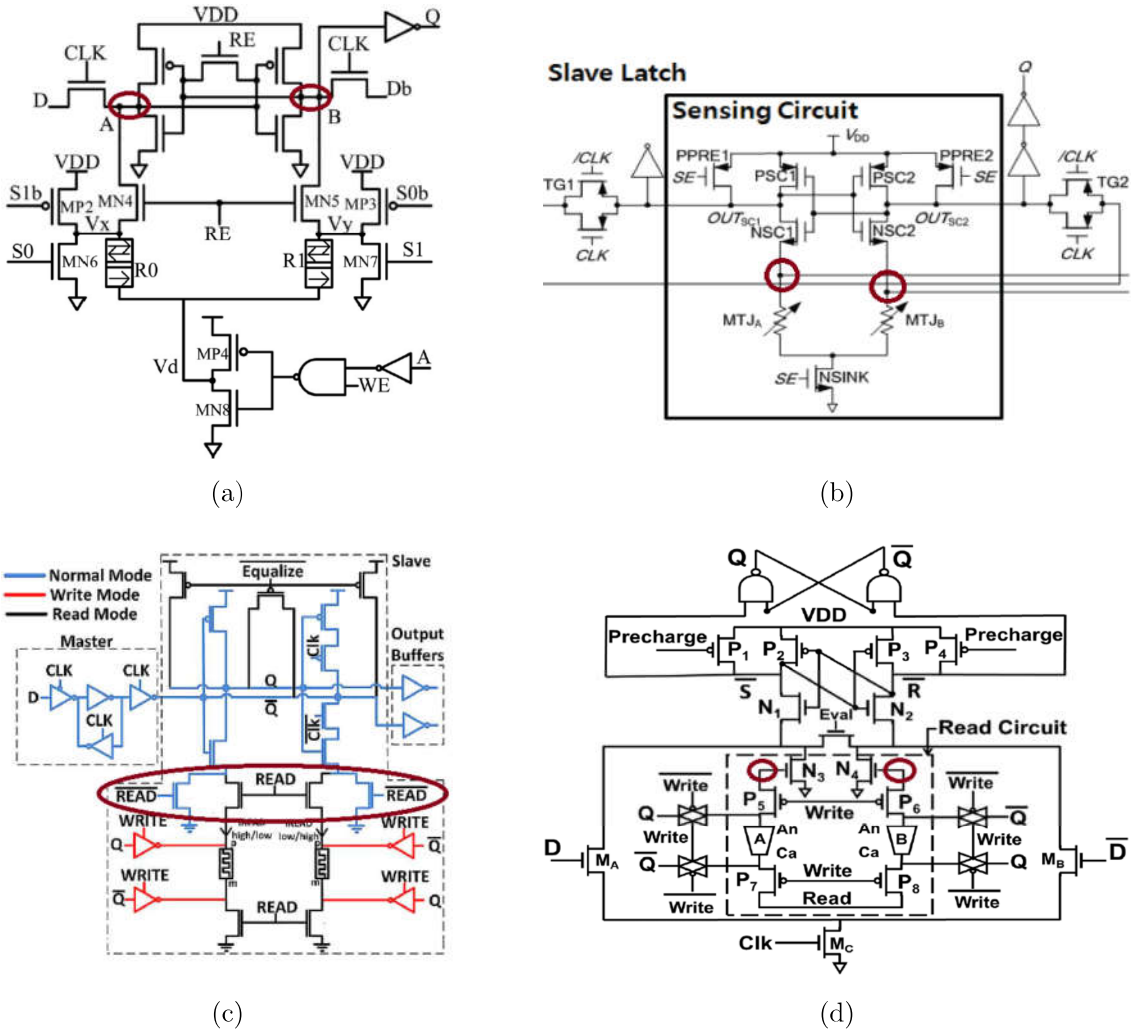
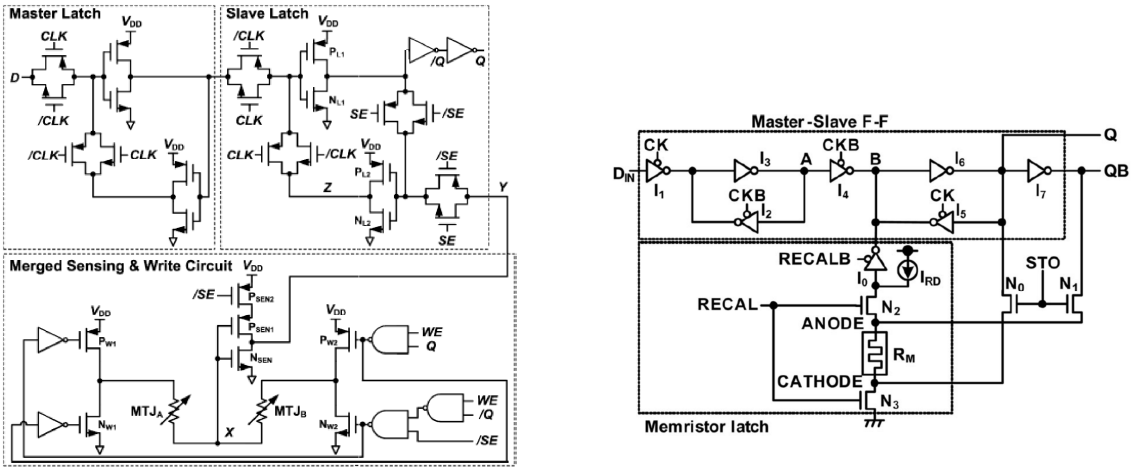


Figure 3.5: 2 NVM differential restore, with NVM branches: (a) outside the latch (figure taken from [58]), (b) inside the latch (figure taken from [55]), (c) inside the latch, with access transistors (figure taken from [70]), (d) tied to voltage sense amplifier [69].

On the other hand, restore operation may rely on **single-ended** sensing. **2 NVMs** can be configured to form a voltage divider, as shown in Figure 3.6a. Similarly, **1 NVM single-ended** restore solution is given in Figure 3.6b. Compared to differential circuits, these architectures may lead to lower yield and slower restore operation.



(a) (b)

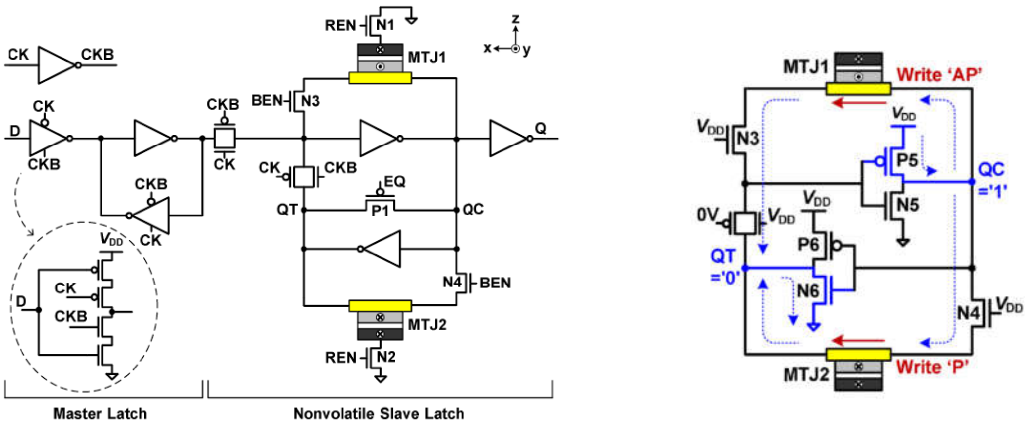
Figure 3.6: Single-ended restore, using: (a) 2 NVM (figure taken from [60]), (b) 1 NVM (figure taken from [64]).

Store operation

In most cases NVM devices have dedicated programming transistors activated only during the store operation (in literature also referred to as: write, backup), thus the store circuit is **separated from flip-flop** (for example, see Figures 3.5a, 3.5c and 3.5d). On the other hand, store circuitry can be **merged with flip-flop** so that the transistors of the latch are used for programming, for a lower transistor count, as in Figure 3.7. Typically, if an NVM device requires higher currents, the latter implies increasing the latch size, which further negatively impacts the propagation delay and the power consumption. Hence, this approach is limited to low-programming current devices (e.g., SHE MRAM).

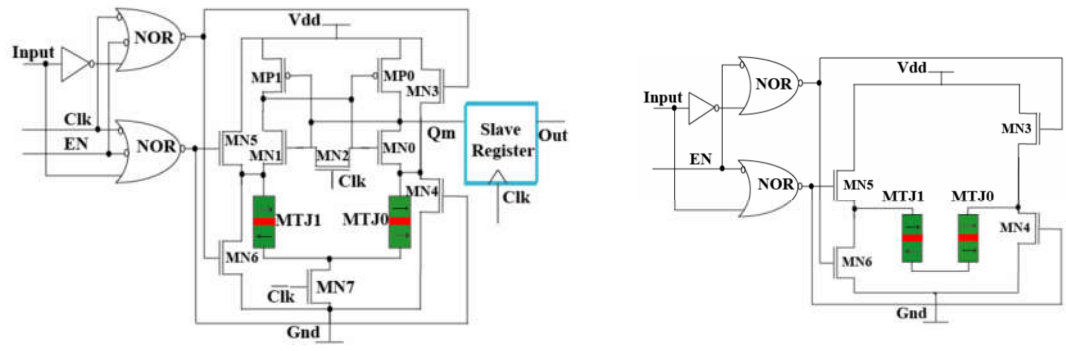
For non-volatile flip-flops that contain two NVMs, three storing methods are encountered in the literature. A majority of MRAM-based NVFFs perform **serial** store where two serially connected devices are programmed simultaneously, as illustrated in Figure 3.8. This solution is not appropriate for voltage-programmed NVM technologies (as ReRAM), since it doubles the required power supply. Then, devices can be programmed in **parallel**, using either the same transistors (Figure 3.7), or separate transistors (Figures 3.5c and 3.5d). To limit the supply current and the number of transistors, at the cost of slower store procedure, some configurations implement **2-step** store, shown in Figure 3.9.

Finally, designs proposed in [52, 61, 64] implement conditional store (compare and write) in order to reduce the store consumption (Figure 3.10a, Figure 3.10b).



(a) (b)

Figure 3.7: Store circuit merged with the slave latch, with parallel programming [21]: (a) full NVFF, (b) latch/programming transistors during store Q=0.

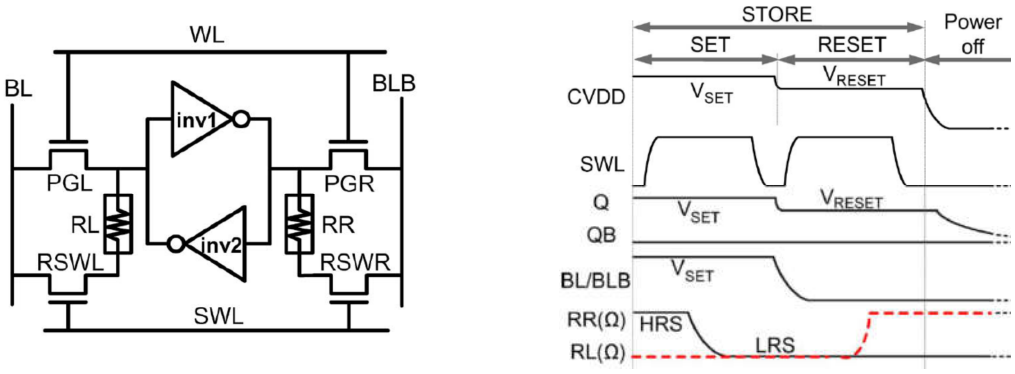


(a) (b)

Figure 3.8: Serial programming (figure taken from [54]): (a) full NVFF, (b) the store circuitry.

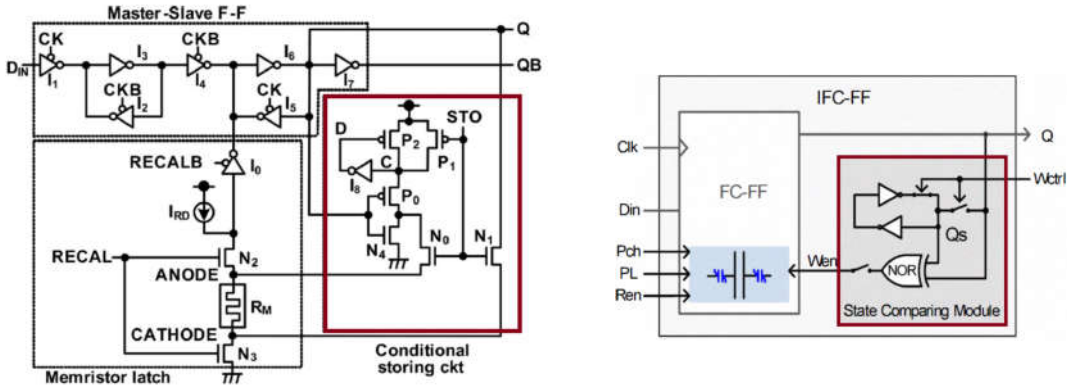
3.3 Non-volatile processors

Incorporating non-volatile memories into processing units offers lots of advantages over the volatile systems. Along with the static power reduction in the standby mode, it increases the system reliability by saving the system state during regular checkpoints or in case of sudden power failure. Then, system can later be returned to the last saved point. If granularity of non-volatile storage is maximal, all processor internal states are saved (including the currently executing instruction, temporary calculation results, pipeline registers, etc.), so the processor can continue working from the exact point it stopped.



(a) (b)

Figure 3.9: 2-step programming (figure taken from [66]): (a) programming circuit, (b) store timing diagram.



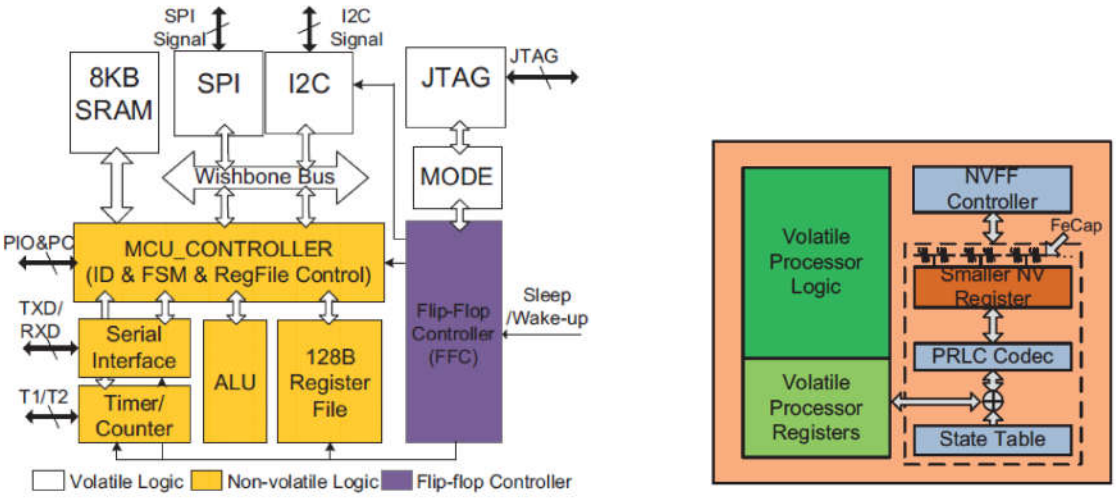
(a) (b)

Figure 3.10: NVFFs with the self enable: (a) [64], (b) [52].

In the current systems, non-volatility is provided by adding **centralized non-volatile module** on-chip or off-chip. As they employ Flash memory, the store/restore transfers are slow and with high energy consumption. Hence, they are not efficient for data-retention in case of power loss or too frequent checkpoints. However, this can be significantly improved if Flash is replaced with some of emerging NVMs. For example, ultra low power microcontroller with FeRAM embedded memory module which has been demonstrated [72] reports 1000x faster write capability at 100x lower power compared to typical Flash-based system. To ensure state back-up, FeRAM power supply has the capacitor sized to guarantee complete memory transfer. Centralized-memory approach does not bring overhead in regular processor performance and consumption, nor increases testability and reliability concerns. Still, in case of complex systems it results in slow sleep/wake-up, as

the transfer bandwidth is limited by the bus size.

Another approach for building non-volatile processor relies on **integrating non-volatile elements at flip-flop level**. Due to current NVM technologies limitations (endurance, speed, consumption), fully NV architecture where everything is directly stored in a non-volatile way is not practical. Thus, flip-flop can be replaced with NVFFs which behave as standard flops in regular operating mode, and can store the data in non-volatile way when it is needed, as described in previous section. Compared to centralized-memory architecture, this can provide almost instantaneous sleep/wake-up, as many cells can be stored/restored simultaneously. First implemented FeRAM-based microcontroller of this kind [73] is shown in Figure 3.11a. Obtained by replacing all FFs in the core (∼1600) and 128B register file with NVFFs [52], it is characterized by zero standby power, 7μs sleep time and 2μs wake-up time. The flip-flop controller is used to generate control signals to both NVFFs and FFs in the system. The authors later present the improvement of the processor [74] illustrated in Figure 3.11b, by implementing the compression-based recovery architecture. Instead of replacing all FFs, they propose adding smaller number of NVFFs which save the compressed difference between the current and the last state.

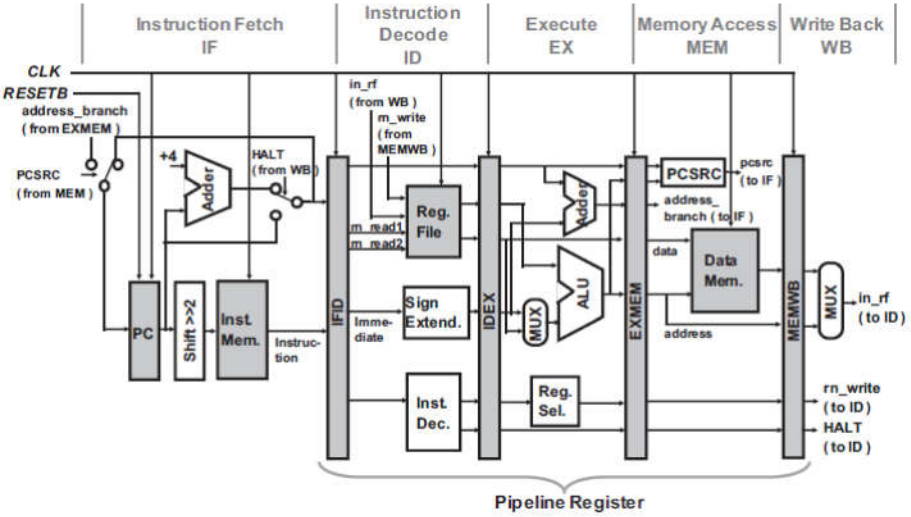


(a) (b)

Figure 3.11: (a) FeRAM-based MCU proposed in [73]. (b) Improved compression-based architecture [74].

Afterward, 32-bit power-gated processing unit using STT-MTJ [75] is demonstrated (Figure 3.12). In this implementation, NVFFs are used for all pipeline registers and program counter, while MRAM modules are used for the instruction memory, data memory and

the register file. Furthermore, 16-bit microcontroller [76] shown in Figure 3.13, fabricated using MRAM and 90nm CMOS technologies, is characterized by 120ns wake-up time. Its processor core has all FFs (∼4000) changed for NVFFs [53], while 64kB RAM/ROM unified macro is also MRAM-based. Apart from a CPU-activated store/restore, it supports software control of NV operations handled by two additional instructions. Its power-management module ensures three low-power modes.



(a)

Power State	Core VDD	Periphery VDD	Core Clock	Periphery Clock	Processor State	Cache Memory	Idle Power	Entry/Exit Delay	Comment	Low Power Technology
C0	Full Supply	Full Supply	On	On	Keep	Full W/R Access	Large	No	Operating	Normal Operation
C3	Off	Off	Off	Off	Keep	Keep Data	Zero	Small	Power Off	Power Gating

(b)

Figure 3.12: MRAM-based MPU proposed in [75]: (a) architecture, (b) power states.

Replacing each FF with its non-volatile counterpart implies large area overhead, as store/restore circuitry is replicated in each flop, and NV power rail (if present) must be routed through the whole processing unit. Also, error correction methods can not be implemented if NVFFs are not organized in a regular array. Thus, a **hybrid approach** is introduced [77]. In this FeRAM-based microcontroller, 256-b mini-arrays which have direct access to the system flip-flops are used to backup their data (Figure 3.14). Volatile data-retention FFs are slightly modified to allow the restore operation. Each array bitcell, with 4 FeRAMs, contains sense amplifier for better restore margin. Control signals are generated in NVL controller that can store/restore to all of the NVL arrays in parallel

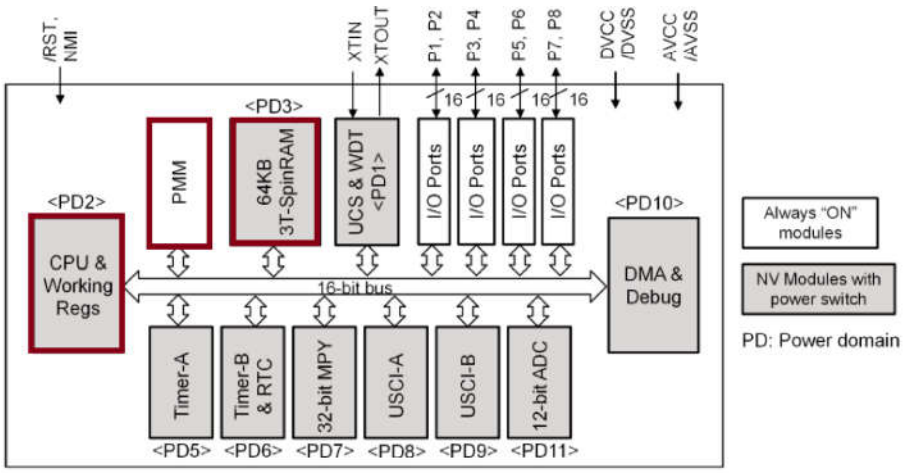
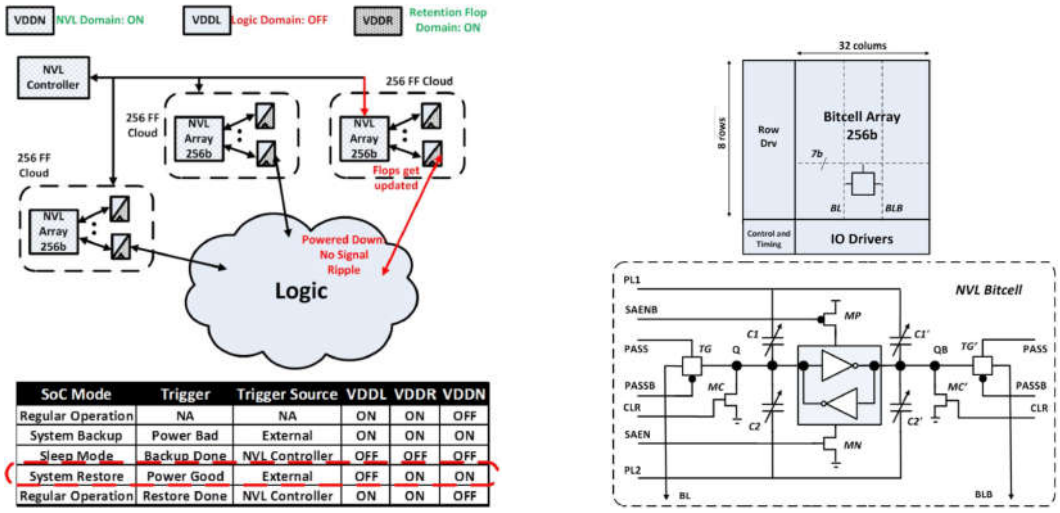


Figure 3.13: MRAM-based MCU proposed in [76].

or serial order depending on the time requirements and power possibilities of the system. Three power domains are present – VDDN for arrays and the controller, VDDR for the slave stage of all FFs (retention), and VDDL for all of the remaining logic, master stage and memories.



SoC Mode	Trigger	Trigger Source	VDDL	VDDR	VDDN
Regular Operation	NA	NA	ON	ON	OFF
System Backup	Power Bad	External	ON	ON	ON
Sleep Mode	Backup Done	NVL Controller	OFF	OFF	OFF
System Restore	Power Good	External	OFF	ON	ON
Regular Operation	Restore Done	NVL Controller	ON	ON	OFF

(a) (b)

Figure 3.14: Details of FeRAM-based MCU proposed in [77]: (a) hybrid mini-NVL array architecture, (b) NVL array and the bitcell.

3.4 Summary

An adequate quantitative comparison of existing NVFF solutions is impossible, due to limited reported results, wide range of used CMOS and NVM technologies, different simulation conditions, etc. However, presented overview indicates that the choice of NVFF topology is strongly affected by the features of NVM devices and their compatibility with the CMOS operation conditions. Different components of these two aspects, design and technology, then together define the overall NVFF characteristics (Figure 3.15).

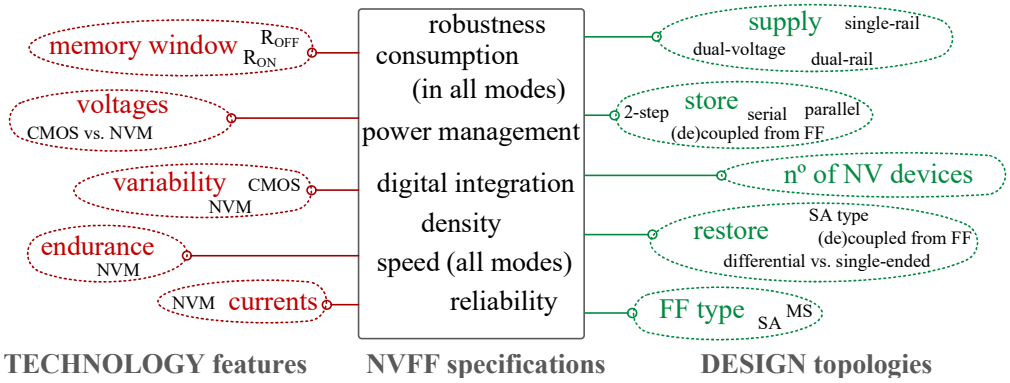


Figure 3.15: NVFF characteristics depend on technology and design.

In the case of ReRAM-based design in advanced CMOS node, programming voltage conditions imply using dual-voltage configuration, which further requires careful consideration of power management and routing of power grid. On the other hand, required programming voltages reduce the possibility of read disturb. This allows for the merged implementation of restore circuit and the flip-flop, leading to a dense solution with small time penalty. Generally, large memory window of ReRAM results in reliable restore operation, thus offering more design flexibility. For example, it enables NVFF architecture with only one NVM device, which is beneficial in terms of area and programming consumption. In addition, the variability of both CMOS and ReRAM can be overcome with the proper choice of sense amplifier circuit.

Good compatibility of ReRAMs with CMOS process facilitates its integration in non-volatile processor. From the aforecited approaches, the one based on the replacement of flip-flop with NVFFs imposes itself as the simplest solution. It requires minimal design time, while offering fast sleep/wake-up transitions and smallest impact on regular flip-flop mode. In addition, the area overhead can be reduced by organizing NVFFs in a regular array with shared store/restore circuitry, wherever it is possible (e.g., register file).

Chapter 4

Design solutions for non-volatile flip-flops

Contents

4.1 Challenges

The main challenges for implementing ReRAM-based circuits in scaled CMOS technologies stem from the programming requirements presented in Chapter 2. First, the results

have shown that **the programming voltages for ReRAM devices are higher than the typical operating voltages of advanced CMOS technologies**. SET/RESET voltages above 1.5V are needed to improve the bit tails in large memory arrays, especially when working at low programming currents, while FORMING operation requires even higher voltages (>2.5V). Moreover, due to the exponential dependence between the switching time and the voltage, increasing the voltage results in lower consumption than increasing the pulse width. Second, the measurements underline that **low programming currents are preferred in order to reduce power consumption and increase device endurance**, at the cost of an increased variability and reduced memory window. Third, **there is a general discrepancy between SET and RESET programming conditions**. In order to target specific R_{ON} and R_{OFF} values with the same pulse widths, different voltages across the device may be required. Finally, **protecting the CMOS devices for reliability is needed** since ReRAM programming voltages can cause the hot carrier degradation of the CMOS transistors. For example, in 28nm FDSOI the voltage of $V_{DS} = V_{GS}$ =1.6V would result in 10% degradation of drain current after less than 1s of accumulated stress [78], leading to the poor endurance of the system. Moreover, forming voltage may cause the oxide breakdown of the transistors.

Regarding the implementation of ReRAM-based non-volatile flip-flops, it is important to **add non-volatile feature with minimal impact on flip-flop performance and consumption**. Another challenge consists of **ensuring the sufficient NVFF yield even with low-memory window technologies and at low voltage**. Given the high variability of ReRAM technologies, which is worsened with low programming currents, R_{OFF}/R_{ON} memory window can be degraded, causing the restore operation failure. For the full compatibility with the digital design flow, it is required to **implement NVFFs as standard cells, using thin-gate oxide transistors**. Finally, **the forming operation should be executed in one step and independent on the value of flip-flop**, for the easier integration of NVFF in larger systems.

In this chapter, two classes of NVFF design with different restore operation are presented: two-NVM (2R NVFF) and novel one-NVM (1R NVFF) architecture. In the context of store operation, two ReRAM programming solutions are proposed: level-shifter-based and current programming-based. Combining them, four NVFF cells and the optimization guidelines to meet all presented challenges are offered. The solutions are compared from the design point of view, and with respect to the existing ReRAM technologies.

4.2 Top level

The proposed NVFF architectures are built by adding non-volatile property to the standard C^{2}MOS master-slave flip-flop (MSFF). On the rising edge of the clock the data is captured in a volatile part, while the back-up is performed only when needed (e.g., before going to sleep mode, or at the checkpoints), thus minimizing NVFF dynamic consumption. Consequently, the switching time of non-volatile memory is not limiting the flip-flop performance, and the required endurance for NVM is significantly smaller than the total number of FF clock cycles.

The common block diagram of implemented solutions, depicted in Figure 4.1, consists of three main parts:

- the adapted MSFF core (MASTER and SLAVE), connected to
- a non-volatile block (NV) to store and restore the flip-flop data, and
- a control logic block (LOGIC) which manages store, restore or forming operations in addition to regular flip-flop operations.

Decoupling the NV block from the flip-flop core enables the separate optimization of ReRAM programming circuit and the slave stage, hence the consumption and performance penalty during standard flip-flop operations are minimized.

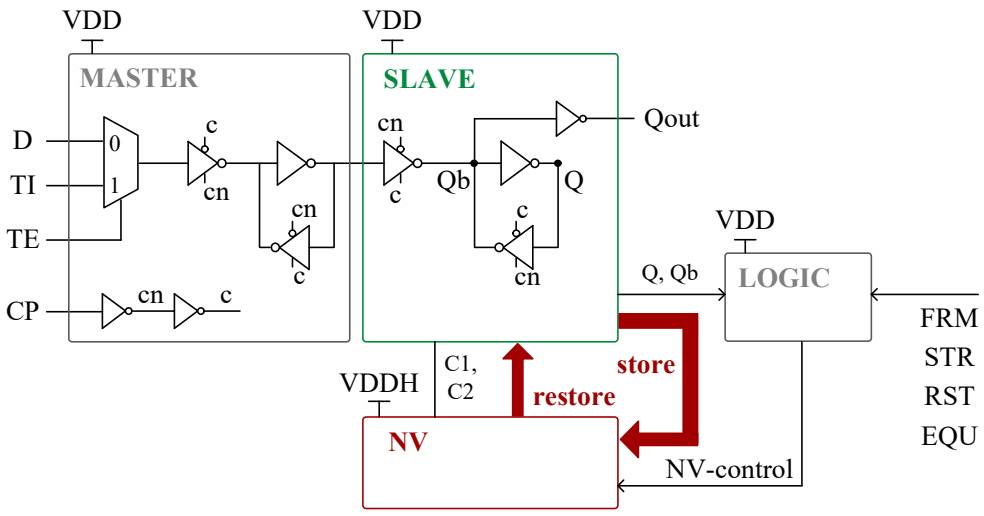


Figure 4.1: Block diagram of NVFF.

Besides the basic data (D), clock (CP), test-enable (TE) and test-in (TI) inputs, and $Qout$ output, NVFFs have additional NV-related inputs: FRM for forming, STR for

store, RST and EQU for restore operation. Added inputs are used together with flip-flop state (Q) for generation of NV-control signals.

The cells use two power rails. Core and control logic block are supplied at nominal CMOS operating voltage (VDD), while NV operates at higher voltage (VDDH) to satisfy the programming requirements for ReRAM FORMING, SET and RESET. Hence, the transistors stacking required for high voltage protection is implemented only in VDDH domain. Compared to the single-rail solution with dynamic voltage scaling [67], the dual-rail scheme minimizes the CMOS degradation, since single-rail design has all transistors of flip-flop exposed to high voltages during store and forming.

NVFFs are designed to work in five modes:

- **active mode** is the regular FF operating mode during which the NV part is disconnected;
- **store mode** stands for saving the flip-flop data in the NVM;
- **sleep mode** indicates that all supplies are off, and can be entered after data-backup;
- **restore mode** stands for recovering the saved context from the NV and it is performed after returning from the sleep mode;
- **forming mode** corresponds to the forming of ReRAMs and it is entered once, before the first use of NVFF.

During the execution of asynchronous NV operations (store, restore and forming) the clock is low, as they require the slave stage to latch the data and be disconnected from the master stage.

The testing of the active mode does not differ from a standard scan flip-flop testing. Moreover, the scan inputs can be used to test store and restore modes after the ReRAM devices have been formed, by performing the following sequence:

1. scan in a first input vector in the chain,
2. store all NVFFs,
3. scan in a second input vector with dual values in the chain,
4. restore all NVFFs,
5. scan out to check the flip-flop values.

Finally, it should be mentioned that NVFF configurations proposed in this chapter can

be applied to various flip-flop designs (e.g., FF without scan inputs, transmission-gate FF, FF with negated output, FF with asynchronous set/reset, etc.). However, the choice of MSFF core may influence the NVFF characteristics, especially in active and restore modes. For example, in Chapter 5 it will be demonstrated that the impact on the propagation delay can be greatly increased if NVFF uses flip-flop shown in Figure 4.2, instead of one in Figure 4.1.

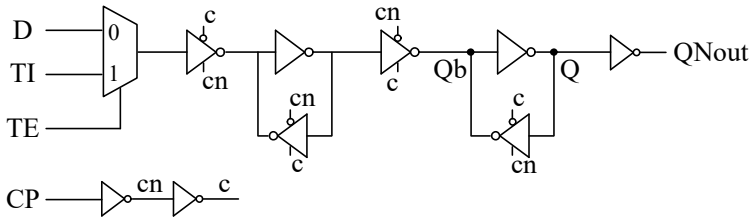


Figure 4.2: Flip-flop core (MSFF_QN) with higher impact on the propagation delay.

4.3 Restore operation

4.3.1 2R and 1R non-volatile flip-flops

In general, restore operation is performed as a differential sensing of two ReRAMs programmed to the opposite states (R_{ON} and R_{OFF}). To save the area, the slave latch is used as a sense amplifier. Therefore, the restore circuit inside the NV block of 2R NVFFs, shown in Figure 4.3, contains devices R_1 and R_2 and restore access transistors N_1, N_2, N_7 and N_8. Additionally, N_3-N_6 are inserted and biased with VDD to protect access transistors during the store mode, when higher voltage is brought to ReRAM top/bottom electrodes. Although not crucial for the restore functionality, N_3-N_6 affect restore yield by increasing CMOS front-end parasitic resistances.

At nodes C_1 and C_2 NV block is tied to the modified NVFF slave latch shown in Figure 4.4. The P_1-P_3 PMOS transistors controlled by an equalizer signal (EQU) are used to precharge Q and Qb nodes at VDD at the beginning of a restore operation. Then, they are released while restoring the data from the NV block. The additional transistors in the latch (P_B, N_B) balance the parasitic resistances of the branches, while the inverter IV_1 balances the parasitic capacitances of the nodes Q and Qb, allowing the same yield to RESTORE 1 and RESTORE 0. Together with IV_2, IV_1 also provides Q_s/Qb_s

outputs which are used to generate NV-control for the store operation instead of Q/Qb nodes, ensuring a minimal impact of the store operation on the latch performance. C_1 and C_2 connections are the sources of NMOS latch transistors, and they are connected to ground by additional N_{G1} and N_{G2} transistors when the cell is not in the restore mode. Figure 4.5 illustrates the timing diagram of the restore operation for the case when the resistance of R_1 is higher than resistance of R_2.

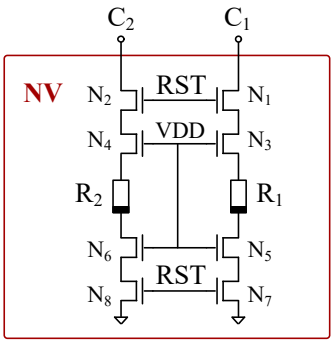


Figure 4.3: NV block of 2R NVFFs (restore operation).

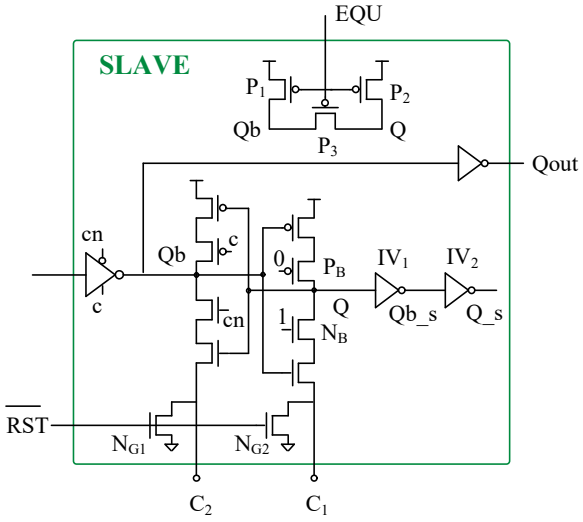


Figure 4.4: Slave stage of NVFF.

