Bruna C. Cagliari, Paulo F. Butzen, and Raphael M. Brum

Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil bruna.cagliari@ufrgs.br, paulo.butzen@ufrgs.br, brum@ufrgs.br

Abstract— The rise of the Internet of Things (IoT) and the constant growth of portable electronics have leveraged the concern with energy consumption. Nonvolatile memory (NVM) emerged as a solution to mitigate the problem due to its ability to retain data on sleep mode without a power supply. Nonvolatile processors (NVPs) may further improve energy saving by using nonvolatile flip-flops (NVFFs) to store system state, allowing the device to be turned off when idle and resume execution instantly after power-on. In view of the potential presented by NVPs, this work describes the initial steps to implement a nonvolatile version of Neander, a hypothetical processor created for educational purposes. First, we implemented Neander in Register Transfer Level (RTL), separating the combinational logic from the sequential elements. Then, the latter was replaced by circuit-level descriptions of volatile flipflops. We then validated this implementation by employing a mixed-signal simulation over a set of benchmarks. Results have shown the expected behavior for the whole instruction set. Then, we implemented circuit-level descriptions of magnetic tunnel junction (MTJ) based nonvolatile flip-flops, using an open-source MTJ model. These elements were exhaustively validated using electrical simulations. With these results, we intend to carry on the implementation and fully equip our processor with nonvolatile features such as instant wake-up.

Index Terms— Nonvolatile Processors, Magnetic Tunnel Junction, Nonvolatile Flip-Flops

I. INTRODUCTION

The Internet of Things has become one of the most prosperous areas within technology as a global network infrastructure that connects the physical and the virtual worlds [1]. One way to do this communication is through a wireless network sensor that passes physical information from the environment to the virtual world [2]. These sensor devices include transductors, networking hardware and microprocessors, build on top of volatile CMOS (complementary metaloxide-semiconductor) flip-flops and memories. Such devices often spend most of their time in sleep mode [3]. This operation mode slowly drains their batteries, even though no useful operation is being performed. Nonvolatile processors (NVPs) appear as a way to guarantee zero standby power: thanks to its non-volatility, the circuit in sleep mode can be disconnected from the power supply without losing the stored data. Overall, battery-powered systems benefit from NVPs due to reduced static power consumption.

Nevertheless, the other problems with the battery include sizing and lifetime. Energy harvesting has become a viable solution for self-powered systems, replacing batteries and providing long life and low maintenance cost [4]. However, the energy from the environment is unstable, causing great concern about data loss. Nonvolatile processors are intrinsically more tolerant to such failures, allowing the use of intermittent sources.

NVPs mitigate energy and performance overheads due to the use of nonvolatile flip-flops (NVFFs) [5]. NVFFs store the system state every n cycles, being able to recall prestored data after a suspension or failure. They perform a parallel local transfer of bits at the lower levels of the memory hierarchy that enables a quick resumption, making NVPs ideal for low power and high-performance computing applications [6]. This approach saves time and energy due to the decrease in data traffic between the memories of the higher layers of the hierarchy that have longer access times.

Due to all the potential presented by NVPs, a review of nonvolatile memories and architectures is presented throughout this work. In summary, this paper describes the first steps taken towards a nonvolatile version of the Neander processor, along with NVP design considerations.

Neander is a hypothetical processor created for educational purposes [7]. The final objective of this project is to implement a nonvolatile Neander. NVMs are still experimental technologies, and the behavior of nonvolatile cells must be carefully tuned to match the desired specification. CMOS logic behavior, on the other hand, is well-known and reliable. For this reason, we chose to implement the NVFFs in circuit-level, while keeping a high-level abstraction for combinational logic. This work's proposal is to validate the initial steps towards the nonvolatile implementation of Neander. The first one is to validate a mixed-signal Neander implementation, combining an analog model for flip-flops with a purely digital implementation of the Neander's random logic. The second is to validate the shadow register described in [8] using electrical simulations. The shadow register proposed is a hybrid scheme that separates volatile and nonvolatile contexts: a NVFF acts in parallel with the existing volatile register.

This paper is organized as follows: an overview on nonvolatile emerging memories is presented in Section II. Important concepts for understanding the purpose of the work are presented in the Section III. The Neander processor is described in Section IV. Section V presents related works. The work proposal is presented in Section VI and Section VII explains our implementation and experimental setup. Results are discussed in Section VIII and final considerations are made in Section IX.

II. NONVOLATILE EMERGING TECHNOLOGIES

The advancement of IoT and the growing use of portable devices accompany the current challenges in low-power solutions. Memories are one of the most energy-consuming components of the system and an essential part in NVPs. Emerging nonvolatile technologies as ferroelectric RAM (FRAM), phase change RAM (PCRAM), magnetoresistive RAM (MRAM), spin-transfer-torque RAM (STT-MRAM), and resistive RAM (RRAM) comes to offer zero standby power, higher endurance, and scalability [9]. A brief comparison between the technologies is shown in Table I.

Table I. Comparison	between	emerging	nonvolatile	memories	[9],[10].
		~ ~ ~			

Technology	Endurance (cycles)	Write time (ns)	Read time (ns)
FRAM	10^{12}	50	20 to 80
PCRAM	$>10^{9}$	~ 50	< 10
RRAM	$> 10^6 \sim 10^{12}$	< 10	< 10
MRAM	$>10^{15}$	3 to 20	3 to 20
STT-RAM	$> 10^{16}$	< 10	< 10

The choice of nonvolatile technology is a critical design decision. Due to the importance of NVMs as one of the main parts of NVPs, an overview of emerging nonvolatile technologies is presented next.

A. FRAM

Ferroelectric random access memory (FRAM) has been commercialized for about 25 years, being the most mature technology among the emerging NVMs [11]. Given the high endurance of FRAM and its well-known technology, it is used in many applications, including IoT devices [12].

A FRAM memory cell consists of a ferroelectric capacitor and a selection transistor that enables the state of the capacitor to be sensed [9]. FRAM explores the polarization properties of a ferroelectric substance, such as lead zirconate titanate (PZT), as a nonvolatile storage mechanism [9]. The operation can be verified in Fig. 1, adapted from [13], where: Vc is the coercive voltage and Pr is the remaining polarization. To switch the logic value '1' to the '0', a voltage greater than the Vc must be applied across the ferroelectric capacitor. To return, sufficient negative voltage must be applied to reverse the polarization. When the power is removed, the polarization moves to one of the stable states: positive (logical '0') or negative (logical '1').



Fig. 1 Ferroelectric P-V hysteresis loop, adapted from [13].

Unfortunately, FRAM has a destructive read cycle [9]. The reading method consists of applying an electric field through the ferroelectric capacitor. If a state change is detected, a peak current will appear, indicating the value that was being maintained in the cell.

B. PCRAM

Phase change random access memory (PCRAM), also known as PCM, is a nonvolatile memory qualified for highdensity applications [14]. PCMs typically are arranged as multi-bit memories: each element can store more than one bit. This approach increases the storage capacity per unit area and therefore reduces the cost of the circuit.

As a potential storage class memory (SCM) [15], PCM reduces the performance gap between nonvolatile storage memories and the volatile ones in the current memory hierarchy, as illustrated in Fig. 2, adapted from [16]. It is also a