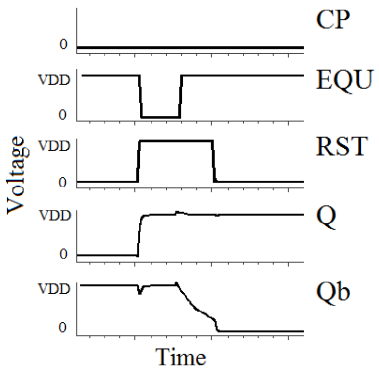


Figure 4.5: Timing diagram for RESTORE 1 (R_1>R_2).

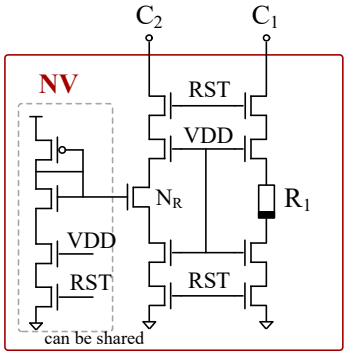


Figure 4.6: NV block of 1R NVFFs (restore operation).

To limit the area overhead and to reduce the store consumption, new 1R architecture derived from 2R is proposed (Figure 4.6). The same differential sensing approach is used to compare a single programmed ReRAM device R_1 to a reference resistance designed with

current mirror. Compared to the state of the art [61], the current mirror reference does not need to be pre-programmed, and it is not limited by poly-resistance area overhead for high resistance values. The area saving for 1R NVFF architecture is maximized when several NVFFs are using the same current source (in dashed line in Figure 4.6).

4.3.2 Restore yield: choosing the slave stage architecture

For both NVFF flavors – 2R and 1R, the robustness of the restore operation is dependent on slave stage architecture and restore mechanism (i.e. the way slave is connected to the NV block). Furthermore, it is sensitive to the CMOS variability and resistance values. Hence, in order to select presented slave-NV configuration (Figure 4.4), three different latch architectures are investigated [79]:

- the **voltage-restore unbalanced** latch (Figure 4.7a), where the NVM devices are tied to the internal Qb/Q nodes of the standard cell;
- the **voltage-restore balanced** latch (Figure 4.7b), with inserted transistors for symmetrical branches;
- the **current-restore balanced** latch (Figure 4.7c), where the NVM devices are tied to the sources of the NMOS transistors of the latch.

The latches, tied to 2R NVFF structure (Figure 4.3) are implemented in 28nm FDSOI technology, and the bit error rates (BER) of restore operations (both cases – RESTORE 1 and RESTORE 0) are estimated for various R_{ON}, R_{OFF} values by performing 10000-sample Monte Carlo simulations, taking into account CMOS process variation. The sizing of the slave is taken from a low-power standard cell library with a drive of X8, while the size of restore access transistors is fixed as a trade off between area and restore yield.

Figure 4.8 shows BER versus R_{OFF}/R_{ON} ratio of voltage-restore unbalanced latch extracted for two R_{ON} values – 3kΩ (red) and 5kΩ (blue). The graph corresponds to RESTORE 0 operation as this represents the worst case scenario – due to different PMOS resistance of asymmetrical latch, charging node Q after the equalization is favored over charging Qb, leading to the higher RESTORE 1 yield. Results show that for a same ratio higher resistance values enable better yield. This is a consequence of lower impact of the parasitic NV access transistors resistances on the sensing.

Figure 4.9 presents the robustness comparison of the three design options. The graph shows the extracted (R_{ON}, R_{OFF}) pairs at 3σ yield for the schematics in Figures 4.7a

(circles), 4.7b (squares) and 4.7c (triangles). It demonstrates that: (i) balancing the structure significantly reduces the memory window required for a same yield, and (ii) current-restore outperforms the voltage-restore architecture. Therefore, the highest yield is obtained at the cost of modifying the slave stage. However, these modifications do not affect the performance and consumption significantly, which will be discussed more in Chapter 5.

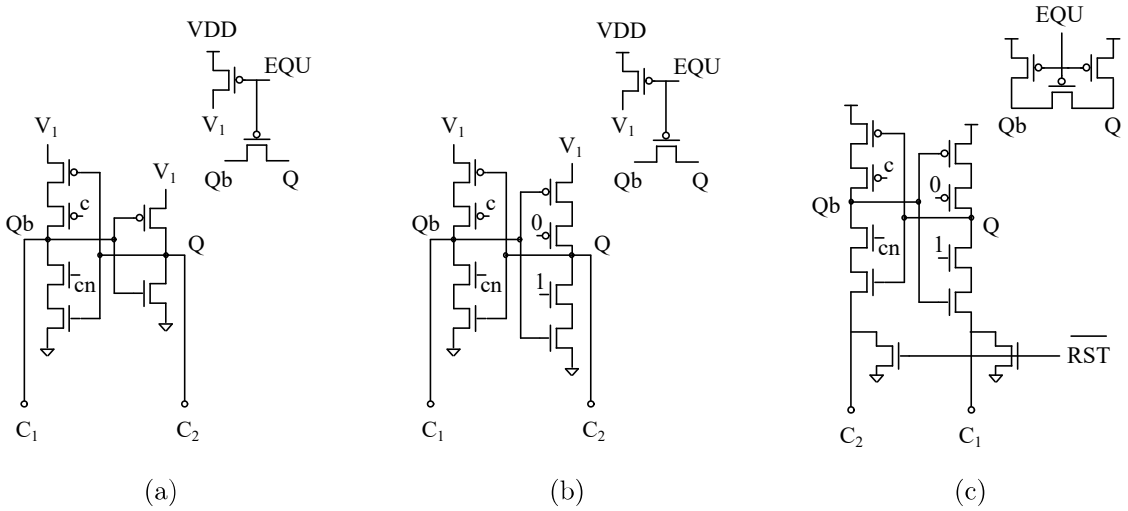


Figure 4.7: (a) Voltage-restore unbalanced latch. (b) Voltage-restore balanced latch. (c) Current-restore balanced latch.

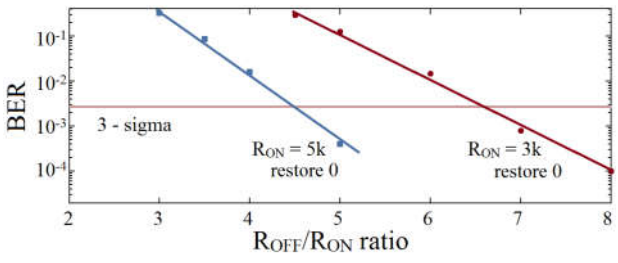


Figure 4.8: Restore BER versus R_{OFF}/R_{ON} ratio of the voltage-restore unbalanced latch for R_{ON}=3kΩ and R_{ON}=5kΩ.

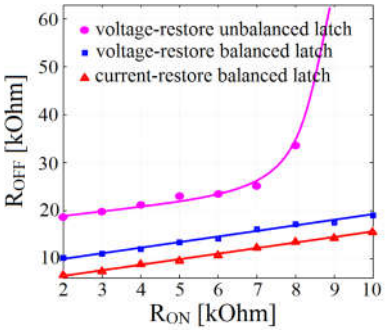
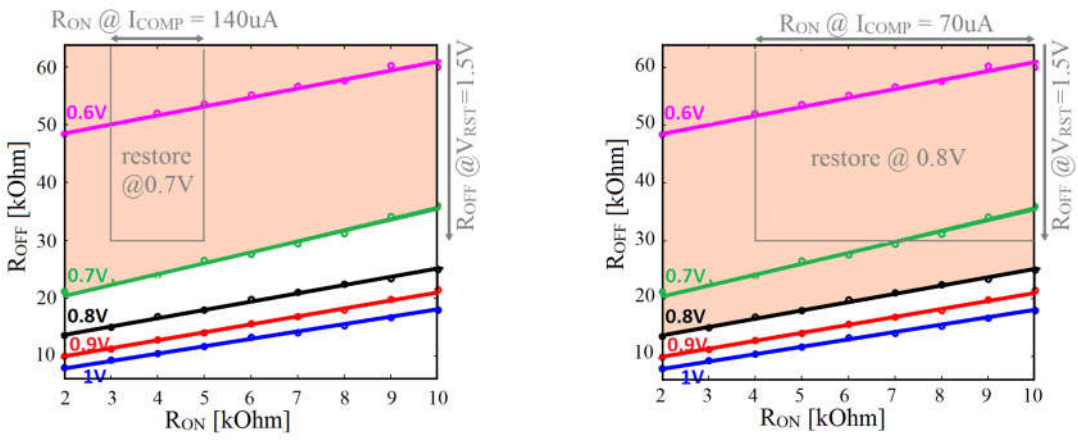


Figure 4.9: Restore memory window (R_{OFF}, R_{ON}) extracted at 3σ yield for the three latches.

4.3.3 Restore yield: OxRAM and CBRAM-based non-volatile flip-flops

Finally, 2R NVFF and 1R NVFF structures (Figures 4.3 and 4.6) tied to the optimal slave stage (Figure 4.4) are simulated, and BER is estimated for restore operations in the [0.6V, 1V] voltage range, considering the CMOS variation, in typical-typical corner. Then, the tails of resistance distribution of available ReRAM technologies are taken into account, and the programming conditions that ensure 3σ yield at the lowest restore voltage are estimated.

For 2R NVFFs, Figures 4.10a and 4.10b show (R_{OFF_MIN}, R_{ON_MAX}) couples which lead to 3σ yield of restore operation – for given R_{ON}, all R_{OFF} values above the curve result in higher yield. With regard to the memory window and programming conditions of OxRAM technology described in Chapter 2 (Figure 2.14), restore at 0.7V (Figure 4.10a) is successful for R_{ON} of 3 to 5kΩ and R_{OFF} higher than 30kΩ, compatible with 140μA of I_{COMP} and 1.5V of V_{RESET}, enabling 10^8 endurance cycles. On the other hand, restore at 0.8V (Figure 4.10b) and more is possible with R_{ON} in 4-10kΩ range for 70μA of I_{COMP}, thus enabling lower programming energy.



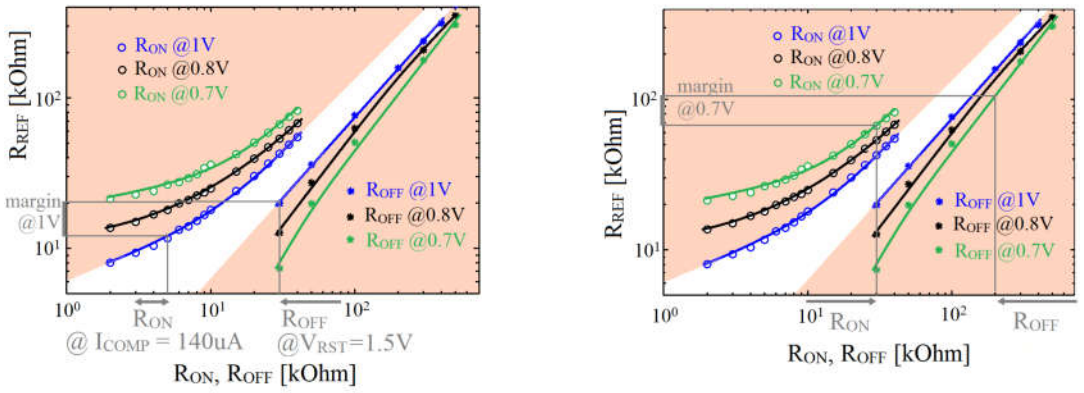
(a) (b)

Figure 4.10: Restore memory window (R_{OFF} vs R_{ON}) extracted at 3σ yield for 2R NVFFs at different supply voltages, and the programming conditions for OxRAM stack [41,49] that ensure successful restore (a) @0.7V, (b) @0.8V.

For 1R NVFFs, Figures 4.11a and 4.11b show the required R_{REF} values versus R_{ON}, R_{OFF} enabling 3σ restore yield, where R_{REF} is the equivalent resistance of transistor N_R

in Figure 4.6. The lower limit of R_{REF} (R_{REF_MIN}) is defined by RESTORE 0 operation (comparison of R_{ON} and R_{REF}), and the upper limit of R_{REF} (R_{REF_MAX}) is defined by RESTORE 1 (comparison of R_{OFF} and R_{REF}). As depicted in Figure 4.11a, OxRAM stack leaves a very small margin for designing R_{REF} at 1V, even with the more aggressive programming conditions (e.g., R_{ON} of 3kΩ to 5kΩ requires R_{REF} higher than 12kΩ, and R_{OFF} of 30kΩ requires R_{REF} less than 20kΩ), while restore @0.8V is not possible (R_{REF_MIN} higher than R_{REF_MAX}).

Instead, the sufficient memory window can be achieved by using CBRAM technologies, as illustrated in Figure 2.8. For example, using CBRAM stacks introduced in [38, 80] gives the R_{REF} margin of around 70kΩ to 100kΩ for the restore at 0.7V (Figure 4.11b). Figures 4.12a and 4.12b present the distribution of the currents for the CBRAM-based 1R NVFF cells which are optimized to restore at 0.7V and 1V, respectively. I_{OFF_MIN} is the current through CBRAM device which is in the lowest OFF state, while I_{ON_MAX} corresponds to the highest ON state. Unlike 2R NVFFs, 1R cells require dedicated optimization for different restore voltages.



(a) (b)

Figure 4.11: Reference resistance (R_{REF}) extracted at 3σ yield for 1R NVFFs at different supply voltages. R_{ON} and R_{OFF} define the lower and the upper limit of R_{REF}, respectively. (a) OxRAM stack leaves small R_{REF} margin for restore @1V. (b) CBRAM [38, 80] programming conditions which provide the sufficient margin for R_{REF} implementation for restore @0.7V.

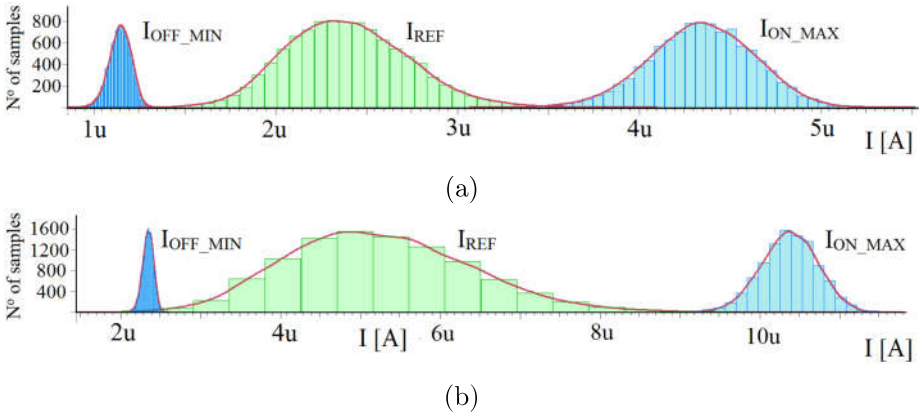


Figure 4.12: The distribution of referent current (I_{REF}) and the current through CBRAM for the border resistance values (I_{ON_MAX} and I_{OFF_MIN}) during the equalization for 1R NVFFs, for: (a) restore @0.7V, (b) restore @1V.

4.4 Store operation

The backup of flip-flop data in the NV block relies on the programming of ReRAM device(s) in accordance with the flip-flop value. Therefore, the circuit which performs SET, RESET, and FORMING of ReRAM depending on the control signals is required. The general concept of proposed programming solutions is illustrated in Figure 4.13. Transistors N_S and P_S are activated during the SET operation, while N_R and P_R are responsible for RESET of ReRAM. This allows for the separate optimization of SET and RESET paths, to satisfy the different programming conditions needed for these operations. Besides, both NMOS and PMOS are engaged in the programming circuit, thus avoiding the V_{TH} drop of the access transistors. Generally, the third operation, FORMING of the device, can be implemented in two ways:

- by including the separate FORMING path (N_F and P_F), controlled by dedicated *form* and $\overline{form}$ signals
- by using the existing SET path, where *set* and $\overline{set}$ signals are redefined to be engaged during both operations.

The optimal implementation of FORMING depends on the whole NVFF cell (e.g., number of devices, the details of programming circuit), and will be discussed in section 4.5.

ReRAM programming paths are supplied at VDDH, while the control signals are generated in the LOGIC block which operates at VDD (Figure 4.1). Considering this, the function of programming circuit is delivering the needed programming voltage and the

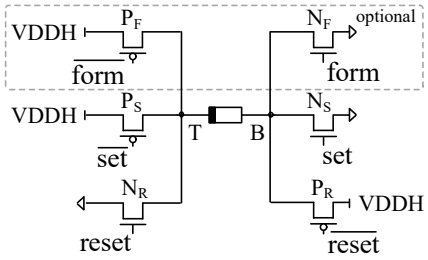


Figure 4.13: Programming of one ReRAM device – principle.

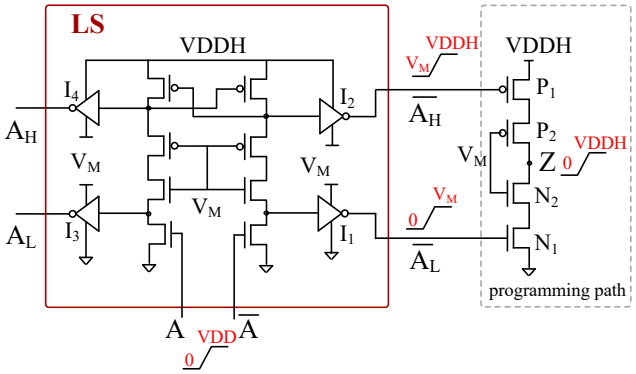


Figure 4.14: Reliable level-shifter and the programming part.

current to the device, while overcoming the voltage gap between VDD and VDDH. In the following, two such programming circuits employing the idea shown in Figure 4.13 are proposed. Additionally, the reliable operations must be ensured, so both solutions have all the transistors in the voltage-critical paths stacked and biased in a way that guarantees limited voltages across their terminals.

4.4.1 Level-shifter programming solution for one ReRAM device

First programming architecture [81] is built by inserting level-shifters between the LOGIC and NV blocks. Used level-shifter [82], depicted in Figure 4.14, consists of two branches of stacked transistors that are differentially switched. It uses high supply VDDH and middle supply V_M to provide the outputs at appropriate voltage levels: the input signals A and $\overline{A}$ in the [0, VDD] range are shifted to the "low range" [0, V_M] (outputs A_L and $\overline{A_L}$), or to the "high range" [V_M, VDDH] (outputs A_H and $\overline{A_H}$). These signals then drive the gates of output stage, placing the signal Z in full [0, VDDH] range.

This concept is applied to the store/forming parts of NV block. Transistors P_1 and N_1 in Figure 4.14 correspond to programming transistors P_S and N_R in Figure 4.13, respectively, while P_2 and N_2 are inserted for reliability. The node Z is connected to the top or bottom electrodes of ReRAM device. Gate signals $\overline{A_H}$ and $\overline{A_L}$ may come from different level-shifters, depending on the required logic. Therefore, only the necessary inverters for reshaping the signals are kept (e.g., I_1 and I_3, or I_2 and I_4). The intermediate voltage level V_M is chosen so that all pin-to-pin transistor voltages are within the safe operating

boundaries, while the level-shifter functionality is guaranteed. V_M is ideally equal to VDDH/2. However, if said conditions are fulfilled V_M=VDD can be used instead, in order to reduce number of supplies.

4.4.2 Current programming solution for one ReRAM device

The second proposed solution [83] is based on the current programming structure depicted in Figure 4.15a. SET of ReRAM is executed by applying signal *set* which activates the current source and turns on the transistors in series with ReRAM. At the beginning of the pulse, device resistance is high and the transistors are sized to provide high $V_R = V_{SET}$. After the resistance switches, the mirror is working and the source current is copied thus providing $I_R = I_{SET}$. Figure 4.16a illustrates the SET behavior. This structure performs the function of level-shifter between VDD and VDDH (*set* signal levels are 0/VDD), while additionally clamps the programming current. The RESET of ReRAM uses the similar circuit, with the inverted top/bottom electrodes. At the beginning of the pulse the mirror is activated and delivers $I_R = V_{RESET}/R_{ON}$ in order to provide $V_R = V_{RESET}$, and after the switching the mirror closes (Figure 4.16b).

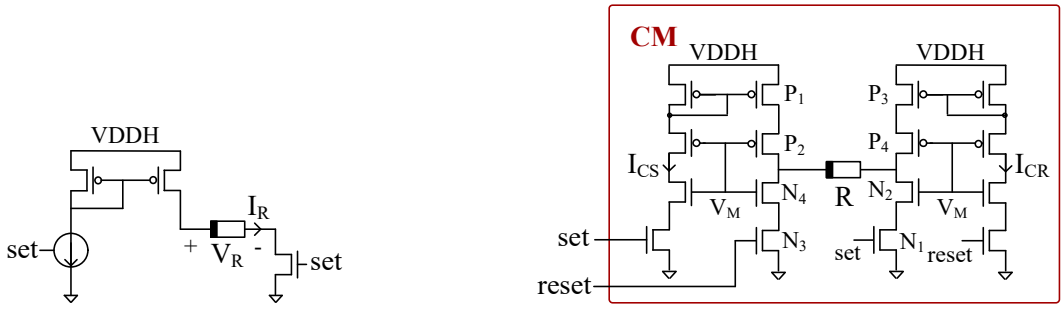


Figure 4.15: Current programming structure: (a) principle (SET operation), (b) full current programming circuit for one ReRAM device.

For complete store operation, device is connected to the pair of current programming structures, as shown in Figure 4.15b. I_{CS} branch and transistors P_1, P_2, N_1 and N_2 are dedicated to SET, while I_{CR} branch together with P_3, P_4, N_3 and N_4 is active during the RESET operation.

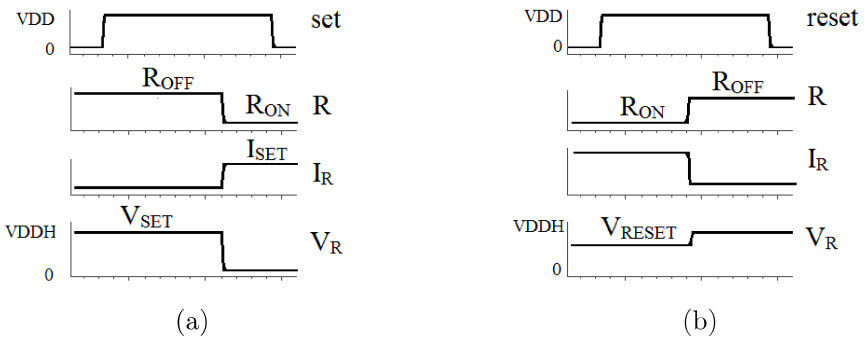


Figure 4.16: Timing diagram of: (a) SET operation, (b) RESET operation.

4.5 NVFF

Based on the presented configurations for store and restore operations, four NVFF architectures are developed: 2R-LS and 1R-LS are NVFF cells that employ level-shifter-based programming circuit, realized using two ReRAMs and one ReRAM device, respectively, while 2R-CM and 1R-CM use current programming circuit. All NVFFs have the dual-rail scheme with NV programming circuits decoupled from the flip-flop core, as explained in Figure 4.1, and the slave stage with the highest restore yield given in Figure 4.4. Consequently, this section describes only NV and LOGIC blocks for all cells.

4.5.1 2R-LS NVFF

Figure 4.17 shows the schematic of NV and LOGIC blocks of 2R-LS NVFF, which are designed to implement the behavior given in Table 4.1.

During the **store mode**, activated by STR signal, R_1 and R_2 are programmed to the opposite states depending on the flip-flop value. Each ReRAM has the dedicated programming circuit allowing for the simultaneous change of state of both devices. NV-control signals are generated as:

$$S_1 = STR \cdot Qb\_s \tag{4.1}$$

$$R_1 = STR \cdot Q\_s \tag{4.2}$$

where Q_s and Qb_s are used instead Q and Qb in order to minimize impact on the flip-flop performance (Figure 4.4). S_1, R_1 and their negations $\overline{S_1}$, $\overline{R_1}$ are used as inputs to two level shifters in order to place the gate signals of programming transistors in suitable

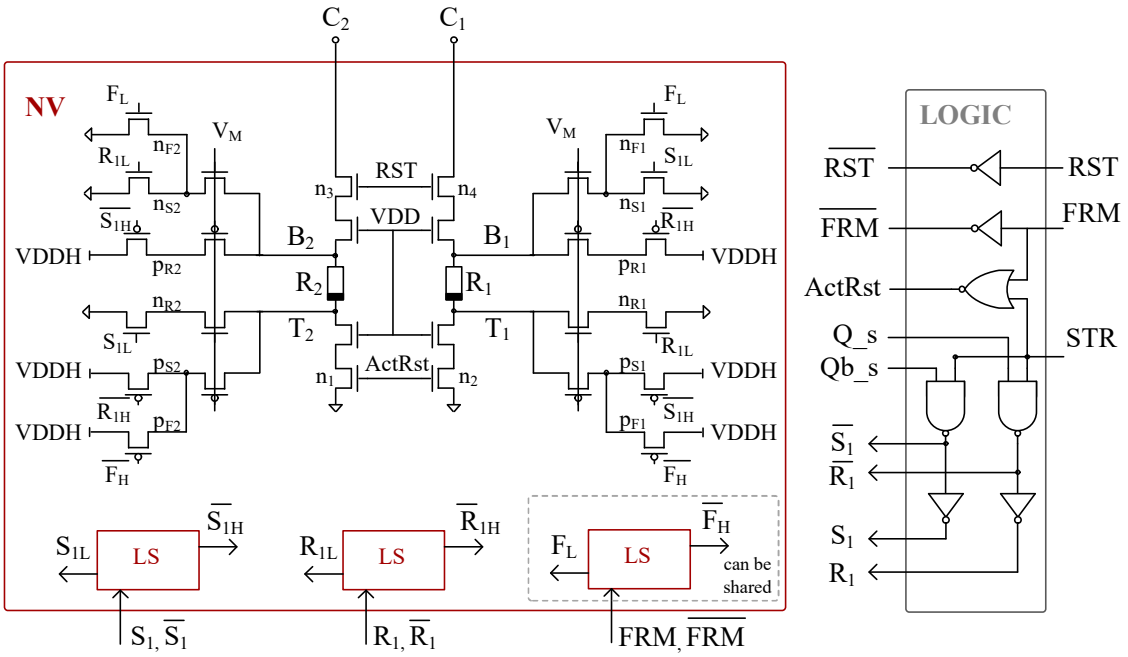


Figure 4.17: 2R-LS NVFF (NV and LOGIC blocks).

Table 4.1: NV signals and supplies for all operating modes of 2R-LS and 1R-LS NVFFs

mode (Q)	$\mathbf{R_1}$	$\mathbf{R_2}$	**NV transistors**	**VDD**	**VDDH**	$\mathbf{V_M}$
forming	SET	SET	n_{F1}, p_{F1}, n_{F2}, p_{F2}: ON other: OFF	$V_{FRM}/2$	V_{FRM}	$V_{FRM}/2$
store (0)	SET	RESET	n_{S1}, p_{S1}, n_{R2}, p_{R2}: ON other: OFF	V_{NOM}	V_{STR}	V_{NOM} ($V_{STR}/2$)
(1)	RESET	SET	n_{R1}, p_{R1}, n_{S2}, p_{S2}: ON other: OFF			
restore	sense	sense	n_1-n_4: ON prog. transistors: OFF	V_{NOM}	V_{STR}	V_{NOM} ($V_{STR}/2$)
				V_{RED}	$2V_{RED}$	V_{RED}
active	/	/	n_1, n_2: ON n_3, n_4: OFF prog. transistors: OFF	V_{NOM}	V_{STR} (0)	V_{NOM} ($V_{STR}/2$)
				V_{RED}	$2V_{RED}$ (0)	V_{RED}
sleep	/	/	all OFF	0	0	0

range – NMOS transistors n_{S1}, n_{R1}, n_{S2} and n_{R2} require S_1 and R_1 in "low" range ([0, V_M] or [0, VDD]), while PMOS transistors p_{S1}, p_{R1}, p_{S2} and p_{R2} require negated signals in "high" range ([V_M, VDDH]). To provide them, level-shifters use two inverters (I_2 and

I_3) instead of four shown in Figure 4.14.

For the **forming mode** which is activated by *FRM* signal, SET is performed on both devices at the same time independently of Q value. It is done in the separate paths consisting of n_{F1}, n_{F2}, p_{F1} and p_{F2}, while the stacked transistors are shared. F_L and $\overline{F_H}$ signals are generated in the additional level-shifter. Hovewer, in case of integrating a large number of NVFF cells in a complex system, multiple NVFFs can be formed at the same time and this level-shifter can be shared between cells. On the other hand, alternative option which merges SET and FORMING paths into one would require more level-shifters per cell for generating the proper control.

During the **restore mode**, resistance difference is sensed and amplified by the slave latch, so all the programming transistors are turned off, while restore access transistors n_1-n_4 are open. $\overline{RST}$ is generated in LOGIC block to be used in the slave stage.

Finally, **active mode** resembles the restore mode, regarding the programming paths that are disengaged. Nevertheless, in order to avoid floating of ReRAM electrodes, signal *ActRst* connects them to 0 by turning on n_1 and n_2 during the active mode.

The proposed configurations of the power supplies VDD, VDDH, and V_M are given in the Table 4.1, where V_{NOM} is the nominal CMOS operating voltage, V_{STR} and V_{FRM} are the values of VDDH during the store and forming, and V_{RED} refers to the reduced CMOS operating voltage value. For the proper functionality of the circuit, it is important that VDDH $> V_M + 2V_{TH}$ during the forming, store, and restore modes. For the long lifetime of the circuit, V_M and VDDH – V_M should be minimized and not exceed V_{NOM} for long periods, to reduce the transistor degradation. Therefore, the optimal solution is V_M = VDDH/2, meaning that the separate power supply for V_M is needed. However, given the V_{NOM} of 28nm FDSOI and the SET/RESET programming voltage range of OxRAM and CBRAM, the co-integration of these technologies allows for using $V_{STR} \sim 2V_{NOM}$. Therefore, using two power supplies is proposed:

$$VDDH = \begin{cases} V_{STR}, & \text{store and restore modes (e.g. 2V)} \\ V_{STR} \text{ or } 0, & \text{active mode} \\ V_{FRM}, & \text{forming mode (e.g. 3V)} \end{cases} \tag{4.3}$$

$$V_{\mathrm{M}} = VDD = \begin{cases} V_{\mathrm{NOM}} & \text{store, restore, active modes (e.g. 1V)} \\ V_{FRM}/2, & \text{forming mode (e.g. 1.5V)} \end{cases} \tag{4.4}$$

Note that NVFF in regular flip-flop mode operates correctly even with other supply combinations. However, VDDH = V_{STR} or VDDH = 0 are suggested in active mode as they ensure simplest power management or minimum consumption, respectively. In order to allow the system operate at reduced voltage, this can be extended to:

$$VDDH = \begin{cases} 2V_{\mathrm{RED}}, & \text{restore mode (e.g. 1.6V)} \\ 2V_{\mathrm{RED}} \text{ or } 0, & \text{active mode} \\ V_{\mathrm{STR}}, & \text{store mode (e.g. 2V)} \\ V_{FRM}, & \text{forming mode (e.g. 3V)} \end{cases} \tag{4.5}$$

$$V_{\mathrm{M}} = VDD = \begin{cases} V_{\mathrm{RED}}, & \text{restore and active mode (e.g. 0.8V)} \\ V_{\mathrm{NOM}} & \text{store mode (e.g. 1V)} \\ V_{FRM}/2, & \text{forming mode (e.g. 1.5V)} \end{cases} \tag{4.6}$$

4.5.2 1R-LS NVFF

1R-LS NVFF cell is given in Figure 4.18. Since it employs the same programming concept as 2R-LS it is based on Table 4.1, but only the circuitry related to device R_1 is implemented. Therefore, only one pair of SET and RESET programming paths is used. However, two level-shifters are still needed to accomplish required behavior during the **store mode**.

Concerning the **forming mode** of 1R cell, it only differs from STORE 0 operation in voltage levels. Hence, *FRM* signal is removed, and forming is performed in two steps: (i) 0 is written to the flip-flop, (ii) signal *STR* is activated. As a result, the forming path is removed from NV block, while LOGIC is simplified.

Restore mode relies on the 1R idea explained in Figure 4.6, while the **active mode** and the power supplies configuration are the same as in 2R-LS NVFF.

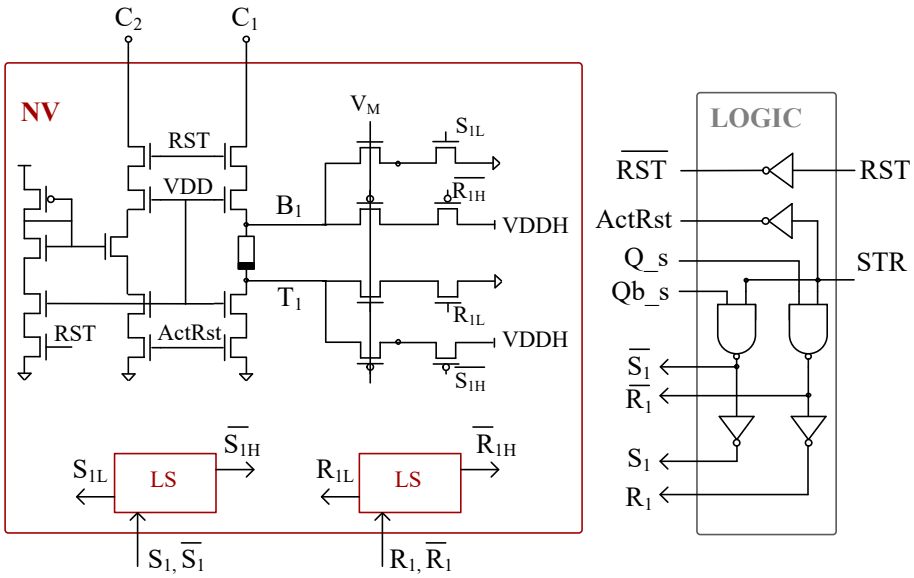


Figure 4.18: 1R-LS NVFF (NV and LOGIC blocks).

4.5.3 2R-CM NVFF

Figure 4.19 depicts the schematic of 2R-CM NVFF, designed to implement the behavior given in Table 4.2. In all operating modes, the required behavior for R_1 and R_2 is identical to the 2R-LS case, and the difference comes from the NV-control signals and the power supplies.

For the **store mode**, both ReRAM devices have one current mirror programming circuit (Figure 4.15b) attached to them and controlled with set_1, $reset_1$ and set_2, $reset_2$ signals, respectively. Unlike 2R-LS implementation, **forming** uses the same programming path as SET operation, which is accomplished by the minor extension of logic functions. Thus, the control signals are generated in LOGIC block at 0/VDD levels as:

$$set_1 = STR \cdot Qb\_s + FRM \tag{4.7}$$

$$set_2 = STR \cdot Q\_s + FRM \tag{4.8}$$

$$reset_1 = STR \cdot Q\_s \tag{4.9}$$

$$reset_2 = STR \cdot Qb\_s \tag{4.10}$$

While the **restore** is same as in 2R-LS, signal *ActRst* is not needed in 2R-CM, given that

the top electrodes are not floating in **active mode**.

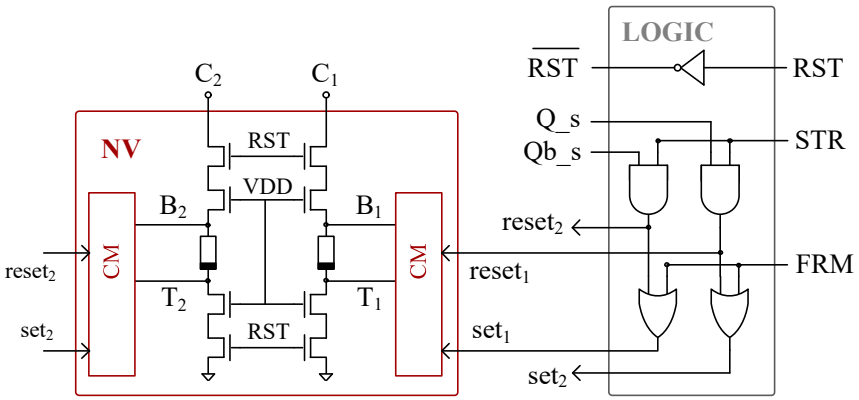


Figure 4.19: 2R-CM NVFF (NV and LOGIC blocks).

Table 4.2: NV control and supplies for all operating modes of 2R-CM and 1R-CM NVFFs

mode (Q)	$\mathbf{R_1}$	$\mathbf{R_2}$	**NV control**	**VDD**	**VDDH**	$\mathbf{V_M}$
forming	SET	SET	$set_1 = set_2 = 1$ $reset_1 = reset_2 = 0$	$V_{FRM}/2$	V_{FRM}	$V_{FRM}/2$
store (0)	SET	RESET	$set_1 = reset_2 = 1$ $reset_1 = set_2 = 0$	V_{NOM}	V_{STR}	V_{NOM} ($V_{STR}/2$)
(1)	RESET	SET	$reset_1 = set_2 = 1$ $set_1 = reset_2 = 0$			
restore	sense	sense	n_1-n_4: ON $set_1 = reset_1 = 0$	V_{NOM}	V_{NOM} (0)	V_{NOM}
			$set_2 = reset_2 = 0$	V_{RED}	V_{RED} (0)	V_{RED}
active	/	/	n_1 - n_4: OFF $set_1 = reset_1 = 0$	V_{NOM}	V_{NOM} (0)	V_{NOM}
			$set_2 = reset_2 = 0$	V_{RED}	V_{RED} (0)	V_{RED}
sleep	/	/	all OFF	0	0	0

The proposed configurations of the power supplies VDD, VDDH, and V_M are given in the Table 4.2, with the same nomenclature as in previous sections. For the proper functionality of the circuit, it is important that VDDH $> V_M + 2V_{TH}$ during the forming and store modes. On the other hand, in the restore mode transistors P_1-P_4 of the programming circuits (Figure 4.15b) must be turned off. Thus, V_M = VDDH/2 relation, which is preferable for LS NVFFs, is forbidden, since it causes PMOS leakage and impacts the

restore yield. As a results, the proposed combination of power supplies is:

$$VDDH = \begin{cases} V_{\text{NOM}} \text{ or } 0, & \text{restore and active modes (e.g. 1V, or 0)} \\ V_{\text{STR}}, & \text{store mode (e.g. 2V)} \\ V_{FRM}, & \text{forming mode (e.g. 3V)} \end{cases} \tag{4.11}$$

$$V_{\text{M}} = VDD = \begin{cases} V_{\text{NOM}} & \text{store, restore, active modes (e.g. 1V)} \\ V_{FRM}/2, & \text{forming mode (e.g. 1.5V)} \end{cases} \tag{4.12}$$

In order to allow the system operate at reduced voltage, this is extended to:

$$VDDH = \begin{cases} V_{\text{RED}} \text{ or } 0, & \text{restore and active mode (e.g. 0.8V, or 0)} \\ V_{\text{STR}}, & \text{store mode (e.g. 2V)} \\ V_{FRM}, & \text{forming mode (e.g. 3V)} \end{cases} \tag{4.13}$$

$$V_{\text{M}} = VDD = \begin{cases} V_{\text{RED}}, & \text{restore and active mode (e.g. 0.8V)} \\ V_{\text{NOM}} & \text{store mode (e.g. 1V)} \\ V_{FRM}/2, & \text{forming mode (e.g. 1.6V)} \end{cases} \tag{4.14}$$

4.5.4 1R-CM NVFF

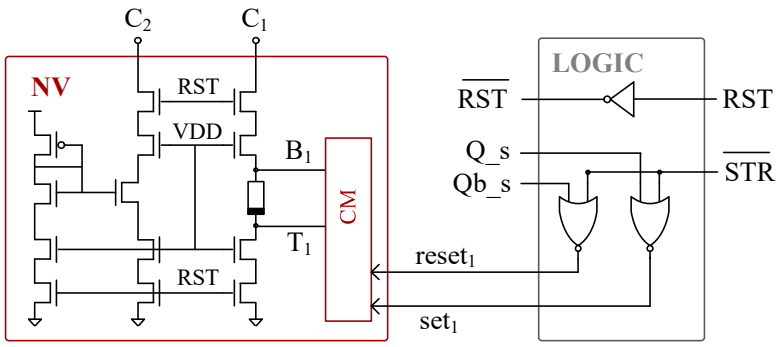


Figure 4.20: 1R-CM NVFF (NV and LOGIC blocks).

1R-CM NVFF cell is given in Figure 4.20. It is implemented according to R_1-related part of Table 4.2. It uses only one programming circuit, and do not require *FRM* signal, since

forming is performed as STORE 0 operation. Restore mode relies on 1R idea explained in Figure 4.6, while the active mode and the power supplies possibilities are the same as in 2R-CM NVFF.

4.6 Summary

Four non-volatile flip-flops presented in this chapter (2R-LS, 1R-LS, 2R-CM, 1R-CM), built by attaching the non-volatile part to the standard master-slave flip-flop, successfully co-integrate ReRAM and 28nm FDSOI technologies. Statistical analysis of the flip-flop core reveals that the current-restore balanced architecture is the most robust in terms of restore yield, and is thus used for the slave stage of NVFFs.

2R NVFF cells are robust and compatible with ReRAM technologies with small memory window. They have high restore yield even with less aggressive programming conditions, leading to high endurance. Particularly, simulation and silicon results suggest implementation of 2R NVFFs using available OxRAM devices, with restore at 0.8V or higher, for low programming energy.

On the other hand 1R NVFFs offer higher integration density, eliminate the *FRM* signal and forming programming part, and allow sharing of the restore current mirror between multiple cells. From the design point of view, 1R NVFFs are more energy efficient in both active and store mode, since they involve only one programming circuit and simpler control block. However, they require higher memory window which limits the choice of ReRAM technology and its programming conditions. Data indicate that using available CBRAM devices can result in 1R NVFFs which can operate at 0.7V in restore mode.

Both programming solutions use only thin-gate oxide transistors, which makes them compatible with digital design flow. Moreover, they are reliable and can be implemented with below-28nm CMOS nodes. NVFFs overcome the gap between SET and RESET programming conditions, and have simple forming operation. Compared to LS NVFFs, CM NVFFs do not require generation of intermediate signals on different voltage levels. Dynamic voltage scaling of VDDH and VDD between store, restore, and active modes is necessary for CM NVFF, while it can be avoided in LS NVFF.

Chapter 5

Evaluation of NVFF

Contents

5.1 Implemented non-volatile flip-flop cells

In order to compare two programming architectures (LS and CM) and two restore configurations (2R and 1R) explained in Chapter 4, three of presented NVFF cells are implemented and evaluated in 28nm FDSOI CMOS technology: 2R-LS, 2R-CM and 1R-CM. To obtain the sufficient restore yield, 1R-CM is implemented with CBRAM, while 2R NVFFs use OxRAM, as discussed in subsection 4.3.3. Programming circuits in the cells are optimized to ensure R_{ON} and R_{OFF} distributions that improve endurance and minimize programming power, listed in Table 5.1. Apart from the nominal voltage, 2R cells can also operate at reduced voltage of 0.8V, determined in subsection 4.3.3 as the minimum voltage for the restore operation (for the chosen ReRAM programming conditions).

On the other hand, although chosen memory window of CBRAM technology allows for reducing the supply, designed 1R cell contains the restore current source optimized for nominal voltage only.

Table 5.1: Parameters of implemented NVFFs

Architecture	**2R-LS, 2R-CM**	**1R-CM**
Technologies	OxRAM [41,49], 28nm FDSOI	CBRAM [38,80], 28nm FDSOI
R_{ON} ($\pm 3\sigma$)	7 kΩ (4-10 kΩ)	<30 kΩ
R_{OFF} ($\pm 3\sigma$)	200 kΩ (30-1000 kΩ)	>200 kΩ
V_{NOM}	1V	1V
V_{STR}	2V	2.2V
V_{RED}	0.8V	/

Besides the comparison of proposed NVFFs among each other, they are benchmarked on the schematic level to: the standard MSFF cell used for building NVFFs (x8 drive), and the dual-rail flip-flop with data-retention latch ("balloon"). Additionally, for MSFF and NVFFs more thorough post-layout simulations are performed. The "balloon" data-retention flip-flop (shown in Figure 5.1) is based on the multiple threshold-voltage solution presented in [8]. In FDSOI technology the native V_{TH} of the transistor is defined by the metal gate and the well doping type while multiple thresholds can be realized by poly-biasing [84]. Based on that, all transistors in this cell are regular-V_{TH} devices, but six retention latch transistors have channel lengths increased by +16nm to provide higher

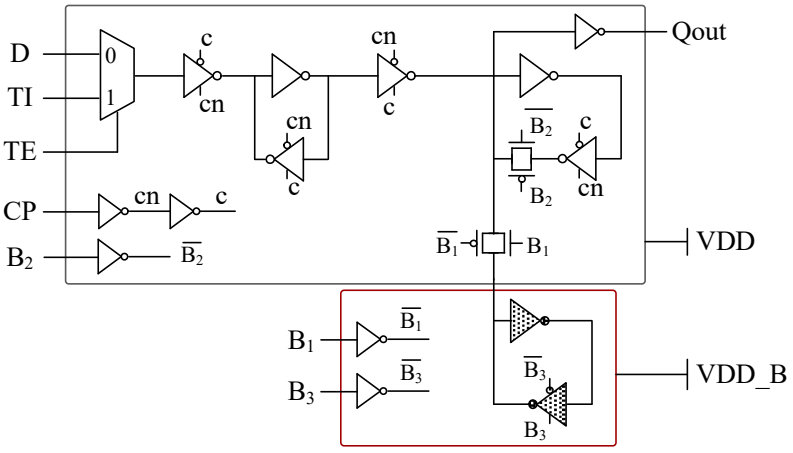


Figure 5.1: "Balloon" cell: dual-rail MSFF with the data-retention latch supplied at VDD_B. Shaded inverters are poly-biased.

V_{TH} for lower standby leakage current. Additionally, control signal B3 has been detached from signal B2 to close the balloon loop in active mode, thus avoiding floating state and reducing consumption.

In this chapter, first the impact of adding NV property to the flip-flop is investigated - the performance and power consumption of NVFFs and FFs are evaluated at nominal and reduced voltages. Second, the consumption of store and restore operations is analyzed and compared to the leakage of volatile FFs, to explore the benefit of replacing standard cells with NVFFs.

5.2 Impact on active mode

5.2.1 Performance

Figure 5.2 shows the propagation delay of NVFF and "balloon" schematics normalized to MSFF, at nominal voltage. The delay is measured from 40% of clock rising edge to 60% of rising and 40% of falling edge of the output signal, for low-to-high (LH) and high-to-low (HL) transitions, respectively. The data is set on the falling clock edge, to avoid setup and hold time violations. On the schematic level, non-volatile flip-flop delay depends only on the core while NV block is decoupled, hence all NVFF cells have the same characteristics. The delays are estimated for various output loads (C_L: from 1 to 12 x4 inverters) and various D/CLK input drivers (BF: x8 buffer, x4 buffer, and x4 buffer driving 4 FFs). Increasing the load and decreasing the drive results in higher propagation delay of both MSFF and NVFF, leading to reduced delay penalty. Compared to MSFF, NVFFs do not exhibit significant performance degradation (less than 3%), which makes them competitive with "balloon" cell.

Having such a small overhead of $t_{CLK\text{-}Q}$ is a consequence of using the MSFF core given in Figure 4.1, in which adding the load on the internal nodes Q and Qb only slightly affects the path to the output $Qout$ by changing the slope in Qb. On the other hand, additional load on the internal nodes in the MSFF_QN core from Figure 4.2 has much higher impact on $t_{CLK\text{-}Q}$ as it is added on the critical path to the output $QNout$. Results shown in Figure 5.3 confirm that using MSFF_QN instead of MSFF core increases delay penalty from 2-3% to 80-90%.

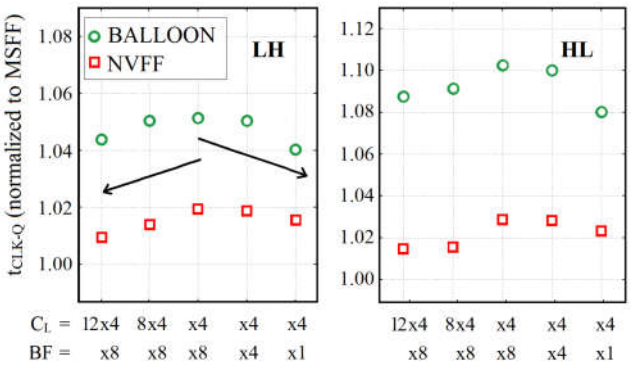


Figure 5.2: $t_{CLK\text{-}Q}$ of "balloon" and NVFF cells compared to MSFF for different output loads (C_L) and input drivers (BF).

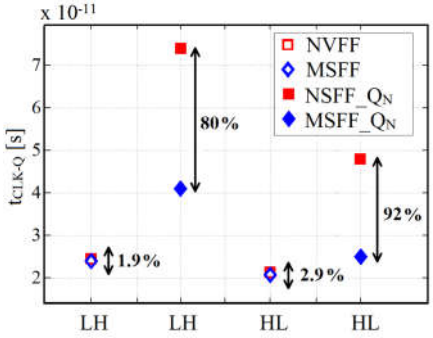


Figure 5.3: $t_{CLK\text{-}Q}$ of MSFF and NVFF, for different flip-flop cores (C_L=x4, BF=x8).

Figures 5.4 and 5.5 show more extensive post-layout performance evaluation of the cells - clock-to-q latency ($t_{CLK\text{-}Q}$) as a function of clock-to-data delay ($t_{CLK\text{-}D}$), at nominal and reduced voltage, respectively. Data/clock drivers and the output load corresponding to the worst case in Figure 5.2 are used (C_L=x4, BF=x8). Obtained results show small discrepancy among NVFFs due to the layout differences, but all cells have less than 8.5% increase of clock-to-q delay compared to MSFF at nominal voltage. Setup and hold times are not impacted since they depend on unmodified MASTER stage.

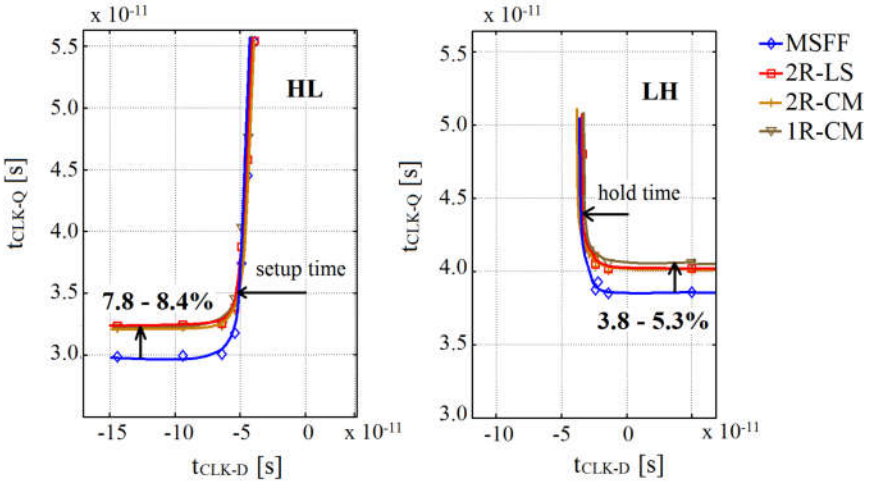


Figure 5.4: $t_{CLK\text{-}Q}$ as a function of $t_{CLK\text{-}D}$, for MSFF and NVFFs, @1V (post-layout simulation).

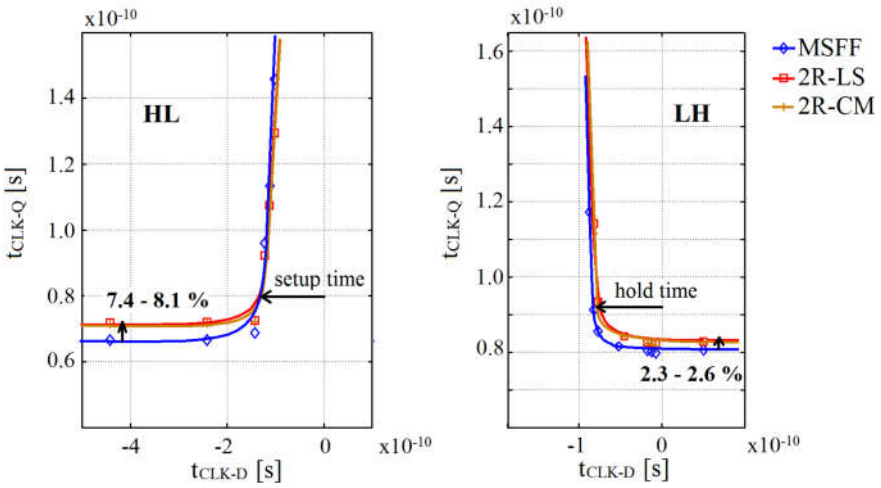


Figure 5.5: t_{CLK-Q} as a function of t_{CLK-D}, for MSFF and 2R NVFFs, @0.7V (post-layout simulation).

5.2.2 Consumption

Figure 5.6 shows the active energy of MSFF, 2R-LS, 2R-CM, and 1R-CM at nominal VDD and zero VDDH, for 10MHz-1GHz frequency range (post-layout simulation results). The consumption is evaluated for 6% data activity and 100% clock activity, using nominal R_{ON} and R_{OFF} values for 2R NVFFs, and nominal R_{ON} for 1R NVFF. The test structure uses x8 input buffers for data and clock, and x4 inverters as an output load. The active energy consists of dynamic part dominant at high frequencies, and static part dependent on the clock period and dominant at low frequencies. Dynamic energy overhead in

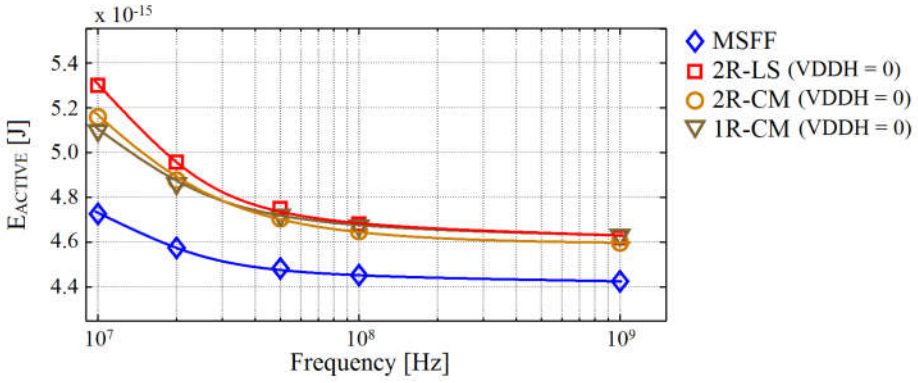


Figure 5.6: Active-mode energy of MSFF and NVFF cells for different frequencies and data activity of 6%, @1V (post-layout simulation).

NVFFs comes from the modifications in SLAVE stage and the LOGIC block, while static consumption overhead additionally includes the leakage in NV block. 2R-LS NVFF has higher consumption than 2R-CM due to stronger leakage in level-shifters. 1R-CM has the smallest consumption at low frequencies due to simpler LOGIC, but it is higher than 2R-CM at high frequencies due to different load of internal signals.

Figures 5.7 and 5.8 show the active energy of NVFFs normalized to MSFF, for the range of data activity levels (2%, 6%, and 10%) and frequencies (1GHz, 10MHz), at nominal and reduced voltages, respectively. Presented results investigate different possibilities of VDDH level during the active mode, as discussed in equations 4.3 – 4.6 and 4.11 – 4.14 of Chapter 4. The impact on the consumption is proportional to the data activity. At high frequencies, the cells show similar overhead (<8% at nominal, and <7% at reduced voltage, @1GHZ), while the difference is more notable at low frequency. In general, turning off VDDH increases energy efficiency by suppressing the leakage of the cells. Particularly, in 2R-LS NVFF using high VDDH can drastically increase consumption (up to 2.5x at nominal, and 1.7x at reduced voltage, @10MHz).

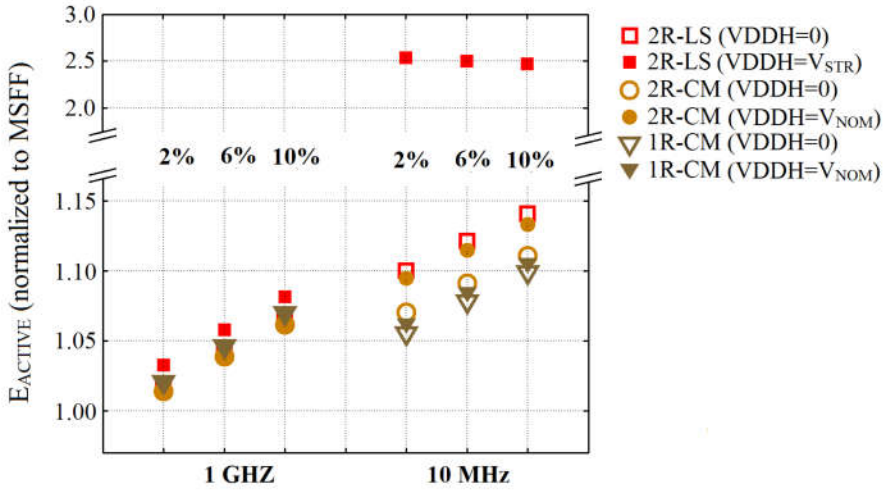


Figure 5.7: Active-mode energy of NVFFs @1V, for different frequencies and data activity levels (post-layout simulation).

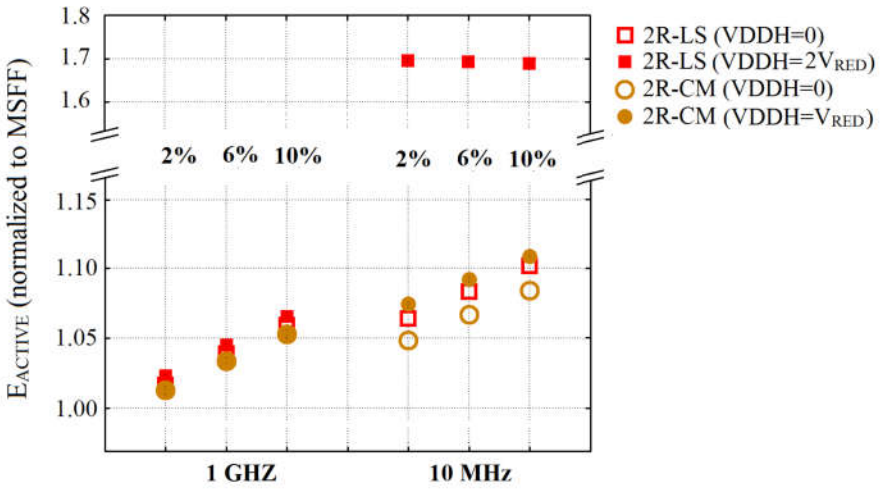


Figure 5.8: Active-mode energy of NVFFs @0.8V, for different frequencies and data activity levels (post-layout simulation).

5.3 Sleep energy

5.3.1 Store operation

For all NVFFs, the energy of the store operation is:

$$E_{STORE} = E_{STORE\_VDD} + E_{STORE\_VDDH} \tag{5.1}$$

Energy in VDD domain (E_{STORE_VDD}) is spent in MS core and LOGIC block during the store pulse. Assuming that clock and test inputs are low, it depends on the flip-flop value that is to be stored (Q) and the input D, and can be represented as:

$$E_{STORE\_VDD}(Q, D) = E_{STORE\_VDD\_dyn} + t_P \cdot P_{STORE\_VDD\_stat} \tag{5.2}$$

where t_P is the width of the store pulse.

Similarly, consumption in VDDH domain (E_{STORE_VDDH}) consists of static and dynamic part. However, considering the programming conditions of ReRAM technologies, dynamic energy needed for activating the programming circuits is negligible, therefore only static energy will be taken into account:

$$E_{STORE\_VDDH}(Q, NV_{old}) \sim E_{STORE\_VDDH\_stat} \tag{5.3}$$

E_{STORE_VDDH} is related to the flip-flop value previously stored in NV block (NV_{old}) and the current flip-flop value (Q), as summarized in Table 5.2. $E_{OFF\text{-}ON}$, $E_{ON\text{-}OFF}$, $E_{ON\text{-}ON}$ and $E_{OFF\text{-}OFF}$ correspond to energy spent in the programming circuit of one ReRAM, for SETing the device which is in OFF state, RESETing the device in ON state, SETing the device which is already ON and RESETing the device which is OFF, respectively. This representation corresponds to 1R-CM NVFF, and 2R-CM NVFF where full programming circuits (current mirrors) of two devices are separated. In 2R-LS NVFF, the programming transistors are separated, while the level-shifters are shared between devices. However, this approximation is justified since the leakage of level-shifters is negligible compared to the currents in programming transistors.

Table 5.2: Store energy in VDDH domain (E_{STORE_VDDH})

NV_{old} Q	2R-LS, 2R-CM	1R-CM
01	$E_{OFF\text{-}ON}$ + $E_{ON\text{-}OFF}$	$E_{ON\text{-}OFF}$
10	$E_{OFF\text{-}ON}$ + $E_{ON\text{-}OFF}$	$E_{OFF\text{-}ON}$
00	$E_{OFF\text{-}OFF}$ + $E_{ON\text{-}ON}$	$E_{ON\text{-}ON}$
11	$E_{OFF\text{-}OFF}$ + $E_{ON\text{-}ON}$	$E_{OFF\text{-}OFF}$

To evaluate VDDH energy, simplified ReRAM programming behavior with abrupt switching and stable resistance before and after switching is assumed and illustrated in Figure 5.9. t_S and t_R are SET and RESET switching times of the device, while t_P is the store pulse width covering device switching time variability. V_R is the voltage across ReRAM,

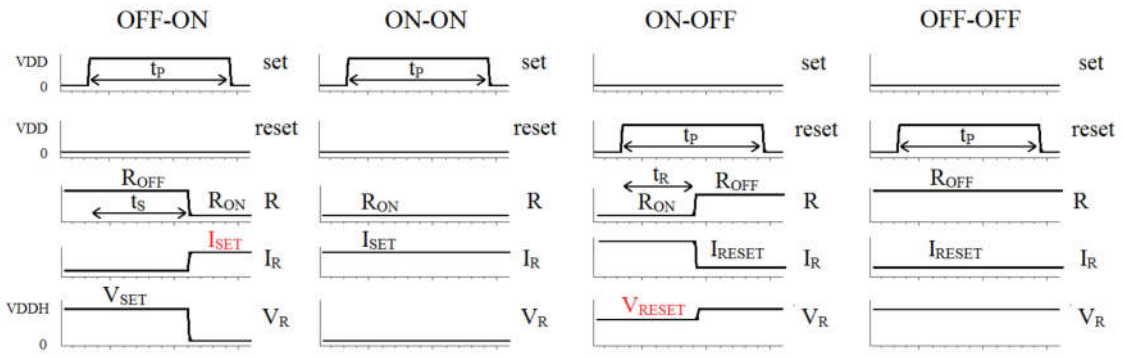


Figure 5.9: Timing diagrams of R, V, and I of one ReRAM during the store pulse, depending on the initial device state (R_{ON} or R_{OFF}) and required operation (SET or RESET).

while I_R is the device current. Based on this model, four presented scenarios are:

$$E_{OFF-ON} = P_{OFF_{SET}} \cdot t_S + P_{ON_{SET}} \cdot (t_P - t_S) \tag{5.4}$$

$$E_{ON-OFF} = P_{ON_{RESET}} \cdot t_R + P_{OFF_{RESET}} \cdot (t_P - t_R) \tag{5.5}$$

$$E_{ON-ON} = P_{ON_{SET}} \cdot t_P \tag{5.6}$$

$$E_{OFF-OFF} = P_{OFF_{RESET}} \cdot t_P \tag{5.7}$$

where $P_{OFF_{SET}}$ is the power of programming circuit when the device is in OFF state and SET path is activated, $P_{ON_{SET}}$ is the power when the device is in ON state and SET path is activated, etc. Therefore:

$$P_{OFF_{SET}} = \frac{V_{SET}}{R_{OFF}} V_{STR} \quad (+ I_{CS} V_{STR}) \tag{5.8}$$

$$P_{ON_{SET}} = I_{SET} V_{STR} \quad (+ I_{CS} V_{STR}) \tag{5.9}$$

$$P_{ON_{RESET}} = \frac{V_{RESET}}{R_{ON}} V_{STR} \quad (+ I_{CR} V_{STR}) \tag{5.10}$$

$$P_{OFF_{RESET}} = I_{RESET} V_{STR} \quad (+ I_{CR} V_{STR}) \tag{5.11}$$

where V_{STR} is the level of VDDH during the store mode, and I_{SET}, I_{RESET}, V_{SET} and V_{RESET} are shown in Figure 5.9. The components in the brackets are used in CM NVFFs and I_{CS}/I_{CR} correspond to SET/RESET current sources.

Table 5.3 summarizes the values of the currents and voltages used in equations for implemented cells (post-layout results), for nominal and extreme resistance values. Since I_{SET} and V_{RESET} are crucial for reaching targeted R_{ON} and R_{OFF}, the programming circuits are optimized to achieve them, while the other two parameters (I_{RESET} and V_{SET}) are merely a result of that optimization.

Figure 5.10 shows store VDD consumption of proposed NVFFs, E_{STORE_VDD}, for all Q/D combinations and the same *STORE* drive (post-layout results). Leakage (left) of 2R-LS is similar to 2R-CM, while 1R-CM requires less static power due to simpler logic block. Dynamic energy (middle) is higher for CM design than LS, and lower for 1R than 2R. However, total consumption (right) of 1R-CM is higher than the total consumption of 2R-LS due to longer store pulse.

Table 5.3: ReRAM programming conditions (post-layout results)

NVFF	**2R-LS**			**2R-CM**			**1R-CM**		
	R_{ONmin}	R_{ONnom}	R_{ONmax}	R_{ONmin}	R_{ONnom}	R_{ONmax}	R_{ONmin}	R_{ONnom}	R_{ONmax}
I_{SET} [μA]	86.5	82.5	79.3	72.8	71.2	69.5	53.0	52.0	47.0
V_{RESET} [V]	1.3	1.6	1.7	1.3	1.6	1.7	1.3	1.6	1.9
	R_{OFFmin}	R_{OFFnom}	R_{OFFmax}	R_{OFFmin}	R_{OFFnom}	R_{OFFmax}	R_{OFFmin}	R_{OFFnom}	R_{OFFmax}
I_{RESET} [μA]	63.0	9.9	2.0	63.0	9.9	2.0	8.7	5.5	2.2
V_{SET} [V]	1.5	1.9	2.0	1.5	1.9	2.0	2.1	2.1	2.2
I_{CS} [μA]	0			11.9			15.6		
I_{CR} [μA]	0			11.9			20.6		
t_S, t_R, t_P [ns]	50, 50, 100			50, 50, 100			250, 250, 500		

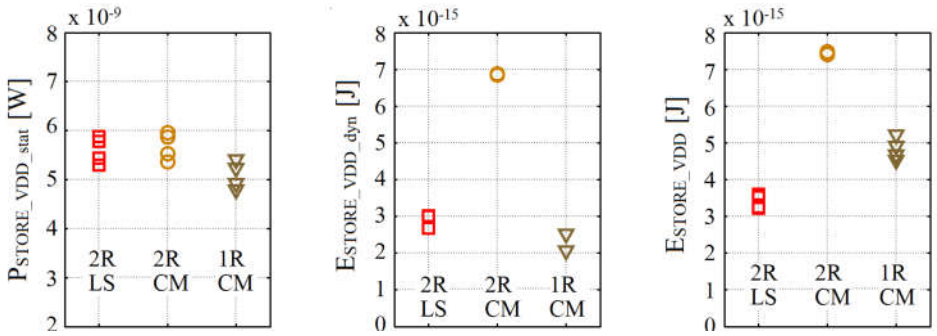


Figure 5.10: Store energy in VDD domain (E_{STORE_VDD}) of NVFFs, for all Q/D combinations: static power, dynamic energy and total consumption (post-layout simulation).

Power consumptions in programming circuit of one ReRAM device, described by equations 5.8 – 5.11, are given in Figure 5.11 for three implemented NVFFs. The results are shown for nominal and extreme resistance values, using the extracted programming conditions listed in Table 5.3. Similarity between 2R-CM and 2R-CM results indicates that CM and LS architectures have comparable consumptions since the cells are optimized for the same programming conditions, and the additional current sources in CM NVFF do not contribute significantly. On the other hand, the difference between 1R-CM and 2R-CM power consumptions reflects the difference in used ReRAM technologies. Generally, 1R-CM results show lower consumption due to higher resistance values (and I_{SET}). The exceptions are $P_{OFF_{SET}}$ and $P_{OFF_{RESET}}$ for high R_{OFF}, as 1R-CM cell is designed with stronger current sources than 2R-CM. Data show that the most power hungry operation is RESET of low-resistance state device, followed by SET of low-resistance state device. Operations performed on the device in OFF state are significant only for low R_{OFF}.

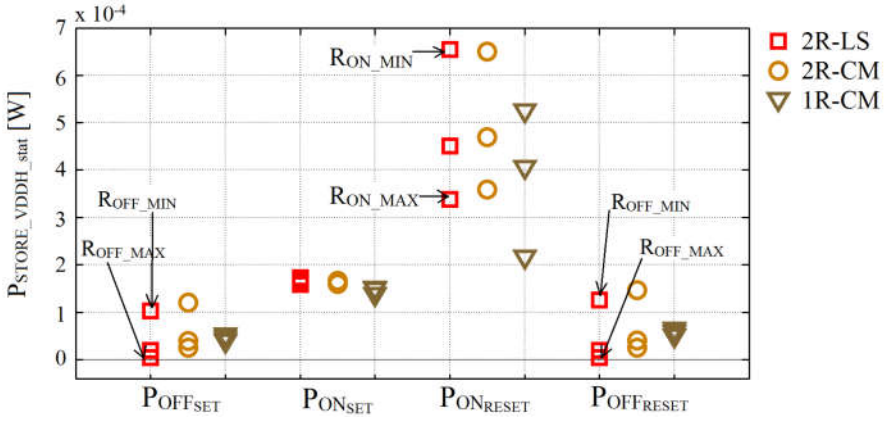


Figure 5.11: Static power consumptions in VDDH domain per ReRAM programming circuit in NVFFs, corresponding to equations 5.8 – 5.11, for the whole resistance range: $P_{OFF_{SET}}$, $P_{ON_{SET}}$, $P_{ON_{RESET}}$, $P_{OFF_{RESET}}$ (post-layout simulation).

Based on previous results, Figure 5.12 shows the estimated store VDDH energy of implemented NVFFs (E_{STORE_VDDH}), obtained using Table 5.2 and equations 5.4 – 5.7, for four NV_{old}/Q possibilities. Results are given for nominal resistance values. 2R NVFFs consume between 20 and 35pJ, and 1R-CM between 30 and 120pJ depending on the Q activity. Even thought 1R-CM NVFF on average has lower power consumption per programming circuit (Figure 5.11), and uses less programming circuits than 2R NVFFs, its energy consumption exceeds 2R cells. This is a consequence of assumed switching times and store pulse widths, which are five times longer for CBRAM than OxRAM technology (see Table 5.3).

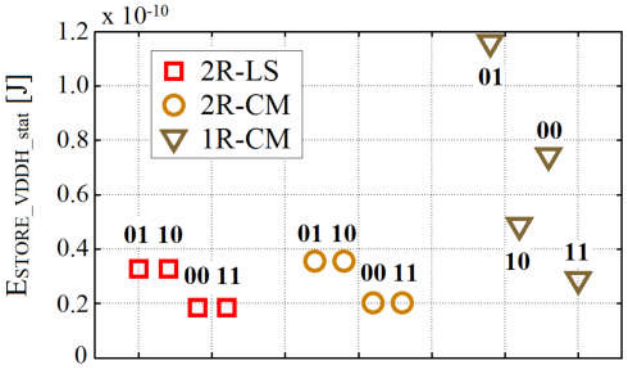


Figure 5.12: Store energy in VDDH domain (E_{STORE_VDDH}) of 2R-LS, 2R-CM, and 1R-CM NVFFs, for all NV_{old}/Q scenarios, corresponding to Table 5.2 (post-layout simulation).

5.3.2 Restore operation

Figure 5.13 shows the simulated restore energy of implemented NVFFs ($E_{RESTORE}$) at nominal voltage, for the range of R_{ON} and R_{OFF} resistances. The consumption depends on stored NV value (RESTORE 1 or RESTORE 0), and flip-flop value (Q). The results are given for the *RESTORE* and *EQU* pulse widths of 1.5ns and 0.5ns, respectively. 1R-CM NVFF consumes the most due to active restore current source, while 2R-CM and 2R-LS NVFFs have similar consumption.

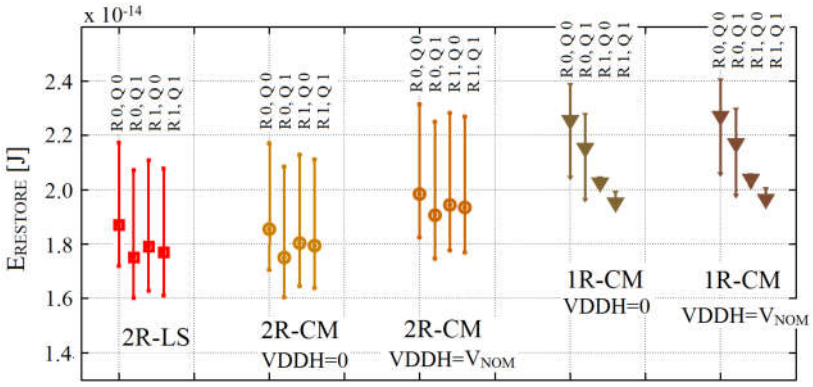


Figure 5.13: $E_{RESTORE}$ of different NVFFs @1V, for all NV/Q scenarios and the whole resistance range (post-layout simulation).

5.3.3 Break-even time

In order to benchmark the sleep energy of NVFFs to MSFF and "balloon" flip-flop, basic scenario in which the flip-flop data is stored each time before going to sleep mode is assumed (Figure 5.14).

Therefore, in one sleep cycle between consecutive active modes, NVFFs spend:

$$E_{SLEEP_{NVFF}}(Q, NV_{old}) = E_{STORE} + E_{RESTORE} \sim E_{STORE\_VDDH\_stat}(Q, NV_{old}) \quad (5.12)$$

This simplification is justified as results in Figures 5.10, 5.12 and 5.13 demonstrate that store consumption in VDD domain (E_{STORE_VDD}) and restore consumption ($E_{RESTORE}$) are negligible compared to the consumption in VDDH domain (E_{STORE_VDDH}), using «1% of the total energy.

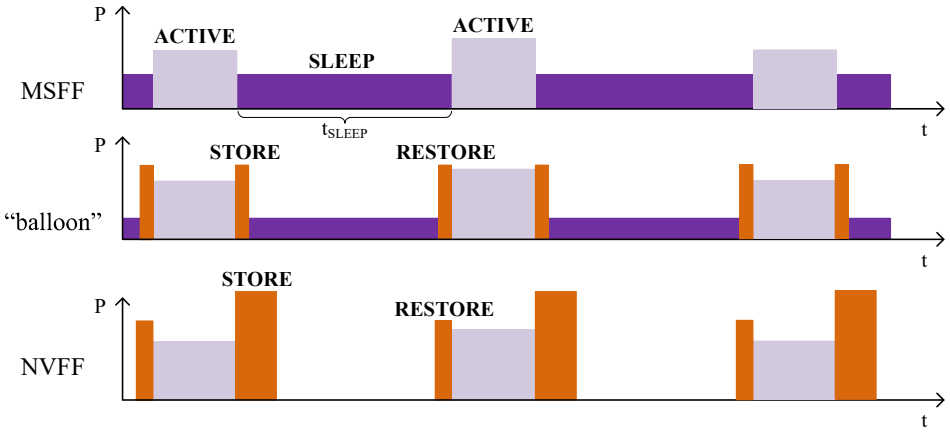


Figure 5.14: Illustration of the power consumption of different cells.

On the other hand, sleep energy of MSFF depends on the time spent in this mode, t_{SLEEP}, and its power supply that is reduced to V_{RET}:

$$E_{SLEEP_{MSFF}}(t_{SLEEP}, V_{RET}, Q) = V_{RET} \cdot I_{leakage}(Q) \cdot t_{SLEEP} \tag{5.13}$$

Finally, the sleep energy of "balloon" cell includes both store/restore transitions and the leakage of the retention part whose supply VDD_B is reduced to V_{RET}:

$$\begin{aligned} E_{SLEEP_{BALLOON}}(t_{SLEEP}, V_{RET}, Q, NV_{old}) = V_{RET} \cdot I_{leakage}(Q) \cdot t_{SLEEP} \\ + E_{STORE}(Q, NV_{old}) + E_{RESTORE}(Q) \end{aligned} \tag{5.14}$$

Break-even time (BET) of NVFF vs. MSFF/"balloon" is defined as t_{SLEEP} for which the MSFF/"balloon" sleep energy is equal to NVFF sleep energy. Thus, replacing MSFF/ "balloon" with NVFF brings energy savings if periods of inactivity are longer than BET. Figures 5.15 and 5.16 show the BET of three NVFFs vs. MSFF and "balloon" cells, respectively, as a function of reduced voltage V_{RET}. The graphs depict the range of curves for four (NV, Q) possibilities, while the bold curve represents the average. For example, if the system uses the "balloon" cells supplied at 0.5V for the data-retention, replacing them with 2R-LS/2R-CM or 1R-CM is beneficial if the periods of inactivity are longer than 0.2 - 0.8 s, or 0.7-3 s.

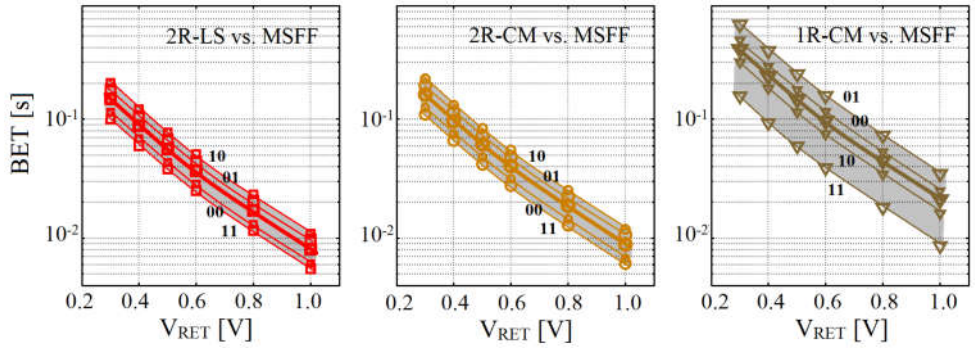


Figure 5.15: BET of NVFFs vs. MSFF as a function of MSFF retention voltage (V_{RET}), for all NV_{old}/Q possibilities (post-layout simulation).

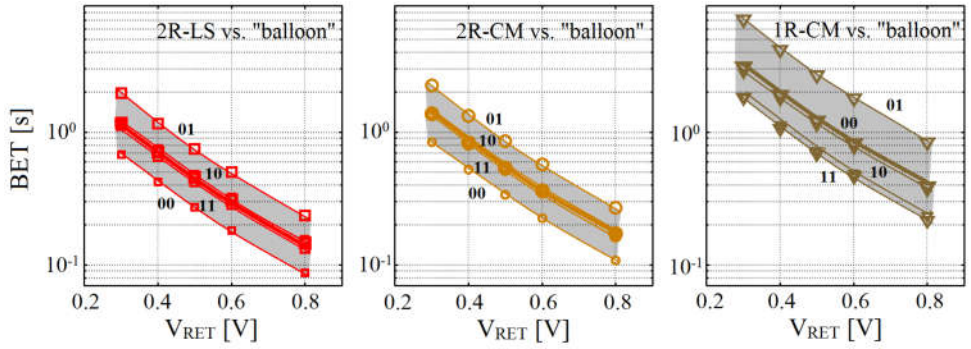


Figure 5.16: BET of NVFFs vs. "balloon" as a function of "balloon" retention voltage (V_{RET}), for all NV_{old}/Q possibilities.

5.4 Physical implementation

Figures 5.17, 5.18, and 5.19 show the layouts of 2R-LS, 2R-CM, and 1R-CM cells. NVFF layout is fully compatible with the digital design flow, using two standard cell rows, with the metal-2 VDDH stripe in the second row. ReRAM cells are added at the back-end-of-line, between metal-8 (B1) and metal-9 (IA) layers.

Compared to the corresponding MSFF from the standard cell library ($1.2\mu m$ x $3.128\mu m$), NVFFs are 5.6x (2R-LS), 4.4x (2R-CM) and 3x (1R-CM) larger. Area overhead of 2R-LS can be reduced by sharing the FORMING level-shifter between several NVFFs, while in 1R-CM case it can be achieved by sharing restore current mirror.

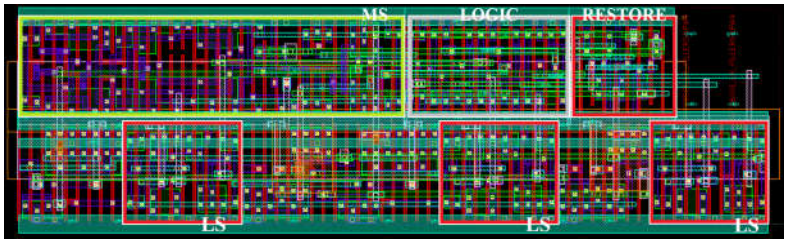


Figure 5.17: 2R-LS NVFF layout.

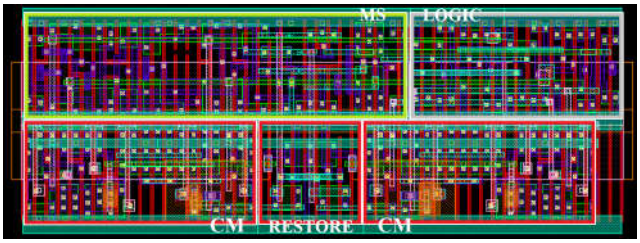


Figure 5.18: 2R-CM NVFF layout.

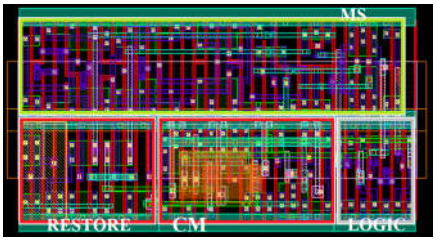


Figure 5.19: 1R-CM NVFF layout.

5.5 Summary

Compared to standard MSFF cell, NVFFs exhibit small increase of propagation delay, lower than the "balloon" data-retention flip-flop. At nominal voltage, this increase is around 5% or 8% for low-to-high or high-to-low $t_{CLK\text{-}Q}$, respectively. Furthermore, results show that the consumption in active mode is the smallest for 1R-CM cell and the highest for 2R-LS NVFF. In the case of LS NVFFs, the same VDDH in all modes is allowed, however not using dynamic voltage scaling to reduce VDDH can significantly increase the leakage of programming part of the cell. Active-mode consumption penalty is proportional to the clock period and the data activity, ranging from 2-4% at 1GHz with 2% activity to 10-14% at 10MHz with 10% activity, at nominal voltage. Both, speed and consumption overheads are lower if cells operate at reduced voltage in active mode.

During the store operation 1R-CM cell has the lower power consumption than other NVFFs, due to higher resistance values and only one programming circuit. However, its required store energy of 30-120pJ is the biggest, which is a consequence of longer switching times of CBRAM devices. 2R-CM and 2R-LS NVFFs have the similar store energy consumption of 20-35pJ. In comparison with MSFF, estimated break-even time of NVFFs is in range of 5-200ms for OxRAM-based 2R cells, or 10-700ms for CBRAM-based 1R cell (the range covers 0.3-1V retention voltage of flip-flop, and all possible values of stored data in consecutive cycles). It increases to 0.1-2s (2R) or 0.2-6s (1R), when NVFFs

are compared to "balloon" flip-flop. Note that given results correspond to the simplest scenario in which the back-up is performed before each sleep period, and that different use cases may bring higher energy savings. For simplicity, presented estimation of standalone cells ignores the transitions of power supplies between the operating modes. This aspect must be taken into account into the savings analysis on the system level.

The biggest penalty of non-volatile over volatile version of flip-flop is in the area of the cell, as the NVFFs are 3x (1R-CM NVFF) to 5.6x (2R-LS NVFF) bigger than MSFF.

Chapter 6

Non-volatile register file

Contents

6.1 Introduction

Apart from the flip-flops which are scattered through the digital logic, microprocessor cores also contain certain number of addressable memory elements (e.g., general-purpose registers) that can be organized in a regular array – register file. Typically, the register files must be capable of simultaneously reading and writing multiple data in each clock cycle, thus they are realized as multi-port architectures [85]. To achieve high-performance, simple and robust solution with minimal design time, processors with a small number of registers and access ports often employ flip-flop-based register files generated using RTL

code. For higher density and lower power consumption, large register files in complex processor cores are generally designed as full-custom SRAM-like arrays.

In order to satisfy all the challenges of co-integration of ReRAM and CMOS technologies, non-volatile flip-flop design solutions presented in Chapter 4 imply large area overhead. However, they can be used to design denser multi-port non-volatile register file (NVRF) by placing the common parts of NVFF outside of the cell and sharing it between multiple cells. Common peripheral circuitry can be then provided for both volatile (multiple read, multiple write), and non-volatile operations (store, restore).

Novel NVRF architecture based on this principle is demonstrated in this chapter. Two-read, one-write (2R1W) access is realized, as this number of ports is compatible with the instruction set of ARM Cortex-M0+ processor. It contains 16 32-bit words, corresponding to 13 general-purpose registers, stack pointer, link and auxiliary registers of M0+. NVRF is designed and laid-out in 130nm CMOS and CBRAM technologies. While increasing the cell density, the register file meets the constrains given for NVFFs in section 4.1: using nominal voltage for the volatile part and higher voltage for ReRAM, optimizing the design to achieve programming currents for lower consumption and increased endurance, overcoming the discrepancy between SET/RESET programming conditions, and ensuring the sufficient restore yield. As the nominal supply of 130nm CMOS node is closer to ReRAM programming voltage, transistor stacking is not implemented, although the solution uses only thin-gate oxide transistors.

6.2 Register file architecture

6.2.1 Top level

Proposed NVRF architecture relies on 1R-CM NVFF described in Chapter 4. Current programming solution (CM) is chosen over level-shifter programming (LS) as it requires less area, while their power and speed characteristics are comparable. Besides, it can be simply integrated into an array by sharing the current sources between several cells. Regarding the restore operation, implementing NVRF with high-memory window CBRAM technology allows for 1R configuration, which is smaller and consumes less than 2R (for the same ReRAM type). Moreover, it simplifies the design as it does not require dedicated forming signal.

Non-volatile register file, depicted in Figure 6.1, consists of the following components:

- one M cell per column, which is identical to the master stage of MSFF,
- array of S-NV bitcells, where S is the adapted slave stage of MSFF and NV stores data from S in a non-volatile way,
- one ST and RST block per column, as the common parts of store and restore circuitry,
- READ blocks, used during read and store operations, and
- DECODERS, to provide required word-line signals.

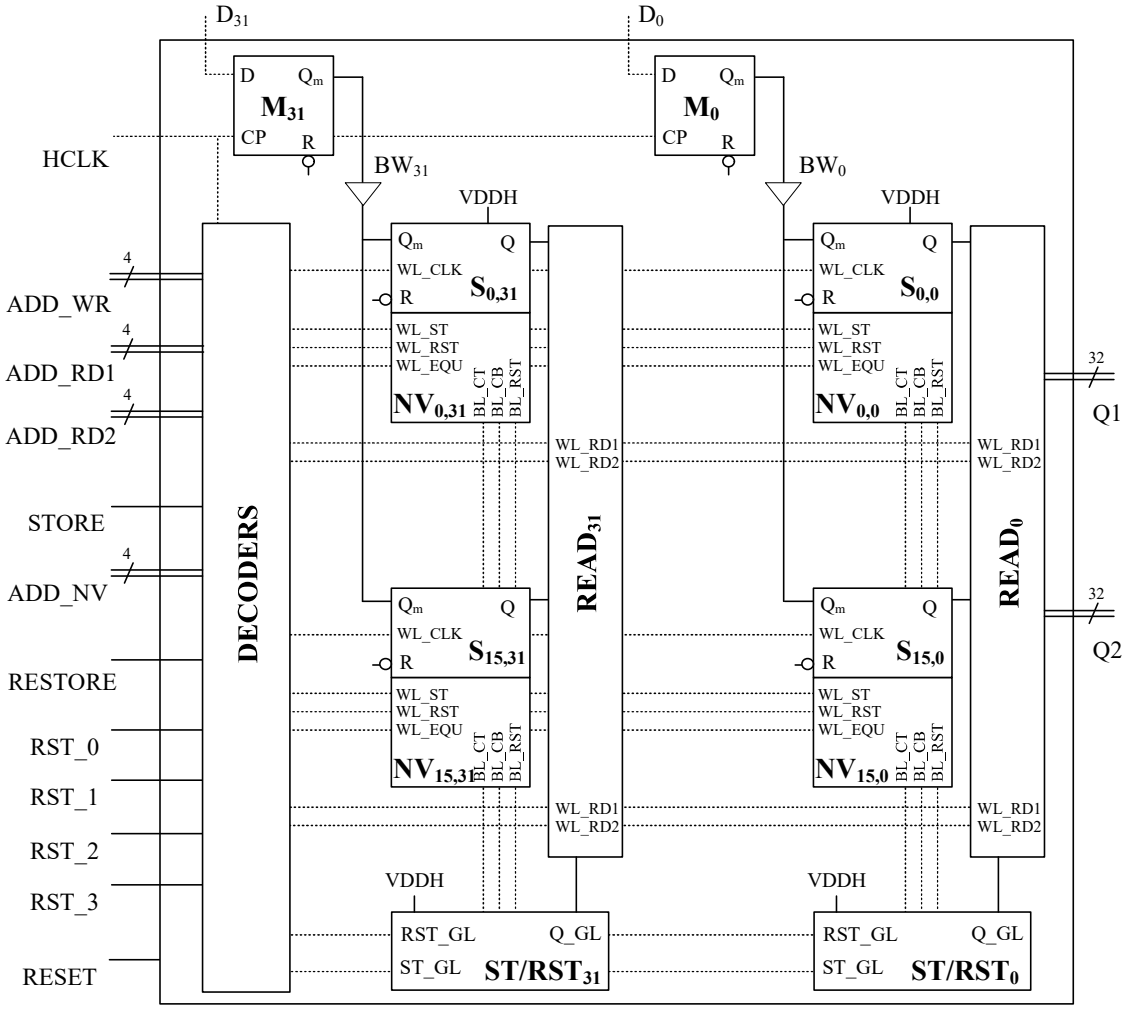


Figure 6.1: Proposed NVRF architecture.

Like NVFF, the solution uses two power supplies – CMOS operating voltage VDD, and VDDH for programming of ReRAM devices. VDDH rail is distributed to the NV part of

bitcell and ST block, while VDD supplies all other blocks. The digital signals are:

- standard write inputs: 32b input data (D), 4b write address (ADD_WR), clock ($HCLK$) and asynchronous reset (R),
- standard read inputs and outputs: two 4b read addresses (ADD_RD1, ADD_RD2) and two 32b output data ($Q1$ and $Q2$),
- NV-related inputs: global $STORE$ and $RESTORE$ control signals, 4b store address (ADD_NV) and four restore signals (RST_1-RST_4).

6.2.2 M and S-NV cells (write operation)

Standard master-slave flip-flop cell used as a base for the register file design is shown in Figure 6.2. The M cell of NVRF is the same as the master stage of MSFF, except the clock inverters which are weaker in the M cell, as their load is now reduced. S-NV bitcell, given in Figure 6.3, employs the slave stage of MSFF. As explained in subsection 4.3.2, slave is connected to the NV restore part through sources of the latch NMOS transistors, for the current-restore which improves the restore yield. Small I_{CP1} and I_{CP2} inverters are added to provide required clock signals. NV part of the bitcell will be explained in the following sections, as it is related to the store and restore operations.

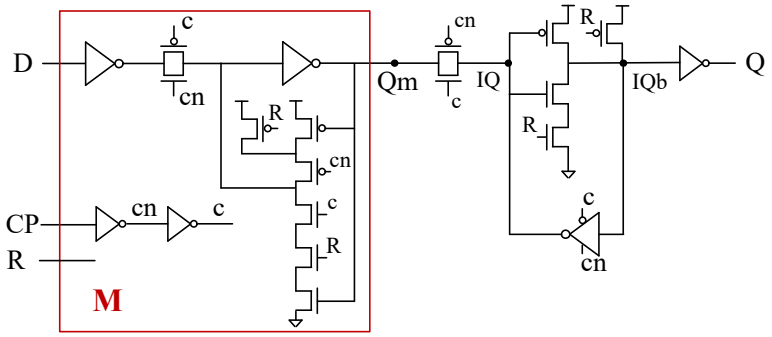


Figure 6.2: Volatile MSFF used for building NVRF.

In order to capture the data D to the register at address ADD_WR on the rising edge of the clock $HCLK$, the address is decoded and corresponding word-line signal WL_CLK which acts as a slave clock is generated. However, deriving the slave clock by simply gating the master clock with the decoded address may cause the write error, as demonstrated in Figure 6.4. The delay between clocks, caused by decoder, leaves both master and slave stages transparent between falling edges, and input D is passed to the output Q. To avoid

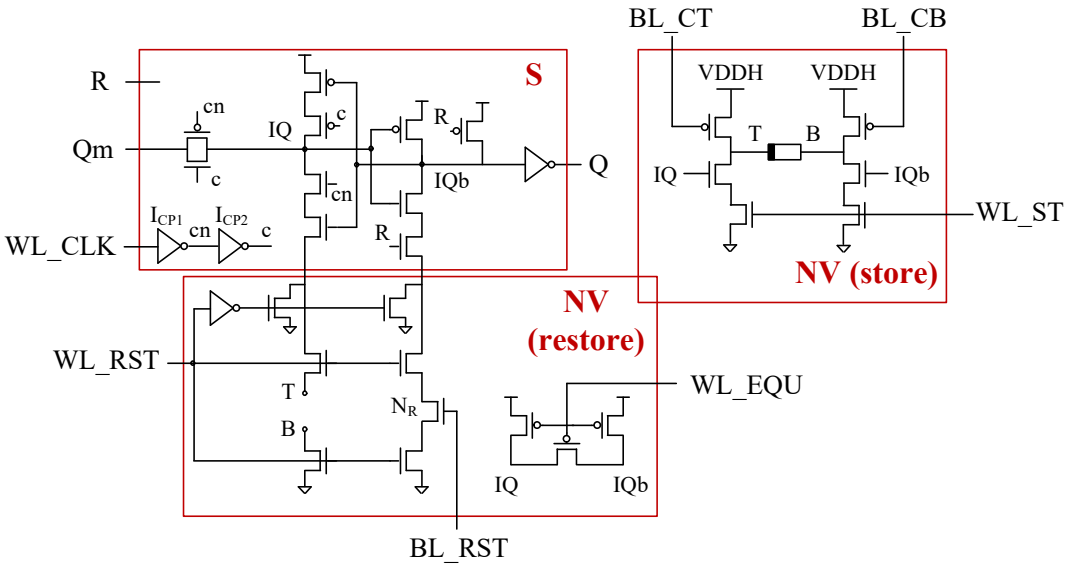


Figure 6.3: S-NV bitcell.

this, signal *WL_CLK* must have high level "inside" the master clock pulse, as shown in Figure 6.5.

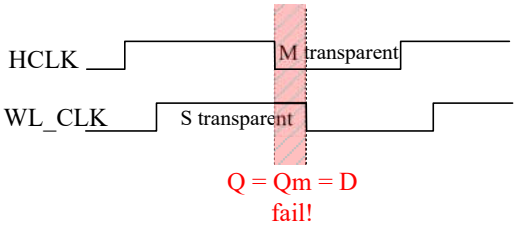


Figure 6.4: If the slave clock (*WL_CLK*) is gated master clock (*HCLK*), wrong value may be written to NVRF.

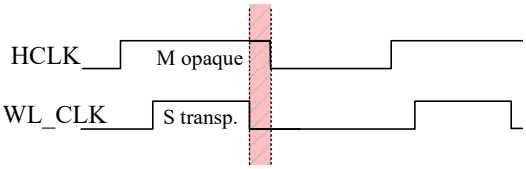


Figure 6.5: Master clock must envelope the slave clock to prevent the error.

To reshape the data signal and enable fast write operation, the buffers BW are inserted between M cell and the column of S-NV bitcells, as depicted in Figure 6.1.

Compared to single MSFF, the delay between master and slave clocks causes the increase of *HCLK-Q* propagation time, while not impacting setup and hold times. However, certain delay may appear in the MSFF-based volatile register file if it uses slower flip-flop cells with enable signal. The inserted buffers do not bring additional delay penalty as it is masked by *WL_CLK* delay.

6.2.3 RST block (restore operation)

Restore operation uses the same method as 1R NVFF – differential sensing is performed to compare single ReRAM device to a reference resistance designed using current mirror. As shown in Figure 6.6, S-NV bitcells contain restore access transistors N_1 – N_4, while

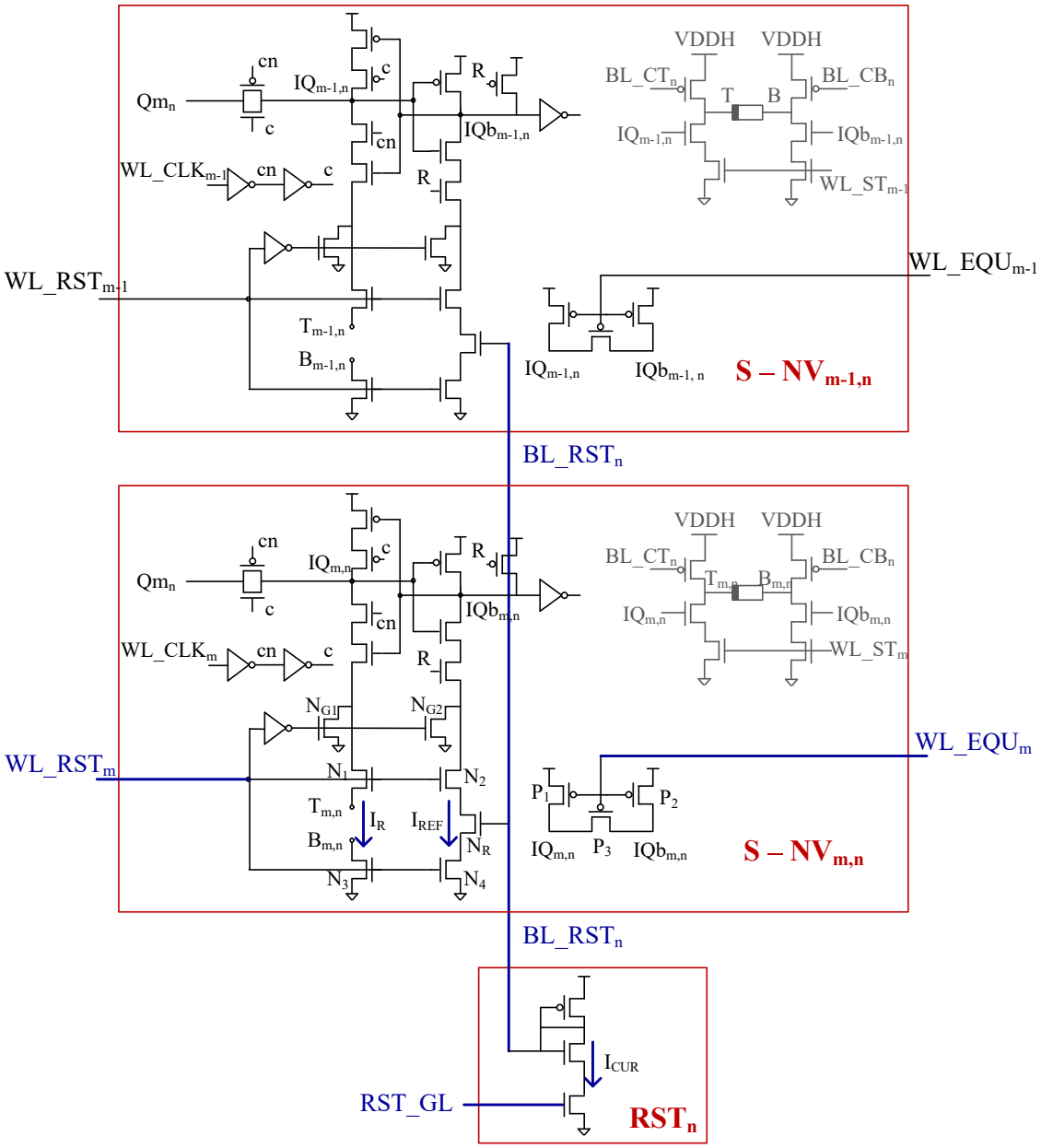
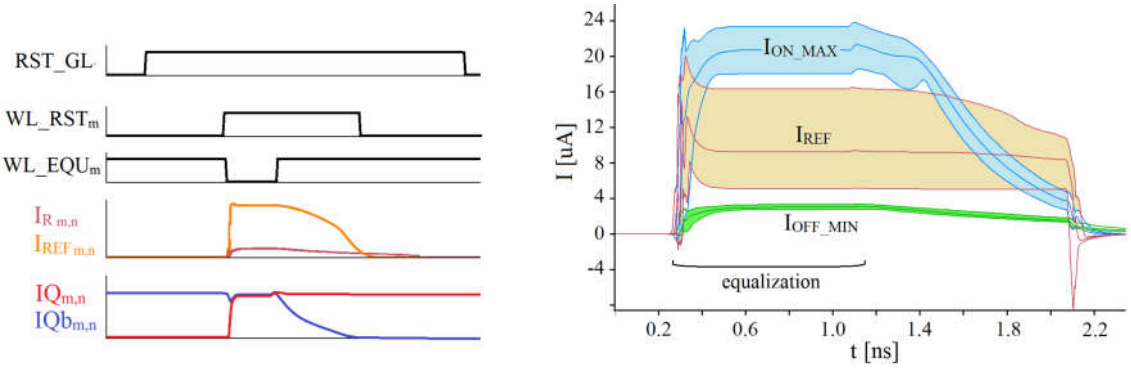


Figure 6.6: RST block with two S-NV bitcells in one column.

the slave latch acts as a sense amplifier. They include P_1-P_3 as precharge transistors, and N_{G1} and N_{G2} to ground the sources of the latch when the cell is not in restore mode.

Each bitcell has one ReRAM device, while the current source used to generate the reference is located in RST block and shared between cells. Global *RST_GL* signal is active during whole restore procedure (for the full array), providing the stable current in the source (I_{CUR}). By activating *WL_EQU* and *WL_RST*, reference current I_{REF} and device current I_R are generated and compared in the selected word, as illustrated in Figure 6.7a.

Note that the latch in the bitcell is not fully balanced, i.e. there are no inserted transistors and inverters which exist in NVFFs (P_B, N_B, and IV_1 in Figure 4.4). The yield of restore operation is sufficient without this modification, since: (i) transistor stacking is not used in the design, leading to reduced resistance of restore access transistors, and (ii) the wider CMOS nodes have less process variation. This is demonstrated in Figure 6.7b, which shows the spread of reference current and ReRAM current in one cell (I_{REF} and I_R in Figure 6.6). The results are shown for the smallest memory window of CBRAM – I_{ON_MAX} is obtained for maximal R_{ON}, while I_{OFF_MIN} corresponds to minimal R_{OFF}. Monte Carlo simulations are performed for 10000 samples, taking into account the process variation in all corners, at nominal VDD voltage of 1.5V. Results show that there is no overlap of compared currents, leading to the correct restore operation.



(a) (b)

Figure 6.7: (a) Timing diagram of RESTORE 1 of selected cell (ReRAM is in OFF state). (b) The distribution of referent current (I_{REF}) and the ReRAM current for the extreme resistance values (I_{ON_MAX} for R_{ON_MAX}, and I_{OFF_MIN} for R_{OFF_MIN}) during the restore @1.5V.

Since RST generates the reference voltage which is the same for the whole array, single block can be used to simultaneously store multiple bitcells. Still, one RST is placed in

each column, as it simplifies the layout design, without the area overhead and results in minimal voltage loss in bit-lines. Moreover, this architecture allows for restoring all bitcells in the array at the same time. However, the limitation is set by the maximum level of the currents. Due to large variability of the ReRAM technology, cells must be designed to provide large currents – I_{REF} must be lower than the lowest I_R, which corresponds to the

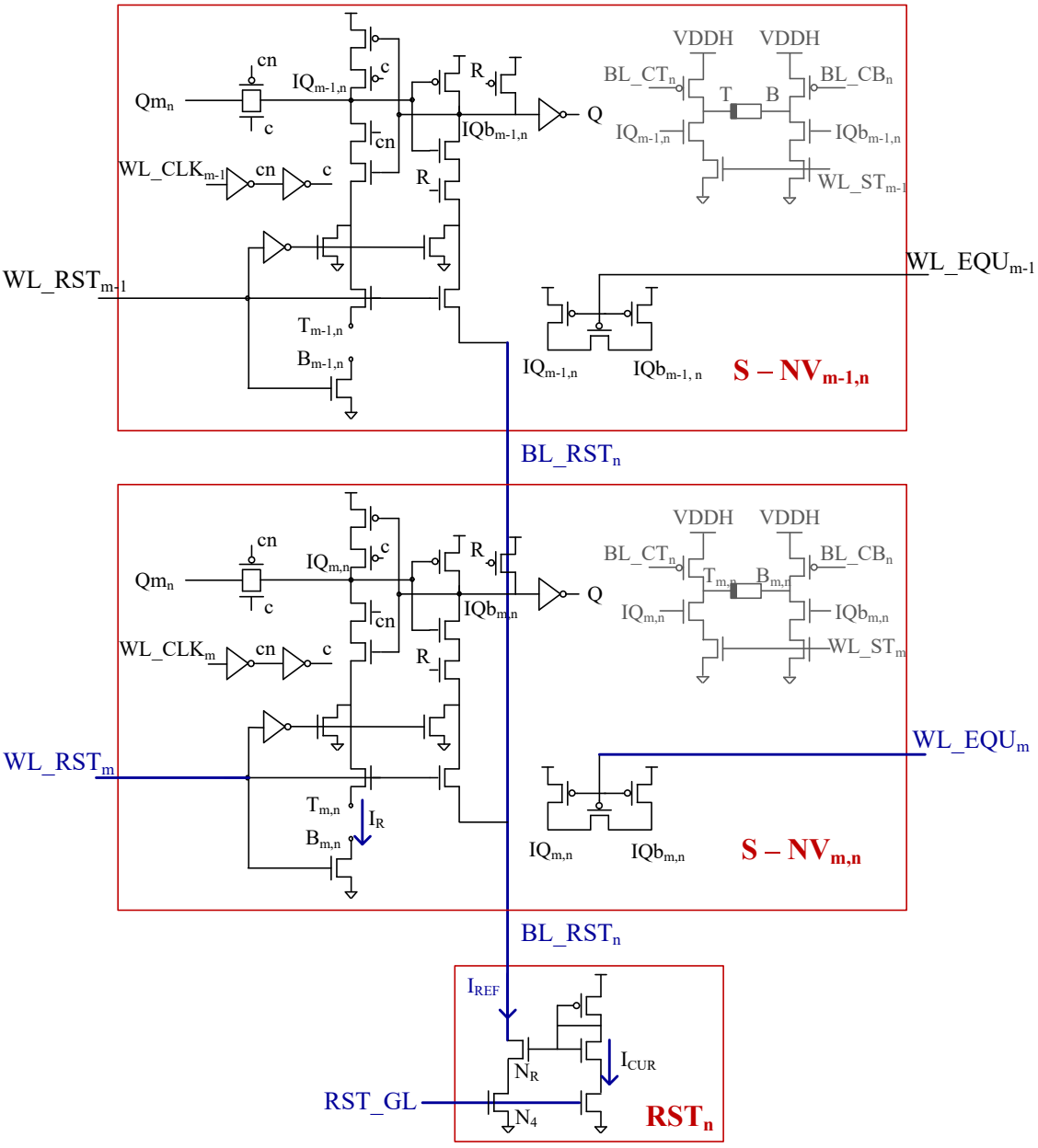


Figure 6.8: RST block with two S-NV bitcells in one column – alternative solution.

highest R_{ON}. Consequently, I_R is even higher for nominal and minimal R_{ON}. Therefore, in the worst case, if all bits of the word are in the minimal R_{ON} state (I_{ON_MIN}~50 μA), restoring 32b word can create up to 2 mA of current during the equalization pulse.

The proposed restore NVRF architecture is chosen over the alternative solution shown in Figure 6.8. The latter is obtained by moving transistors N_R and N_4 outside of bitcell to the RST block, to generate global reference current. Although this reduces the bitcell area, it is more sensitive to the parasitic resistance of the bit-line. Furthermore, to restore more than one word at the same time, it is necessary to provide multiple reference currents, which increases the size of RST block.

6.2.4 ST block (store operation)

Store operation in NVRF is realized using the programming principle described in Figure 4.15a – SET and RESET current mirrors/sources use the control signals at VDD level to provide higher programming voltage (VDDH) and required current. The architecture is depicted in Figure 6.9 which shows ST block, and two S-NV bitcells that belong to the same column. Each cell has the device programming transistors – P_5, N_6 and N_8 on the SET path, and P_6, N_5 and N_7 on the RESET path. Common SET/RESET current sources (P_9, P_{11} and N_9 – N_{12}) are located in the ST block. Transistors N_{10} and N_{12} in ST are inserted for the mirror symmetry and to reduce the current variation. Both bitcells and ST blocks are supplied by VDDH.

To activate appropriate current source, control signals in ST block (GL_RESET and GL_SET) are generated using global signal $STORE\_GL$ which is high during whole store procedure (for the full array), and the Q value of the selected cell. The latter (Q_GL) is obtained using already existing read circuit. On the other hand, the programming paths inside bitcells are selected by internal flip-flop nodes (IQ, IQb) and word-line WL_ST. Figure 6.10 illustrates the control signals and the voltage, current and the resistance state of ReRAM devices during the store operation performed on two bitcells.

As ST block programs one selected cell at a time, one ST per column architecture enables row-by-row store procedure for the whole NVRF. Still, it is important to estimate the possible current peak during this operation. Based on the results of NVFF cells shown in section 5.3, the highest current appears when performing RESET on the device in ON state. Thus, in the worst case, if all CBRAMs of the word have minimal R_{ON} value and

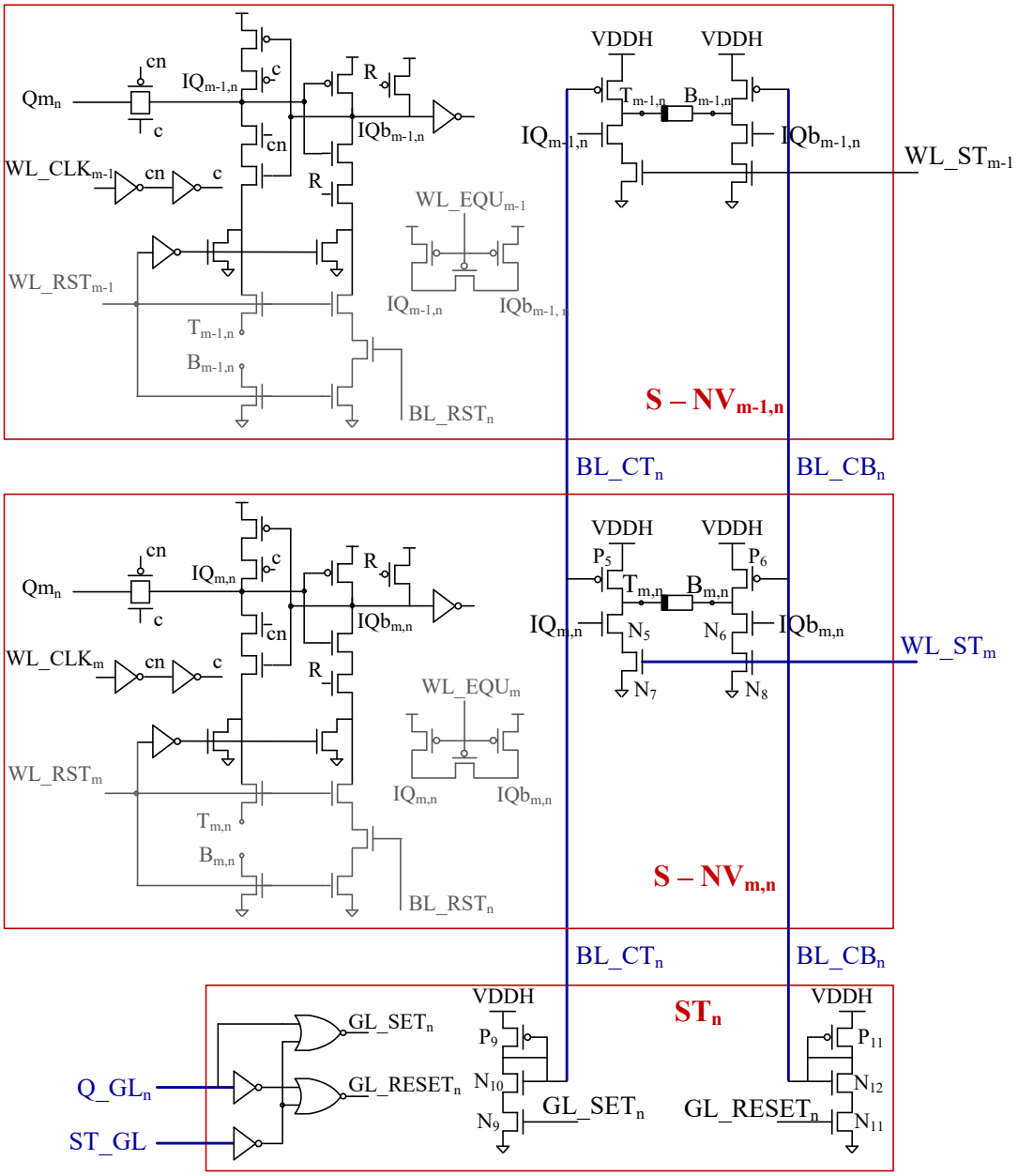


Figure 6.9: ST block with two S-NV bitcells in one column.

should change the state, storing 32b can reach up to 8mA.

In the proposed store NVRF architecture it is necessary to distribute VDDH supply to the whole array, which may complicate the routing of the power grid. To avoid this, alternative schematics shown in Figure 6.11 can be used. It is obtained by moving whole

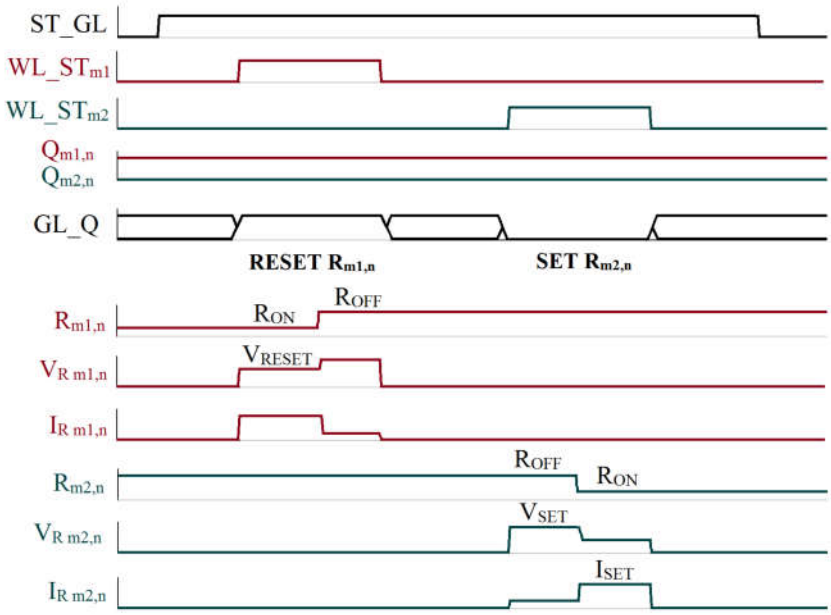


Figure 6.10: Timing diagram of store operation of S-NV$_{m1,n}$ cell (RESET is performed on ReRAM which is in ON state) and S-NV$_{m2,n}$ cell (SET is performed on ReRAM which is in OFF state)

current mirrors, as well as VDDH power supply, outside of bitcell to the ST block. However, in order to keep the initial approach and avoid V_{TH} drop of the access transistors, both PMOS and NMOS must be used as selectors. Consequently, PMOS selectors require level-shifted word-lines, which complicates the decoder circuits. Besides, it is more sensitive to the parasitic resistance of the bit-line.

6.2.5 READ block (read operation)

Figure 6.12 presents the READ block used in NVRF. The output Q of each bitcell is tied to the bit-lines by NMOS selector controlled by word-line, and bit-line value is amplified with PMOS keeper and inverter. This circuit is then duplicated to provide two read operations in parallel.

To achieve higher density, this architecture is chosen over the solution which employs the standard multiplexer cells. For example, READ block can be realized in two stages of 4$\longrightarrow$1 multiplexers, as shown in Figure 6.13, or in four stages of 2$\longrightarrow$1 multiplexers. As the multiplexer cells use the address bits directly, they do not require decoders in

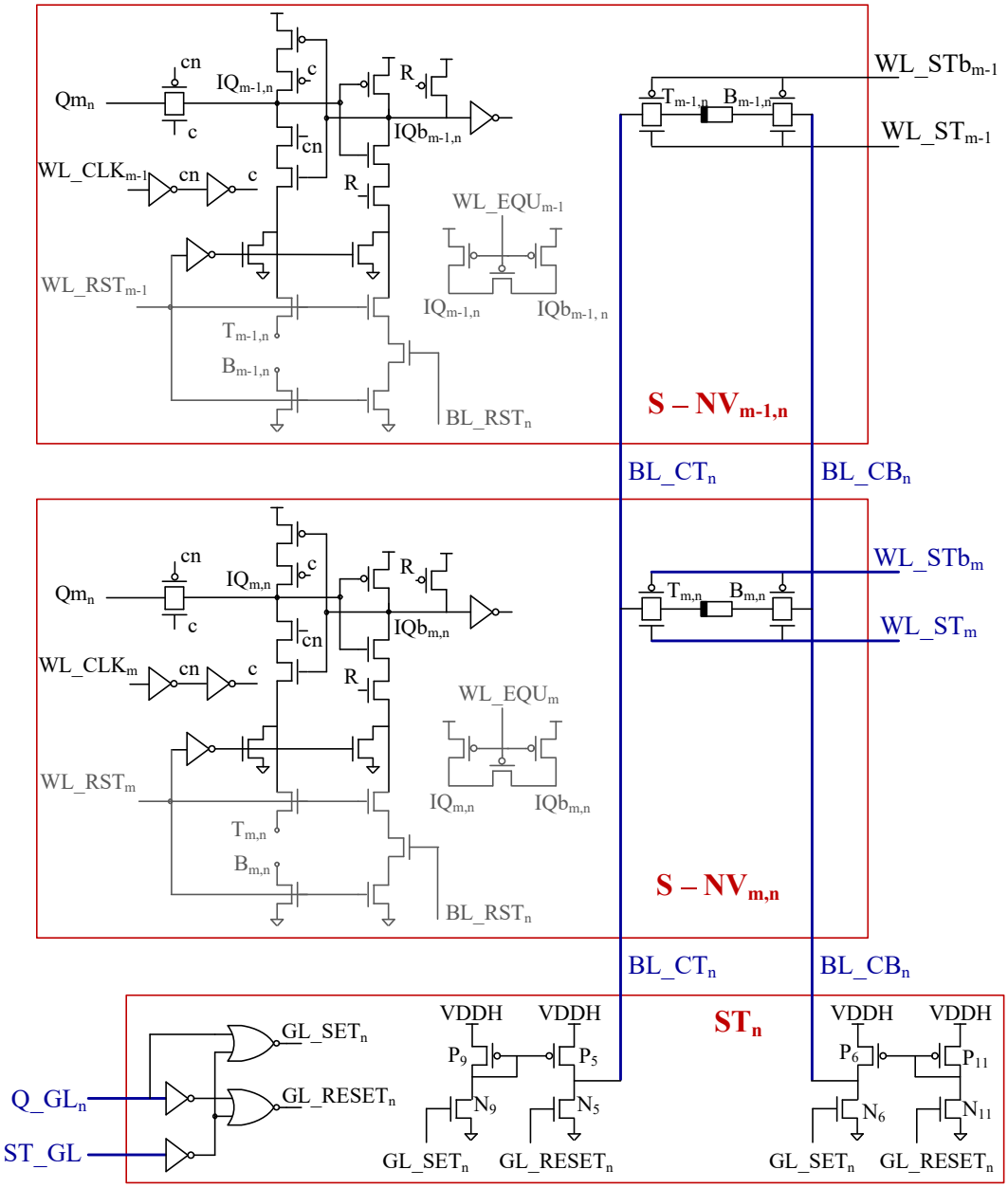


Figure 6.11: ST block with two S-NV bitcells in one column – alternative solution.

front of the array. According to the estimation shown in Table 6.1, the implemented solution reduces NVRF width (and consequently, NVRF area) 29% to 39% compared to the multiplexer-based implementation.

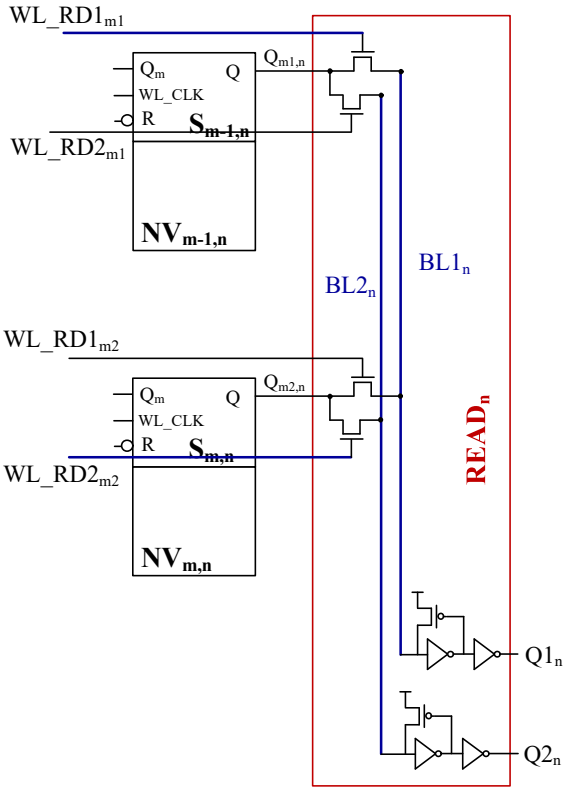


Figure 6.12: READ block with two S-NV bitcells in one column, realized using NMOS selectors and keepers.

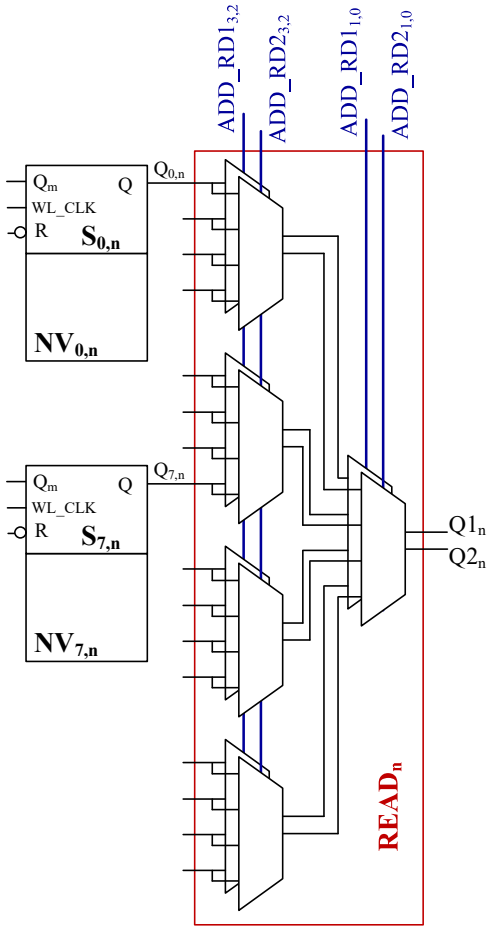


Figure 6.13: READ block with two S-NV bitcells in one column, realized using standard 4⟶1 multiplexer cells.

Table 6.1: Estimation of the layout width for different READ possibilities

	NMOS selector	**mux 4⟶1**	**mux 2⟶1**
2-read column pitch [μm]	1.94	10	8
Decoders width [μm]	∼20	0	0
32b 2-read circuitry [μm]	∼82	320	256
32b 2-read NVRF [μm]	620	860	800

6.2.6 DECODERS

To allow parallel execution of 2 read and 1 write operations, three address decoders are needed, as presented in Figure 6.14. In order to meet the requirement for the slave clock *WL_CLK* discussed in subsection 6.2.2, the pulse *HCLK_aux* is generated based on the main clock *HCLK*, and then gated by the decoded address *ADD_WR*, as illustrated in Figure 6.15.

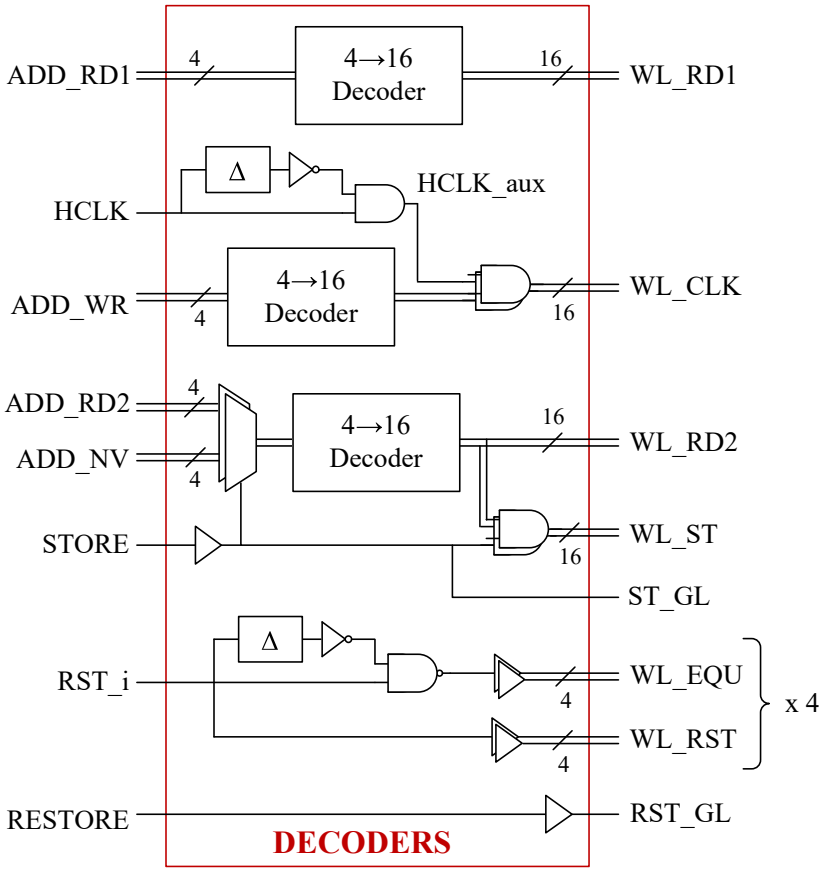


Figure 6.14: DECODERS block.

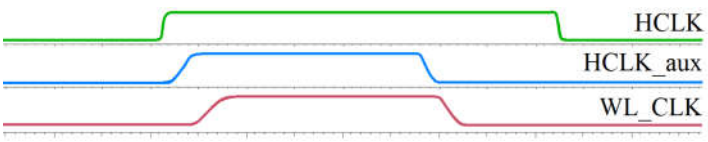


Figure 6.15: Signals related to WRITE operation.

The separate decoders are not necessary for NV address, as NV operations are not performed at the same time as read/write. Particularly, as the store operation requires the

read from the same address, they can share the decoding circuitry. Thus, during the store operation generated *WL_RD2* and *WL_ST* correspond to decoded *ADD_NV*, while during the active mode, *WL_RD2* corresponds to decoded *ADD_RD2* and all *WL_ST* are low.

As restore operation can be performed on several rows at the same time, it is controlled by signals *RST_1-RST_4*, each restoring four words simultaneously. Hence, there is no restore address nor decoder. Each signal is used to generate the corresponding equalizer pulse required for the restore operation. This partitioning is chosen to have the same maximal current peak as the store operation.

6.2.7 Operating modes

NVRF works in the following modes, according to the timing diagrams of NVRF input/output signals and power supplies, shown in Figures 6.16 and 6.17.

- In **active mode**, two read and one write operation (2R1W) on 32b words are performed in parallel at *ADD_RD1*, *ADD_RD2* and *ADD_WR* addresses, respectively. *STORE*, *RESTORE*, and *RST_1-RST_4* must be low, to deactivate NV part. VDDH can be at V_{NOM} level, or at 0 to reduce the consumption.
- During the **store mode** 32b words defined by *ADD_NV* are backed-up. Clock *HCLK* must be low to latch the slave and prevent the write operation, as well as *RESTORE* and *RST_1-RST_4*. Output *Q1* is still reading from the *ADD_RD1*, while *Q2* shows the values which are currently stored (from *ADD_NV*). VDDH is increased to V_{STR}.
- In the **sleep mode** all supplies are off.
- During the **restore mode**, four 32b words can be restored in parallel. Clock *HCLK* must be low to latch the slave and prevent the write operation, as well as *STORE* signal. VDDH must be at V_{NOM}, as VDDH=0 causes leakage in the programming part and reduces restore yield significantly. Note that this differs from NVFF, due to the design without protection transistor stacking.
- **Forming mode** starts by reseting the whole array to write 0 in each bitcell, which is followed by store operation with power supply increased to V_{FRM}.

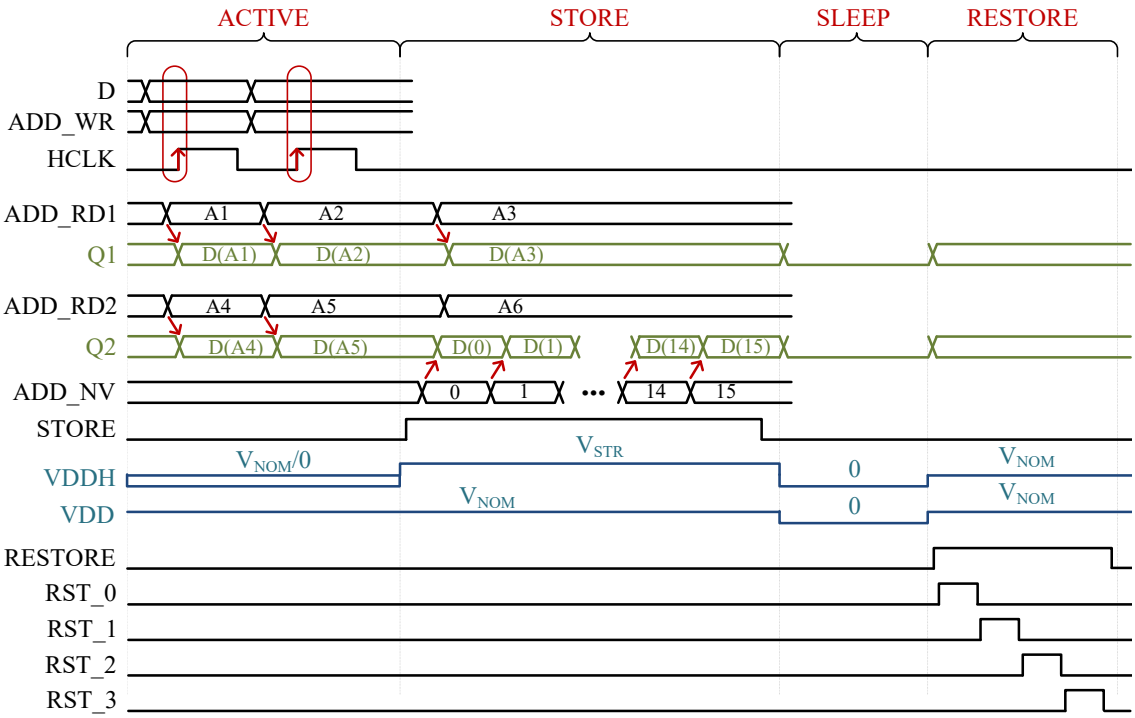


Figure 6.16: Timing diagram of NVRF modes: active, store, sleep, restore.

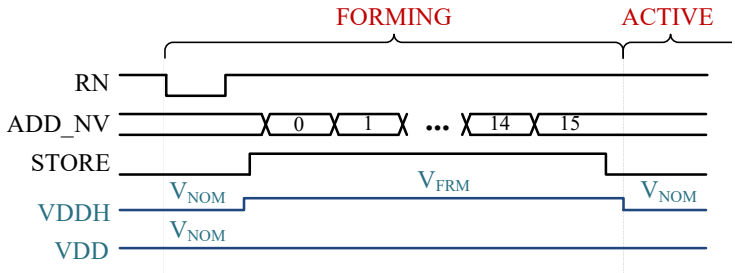


Figure 6.17: Timing diagram of NVRF modes: forming.

Technologies	CBRAM, 130nm
V_{NOM}	1.5V
V_{STR}	2.4V
R_{ON}	<30kΩ
R_{OFF}	>200kΩ

Figure 6.18: Parameters of implemented NVRF.

6.2.8 Implementation

The parameters of 16x32 CBRAM-based NVRF in 130nm CMOS node are listed in Figure 6.18, and the layout is shown in Figure 6.19. CBRAM device is located between Metal 4 and Metal 5 layers. To avoid the routing congestion caused by three-supplies power grid on top of the logic and CBRAMs, the memory uses only Metal 1 – Metal 5 layers. Then, the power ring for VDD, GND and VDDH in Metal 5 and Metal 6 is placed outside the NVRF.

The total area of non-volatile register file, including the power ring and inserted decoupling capacitors is 638μm x 104μm, while the array and the decoders occupy 620μm x 72μm. S-NV cell is 1.85x bigger than the corresponding volatile flip-flop of the same driveability. However, compared to RTL-generated volatile register file, the area penalty of adding non-volatility is minimal, as custom READ block is significantly smaller than the one made of standard cells, according to the Table 6.1.

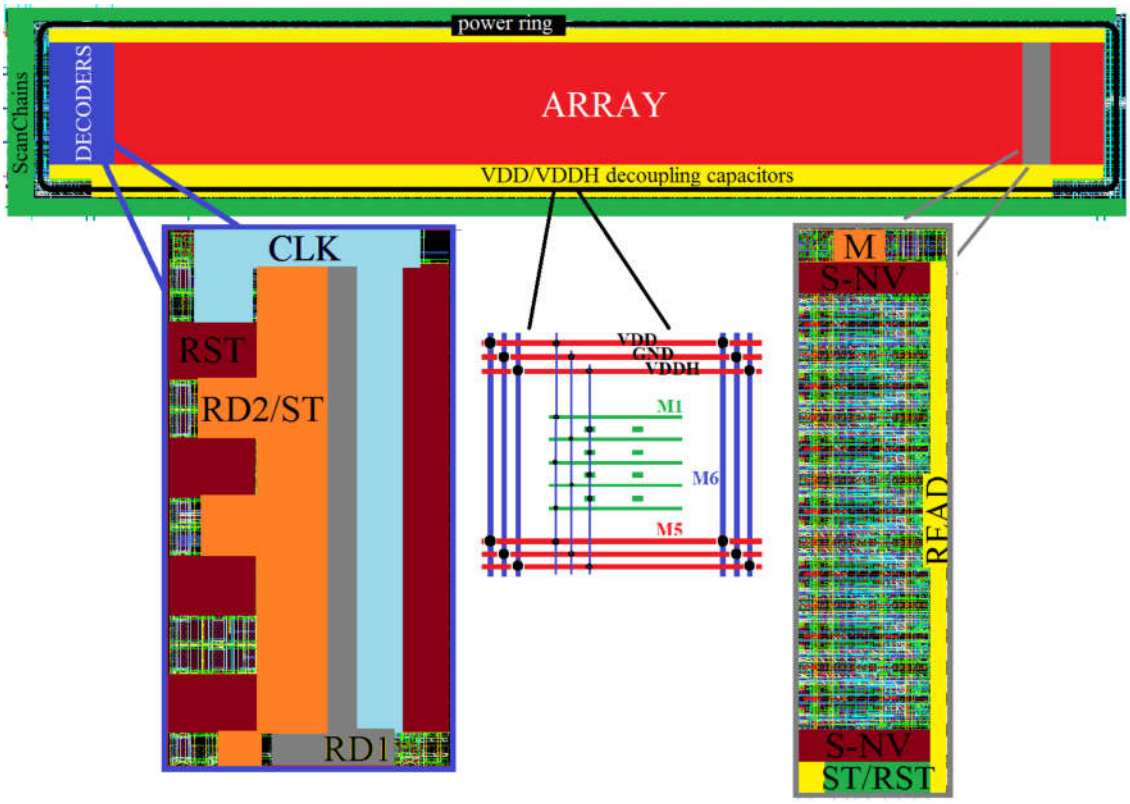


Figure 6.19: Layout of NVRF (top), and a zoom-in on the decoders (bottom, left), the column (bottom, right), and the power grid arrangement (bottom, middle).

6.3 Summary

Presented design demonstrate that the area overhead in non-volatile flip-flops can be minimized by organizing them in a register file array, without impacting the robustness of store and restore operations. Implemented solution enables the parallel execution of write-to-one and read-from-two addresses. Moreover, the number of ports can be easily

extended without the modification of S-NV bitcell. Along with supplementary address decoder, each additional read or write operation require one NMOS transistor in the READ block, or new M cell, respectively. This NVRF architecture supports very fast wake-up transition, as restore operation can be performed on several rows simultaneously.

Finally, proposed NVRF design can be ported to 28nm and below CMOS node using only thin-gate oxide transistors, by following the approach implemented in NVFFs in chapter 4. Thus, the transistor stacking must be implemented in NV part of S-NV bitcell and in ST block, and VDD/VDDH supply levels must correspond to the configuration proposed in Table 4.2.

Chapter 7

Conclusion

Non-volatile sequential logic circuits

To achieve an adequate ReRAM-based design and exploit all the benefits that this non-volatile technology brings, it is important to understand the relations between different NVM device properties, and the ways they can be adjusted. In this thesis, the **influence of ReRAM programming voltages and currents on R_{ON} and R_{OFF} values, switching speed, programming energy and device endurance has been analyzed**. Moreover, different elements of NVFF design topologies (store/restore circutry, power supply, etc.) are investigated with respect to the properties of used NVM and CMOS technologies, such as memory window, programming conditions, variability. Gathered design guidelines are then used for the implementation of novel non-volatile ReRAM-based flip-flops and register file.

First, **two dual-voltage circuits for the programming of ReRAM devices have been proposed**: level shifter-based and current programming-based. Using only thin-gate oxide transistors for the compatibility with digital design flow, both solutions **successfully solve the problem of reliability**, which is a main issue encountered in co-design of ReRAM and advanced CMOS nodes. Additionally, current programming circuit does not require generation of intermediate signals on different voltage levels. Second, **two restore architectures have been investigated**: a two-device architecture, compatible with wide range of ReRAM technologies, and a novel single-device architecture which allows for better integration density and reduced consumption. Robust **low-voltage restore operations** are achieved by choosing the appropriate ReRAM technologies and their programming conditions, after the statistical analysis which take into account both CMOS and ReRAM variations. Third, the core of NVFF, particularly **the slave stage architecture, is examined and defined** to reach the highest restore yield and performance in regular flip-flop mode.

To investigate the difference between store/restore architectures, three out of four proposed **NVFFs have been simulated and laid-out in 28nm FDSOI** – 2R-LS, 2R-CM and 1R-CM. Designed as **standard cells**, all solutions are optimized to satisfy the following challenges: using ReRAM programming conditions which improve endurance and minimize power, achieving minimal impact on flip-flop operations, and overcoming the gap between SET and RESET programming conditions which allows the same programming-pulse widths for both operations. **2R-CM and 2R-LS are OxRAM-based** and can perform restore operation at reduced voltage of 0.8V, while **1R-CM is CBRAM-based** and have successful restore at nominal voltage.

Post-layout simulations have shown that, compared to the master-slave flip-flop from the standard cell library, all NVFFs are characterized with **similar, small propagation delay penalty** (8.5% in the worst case), while their setup and hold times are not affected. Regarding the **active-mode consumption**, NVFFs require **10-14% more energy** than MSFF at 10MHz with 10% data activity at nominal voltage. This overhead can be reduced down to 2% at higher frequencies, lower data activity and lower operating voltage. Among proposed solutions, **1R restore and CM store are the most energy efficient architectures in active mode**.

To evaluate the consumption of NVFFs in the store mode, simplified electrical model of ReRAM behavior has been used, covering different store scenarios which depend on the flip-flop data. Results indicate that the **consumption for LS and CM programming circuits is similar** when they are optimized for the same technology and the programming conditions. In general, NVFFs with **1R restore are more energy efficient than 2R NVFFs** with the same ReRAMs, due to only one programming circuit and simpler control block. However, their higher-memory window requirement limits the choice of ReRAM technology or implies more aggressive programming conditions (particularly, longer pulse widths), increasing the programming energy. Hence, designed **2R-CM and 2R-LS NVFFs based on faster OxRAM devices have the lower store energy consumption than CBRAM-based 1R-CM NVFF**. Evaluation presented in this work suggests that for normally-off systems characterized by long standby modes **NVFF solutions can bring considerable energy savings**. For example, existing data-retention flop supplied at 0.5V exceeds the consumption of 2R-LS/2R-CM or 1R-CM after inactivity periods of 0.2-0.8s, or 0.7-3s, respectively.

NVFF layout is **fully compatible with the digital design flow**, with ReRAM devices at the back-end-of-line between two last metal layers. Compared to the area of standard

MSFF cell, NVFFs range from **3x larger** (1R-CM) to 5.6x larger (2R-LS). However, both cells can be reduced, as they can share restore current source or forming level-shifter with other NVFFs on chip.

Presented results have been obtained for the standard MSFF cell of the smallest drive (x8) and corresponding NVFFs. As the NVFFs with more complex flip-flop cores (e.g., with higher drive, or with asynchronous set/reset) have the same NV block, their active-mode performance and consumption, as well as the area, will have smaller penalty.

Finally, it has been shown that the main drawback of NVFFs, **area overhead, can be minimized by creating the memory array** and placing the common circuitry for write, read, store and restore on the periphery. In this work, **two-read one-write multi-port non-volatile register file has been implemented** in 130nm CMOS node. The array relies on the CBRAM-based 1R-CM NVFF. Suggested solution offers higher density than NVFF with the same wake-up transition, and can be easily adjusted to support more read and write ports.

Future work: non-volatile processor

After designing ReRAM-based NVFF, the next natural step is to apply the same approach on a larger-scale circuit. In fact, there is already an ongoing work in implementing the solution presented in this thesis is the bigger NVP project. In the first attempt to realize ReRAM-based non-volatile microcontroller unit, NVFF based on 1R-CM architecture is included in the ARM Cortex-M0+ microprocessor. All flip-flops of the volatile version of the processor core are replaced with their non-volatile counterparts, as illustrated in the Figure 7.1. Consequently, dedicated power management unit (PMU) with the finite-state machine which handles different sleep modes and power supply requirements is realized. In the deep sleep mode, only PMU and wake-up controller (WIC) stay powered, while the rest of the circuit can be cut-off. Thus, the non-volatile version enables high energy savings compared to the standard MCU, with minimal go-to-sleep/wake-up penalty. The layout of the non-volatile MCU is shown in Figure 7.2.

One of the major challenges and the following step in the development of non-volatile processor is reducing its area. To address this issue, a plausible solution may be including non-volatile register file together with non-volatile flip-flops. For the architectures such as M0+, which do not contain the register file and have flip-flop-based general-purpose

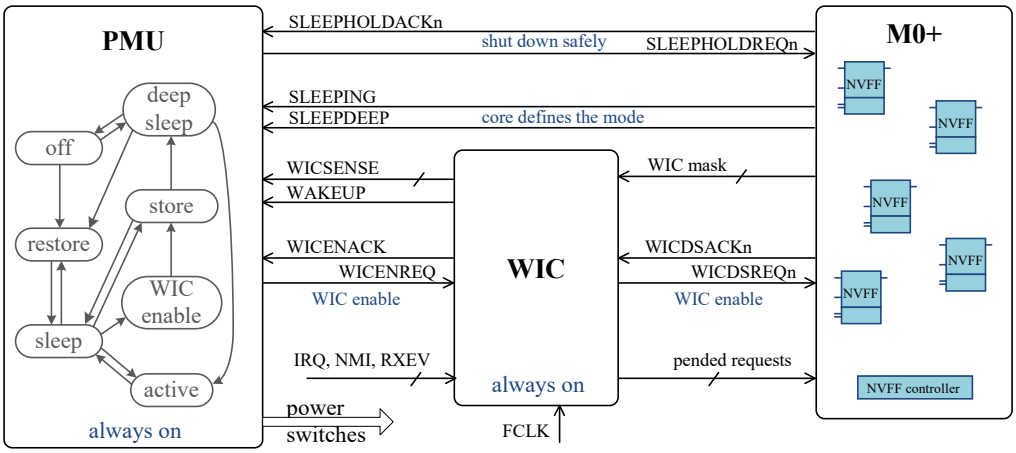
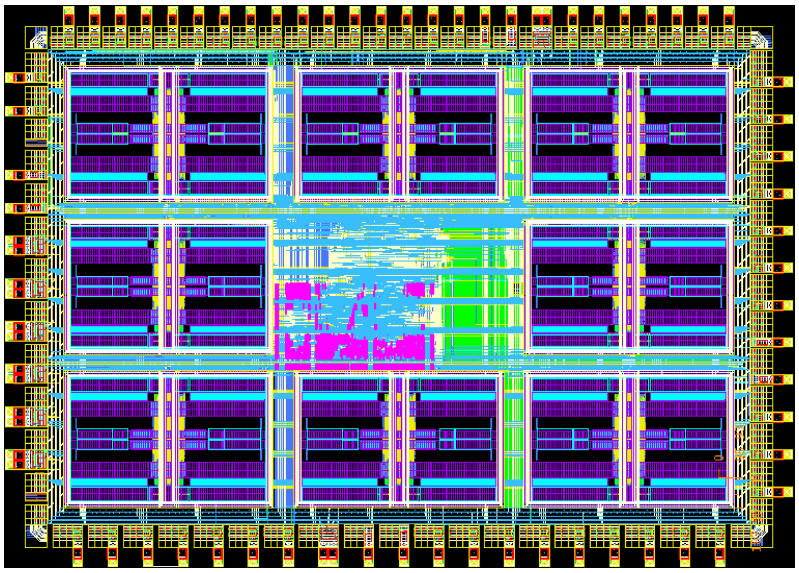


Figure 7.1: NVFFs are integrated in non-volatile MCU.



(a)

(b)

Figure 7.2: Layout of NV M0+: (a) full circuit with highlighted NVFFs (pink), (b) zoom-in on the power grid.

registers, the implementation of NVRF requires slight modification of the processor core. Consequently, the power management unit and NVFF/NVRF controller must be adjusted to support this modification.

Once the concept with "fully" non-volatile hardware is proven, new directions for NVP improvement will arise. For example, instead of straightforwardly making all flops non-volatile, deeper system examination can suggest appropriate subset of FFs to be replaced with NVFFs. Finally, the development can also be continued on the software level. In order to better support NV capability, new ways to adapt the system software can be explored, such as: additional control of go-to-sleep/wake-up procedures, optimum choice of sleep mode, etc. Thus, proper hardware/software co-design can exploit all the benefits of adding non-volatility to the IoT system.

Publications

- **N. Jovanovic**, O. Thomas, E. Vianello, B. Nikolic, and L. Naviner. *"Design Considerations for Reliable OxRAM-based Non-Volatile Flip-Flops in 28nm FD-SOI Technology"*. Internatial Symposium on Circuits and Systems (ISCAS), IEEE, 2016.
- **N. Jovanovic**, E. Vianello, O. Thomas, B. Nikolic, and L. Naviner. *"Design insights for reliable energy efficient OxRAM-based flip-flop in 28nm FD-SOI"*. SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2015 IEEE, pages 1–2, Oct 2015., *best paper award.*
- **N. Jovanovic**, O. Thomas, E. Vianello, J.-M. Portal, B. Nikolic, and L. Naviner. *"OxRAM-based non volatile flip-flop in 28nm FDSOI"*. New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International, pages 141–144, June 2014.
- E. Vianello, O. Thomas, G. Molas, O. Turkyilmaz, **N. Jovanovic**, D. Garbin, G. Palma, M. Alayan, C. Nguyen, J. Coignus, B. Giraud, T. Benoist, M. Reyboz, A. Toffoli, C. Charpin, F. Clermidy, and L. Perniola. *"Resistive memories for ultra-low-power embedded computing design"*. Electron Devices Meeting (IEDM), 2014 IEEE International, pages 6.3.1–6.3.4, Dec 2014.
- E. Vianello, D. Garbin, **N. Jovanovic**, O. Bichler, O. Thomas, B. Salvo, and L. Perniola. *"Oxide based resistive memories for low power embedded applications and neuromorphic systems"*. ECS Transactions, 69(3):3–10, 2015.
- F. Longnos, M. Reyboz, **N. Jovanovic**, A. Levisse, T. Benoist, G. Suraci, O. Thomas, E. Vianello, G. Molas, B. Salvo, and L. Perniola. *"CBRAM corner analysis for robust design solutions"*. Non-Volatile Memory Technology Symposium (NVMTS), 2014 14th Annual, pages 1–4, Oct 2014.
- M. Reyboz, **N. Jovanovic**, F. Longnos, E. Vianello, O. Thomas, F. Clermidy, G. Molas, S. Onkaraiah, J.-M. Portal, and C. Muller. *"From compact model to innovative circuit design of Ag-GeS_2 conductive bridge memories"*. Memory Workshop (IMW), 2014 IEEE 6th International, pages 1–4, May 2014.
- F. Clermidy, **N. Jovanovic**, S. Onkaraiah, H. Oucheikh, O. Thomas, O. Turkyilmaz, E. Vianello, J.-M. Portal, and M. Bocquet. *"Resistive memories: Which applications?"*. Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, pages 1–6, March 2014.

Bibliography

[1] H. Jayakumar, K. Lee, W.S. Lee, A. Raha, Y. Kim, and V. Raghunathan. *"Powering the Internet of Things"*. Low Power Electronics and Design (ISLPED), 2014 IEEE/ACM International Symposium on, pages 375–380, Aug 2014. (Cited on page 1.)

[2] *"Intel IoT Platform."*. http://www.intel.com/content/www/us/en/internet-of-things/overview.html. (Cited on page 1.)

[3] M. Hayashikoshi, Y. Sato, H. Ueki, H. Kawai, and T. Shimizu. *"Normally-off MCU architecture for low-power sensor node"*. Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, pages 12–16, Jan 2014. (Cited on page 2.)

[4] N.S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan. *"Leakage current: Moore's law meets static power"*. Computer, 36(12):68–75, Dec 2003. (Cited on pages 2 and 22.)

[5] K.J. Nowka, G.D. Carpenter, E.W. MacDonald, H.C. Ngo, B.C. Brock, K.I. Ishii, T.Y. Nguyen, and J.L. Burns. *"A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling"*. Solid-State Circuits, IEEE Journal of, 37(11):1441–1447, Nov 2002. (Cited on page 3.)

[6] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada. *"1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS"*. Solid-State Circuits, IEEE Journal of, 30(8):847–854, Aug 1995. (Cited on page 3.)

[7] H. Nakamura, T. Nakada, and S. Miwa. *"Normally-off computing project: Challenges and opportunities"*. Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, pages 1–5, Jan 2014. (Cited on pages 3 and 4.)

[8] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tanabe, and J. Yamada. *"A 1-V high-speed MTCMOS circuit scheme for power-down application circuits"*. Solid-State Circuits, IEEE Journal of, 32(6):861–869, Jun 1997. (Cited on pages 3, 22 and 58.)

[9] P.-F. Chiu, M.F. Chang, C.-W. Wu, C.-H. Chuang, S.-S. Sheu, Y.-S. Chen, and M.-J. Tsai. *"Low Store Energy, Low VDDmin, 8T2R Nonvolatile Latch and SRAM With Vertical-Stacked Resistive Memory (Memristor) Devices for Low Power Mobile*

Applications". Solid-State Circuits, IEEE Journal of, 47(6):1483–1496, June 2012. (Cited on page 4.)

[10] N. Planes, O. Weber, V. Barral, S. Haendler, D. Noblet, D. Croain, M. Bocat, P. Sassoulas, X. Federspiel, A. Cros, A. Bajolet, E. Richard, B. Dumont, P. Perreau, D. Petit, D. Golanski, C. Fenouillet-Beranger, N. Guillot, M. Rafik, V. Huard, S. Puget, X. Montagner, M.-A. Jaud, O. Rozeau, O. Saxod, F. Wacquant, F. Monsieur, D. Barge, L. Pinzelli, M. Mellier, F. Boeuf, F. Arnaud, and M. Haond. *"28nm FDSOI technology platform for high-speed low-voltage digital applications"*. VLSI Technology (VLSIT), 2012 Symposium on, pages 133–134, June 2012. (Cited on page 5.)

[11] J.S. Meena, S.M. Sze, U. Chand, and T.-Y. Tseng. *"Overview of emerging nonvolatile memory technologies"*. Nanoscale Research Letters, 9(1):1–33, 2014. (Cited on pages 8, 9 and 11.)

[12] A. Makarov, V. Sverdlov, and S. Selberherr. *"Emerging memory technologies: Trends, challenges, and modeling methods"*. Microelectronics Reliability, 52(4):628–634, 2012. (Cited on pages 8 and 12.)

[13] L. Torres, R.M. Brum, L.V. Cargnini, and G. Sassatelli. *"Trends on the application of emerging nonvolatile memory to processors and programmable devices"*. Circuits and Systems (ISCAS), 2013 IEEE International Symposium on, pages 101–104, May 2013. (Cited on page 9.)

[14] M. Qazi, A. Amerasekera, and A.P. Chandrakasan. *"A 3.4-pJ FeRAM-Enabled D Flip-Flop in 0.13- μm CMOS for Nonvolatile Processing in Digital Systems"*. Solid-State Circuits, IEEE Journal of, 49(1):202–211, Jan 2014. (Cited on pages 9, 22 and 24.)

[15] V. Rzehak. *"Low-power FRAM microcontrollers and their applications"*. Technical report, Texas Instruments, 2011. http://www.ti.com/lit/wp/slaa502/slaa502.pdf. (Cited on page 10.)

[16] P. Ramkumar. *"MSP430 FRAM microcontrollers with CapTIvate technology"*. Technical report, Texas Instruments, 2015. http://www.ti.com/lit/wp/slay044/slay044.pdf. (Cited on page 10.)

[17] M. Terao, T. Morikawa, and T. Ohta. *"Electrical Phase-Change Memory: Fundamentals and State of the Art"*. Japanese Journal of Applied Physics, 48(8R):080001,

2009. (Cited on pages 10 and 11.)

[18] Y. Choi, I. Song, M.-H. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo, J. Shin, Y. Rho, C. Lee, M.G. Kang, J. Lee, Y. Kwon, S. Kim, J. Kim, Y.-J. Lee, Q. Wang, S. Cha, S. Ahn, H. Horii, J. Lee, K. Kim, H. Joo, K. Lee, Y.-T. Lee, J. Yoo, and G. Jeong. *"A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, pages 46–48, Feb 2012. (Cited on page 11.)

[19] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano. *"A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram"*. Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International, pages 459–462, Dec 2005. (Cited on page 11.)

[20] U.K. Klostermann, M. Angerbauer, U. Griming, F. Kreupl, M. Ruhrig, F. Dahmani, M. Kund, and G. Muller. *"A Perpendicular Spin Torque Switching based MRAM for the 28 nm Technology Node"*. Electron Devices Meeting, 2007. IEDM 2007. IEEE International, pages 187–190, Dec 2007. (Cited on page 12.)

[21] K.-W. Kwon, S.H. Choday, Y. Kim, X. Fong, S.P. Park, and K. Roy. *"SHE-NVFF: Spin Hall Effect-Based Nonvolatile Flip-Flop for Power Gating Architecture"*. Electron Device Letters, IEEE, 35(4):488–490, April 2014. (Cited on pages 12, 22 and 29.)

[22] L. Liu, C.-F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman. *"Spin-Torque Switching with the Giant Spin Hall Effect of Tantalum"*. Science, 336(6081):555–558, 2012. (Cited on page 12.)

[23] R. Waser, R. Dittmann, G. Staikov, and K. Szot. *"Redox-Based Resistive Switching Memories – Nanoionic Mechanisms, Prospects, and Challenges"*. Advanced Materials, 21(25-26):2632–2663, 2009. (Cited on pages 13, 15 and 17.)

[24] F. Clermidy, N. Jovanovic, S. Onkaraiah, H. Oucheikh, O. Thomas, O. Turkyilmaz, E. Vianello, J.-M. Portal, and M. Bocquet. *"Resistive memories: Which applications?"*. Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, pages 1–6, March 2014. (Cited on page 13.)

[25] A.D. Kent and D.C. Worledge. *"A new spin on magnetic memories"*. Nature nanotechnology, 10(3):187–191, Mar 2015. (Cited on page 14.)

[26] S. Dietrich, M. Angerbauer, M. Ivanov, D. Gogl, H. Hoenigschmid, M. Kund, C. Liaw, M. Markert, R. Symanczyk, L. Altimime, S. Bournat, and G. Mueller. *"A Nonvolatile 2-Mbit CBRAM Memory Core Featuring Advanced Read and Program Control"*. Solid-State Circuits, IEEE Journal of, 42(4):839–845, April 2007. (Cited on page 14.)

[27] M.-F. Chang, S.-S. Sheu, K.-F. Lin, C.-W. Wu, C.-C. Kuo, P.-F. Chiu, Y.-S. Yang, Y.-S. Chen, H.-Y. Lee, C.-H. Lien, F.T. Chen, K.-L. Su, T.-K. Ku, M.-J. Kao, and M.-J. Tsai. *"A High-Speed 7.2-ns Read-Write Random Access 4-Mb Embedded Resistive RAM (ReRAM) Macro Using Process-Variation-Tolerant Current-Mode Read Schemes"*. Solid-State Circuits, IEEE Journal of, 48(3):878–891, March 2013. (Cited on page 14.)

[28] W. Otsuka, K. Miyata, M. Kitagawa, K. Tsutsui, T. Tsushima, H. Yoshihara, T. Namise, Y. Terao, and K. Ogata. *"A 4Mb conductive-bridge resistive memory with 2.3GB/s read-throughput and 216MB/s program-throughput"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International, pages 210–211, Feb 2011. (Cited on page 14.)

[29] T.-Y. Liu, T.H. Yan, R. Scheuerlein, Y. Chen, J.K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, T. Sasaki, S. Addepalli, A. Al-Shamma, C.-Y. Chen, M. Gupta, G. Hilton, S. Joshi, A. Kathuria, V. Lai, D. Masiwal, M. Matsumoto, A. Nigam, A. Pai, J. Pakhale, C.H. Siau, X. Wu, R. Yin, L. Peng, J.Y. Kang, S. Huynh, H. Wang, N. Nagel, Y. Tanaka, M. Higashitani, T. Minvielle, C. Gorla, T. Tsukamoto, T. Yamaguchi, M. Okajima, T. Okamura, S. Takase, T. Hara, H. Inoue, L. Fasoli, M. Mofidi, R. Shrivastava, and K. Quader. *"A 130.7mm2 2-layer 32Gb ReRAM memory device in 24nm technology"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International, pages 210–211, Feb 2013. (Cited on page 14.)

[30] R. Fackenthal, M. Kitagawa, W. Otsuka, K. Prall, D. Mills, K. Tsutsui, J. Javanifard, K. Tedrow, T. Tsushima, Y. Shibahara, and G. Hush. *"A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, pages 338–339, Feb 2014. (Cited on page 14.)

[31] J. Zahurak, K. Miyata, M. Fischer, M. Balakrishnan, S. Chhajed, D. Wells, H. Li, A. Torsi, J. Lim, M. Korber, K. Nakazawa, S. Mayuzumi, M. Honda, S. Sills,

S. Yasuda, A. Calderoni, B. Cook, G. Damarla, H. Tran, B. Wang, C. Cardon, K. Karda, J. Okuno, A. Johnson, T. Kunihiro, J. Sumino, M. Tsukamoto, K. Aratani, N. Ramaswamy, W. Otsuka, and K. Prall. *"Process integration of a 27nm, 16Gb Cu ReRAM"*. Electron Devices Meeting (IEDM), 2014 IEEE International, pages 6.2.1–6.2.4, Dec 2014. (Cited on page 14.)

[32] H.-S.P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F.T. Chen, and M.-J. Tsai. *"Metal–Oxide RRAM"*. Proceedings of the IEEE, 100(6):1951–1970, June 2012. (Cited on pages 15 and 16.)

[33] Q. Liu, M. Liu, S. Long, W. Wang, M. Zhang, Q. Wang, and J. Chen. *"Improvement of resistive switching properties in ZrO2- based ReRAM with implanted metal ions"*. Solid State Device Research Conference, 2009. ESSDERC '09. Proceedings of the European, pages 221–224, Sept 2009. (Cited on page 15.)

[34] M. Tada, T. Sakamoto, K. Okamoto, M. Miyamura, N. Banno, Y. Katoh, S. Ishida, N. Iguchi, N. Sakimura, and H. Hada. *"Polymer solid-electrolyte (PSE) switch embedded in 90nm CMOS with forming-free and 10nsec programming for low power, nonvolatile programmable logic (NPL)"*. Electron Devices Meeting (IEDM), 2010 IEEE International, pages 16.5.1–16.5.4, Dec 2010. (Cited on page 15.)

[35] M. Tada, T. Sakamoto, N. Banno, K. Okamoto, M. Miyamura, N. Iguchi, and H. Hada. *"Improved reliability and switching performance of atom switch by using ternary Cu-alloy and RuTa electrodes"*. Electron Devices Meeting (IEDM), 2012 IEEE International, pages 29.8.1–29.8.4, Dec 2012. (Cited on page 15.)

[36] E. Vianello, G. Molas, F. Longnos, P. Blaise, E. Souchier, C. Cagli, G. Palma, J. Guy, M. Bernard, M. Reyboz, G. Rodriguez, A. Roule, C. Carabasse, V. Delaye, V. Jousseaume, S. Maitrejean, G. Reimbold, B. De Salvo, F. Dahmani, P. Verrier, D. Bretegnier, and J. Liebault. *"Sb-doped GeS2 as performance and reliability booster in Conductive Bridge RAM"*. Electron Devices Meeting (IEDM), 2012 IEEE International, pages 31.5.1–31.5.4, Dec 2012. (Cited on page 15.)

[37] A. Belmonte, W. Kim, B. Chan, N. Heylen, A. Fantini, M. Houssa, M. Jurczak, and L. Goux. *"90nm WAl2O3TiWCu 1T1R CBRAM cell showing low-power, fast and disturb-free operation"*. Memory Workshop (IMW), 2013 5th IEEE International, pages 26–29, May 2013. (Cited on page 15.)

[38] M. Barci, G. Molas, A. Toffoli, M. Bernard, A. Roule, C. Cagli, J. Cluzel, E. Vianello, B. De Salvo, and L. Perniola. *"Bilayer Metal-Oxide CBRAM Technology for Improved Window Margin and Reliability"*. Memory Workshop (IMW), 2015 IEEE International, pages 1–4, May 2015. (Cited on pages 15, 44 and 58.)

[39] G. Palma, E. Vianello, O. Thomas, M. Suri, S. Onkaraiah, A. Toffoli, C. Carabasse, M. Bernard, A. Roule, O. Pirrotta, G. Molas, and B. De Salvo. *"Interface Engineering of Ag-GeS_2 -Based Conductive Bridge RAM for Reconfigurable Logic Applications"*. Electron Devices, IEEE Transactions on, 61(3):793–800, March 2014. (Cited on page 15.)

[40] F.M. Lee, Y.Y. Lin, W.C. Chien, D.Y. Lee, M.H. Lee, W.C. Chen, H.L. Lung, K.Y. Hsieh, and C.Y. Lu. *"A novel conducting bridge resistive memory using a semiconducting dynamic E-field moderating layer"*. VLSI Technology (VLSIT), 2013 Symposium on, pages T104–T105, June 2013. (Cited on page 15.)

[41] E. Vianello, O. Thomas, G. Molas, O. Turkyilmaz, N. Jovanovic, D. Garbin, G. Palma, M. Alayan, C. Nguyen, J. Coignus, B. Giraud, T. Benoist, M. Reyboz, A. Toffoli, C. Charpin, F. Clermidy, and L. Perniola. *"Resistive Memories for Ultra-Low-Power embedded computing design"*. Electron Devices Meeting (IEDM), 2014 IEEE International, pages 6.3.1–6.3.4, Dec 2014. (Cited on pages 15, 17, 18, 43 and 58.)

[42] A. Kawahara, K. Kawai, Y. Ikeda, Y. Katoh, R. Azuma, Y. Yoshimoto, K. Tanabe, Z. Wei, T. Ninomiya, K. Katayama, R. Yasuhara, S. Muraoka, A. Himeno, N. Yoshikawa, H. Murase, K. Shimakawa, T. Takagi, T. Mikawa, and K. Aono. *"Filament scaling forming technique and level-verify-write scheme with endurance over 107 cycles in ReRAM"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International, pages 220–221, Feb 2013. (Cited on page 15.)

[43] W. Kim, S.I. Park, Z. Zhang, Y. Yang-Liauw, D. Sekar, H.-S.P. Wong, and S.S. Wong. *"Forming-free nitrogen-doped AlOX RRAM with sub- μA programming current"*. VLSI Technology (VLSIT), 2011 Symposium on, pages 22–23, June 2011. (Cited on page 15.)

[44] Y.Y. Chen, R. Roelofs, A. Redolfi, R. Degraeve, D. Crotti, A. Fantini, S. Clima, B. Govoreanu, M. Komura, L. Goux, L. Zhang, A. Belmonte, Q. Xie, J. Maes, G. Pourtois, and M. Jurczak. *"Tailoring switching and endurance / retention reliab-*

ility characteristics of HfO2 / Hf RRAM with Ti, Al, Si dopants". VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on, pages 1–2, June 2014. (Cited on page 15.)

[45] A. Fantini, L. Goux, A. Redolfi, R. Degraeve, G. Kar, Y.Y. Chen, and M. Jurczak. *"Lateral and vertical scaling impact on statistical performances and reliability of 10nm TiN/Hf(Al)O/Hf/TiN RRAM devices"*. VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on, pages 1–2, June 2014. (Cited on page 15.)

[46] L. Goux, A. Fantini, A. Redolfi, C.Y. Chen, F.F. Shi, R. Degraeve, Y.Y. Chen, T. Witters, G. Groeseneken, and M. Jurczak. *"Role of the Ta scavenger electrode in the excellent switching control and reliability of a scalable low-current operated TiNTa2O5Ta RRAM device"*. VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on, pages 1–2, June 2014. (Cited on pages 15 and 17.)

[47] I. Valov, R. Waser, J.R. Jameson, and M.N. Kozicki. *"Electrochemical metallization memories–fundamentals, applications, prospects"*. Nanotechnology, 22(25):254003, 2011. (Cited on page 16.)

[48] M. Kund, G. Beitel, C.-U. Pinnow, T. Rohr, J. Schumann, R. Symanczyk, K.-D. Ufert, and G. Muller. *"Conductive bridging RAM (CBRAM): an emerging non-volatile memory technology scalable to sub 20nm"*. Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International, pages 754–757, Dec 2005. (Cited on page 17.)

[49] A. Benoist, S. Blonkowski, S. Jeannot, S. Denorme, J. Damiens, J. Berger, P. Candelier, E. Vianello, H. Grampeix, J.F. Nodin, E. Jalaguier, L. Perniola, and B. Allard. *"28nm advanced CMOS resistive RAM solution as embedded non-volatile memory"*. Reliability Physics Symposium, 2014 IEEE International, pages 2E.6.1–2E.6.5, June 2014. (Cited on pages 17, 20, 43 and 58.)

[50] M. Bocquet, D. Deleruyelle, H. Aziza, C. Muller, J.-M. Portal, T. Cabout, and E. Jalaguier. *"Robust Compact Model for Bipolar Oxide-Based Resistive Switching Memories"*. Electron Device, IEEE Transactions on, 61(3):674–681, March 2014. (Cited on page 18.)

[51] S. Masui, W. Yokozeki, M. Oura, T. Ninomiya, K. Mukaida, Y. Takayama, and T. Teramoto. *"Design and applications of ferroelectric nonvolatile SRAM and flip-*

flop with unlimited read/program cycles and stable recall". Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE 2003, pages 403–406, Sept 2003. (Cited on page 22.)

[52] J. Wang, Y. Liu, H. Yang, and H. Wang. *"A compare-and-write ferroelectric nonvolatile flip-flop for energy-harvesting applications"*. Green Circuits and Systems (ICGCS), 2010 International Conference on, pages 646–650, June 2010. (Cited on pages 22, 28, 30 and 31.)

[53] N. Sakimura, T. Sugibayashi, R. Nebashi, and N. Kasai. *"Nonvolatile Magnetic Flip-Flop for Standby-Power-Free SoCs"*. Solid-State Circuits, IEEE Journal of, 44(8):2244–2250, Aug 2009. (Cited on pages 22, 23, 25 and 32.)

[54] W. Zhao, E. Belhaire, and C. Chappert. *"Spin-MTJ based Non-volatile Flip-Flop"*. Nanotechnology, 2007. IEEE-NANO 2007. 7th IEEE Conference on, pages 399–402, Aug 2007. (Cited on pages 22, 25 and 29.)

[55] Y. Jung, J. Kim, K. Ryu, S.-O. Jung, J.P. Kim, and S.H. Kang. *"MTJ based nonvolatile flip-flop in deep submicron technology"*. SoC Design Conference (ISOCC), 2011 International, pages 424–427, Nov 2011. (Cited on pages 22, 25 and 27.)

[56] S. Yamamoto, Y. Shuto, and S. Sugahara. *"Nonvolatile flip-flop using pseudo-spin-transistor architecture and its power-gating applications"*. Semiconductor Conference Dresden-Grenoble (ISCDG), 2012 International, pages 17–20, Sept 2012. (Cited on page 22.)

[57] P. Wang, X. Chen, Y. Chen, H. Li, S. Kang, X. Zhu, and W. Wu. *"A 1.0V 45nm nonvolatile magnetic latch design and its robustness analysis"*. Custom Integrated Circuits Conference (CICC), 2011 IEEE, pages 1–4, Sept 2011. (Cited on page 22.)

[58] K. Huang and Y. Lian. *"A Low-Power Low-VDD Nonvolatile Latch Using Spin Transfer Torque MRAM"*. Nanotechnology, IEEE Transactions on, 12(6):1094–1103, Nov 2013. (Cited on pages 22 and 27.)

[59] K. Ryu, J. Kim, J. Jung, J.P. Kim, S.H. Kang, and S.-O. Jung. *"A Magnetic Tunnel Junction Based Zero Standby Leakage Current Retention Flip-Flop"*. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 20(11):2044–2053, Nov 2012. (Cited on page 22.)

[60] T. Na, K. Ryu, J. Kim, S.-O. Jung, J.P. Kim, and S.H. Kang. *"High-performance low-power magnetic tunnel junction based non-volatile flip-flop"*. Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, pages 1953–1956, June 2014. (Cited on pages 22 and 28.)

[61] D. Chabi, W. Zhao, E. Deng, Y. Zhang, N. Ben Romdhane, J.-O. Klein, and C. Chappert. *"Ultra Low Power Magnetic Flip-Flop Based on Checkpointing/Power Gating and Self-Enable Mechanisms"*. Circuits and Systems I: Regular Papers, IEEE Transactions on, 61(6):1755–1765, June 2014. (Cited on pages 22, 23, 24, 26, 28 and 41.)

[62] H. Cai, Y. Wang, W. Zhao, and L.A. de Barros Naviner. *"Multiplexing Sense-Amplifier-Based Magnetic Flip-Flop in a 28-nm FDSOI Technology"*. Nanotechnology, IEEE Transactions on, 14(4):761–767, July 2015. (Cited on pages 22 and 23.)

[63] K. Jabeur, G. Di Pendina, F. Bernard-Granger, and G. Prenat. *"Spin Orbit Torque Non-Volatile Flip-Flop for High Speed and Low Energy Applications"*. Electron Device Letters, IEEE, 35(3):408–410, March 2014. (Cited on page 22.)

[64] C.-M. Jung, K.-H. Jo, E.-S. Lee, H.M. Vo, and K.-S. Min. *"Zero-Sleep-Leakage Flip-Flop Circuit With Conditional-Storing Memristor Retention Latch"*. Nanotechnology, IEEE Transactions on, 11(2):360–366, March 2012. (Cited on pages 22, 28 and 30.)

[65] S. Onkaraiah, M. Reyboz, F. Clermidy, J. Portal, M. Bocquet, C. Muller, H. Hraziia, C. Anghel, and A. Amara. *"Bipolar ReRAM Based non-volatile flip-flops for low-power architectures"*. New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International, pages 417–420, June 2012. (Cited on page 22.)

[66] M.-F. Chang, C.-H. Chuang, M.-P. Chen, L.-F. Chen, H. Yamauchi, P.-F. Chiu, and S.-S. Sheu. *"Endurance-aware circuit designs of nonvolatile logic and nonvolatile sram using resistive memory (memristor) device"*. Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific, pages 329–334, Jan 2012. (Cited on pages 22, 23 and 30.)

[67] I. Kazi, P. Meinerzhagen, P.-E. Gaillardon, D. Sacchetto, Y. Leblebici, A. Burg, and G. De Micheli. *"Energy/Reliability Trade-Offs in Low-Voltage ReRAM-Based Non-Volatile Flip-Flop Design"*. Circuits and Systems I: Regular Papers, IEEE Transactions on, 61(11):3155–3164, Nov 2014. (Cited on pages 22 and 38.)

[68] R. Patel, S. Kvatinsky, E.G. Friedman, and A. Kolodny. *"Multistate Register Based on Resistive RAM"*. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 23(9):1750–1759, Sept 2015. (Cited on pages 22, 23 and 24.)

[69] D. Mahalanabis, V. Bharadwaj, H.J. Barnaby, S. Vrudhula, and M.N. Kozicki. *"A Nonvolatile Sense Amplifier Flip-Flop Using Programmable Metallization Cells"*. Emerging and Selected Topics in Circuits and Systems, IEEE Journal on, 5(2):205–213, June 2015. (Cited on pages 22, 23 and 27.)

[70] I. Kazi, P. Meinerzhagen, P.-E. Gaillardon, D. Sacchetto, A. Burg, and G. De Micheli. *"A ReRAM-based non-volatile flip-flop with sub-VT read and CMOS voltage-compatible write"*. New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International, pages 1–4, June 2013. (Cited on pages 25 and 27.)

[71] T. Na, K. Ryu, J. Kim, S.H. Kang, and S.-O. Jung. *"A comparative study of STT-MTJ based non-volatile flip-flops"*. Circuits and Systems (ISCAS), 2013 IEEE International Symposium on, pages 109–112, May 2013. (Cited on page 26.)

[72] M. Zwerg, A. Baumann, R. Kuhn, M. Arnold, R. Nerlich, M. Herzog, R. Ledwa, C. Sichert, V. Rzehak, P. Thanigai, and B.O. Eversmann. *"An 82 μA/MHz microcontroller with embedded FeRAM for energy-harvesting applications"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International, pages 334–336, Feb 2011. (Cited on page 30.)

[73] Y. Wang, Y. Liu, S. Li, D. Zhang, B. Zhao, M.-F. Chiang, Y. Yan, B. Sai, and H. Yang. *"A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops"*. ESSCIRC (ESSCIRC), 2012 Proceedings of the, pages 149–152, Sept 2012. (Cited on page 31.)

[74] Y. Wang, Y. Liu, Y. Liu, D. Zhang, S. Li, B. Sai, M.-F. Chiang, and H. Yang. *"A compression-based area-efficient recovery architecture for nonvolatile processors"*. Design, Automation Test in Europe Conference Exhibition (DATE), 2012, pages 1519–1524, March 2012. (Cited on page 31.)

[75] H. Koike, T. Ohsawa, S. Ikeda, T. Hanyu, H. Ohno, T. Endoh, N. Sakimura, R. Nebashi, Y. Tsuji, A. Morioka, S. Miura, H. Honjo, and T. Sugibayashi. *"A power-gated MPU with 3-microsecond entry/exit delay using MTJ-based nonvolatile flip-flop"*. Solid-State Circuits Conference (A-SSCC), 2013 IEEE Asian, pages 317–320, Nov 2013. (Cited on pages 31 and 32.)

[76] N. Sakimura, Y. Tsuji, R. Nebashi, H. Honjo, A. Morioka, K. Ishihara, K. Kinoshita, S. Fukami, S. Miura, N. Kasai, T. Endoh, H. Ohno, T. Hanyu, and T. Sugibayashi. *"10.5 A 90nm 20MHz fully nonvolatile microcontroller for standby-power-critical applications"*. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, pages 184–185, Feb 2014. (Cited on pages 32 and 33.)

[77] S. Khanna, S.C. Bartling, M. Clinton, S. Summerfelt, J.A. Rodriguez, and H.P. McAdams. *"An FRAM-Based Nonvolatile Logic MCU SoC Exhibiting 100Digital State Retention at* VDD*= 0 V Achieving Zero Leakage With < 400-ns Wakeup Time for ULP Applications"*. Solid-State Circuits, IEEE Journal of, 49(1):95–106, Jan 2014. (Cited on pages 32 and 33.)

[78] G. Besnard, X. Garros, F. Andrieu, P. Nguyen, W. Van Den Daele, P. Reynaud, W. Schwarzenbach, D. Delprat, K.K. Bourdelle, G. Reimbold, and S. Cristoloveanu. *"Superior performance and Hot Carrier reliability of Strained FDSOI nMOSFETs for advanced CMOS technology nodes"*. Solid State Device Research Conference (ESSDERC), 2014 44th European, pages 226–229, Sept 2014. (Cited on page 36.)

[79] N. Jovanovic, E. Vianello, O. Thomas, B. Nikolic, and L. Naviner. *"Design insights for reliable energy efficient OxRAM-based flip-flop in 28nm FD-SOI"*. SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2015 IEEE, pages 1–2, Oct 2015. (Cited on page 41.)

[80] A. Redolfi, L. Goux, N. Jossart, F. Yamashita, E. Nishimura, D. Urayama, K. Fujimoto, T. Witters, F. Lazzarino, and M. Jurczak. *"A novel CBRAM integration using subtractive dry-etching process of Cu enabling high-performance memory scaling down to 10nm node"*. VLSI Technology (VLSI Technology), 2015 Symposium on, pages T134–T135, June 2015. (Cited on pages 44 and 58.)

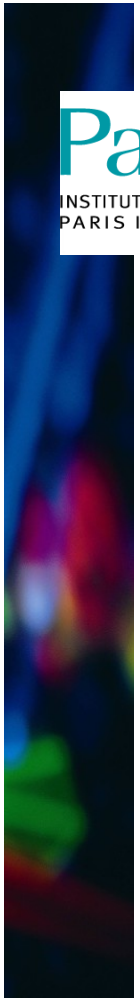
[81] N. Jovanovic, O. Thomas, E. Vianello, J.-M. Portal, B. Nikolic, and L. Naviner. *"OxRAM-based non volatile flip-flop in 28nm FDSOI"*. New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International, pages 141–144, June 2014. (Cited on page 46.)

[82] B. Serneels, M. Steyaert, and W. Dehaene. *"A High speed, Low Voltage to High Voltage Level Shifter in Standard 1.2V 0.13 μ CMOS"*. Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference on, pages 668–671, Dec 2006. (Cited on page 46.)

[83] N. Jovanovic, O. Thomas, E. Vianello, B. Nikolic, and L. Naviner. *"Design Considerations for Reliable OxRAM-based Non-Volatile Flip-Flops in 28nm FD-SOI Technology"*. Internatial Symposium on Circuits and Systems (ISCAS), IEEE, 2016. (Cited on page 47.)

[84] J.-P. Noel, O. Thomas, M. Jaud, O. Weber, T. Poiroux, C. Fenouillet-Beranger, P. Rivallin, P. Scheiblin, F. Andrieu, M. Vinet, O. Rozeau, F. Boeuf, O. Faynot, and A. Amara. *"Multi- V_2 UTBB FDSOI Device Architectures for Low-Power CMOS Circuit"*. Electron Devices, IEEE Transactions on, 58(8):2473–2482, Aug 2011. (Cited on page 58.)

[85] A. Chandrakasan, W. Bowhill, and F. Fox. *Register Files and Caches*, pages 284–308. Wiley-IEEE Press, 2001. (Cited on page 73.)



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0023

EDITE ED 130

Doctorat ParisTech

RÈSUME DE THÈSE

pour obtenir le grade de docteur délivré par

Télécom ParisTech

Spécialité “Communications et Electronique”

présentée et soutenue publiquement par

Natalija JOVANOVIĆ

23 Mars 2016

Conception d’éléments séquentiels non volatiles pour la conception de circuits numériques fortement intégrés

Directeur de thèse : **Lirida ALVES DE BARROS NAVINER**
Co-encadrement de la thèse : **Olivier THOMAS, Borivoje NIKOLIĆ**

Jury

M. Jean-Michel PORTAL, IM2NP, Aix-Marseille Université,	Président
M. Philippe CANDELIER, STMicroelectronics, Crolles	Examinateur
M. Lionel TORRES, LIRMM, Montpellier	Rapporteur
M. Jean-Didier LEGAT, UCL, Louvain	Rapporteur
Mme. Lirida ALVES DE BARROS NAVINER, Telecom ParisTech, Paris	Directeur de thèse
M. Olivier THOMAS, CEA-LETI, Grenoble	Co-encadrement
M. Borivoje NIKOLIĆ, BWRC, Berkeley	Co-encadrement

THÈSE

Télécom ParisTech
école de l’Institut Mines Télécom – membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

1. Introduction

L'évolution de la technologie, en particulier la miniaturisation des puces et des capteurs électroniques, les améliorations apportées à la communication sans fil et leur prix relativement faible, ont entraîné une énorme expansion de petits appareils électroniques. En effet, les périphériques connectés (intelligents), appelés Internet of Things (IoT), peuvent trouver des applications dans différents dans la vie moderne et, dans les prochaines années, représenteront une part considérable du nombre total de périphériques.

Une caractéristique importante de ces systèmes est leur nature normalement désactivée, ce qui signifie qu'ils passent la plus grande partie de leur vie en mode veille, avec uniquement de courtes périodes d'activité.

La réduction des dimensions physiques du transistor CMOS augmente le courant de fuite statique [4]. Ainsi, les périodes de veille des microcontrôleurs peuvent constituer la composante dominante de la consommation totale d'énergie d'un appareil connecté. Dans la plupart des cas, les périphériques intelligents sont alimentés par batterie et, pour des raisons pratiques, ils devraient être conçus pour supporter une longue durée de vie opérationnelle sans avoir besoin de remplacement de la batterie. Par conséquent, l'un des principaux défis de la conception d'IoT est de créer des systèmes éconergétiquement efficaces qui peuvent fonctionner à de faibles budgets de puissance. En particulier, **il y a une forte demande pour une ultra-faible consommation durant les phases de veille, sans dégrader les performances en mode actif.** De plus, la conception doit être en **mesure de supporter** les **défaillances soudaines de l'appareil,** qui peuvent être fréquentes à cause de l'instabilité des sources d'alimentation externes en cas d'alimentation basée sur la récuperation d'énergie.

Il existe de nombreuses solutions pour atteindre un rendement énergétique élevé en mode veille et en mode actif (par exemple, avec une tension dynamique et mise à l'échelle de fréquence, horloge et récupération d'énergie, entre autres). Ces solutions sont implémentées dans des microcontrôleurs commerciaux actuels sous la forme de différents modes de fonctionnement et de sommeil avec une gestion d'énergie complexe.

Dans la mise en oeuvre efficace de calculateurs devant être normalement en mode veille, un rôle important consiste à l'optimisation du portionnage d'énergie et, par conséquent, à **l'organisation des registres et de la mémoire responsables pour la sauvegarde** de **l'état du système** [7].

Pendant le mode de veille, dans les architectures MCU classiques, les registres volatiles et la mémoire restent alimentés, de préférence à tension réduite, tandis que le reste du circuit est déconnecté (Figure 1.1a, haut). Pour une économie plus significative, les registres de retention des données peuvent être utilisés, et la source toujours passante est appliquée uniquement à la partie où les données sont préservées, comme dans un latch "ballon" attaché à l'étage esclave secondaire [8] (cf. Figure 1.1a, en bas). Bien que cette solution offre des transitions très rapides entre mode actif / dormant, la dissipation de puissance toujours présente dans une bascule CMOS

peut atteindre des niveaux non négligeables dans les modes de sommeil longue durée. Alternativement, en mode veille, les données peuvent être conservées dans des mémoires externes non-volatiles, annulant complètement la dissipation de puissance (Figure 1.1b). Cependant, les transferts de données cœur-bus-mémoire sont lents et ont un impact négatif dans la consommation d'énergie, surtout lorsque le module non volatile est basé sur la mémoire Flash. Par conséquent, seule une quantité limitée de données et d'état du processeur est maintenue, conformément au temps de mise en veille / réveil et aux besoins énergétiques.

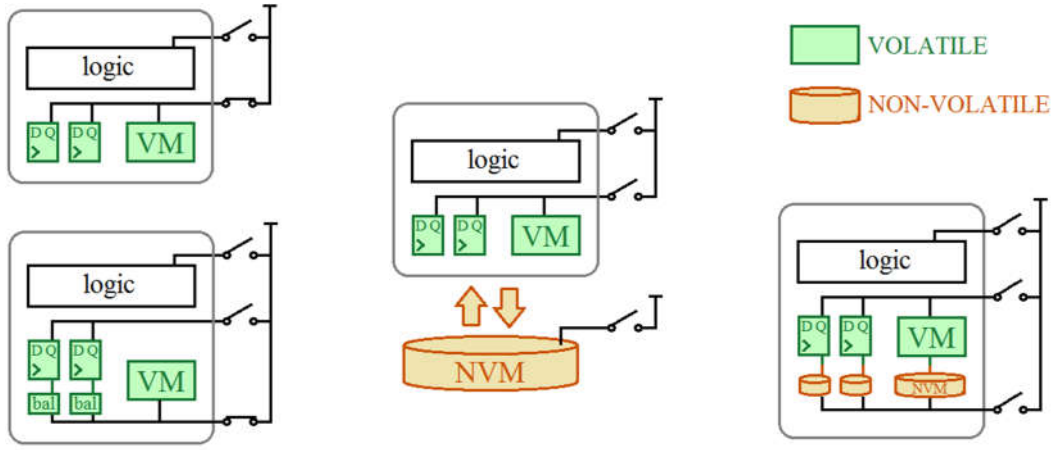


Figure 1.1: Enregistrement du contexte en mode veille dans les systèmes informatiques normalisés: (a) dans les bascules volatiles, (b) dans la mémoire externe non-volatile, (c) dans les bascules non-volatiles.

Aujourd'hui, avec les recherches approfondies sur les nouveaux types de mémoires non-volatiles (NVM), telles que des **mémoires résistives à accès aléatoire (ReRAMs),** d'autres possibilités sont apparues. Initialement envisagées comme une solution de remplacement de Flash / DRAM, elles peuvent s'adapter au concept précédent en tant que mémoires externes non-volatiles, ce qui conduit à de meilleures caractéristiques vitesse/consommation du système. Plus important encore, elles apportent la possibilité d'une intégration facile avec la technologie CMOS, ce qui permet d'intégrer la fonctionnalité non-volatile directement dans la logique du processeur, comme le montre la figure 1.1c. Pour l'instant, les NVM émergentes sont encore trop lentes et consommatrices de puissance pour remplacer complètement les SRAM. Cependant, elles peuvent être combinées avec ces dernières (par exemple, NV-SRAM [9]). Ainsi, les mémoires volatiles rapides peuvent satisfaire les exigences critiques en matière de performance, tandis que les NVM peuvent assurer un stockage sans consommation d'énergie en mode veille. Les éléments essentiels de cette approche, utilisée pour sauvegarder l'ensemble du contexte du microcontrôleur et permettre des transitions rapides entre les modes, sont **des bascules non-volatiles (NVFF).**

De nombreuses solutions à bascule non-volatile basées sur plusieurs technologies NVM sont reportées dans la littérature. Parmi elles, les ReRAM basées sur NVFF ont été implémentées dans des nœuds CMOS de 90 nm ou plus anciens. Cependant, dans les processus CMOS, la

tension de fonctionnement maximale devient inférieure aux tensions d'initialisaiton et de programmation requises pour les ReRAM. Cet écart de tension génère des problèmes de fiabilité dans les conceptions qui utilisent des transistors à oxyde de grille mince. En outre, le choix de la topologie NVFF est fortement affecté par les caractéristiques des périphériques NVM, telles que les tensions de programmation, les courants, la fenêtre de mémorisation, etc. Ainsi, les architectures NVFF proposées jusqu'à présent et utilisant des périphériques NVM autres que ReRAM ne peuvent pas être directement appliquées aux cellules ReRAM .

Objectif et contribution de la thèse

Cette thèse explore la co-intégration des ReRAM avec des technologies CMOS de 28 nm et moins dans la conception de bascules non volatiles. La variabilité du procédé et les courants de fuite en mode actif, qui sont les points faibles de noeuds CMOS avancés, sont réduits au minimum en utilisant la technologie FDSOI [10], ce qui permet également de hautes performances et un fonctionnement à basse tension. Deux nouvelles solutions de conception pour la programmation de dispositifs ReRAM basées sur oxyde de grille mince sont proposées, puis appliquées dans une architecture à double tension à deux NVFF ReRAM. Afin de diminuer le surcoût en surface et réduire la consommation, une nouvelle architecture ReRAM est également présentée. Les bascules NVFF sont conçues comme des cellules standard pour assurer la compatibilité avec le flot de conception numérique. L'une des architectures NVFF est par ailleurs utilisée dans la mise en œuvre d'une version non-volatile du microprocesseur ARM Cortex-M0+.

Parallèlement aux nouvelles solutions de conception, les caractéristiques électriques des technologies ReRAM disponibles sont explorées afin de choisir l'architecture la plus appropriée et de l'optimiser en conséquence. Lors de la définition des valeurs d'état résistif et des conditions de programmation optimales de ReRAM, l'accent est principalement dans: (i) l'amélioration de l'endurance et la réduction de la puissance de programmation, (ii) la réalisation d'une récupération de données fiable et à basse tension. Cette dernière tâche , rendue difficile à cause de la variabilité des dispositifs ReRAM et CMOS, est réalisée par l'évaluation statistique des différentes architectures bascules. Les bascules NVFF sont mises en œuvre dans une technologie CMOS FDSOI de 28 nm et comparées à une bascule d'une bibliothèque standard et à une bascule pour data retention, afin d'évaluer l'impact de l'ajout de non-volatilité sur le mode bascule normal et d'estimer quand il est opportun d'employer les NVFF au lieu de solutions volatiles standards. Enfin, pour obtenir une densité plus élevée sans affecter la robustesse des opérations non-volatiles, un nouveau file register non volatile (NVRF : non volatile register file) multi-port basé sur l'une des solutions NVFF est proposé.

2. Mémoires non-volatiles

En fonction de leur capacité à retenir les données lors de la déconnexion de la source d'alimentation, des mémoires à semi-conducteurs peuvent être séparées en deux catégories (Figure 2.1). **Les mémoires volatiles,** qui nécessitent une alimentation pour préserver l'état stocké, sont largement présentes dans les systèmes informatiques modernes sous forme de SRAM et DRAM. Dans la classe de **mémoires non-volatiles,** qui ne perdent pas les données lorsque l'alimentation est coupée, la plus dominante est aujourd'hui la mémoire Flash. Elle se caractérise par un faible coût de fabrication et une densité plus élevée que la DRAM. Cependant, elle a une vitesse de lecture/écriture beaucoup plus faible et son endurance est limitée à environ 10^6 cycles. En outre, la mémoire Flash nécessite des tensions de programmation beaucoup plus élevées que la tension de fonctionnement des dispositifs CMOS, ce qui entraîne une forte consommation pour l'écriture et pose une forte limitation pour son intégration avec la logique CMOS. Ainsi, dans les systèmes actuels, les mémoires Flash sont utilisées comme stockage secondaire pour le code et les données.

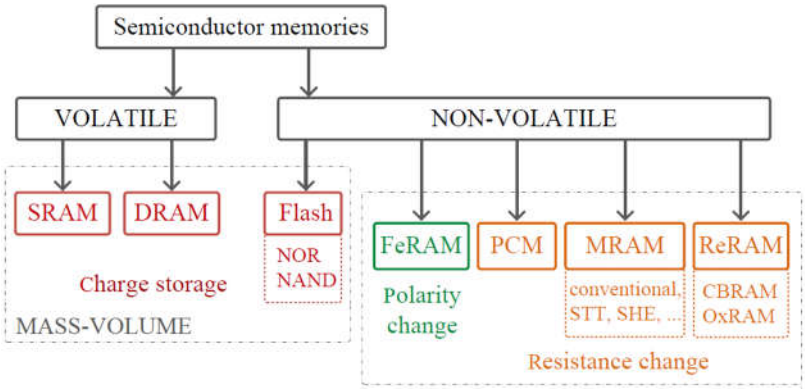


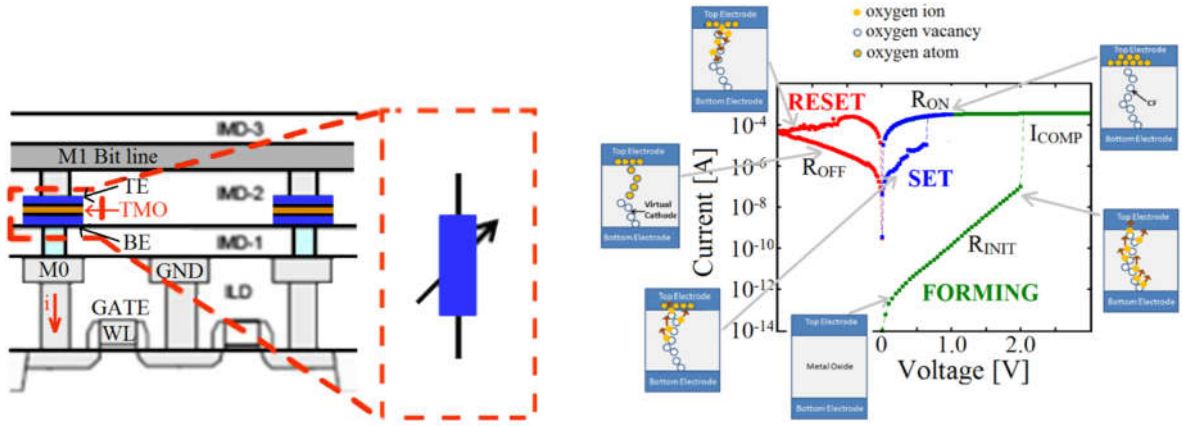
Figure 2.1: Mémoires à semi-conducteurs.

Afin d'améliorer l'intégration et la densité de la mémoire, de nombreuses solutions ont été étudiées dans les dernières années [11]. Parmi les différentes technologies NVM, les **mémoires résistives à accès aléatoire (ReRAMs)** sont particulièrement intéressantes, car elles se basent sur des tensions de fonctionnement faibles et permettent programmation et lecture rapides, ont une faible consommation d'énergie et se prêtent à la réduction d'échelle. En outre, elles présentent une endurance adéquate pour les applications IoT. En raison de leur structure simple et de leur compatibilité avec le processus de back-end-line CMOS, les ReRAM offrent l'intégration la plus simple avec la logique numérique CMOS à faible coût.

ReRAM contexte et caractérisation électrique

Une ReRAM est une structure métal-isolant-métal à deux terminaux (Figure 2.2 (a)) qui présente une résistance de commutation réversible. L'application de tensions/courants appropriés à travers les électrodes provoque des réactions qui forment ou suppriment le(s) filament(s) conducteur(s)

entre deux bornes, entraînant un changement de résistance (Figure 2.2 (b)). La création d'un chemin conducteur dans l'isolant est obtenue pendant l'opération SET, et il en résulte un faible état résistif du dispositif (R_{ON}). L'absence de piste conductrice est obtenue pendant l'opération de RESET et conduit à un état hautement résistif (R_{OFF}). Le premier cycle de programmation, appelée FORMING, nécessite généralement une tension de seuil supérieure à celle de l'opération SET car la résistance du dispositif frais a une valeur initiale très élevée ($R_{INIT} >> R_{OFF}$). Au cours de SET et FORMING, il est important de limiter le courant à travers le dispositif en définissant le courant de conformité (ou courant de programmation, I_{COMP}) pour empêcher la dégradation et la défaillance du dispositif.



(a) (b)

Figure 2.2: (a) L' intégration ReRAM BEOL, (b) Caractéristique IV quasi-statique obtenue à partir des données expérimentales OxRAM, et l'illustration du processus de commutation dans les OxRAM binaires simples [32].

En fonction des matériaux utilisés, le mécanisme de création/rupture d'un chemin conducteur peut être différent [23]. Ainsi, deux types de ReRAM se distinguent : **pont conducteur** de **mémoire à accès aléatoire (CBRAM),** également connu comme cellule de métallisation programmable (PMC), et **mémoire à accès aléatoire métal-oxyde (OxRAM).** Cependant, même au sein d'une famille ReRAM, un large éventail de caractéristiques électriques est rencontré, comme l'indique la Figure 2.3 qui montre le rapport des résistances (R_{OFF}/ R_{ON}) par rapport à la performance en terme de nombre de cycles pour différentes technologies.

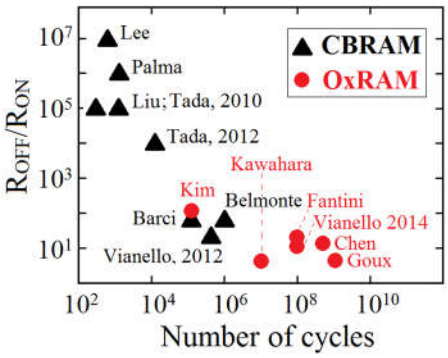
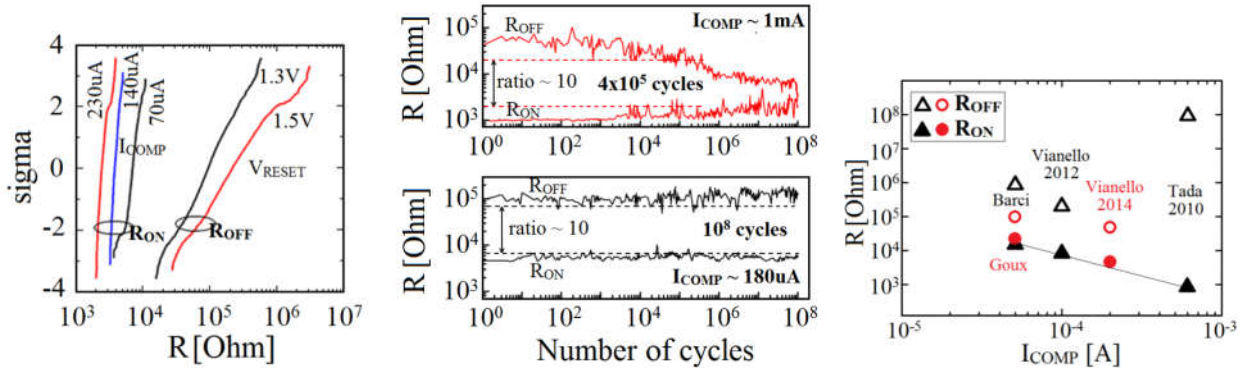


Figure 2.3: Fenêtre de mémorisation en fonction de l'endurance pour CBRAM [33-40] et OxRAM [41-46].

En général, pour une technologie ReRAM donnée, il existe un compromis entre la tension de programmation (V_{SET}, V_{RESET}, $V_{FORMING}$), le courant de programmation (I_{COMP}), le rapport de résistances (R_{OFF}/R_{ON} ratio), l'endurance du dispositif et la consommation d'énergie. Par conséquent, il est important de prendre en compte ces relations dans la conception basée sur ReRAM.

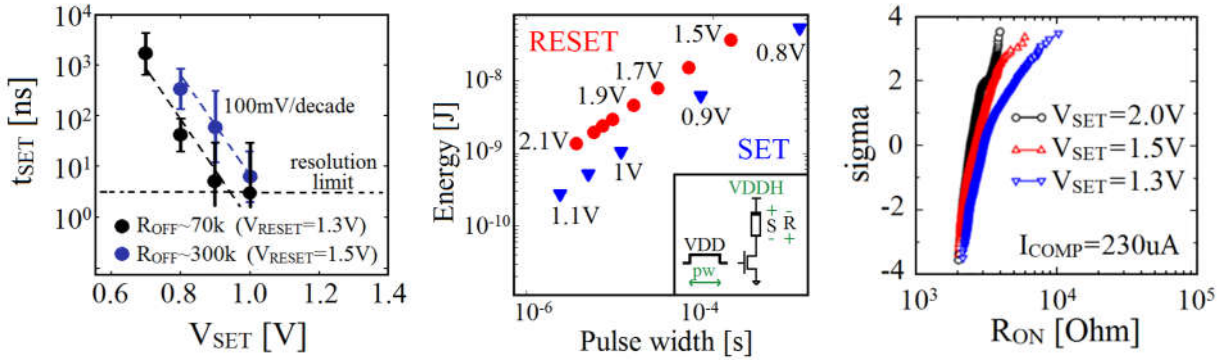
Par exemple, les valeurs R_{ON} et R_{OFF} peuvent être ajustées en réglant le courant de conformité SET et la tension RESET, respectivement (Figure 2.4 (a)). Pour **réduire la consommation d'énergie de programmation,** il est souhaitable **d'utiliser de faibles courants de conformité.** En outre, le faible I_{COMP} se traduit par une **meilleure endurance,** aussi I_{COMP} élevé conduit progressivement à la dégradation du rapport de résistances en diminuant R_{OFF} (Figure 2.4 (b)). Malheureusement, la réduction de du courant de conformité augmente R_{ON} (Figure 2.4 (c)), ce qui réduit la fenêtre de mémorisation. Toutefois, cette fenêtre de mémorisation inférieure peut être compensée par un **V_{RESET} plus élevé** permettant d'augmenter R_{OFF}.



(a) (b) (c)

Figure 2.4: (a) La distribution de la valeur de résistance par rapport à des conditions de programmation (R_{ON} vs I_{COMP}, R_{OFF} vs V_{RAZ}), pour les cellules TiN / Ti / HfO / TiN OxRAM, (b) Test d'endurance (pulse cycling) pour différents courants de conformité SET [49], (c) Valeurs RON et ROFF en fonction du courant de programmation pour différentes piles ReRAM.

Comme l'une des caractéristiques essentielles des cellules ReRAM est la dépendance exponentielle entre le temps de commutation et la tension appliquée (Figure 2.5 (a)), la **maximisation** des **tensions de programmation permet** de **réduire la largeur d'impulsion de programmation,** et ainsi **réduire l'énergie de programmation** (Figure 2.5 (b)). En outre, des tensions de programmation élevées surmontent la variabilité du dispositif dans les grands tableaux de mémoire (Figure 2.5 (c)). Enfin, les dispositifs ReRAM ont besoin de l'opération de FORMING, qui est réalisée avec une tension supérieure à V_{SET} et V_{RESET}. Ainsi, les **cellules ReRAM nécessitent** des **tensions de programmation supérieures aux tensions de fonctionnement typiques des technologies CMOS avancées.**

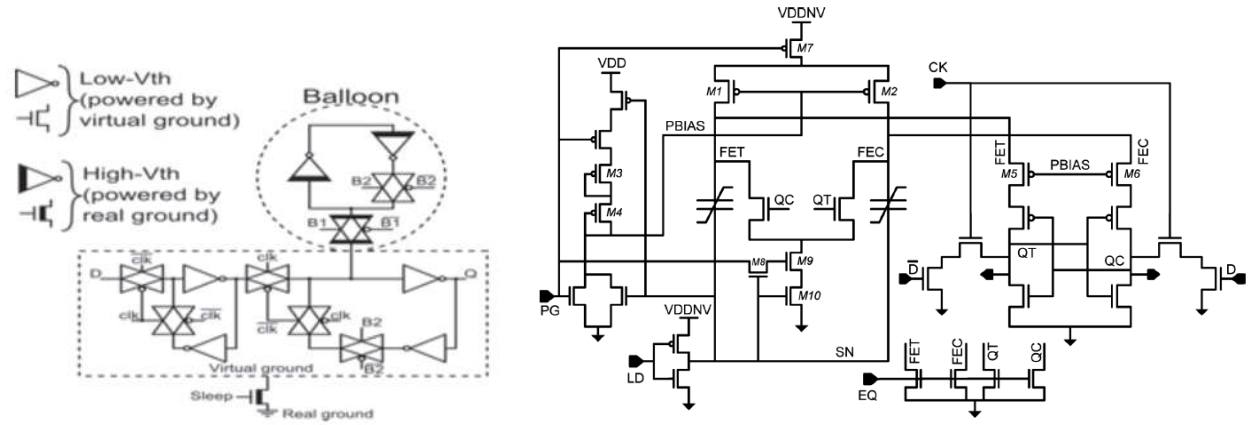


(a) (b) (c)

Figure 2.5: Impact de la tension de programmation: (a) la dépendance exponentielle entre le temps de commutation et V $_{SET}$ [41], (b) l'énergie simulée SET / RESET vs. temps de commutation pour différentes VDDH, (c) distribution de R pour différents V $_{SET,}$ en utilisant largeur d'impulsion et courant de programmation fixes.

3. Travaux connexes

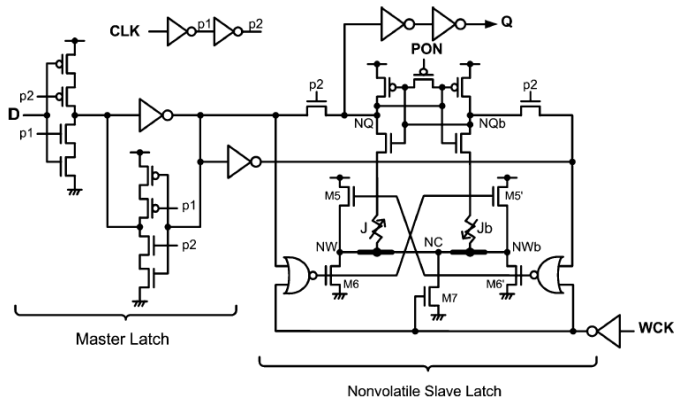
Avec la mise à l'échelle de la technologie CMOS, la consommation due aux courants de fuite pendant le mode inactif du système peut prendre une part très importante de la consommation totale [4]. Afin de réduire la consommation statique, la bascule pour retention de données [8] a été mise au point pour remplacer les bascules standards dans l'unité de traitement (Figure 3.1 (a)). Le latch à faible retention attaché à la bascule peut prendre son contenu (opération de stockage ou « store »), le conserver pendant le mode veille et le renvoyer à la sortie du mode veille (opération de restauration ou « restore »). Par conséquent, l'alimentation de la bascule principale peut être désactivée pendant le sommeil, tandis que l'alimentation du latch de retention est maintenue à un niveau réduit.



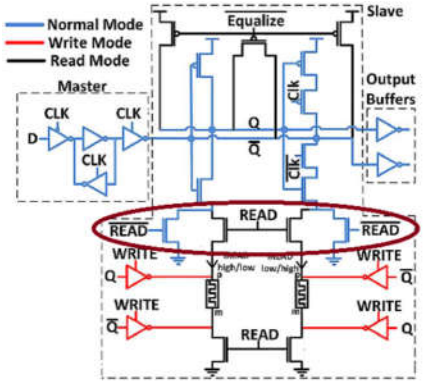
(a) (b)

Figure 3.1 – (a) Bascule de rétention de données volatile (latch "ballon") [8]; (b) Exemple de bascule non-volatile à base de FeRAM [14].

Pour atteindre une consommation encore plus faible, une bascule non-volatile semble être une bonne substitution pour la bascule de retention de données, car elle offre un mode de veille à consommation nulle, tout en permettant des transitions store/restore relativement rapides. Récemment, diverses solutions NVFF créées en ajoutant des NVM à des bascules ont été proposées. Du fait d'être la plus mature des nouvelles mémoires non-volatiles, la FeRAM est utilisée dans plusieurs NVFFs fabriquées [14, 51, 52] (par exemple, Figure 3.1 (b)). D'autres conceptions basées sur MRAM ont également été introduites: NVFF en utilisant un MTJ [53], de nombreux NVFFs employant STT-MTJs [54-62], et autres comptant sur SHE MRAM [21, 63] (par exemple, Figure 3.2 (a)). Enfin, des NVFFs à base de ReRAM ont été décrits en [64-69] (par exemple, Figure 3.2 (b)).



(a)



(b)

Figure 3.2: (a) Exemple de bascule non-volatile à base de MRAM [53], (b) Exemple de bascule non-volatile à base de ReRAM [67].

Une comparaison quantitative et pertinente des solutions NVFF existantes est impossible, en raison de l'insuffisance de résultats publiés, d'une large gamme de technologies CMOS et NVM utilisées, de différentes conditions de simulation, etc. Cependant, la littérature présentée indique que le choix de la topologie NVFF est fortement influencé par les caractéristiques des dispositifs NVM et leur compatibilité avec les conditions de fonctionnement CMOS. Les différentes facettes de la conception et de la technologie, définissent alors ensemble les caractéristiques NVFF globales (Figure 3.3).

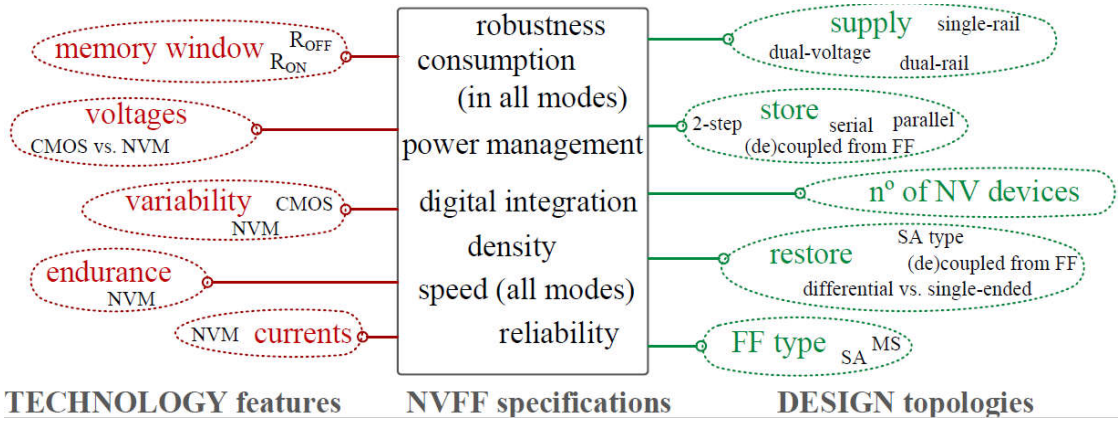


Figure 3.3: Les caractéristiques NVFF dépendent de la technologie et de la conception.

Dans le cas de la conception basée sur les ReRAM dans un nœud CMOS avancé, les conditions de tension de programmation impliquent une configuration à double tension, ce qui nécessite également un examen attentif de la gestion de la consommation et du routage du réseau d'alimentation. D'autre part, les tensions de programmation requises réduisent la possibilité de problèmes de lecture. Cela permet l'implémentation conjointe du circuit de restauration et de la bascule, ce qui entraîne une solution dense avec de faibles pénalités de temps. Généralement, la large fenêtre de mémorisation des ReRAMs permet une opération de restauration fiable, offrant ainsi plus de souplesse de conception. Par exemple, il est possible d'obtenir une architecture NVFF avec un seul dispositif NVM, ce qui est bénéfique en termes de surface et consommation pour la programmation. En outre, les problèmes de variabilité des technologies CMOS et ReRAM peuvent être surmontés en choisissant judicieuxement le circuit sense amplifier.

La bonne compatibilité des ReRAM avec la technologie CMOS facilite son intégration dans un processeur non-volatile. Parmi les approches précédentes, celle basée sur le remplacement de la bascule traditionnelle par une bascule NVFF s'impose comme la solution la plus simple. Elle nécessite un temps de conception faible, tout en offrant des transitions rapides de veille/réveil et présentant un faible impact sur le mode de fonctionnement bascule « normal ». De plus, les surfaces requises peuvent être réduites en organisant des NVFF dans un réseau régulier avec des circuits de stockage/restauration partagés (cela est possible, par exemple, dans les register file).

4. Solutions de conception pour flip-flops non-volatiles

Défis

Les principaux défis pour la mise en œuvre des circuits basés sur ReRAM dans les technologies CMOS fortement submicroniques proviennent des exigences de programmation. Tout d'abord, les résultats ont montré que **les tensions de programmation pour** les **mémoires ReRAM sont plus élevées que les tensions de fonctionnement typiques des technologies CMOS avancées.**

Des tensions SET/RESET supérieures à 1,5 V sont nécessaires pour améliorer les retards dans les grands réseaux de mémoire, surtout lorsqu'ils fonctionnent à des courants de programmation faibles, alors que l'opération FORMING nécessite des tensions encore plus élevées (> 2,5 V). De plus, en raison de la dépendance exponentielle entre le temps de commutation et la tension, l'augmentation de la tension entraîne une consommation plus faible que l'augmentation de la largeur d'impulsion. En second lieu, les mesures mettent en évidence que les **courants faibles de programmation sont préférés afin de réduire** la **consommation d'énergie et augmenter** l'**endurance de** du dispositif**,** au prix d'une augmentation de la variabilité et d'une réduction de la fenêtre de mémorisation. Troisièmement, **il y a un écart entre les conditions** de **programmation SET et RESET.** Afin de cibler des valeurs spécifiques R_{ON} et R_{OFF} ayant les mêmes largeurs d'impulsions, des tensions différentes aux bornes du dispositif peuvent être nécessaires. Enfin, **il est nécessaire d'améliorer la fiabilité des circuits CMOS** puisque les tensions de programmation ReRAM peuvent provoquer une dégradation des transistors CMOS par porteurs chauds. Par exemple, dans la technologie FDSOI 28nm la tension $V_{DS} = V_{GS} = 1,6V$ se traduirait par une dégradation de 10% du courant de drain après moins de 1s de stress accumulé [78], conduisant à une mauvaise endurance du système. En outre, la tension de FORMING peut provoquer la rupture d'oxyde des transistors.

En ce qui concerne la mise en œuvre de bascules non-volatiles à base de ReRAM, il est important **de garantir que la propriété de non-volatilité se fait avec un impact minimal sur les performances et la consommation de la bascule.** Un autre défi consiste à **assurer le rendement de NVFF suffisant même avec** les **technologies** à **faible fenêtre de mémorisation et à basse tension.** Compte tenu de la forte variabilité des technologies ReRAM, par ailleurs aggravée par les faibles courants de programmation, la fenêtre de mémorisaiton R_{OFF}/R_{ON} peut être dégradée, ce qui provoque l'échec de l'opération de restauration. Pour la pleine compatibilité avec le flot de conception numérique, il est nécessaire que les **NVFFs soient** disponibles en tant que **cellules standards en utilisant** des transistors à oxyde de grille mince**.** Enfin, pour l'intégration plus facile des NVFF dans les grands systèmes, **l'opération FORMING doit être exécutée en une seule étape et doit être indépendante de la valeur de la bascule.**.

Solution proposées

Dans cette thèse, quatre cellules flip-flop non-volatiles qui co-intégrent avec succès les technologies ReRAM et FDSOI de 28 nm sont présentées. Ces cellules comprennent deux architectures de restauration différentes (restauration 1R et 2R) et deux architectures de stockage différentes (LM et CM). Toutes les bascules NVFF sont construites en attachant la partie non-volatile à la bascule standard maître-esclave (Figure 4.1) et se composent de trois parties:
- le cœur de la bascule MSFF adaptée (c'est-à-dire, parties MASTER et SLAVE), connecté à
- un bloc non-volatile (NV) pour stocker et restaurer les données de la bascule, et
- un bloc logique de contrôle (LOGIC) qui gère les opérations de stockage, de restauration ou de FORMING en plus des opérations classiques de la bascule.

Les cellules utilisent deux rails d'alimentation. Les blocs logiques de base et de commande sont alimentés à la tension nominale de fonctionnement CMOS (VDD), tandis que le bloc NV fonctionne à une tension plus élevée (VDDH) pour satisfaire les exigences de programmation du dispositif ReRAM (FORMING, SET et RESET). Par conséquent, l'empilage des transistors requis pour la protection haute tension est implémenté uniquement dans la zone soumise à VDDH.

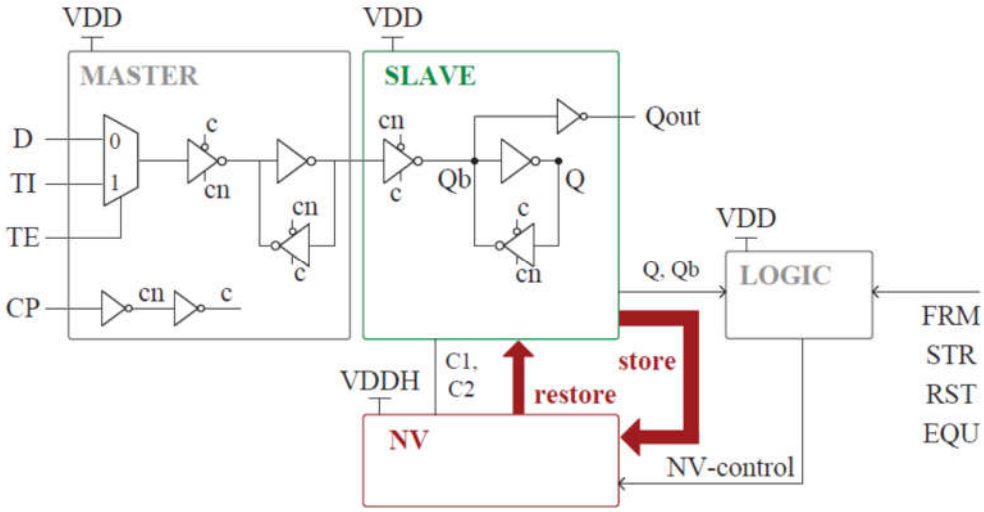
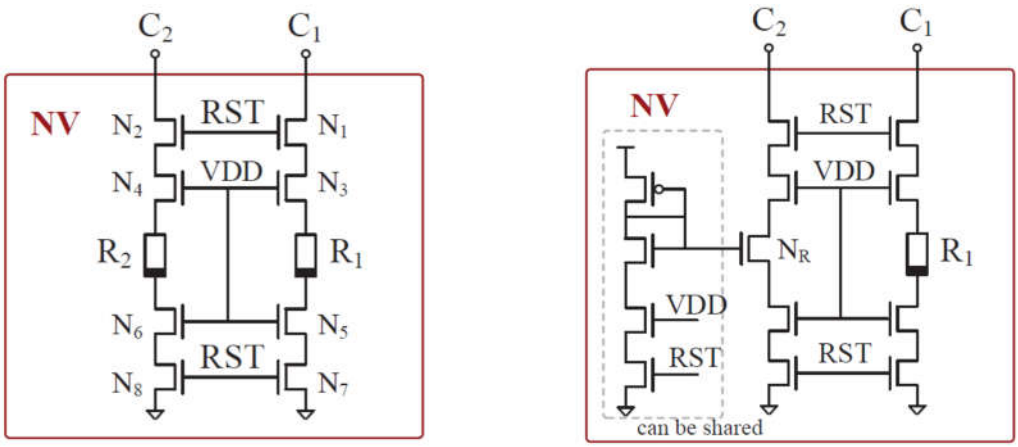


Figure 4.1: Schéma de principe de NVFF

Les NVFF sont conçus pour fonctionner en cinq modes:
- le mode **actif** est le mode de fonctionnement FF normal, pendant lequel la partie NV est désactivée;
- le mode **store** permet d'enregistrer les données de la bascule dans le NVM;
- mode **veille** indique que toutes les alimentations sont éteintes et peut être activé après la sauvegarde des données;
- le mode de **restauration** permet de récupérer le contexte enregistré dans la NV et il est activé après le retour du mode veille;
- le mode **initialisation** correspond à l'opération FORMING des ReRAMs et est activé une fois, avant la première utilisation de NVFF.

Opération de restauration

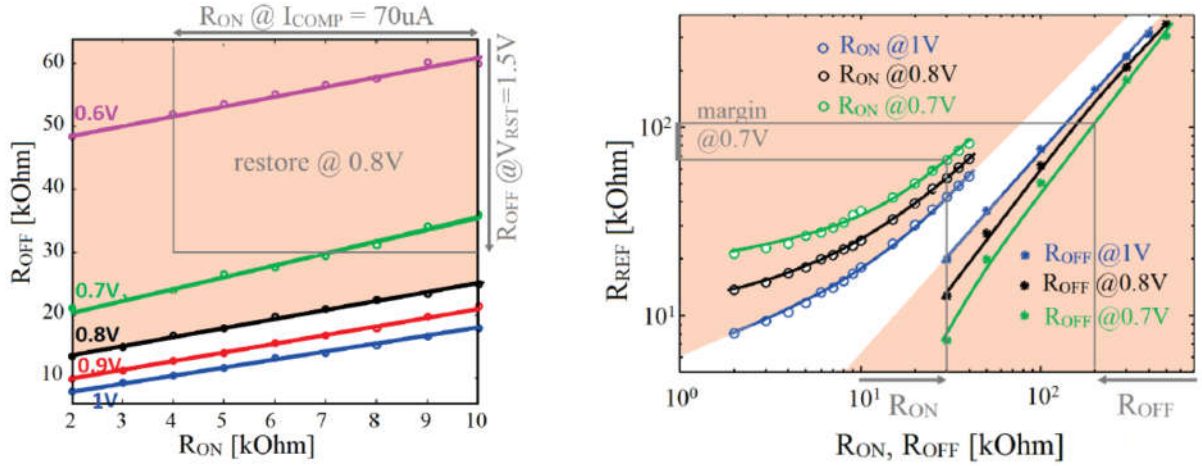
Dans les solutions NVFF 2R (Figure 4.2 (a)), l'opération de restauration est exécutée en tant que détection de différentiel de deux ReRAMs programmées pour les états opposés (R_{ON} et R_{OFF}), alors que le latch esclave est utilisé en tant qu'amplificateur de lecture. Afin de limiter le surcoût en surface et réduire la consommation pour la mémorisation, une nouvelle architecture NVFF 1R dérivée de NVFF 2R est proposée (Figure 4.2 (b)), où un seul dispositif ReRAM programmable R_1 est comparé à la résistance de référence basée sur un miroir de courant.



(a) (b)

Figure 4.2: Bloc NV (opération de restauration) de (a) 2R NVFF, (b) 1R NVFF.

Les architectures proposées sont mises en œuvre en technologie FDSOI de 28 nm, et leur robustesse est estimée en extrayant le taux binaire d'erreur (BER) de l'opération de restauration. Ceci est fait pour diverses valeurs de R_{ON} et R_{OFF}, dans la plage de tension [0.6V, 1V], en tenant compte des variations du process CMOS. Ensuite, la distribution de la résistance des technologies ReRAM disponibles est prise en compte, et les conditions de programmation qui assurent le rendement 3σ à la tension de restauration la plus basse sont estimées (Figure 4.3).



(a) (b)

Figure 4.3: (a) Fenêtre de mémorisation (R_{OFF} vs R_{ON}) extraite avec un rendement de 3σ pour 2R NVFFs à différentes tensions d'alimentation, ainsi que les conditions de programmation pour OxRAM Stack [41, 49] qui assurent une restauration réussie à 0,8V. (b) Résistance de référence (R_{REF}) extraite avec un rendement de 3σ pour 1R NVFFs à différentes tensions d'alimentation, et CBRAM [38, 80] avec conditions de programmation qui fournissent une marge suffisante pour la mise en oeuvre de R_{REF} pour restaurer à 0.7V.

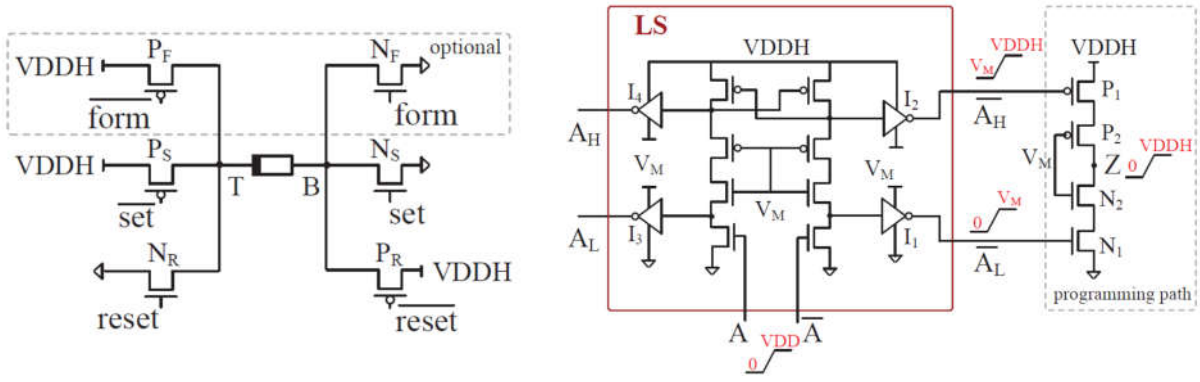
Les cellules NVFF 2R sont robustes et compatibles avec les technologies ReRAM à faible fenêtre de mémorisation. Elles ont un rendement de restauration élevé, même avec des conditions de programmation moins agressives, ce qui entraîne une endurance élevée. En

particulier, les résultats de simulation et de mesures suggèrent la mise en œuvre de NVR 2R en utilisant des dispositifs OxRAM disponibles, avec une restauration à 0.8V ou plus, pour une faible énergie de programmation (Figure 4.3(a)).

D'autre part les NVFF 1R offrent une densité d'intégration plus élevée, n'ont pas besoin du signal de FRM ou la partie FORMING et permettent d'utiliser un seul miroir de courant de restauration pour plusieurs cellules. Du point de vue du design, les bascules NVFF 1R sont plus éconergétiques à la fois en mode actif et en mode stockage, car ils ne comportent qu'un seul circuit de programmation et un bloc de contrôle plus simple. Cependant, elles nécessitent une fenêtre de mémorisation plus large, ce qui limite le choix de la technologie ReRAM et de ses conditions de programmation. Les données indiquent que l'utilisation des dispositifs CBRAM disponibles peut conduire à des NVFF 1R pouvant fonctionner à 0.7V en mode restauration (Figure 4.3 (b)).

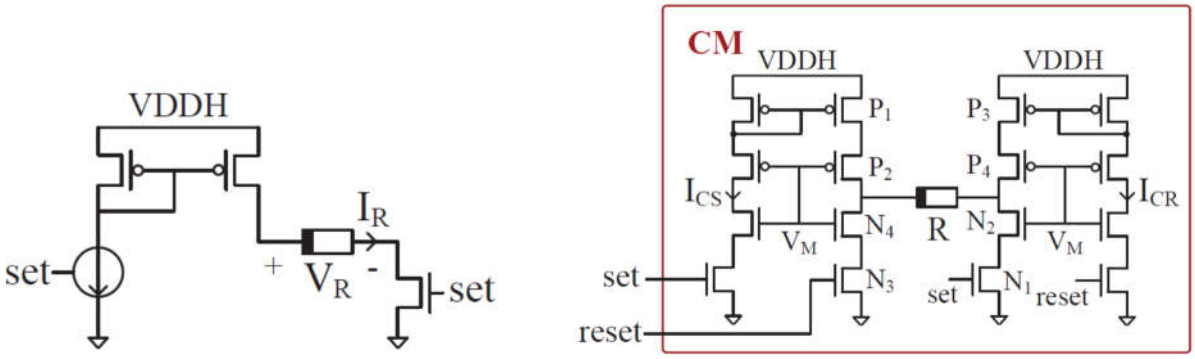
Operation de stockage

La sauvegarde des données d'une bascule dans le bloc NV repose sur la programmation du (des) circuit(s) ReRAM selon la valeur contenue dans cette bascule. Par conséquent, il faut un circuit qui effectue le SET, RESET et FORMING de la ReRAM en fonction des signaux de commande. L'idée générale des solutions de programmation proposées est illustrée dans la figure 4.4(a). Les chemins liés à la programmation ReRAM sont alimentés à VDDH, tandis que les signaux de commande sont générés dans le bloc LOGIC qui fonctionne à VDD. La fonction du circuit de programmation est alors de fournir le courant et la tension de programmation nécessaires au dispositif, tout en gérant l'écart de tension entre VDD et VDDH. Deux circuits de programmation sont proposés: l'un basé sur le décalage de niveau « LS » (Figure 4.4(b)) et l'autre sur miroir de courant « CM » (Figure 4.5).



(a) (b)

Figure 4.4: (a) Programmation d'un dispositif ReRAM - principe, (b) Détecteur de niveau fiable et partie de programmation.



(a) (b)

Figure 4.5: Structure de programmation à miroir de courant: (a) principe (opération SET), (b) circuit de programmation complet pour un périphérique ReRAM.

Les deux solutions de programmation utilisent uniquement des transistors à oxyde de grille mince, ce qui les rend compatibles avec le flot de conception numérique. En outre, ils sont fiables et peuvent être implémentés avec des noeuds CMOS inférieurs à 28 nm. Les NVFF gèrent l'écart entre les conditions de programmation SET et RESET et ont une simple opération de FORMING. Par rapport aux NVFF LS, les NVFF CM ne nécessitent pas la génération de signaux intermédiaires sur différents niveaux de tension. La réduction de tension dynamique de VDDH et VDD entre les modes de stockage, de restauration et actif est nécessaire pour les NVFF CM, alors qu'elle peut être évitée dans le cas des NVFF LS.

5. Evaluation de NVFF

Cellules NVFF mises en oeuvre

Afin de comparer deux architectures de programmation (LS et CM) et deux configurations de restauration (2R et 1R), trois des cellules NVFF présentées sont mises en œuvre et évaluées dans la technologie CMOS FDSOI de 28 nm: 2R-LS, 2R-CM et 1R-CM. Pour obtenir le rendement de restauration suffisant, la cellule 1R-CM est mise en œuvre avec des CBRAM, tandis que les cellules NVFF 2R utilisent des OxRAM. Les circuits de programmation dans les cellules sont optimisés pour répartir R_{ON} et R_{OFF} améliorant l'endurance et réduisant au minimum la puissance de programmation (Tableau 5.1). Les cellules 2R peuvent fonctionner à la tension nominale et à une tension réduite de 0,8 V (cf. Section 4), comme tension minimale pour l'opération de restauration (pour les conditions de programmation ReRAM choisies).

Les bascules NVFF proposées sont également évaluées sur le plan schématique par rapport à la cellule MSFF standard utilisée pour la construction de NVFF et à la bascule à double rail avec latch de retention de données ("ballon"). En outre, des simulations post-layout plus complètes

sont effectuées pour MSFF et NVFF. Les caractéristiques des NVFF simulées et présentées ci-après sont: le délai de propagation en mode actif, la consommation d'énergie en mode actif, la consommation d'énergie en mode stockage et le break-even sleep time des NVFF par rapport aux cellules volatiles.

Table 5.1: Paramètres des bascules NVFF implémentées

Architecture	**2R -LS, 2R-CM**	**1R-CM**
Technologies	OxRAM [41, 49], 28nm FDSOI	CBRAM [38, 80], 28nm FDSOI
R_{ON} (±3σ)	7 kΩ (4-10 kΩ)	<30 kΩ ON
R_{OFF} (±3σ)	200 kΩ (30-1000 kΩ)	>200 kΩ
V_{NOM}	1V	1V
V_{STR}	2V	2.2V
V_{RED}	0.8V	/

Résultats

Par rapport à la cellule MSFF standard, les NVFF fonctionnant en mode actif présentent une faible augmentation du délai de propagation, inférieure à la bascule avec retention de données "ballon". À la tension nominale, cette augmentation est de l'ordre de 5% ou 8% pour des temps transitions t_{CLK-Q} 0->1 ou 1->0 , respectivement (Figure 5.1).

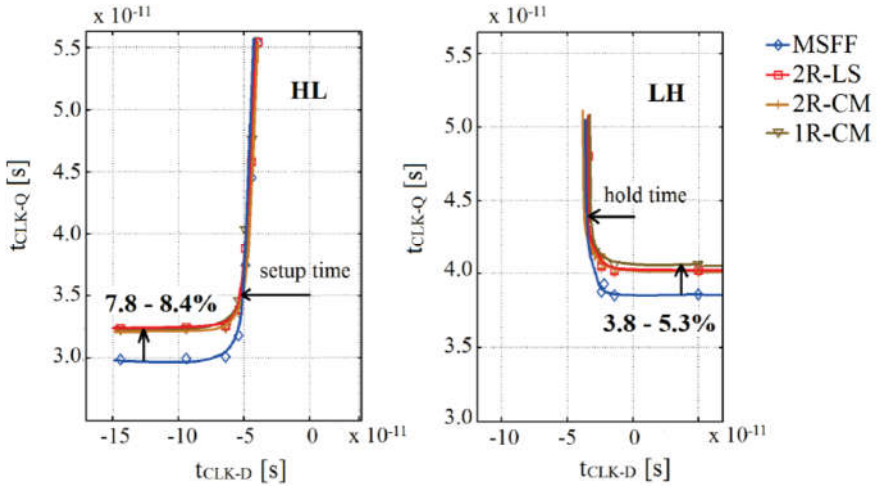


Figure 5.1: t_{CLK_Q} en fonction de t_{CLK_D}, pour MSFF et NVFFs, @ 1V (simulation post-layout).

En outre, les résultats montrent que la consommation d'énergie en mode actif est la plus petite pour la cellule NVFF 1R-CM et la plus élevée pour NVFF 2R-LS (Figure 5.2). Dans le cas de NVFF LS, le même VDDH dans tous les modes est autorisé, mais ne pas utiliser la palette de tension dynamique pour réduire le VDDH peut considérablement augmenter les fuites dans la

partie programmation de la cellule. La pénalité sur la consommation en mode actif est proportionnelle à la période d'horloge et à le taux d'activité de données, allant de 2-4% à 1GHz avec 2% d'activité jusqu'à 10-14% à 10MHz avec 10% d'activité, à la tension nominale. Les pénalités en vitesse et en consommation sont plus faibles si les cellules fonctionnent à tension réduite en mode actif.

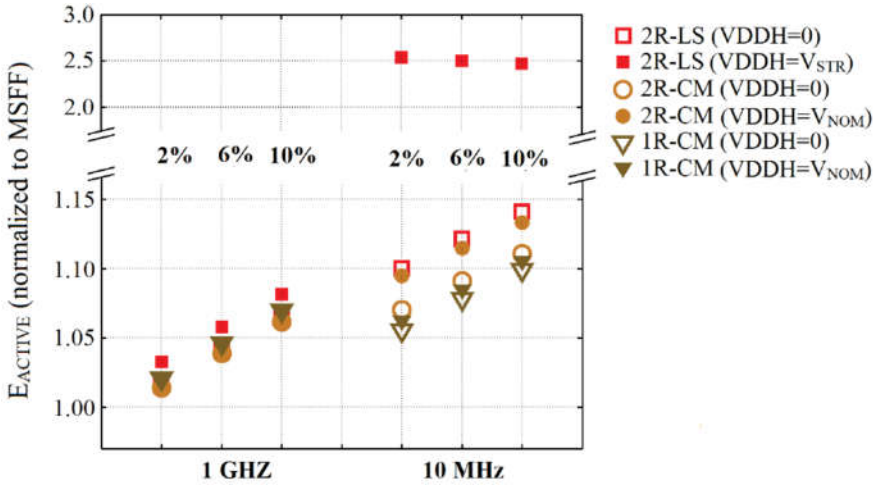


Figure 5.2: Energie des NVFF en mode actif @ 1V, pour différentes fréquences et niveaux d'activité de données (simulation post-layout).

Pour évaluer la consommation des NVFFs en mode stockage, un modèle électrique simplifié du comportement ReRAM a été utilisé, couvrant différents scénarios de stockage qui dépendent de la donnée dans la bascule. Pendant l'opération de stockage, la cellule 1R-CM présente une consommation d'énergie inférieure à celle d'autres NVFF, en raison de valeurs de résistance plus élevées et d'un seul circuit de programmation. Cependant, son énergie nécessaire au stockage de 30-120pJ est la plus importante, ce qui est une conséquence des temps de commutation plus longs des dispositifs CBRAM (Figure 5.3). Les NVR 2R-CM et 2R-LS possèdent une consommation d'énergie similaire de 20 à 35 pJ.

Le "Break-even time" (BET) des NVFF par rapport aux MSFF/«ballon» est défini comme la valeur t_{SLEEP} pour laquelle l'énergie du sommeil MSFF/«ballon» est égale à l'énergie du sommeil NVFF. Ainsi, cette valeur indique le point où le remplacement de la cellule volatile par NVFF apporte des économies d'énergie si les périodes d'inactivité sont supérieures à BET. En comparaison avec MSFF, le BET estimé de NVFF est dans la gamme de 5-200ms pour les cellules 2R basées sur OxRAM, ou 10-700 ms pour la cellule 1R à base de CBRAM, comme le montre la Figure 5.4 (la gamme couvre la tension de retention de 1-3-1V de la bascule et toutes les valeurs possibles des données stockées en cycles consécutifs). Il augmente à 0.1-2s (2R) ou 0.2-6s (1R), lorsque les NVFF sont comparées à la bascule "balloon". Il faut noter que les résultats donnés correspondent au scénario le plus simple dans lequel le back-up est effectué avant chaque période de sommeil, et que les cas d'utilisation différents peuvent entraîner des économies d'énergie. Dans un but de simplification, l'estimation présentée ne tient pas compte

des transitions des alimentations entre les modes de fonctionnement. Cet aspect doit être pris en compte dans l'analyse des économies au niveau du système.

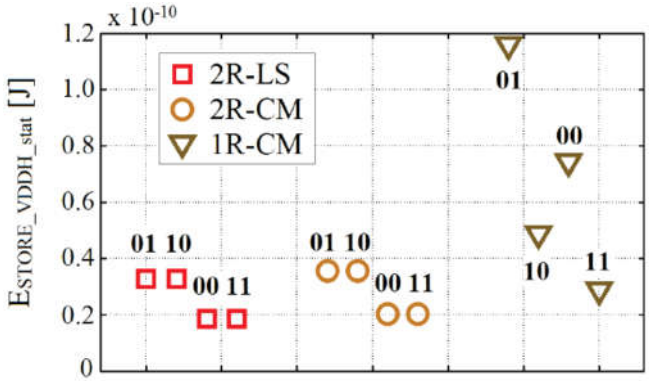


Figure 5.3: Energie (mode stockage) dans le domaine VDDH (E_{STORE_VDDH}) des NVFF 2R-LS, 2R-CM et CM-1R, pour tous les scénarios Q_{old}/NV (simulation post-layout).

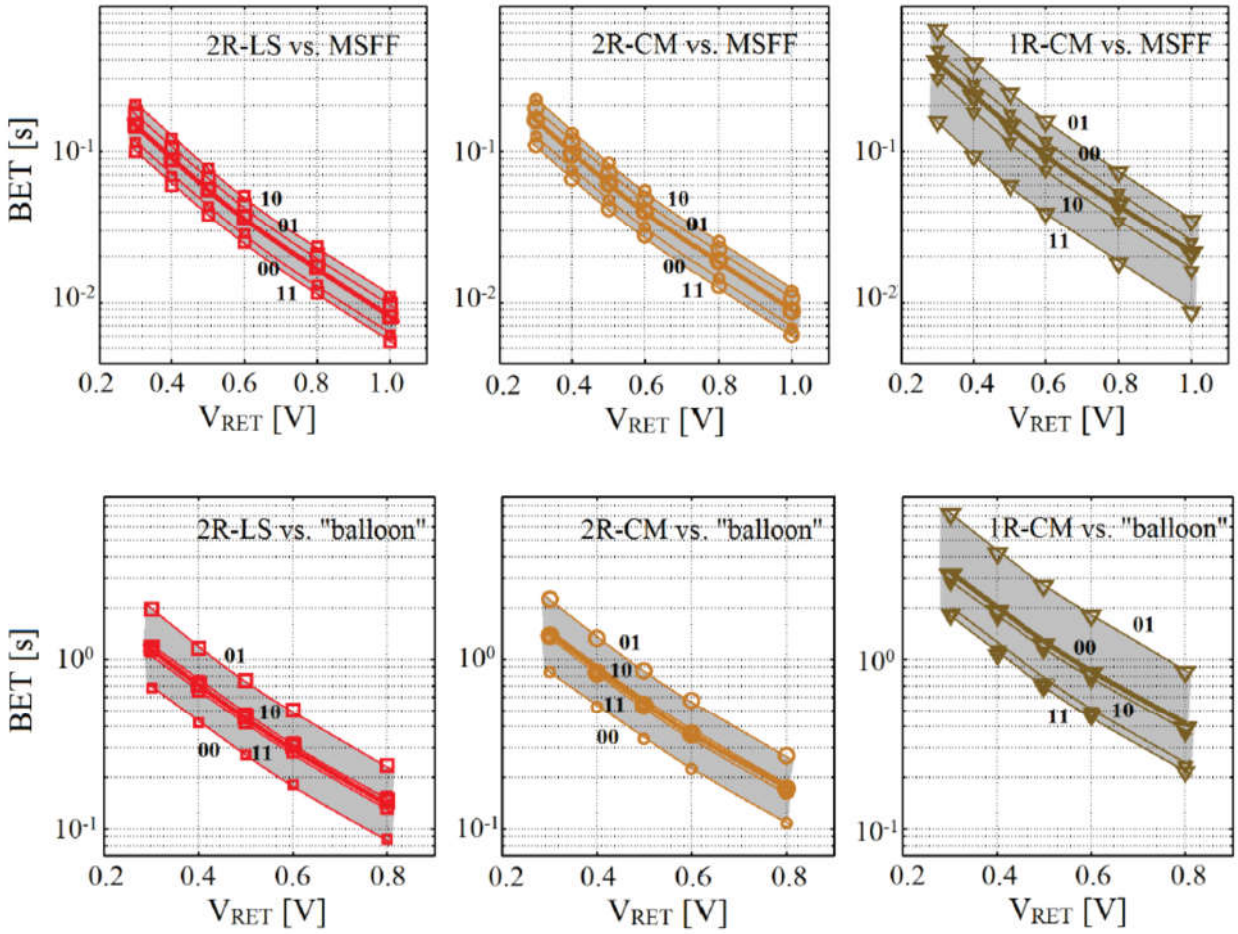
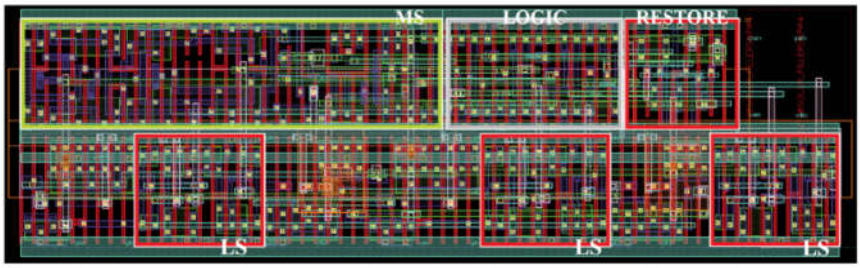
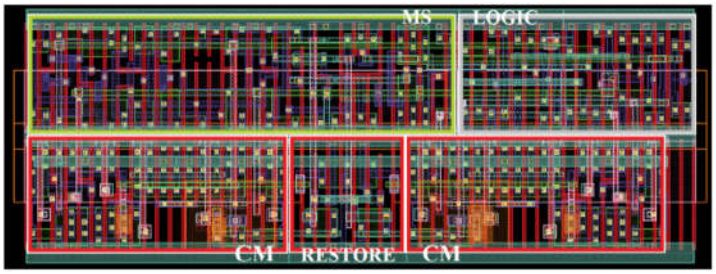


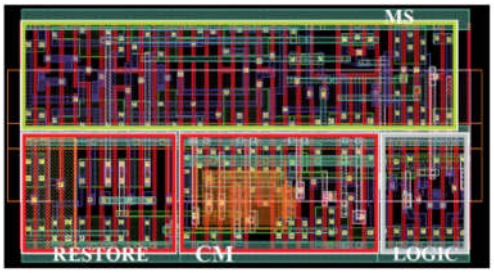
Figure 5.4: (en haut) BET de NVFFs vs. MSFF en fonction de la tension de rétention MSFF ($V_{RET,}$), (en bas) BET de NVFFs vs « ballon » en fonction de la tension de maintien de « ballon » (V_{RET}) pour toutes les possibilités de V_{RET}/Q (simulation post-layout)

La pénalité la plus importante de la version non-volatile par rapport à la version volatile de la bascule est dans la surface requise pour la cellule, car les NVFF sont 3x (1R-CM NVFF) à 5,6x (2R-LS NVFF) plus gros que les MSFF (Figure 5.5).



(a)





(b) (c)

Figure 5.5: Layout de (a) 2R-LS NVFF, (b) 2R-CM NVFF, (c) 1R-CM NVFF.

6. File register non-volatile

Outre les bascules qui sont dispersées dans la logique numérique, les coeurs de microprocesseur contiennent également un certain nombre d'éléments de mémoire adressables (par exemple, les registres à usage général) qui peuvent être organisés dans un fichier de registre ordinaire (« register file »). En règle générale, les register files doivent pouvoir lire et écrire simultanément plusieurs données à chaque cycle d'horloge, ainsi ils sont réalisés en tant qu'architectes multi-ports [85]. Pour obtenir une solution haute performance, simple et robuste avec un temps de conception minimal, les processeurs avec un petit nombre de registres et ports d'accès utilisent souvent des register files basés sur des bascules générés à l'aide du code RTL. Pour une densité plus élevée et consommation d'énergie limitée, les gros register files dans les processeurs complexes sont généralement conçus comme des tableaux de type SRAM entièrement personnalisés.

Afin de satisfaire tous les défis de la co-intégration des technologies ReRAM et CMOS, les solutions de conception de bascule non-volatile présentées dans la Section 4 impliquent des

surfaces utilisées élevées. Cependant, elles peuvent être utilisées pour la conception de register files multi-port non-volatiles (NVRF) plus denses en plaçant les parties NVFF communes en dehors de la cellule et en les partageant entre plusieurs cellules. Une circuiterie périphérique commune peut être utilisée ensuite pour les opérations volatiles (lecture multiple, écriture multiple) et les opérations non-volatiles (stockage, restauration).

Une nouvelle architecture NVRF basée sur ce principe est démontrée dans cette thèse. Un accès double lecture simple écriture (2R1W) est considéré, étant donné que ce nombre de ports est compatible avec le jeu d'instructions du processeur ARM Cortex-M0 + processeur. Il contient 16 mots de 32 bits, ce qui correspond à 13 registres généraux, le pointeur de pile, le registre de lien et les registres auxiliaires de M0 +. Le NVRF est conçu et dessiné en technologies CBRAM et CMOS 130 nm. Tout en augmentant la densité cellulaire, le register file répond aux contraintes données pour NVFFs:

- en utilisant la tension nominale pour la partie volatile et une tension plus élevée pour ReRAM,
- en optimisant la conception pour avoir des courants de programmation qui conduisent à une consommation plus faible et une augmentation de l'endurance,
- en surmontant l'écart entre les conditions de programmation SET/RESET,
- en assurant un rendement de restauration suffisant.

Comme l'alimentation nominale du noeud CMOS 130 nm est plus proche de la tension de programmation ReRAM, il n'y a pas d'empilement de transistors, bien que la solution utilise uniquement des transistors à oxyde de grille mince.

L'architecture NVRF proposée repose sur la cellule NVFF 1R-CM. Une solution de programmation de type CM plutôt que LS est choisie car elle nécessite moins de place, alors que les consommations et les vitesses de ces deux approches sont comparables. Par ailleurs, elle peut être simplement intégrée dans un tableau en partageant les sources de courant entre plusieurs cellules. En ce qui concerne la restauration, la mise en œuvre du NVRF avec la technologie CBRAM à haute fenêtre de mémorisation permet la configuration 1R, plus compacte et moins consommatrice que la configuration 2R (pour le même type de ReRAM). De plus, cela simplifie la conception car le signal de FORMING dédié n'est pas nécessaire.

Le register file non-volatile, représenté dans la Figure 6.1, se compose des éléments suivants:
- une cellule M par colonne, identique à l'étage maître de MSFF,
- le tableau de cellules S-NV, où S est l'étage esclave adapté de MSFF, et NV stocke les données de S d'une manière non-volatile (Figure 6.2),
- un bloc ST et RST par colonne, soit les parties communes de circuiterie pour stockage et restauration (Figure 6.3),
- des blocs de lecture READ, utilisés pendant les opérations de lecture et de stockage, et
- des blocs DECODERS, pour fournir des signaux d'adressage requis.

Comme pour la cellule NVFF, la solution utilise deux alimentation - la tension VDD d'opération CMOS et et la tension VDDH pour programmation des dispositifs ReRAM . VDDH est distribué vers les parties NV des bitcells et le block ST, tandis que VDD alimente tous les autres blocs.

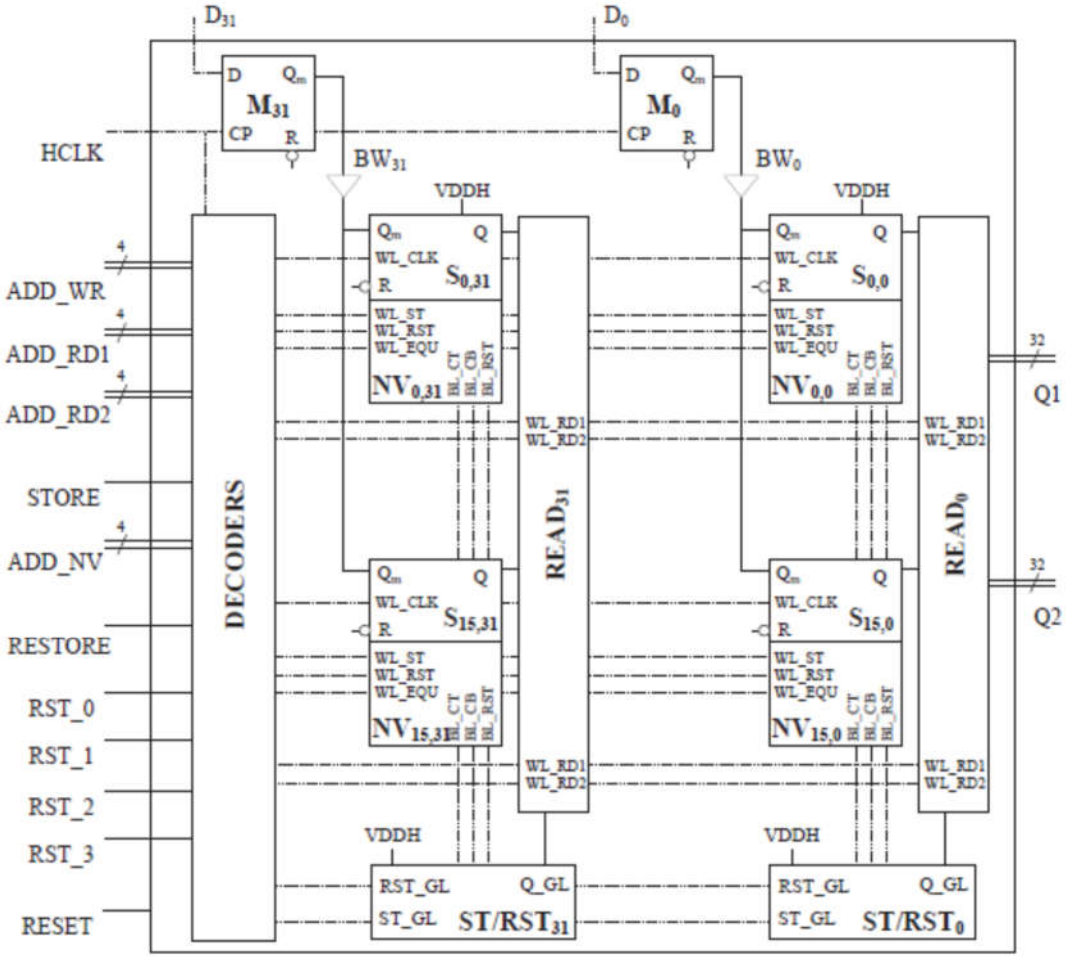


Figure 6.1. L'architecture NVRF proposée.

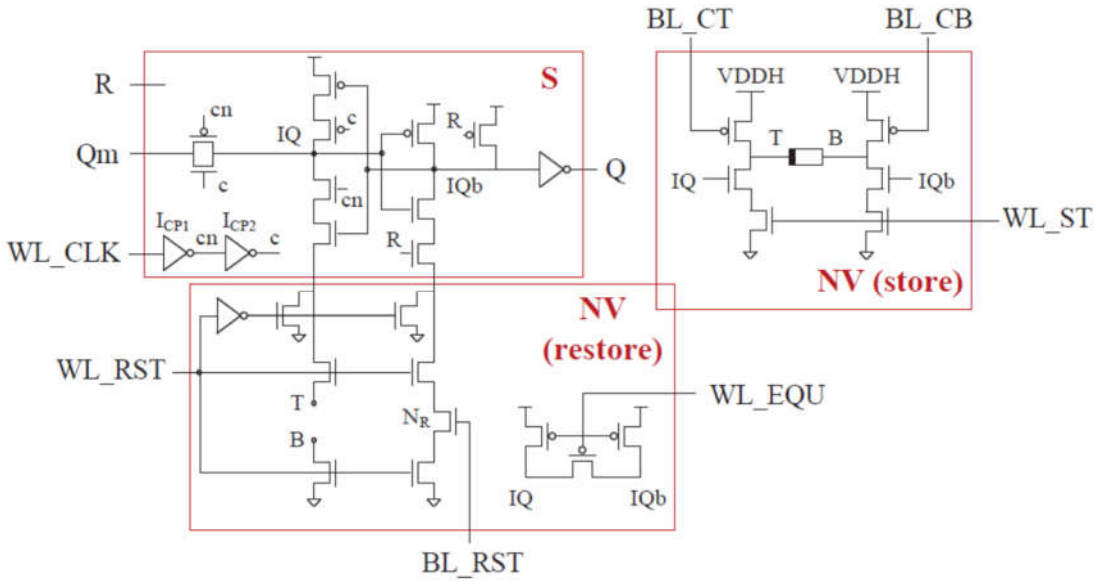


Figure 6.2. – S-NV cellule d'information.

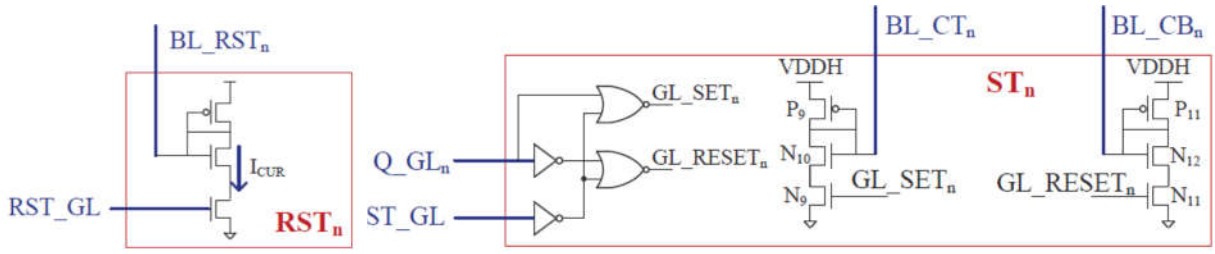


Figure 6.3. – Les cellules RST (à gauche) et ST (à droite).

La conception présentée démontre que les surfaces silicium des bascules non-volatiles peuvent être minimisées en les organisant dans un tableau de type file register, sans impact sur la robustesse du stockage et de restauration. La solution mise en oeuvre permet l'exécution en parallèle d'écriture à une adresse et de lecture en deux adresses. De plus, le nombre de ports peut être facilement étendu sans modification de la cellulle élémentaire S-NV. Avec ajout de décodeur d'adresse, chaque opération de lecture ou d'écriture requiert, respectivement, un transistor NMOS dans le bloc READ ou une nouvelle cellule M. Cette architecture NVRF permet une transition sommeil-réveil très rapide, puisque l'opération de restauration peut être effectuée simultanément sur plusieurs lignes.

Enfin, la conception NVRF proposée peut être portée à la technologie CMOS 28 nm et même des nœuds en-dessous en utilisant uniquement des transistors à oxyde de grille mince, conformément à l'approche mise en œuvre dans les NVFFs. Ainsi, l'empilement de transistors doit être mis en œuvre dans la partie NV de la cellule d'information S-NV et dans le bloc ST.

7. Conclusion

Le travail effectué au cours de cette thèse peut être résumé comme suit:

- L'influence des tensions et des courants de programmation ReRAM sur les valeurs de R_{ON} et de R_{OFF}, vitesse de commutation, l'énergie de programmation et d'endurance dispositif a été étudiée.

- Différents aspects de la conception de bascules NVFF (circuits de stockage / restauration, alimentation, etc.) ont été explorés en ce qui concerne les propriétés des technologies NVM et CMOS utilisées (fenêtre de mémorisation, conditions de programmation, variabilité, etc.). Des lignes directrices de conception en ont été extraites pour la mise en œuvre de nouvelles bascules et register files à base ReRAM-non-volatiles.

- Deux circuits à double tension pour la programmation des dispositifs ReRAM ont été proposés: l'un à base de convertisseur de niveau et l'autre à base de programmation à miroir de courant. Les deux solutions sont compatibles avec la technologie CMOS, et utilisent uniquement des transistors à oxyde de grille mince pour la compatibilité avec le flot de la conception numérique.

- Deux architectures de restauration ont été étudiées: l'une à deux dispositifs et compatible avec une large gamme de technologies ReRAM, et l'autre à dispositif unique permettant une meilleure densité d'intégration et une plus faible consommation. Les opérations de restauration robustes à faible tension sont obtenues en choisissant judicieusement les technologies de ReRAM et leurs conditions de programmation, après l'analyse statistique qui prend en compte les variations CMOS et ReRAM.

- Le coeur du NVFF, en particulier l'architecture de l'étage esclave, a été étudié et concu pour atteindre le plus haut rendement de restauration et des performances en mode bascule normal.

- Les NVFFs proposés ont été simulés et déssinés en technologie FDSOI 28nm. Conçues comme cellules standards, toutes les solutions sont optimisées pour utiliser des conditions de programmation ReRAM qui améliorent l'endurance et réduisent la consommation, tout en surmontand l'écart entre les conditions de programmation SET et RESET.

- L'évaluation post-layout des masques des NVFFs a été effectuée. Elle a montré que, par rapport à la bascule MSFF de la bibliothèque de cellules standard, tous les NVFFs introduisent un délai minimal et une pénalité de consommation en mode actif. Parmi les solutions proposées, les circuits de restauration 1R et de stockage CM correspondent aux architectures les plus efficaces pour ce qui concerne l'énergie en mode actif.

- En mode stockage, les résultats indiquent que la consommation des circuits LS et CM est similaire quand ils sont optimisés pour la même technologie et conditions de programmation. En général, les NVFF 1R sont plus efficaces en énergie que les NVFF 2R avec les mêmes ReRAMs, en raison d'un seul circuit de programmation et d'un bloc de contrôle plus simple. Cependant, leur contrainte de fenêtre de mémorisation plus élevée limite le choix de la technologie ReRAM ou implique des conditions de programmation plus agressives (en particulier, des largeurs

d'impulsion plus grandes), ce qui augmente l'énergie de programmation. Par conséquent, les bascules NVFF 2R-CM et NVFF 2R-LS conçues avec des OxRAM plus rapides ont une consommation d'énergie de stockage inférieure à celles des NVFF 1R-CM à base de CBRAM.

- L'évaluation présentée dans ce travail suggère que, pour des systèmes caractérisés par de longues périodes de veille, l'utilisation de NVFF peut apporter des économies d'énergie considérables. Par exemple, la rétention de données dans les bascules existantes alimentées à 0,5V est supérieure à la consommation des solutions 2R-LS/2R-CM ou 1R-CM après des périodes d'inactivité de 0.2-0.8s ou 0.7-3s, respectivement.

- Le layout de la bascule NVFF est entièrement compatible avec le flot de conception numérique, en plaçant les ReRAM entre les deux dernières couches métalliques de la technologie. Par rapport à la surface de cellule standard MSFF, les cellules NVFF sont de 3 (1R-CM) à 5 ou 6 (2R-LS) fois plus grandes. Cependant, les surfaces de ces cellules peuvent être réduites, car elles peuvent partager plusieurs ressources avec d'autres NVFFs sur le circuit intégré.

- Il a été démontré que le principal inconvénient des NVFFs (les surfaces de silicium utilisées), peut être réduit au minimum en créant un réseau de mémoire et en plaçant la circuiterie commune (pour écrire, lire, stocker et restaurer) à la périphérie. A titre d'exemple, un register file multi-ports non-volatile de type deux lectures et une écriture a été implémenté dans le noeud CMOS 130nm. Le réseau repose sur la cellule 1R-CM à base CBRAM. La solution proposée offre une densité supérieure à celle de la cellule NVFF avec la même performance de transition de réveil, et peut être facilement ajustée pour prendre en charge plus de ports de lecture et d'écriture.

Les travaux futurs: processeur non-volatile

Après la conception de cellules NVFF à base de ReRAM, la prochaine étape naturelle consiste à appliquer la même approche sur un circuit à plus grande échelle. En fait, il existe déjà un travail en cours concernant l'application des solutions présentées dans cette thèse à un processeur non volatile. Dans cette premiere tentative de réaliser une unité de microcontrôleur non-volatile à base ReRAM, la bascule NVFF basée sur une architecture 1R-CM est incluse dans le processeur ARM Cortex-M0+ microprocesseur. Toutes les bascules de la version volatile du coeur de processeur sont remplacées par leurs homologues non-volatiles. Par conséquent, l'unité de gestion de l'alimentation dédiée (PMU) est réalisée avec une machine à états qui gère les différents modes sommeil et contraintes d'alimentation. Dans le mode de sommeil profond, seule l'unité de gestion PMU et le controleur de réveil (WIC) sont alimentés, tandis que le reste du circuit peut être coupé. Ainsi, la version non-volatile permet des économies d'énergie élevées par rapport à la norme MCU, ce qui réduit le désavantage de la mise en sommeil et du réveil.

L'un des défis majeurs et l'étape suivante dans le développement du processeur non-volatile porte sur la réduction de sa surface. Pour résoudre ce problème, une solution plausible peut être notamment de mettre ensemble le register file non-volatile et les bascules non-volatiles. Pour les

architectures telles que M0+, qui ne contiennent pas de register file et ont les registres à usage général à base de bascules, la mise en œuvre de NVRF nécessite une légère modification du coeur du processeur. Par conséquent, l'unité de gestion de l'alimentation et le contrôleur NVFF / NVRF doivent être adaptés pour valider cette modification. Après la preuve du concept avec «uniquement» du matériel non-volatile, de nouvelles orientations pour l'amélioration du NVP surgiront. Par exemple, au lieu de considérer que toutes les bascules sont non-volatiles, l'examen plus approfondi du système peut conduire à la sélection d'un sous-ensemble plus approprié de FFs à être remplacé par des NVFFs. Enfin, le développement peut se poursuivre au niveau du logiciel. Pour une meilleure exploitation des propriétés NV, de nouvelles façons d'adapter le logiciel du système peuvent être explorées. Quelques pistes pour cette réflexion sont le contrôle supplémentaire des procédures de mise en sommeil/réveil, le choix optimal du mode veille, etc. En conclusion, la co-conception matériel/logiciel appropriée permettra de tirer le meilleur profit de l'ajout de la non-volatilité au système IoT.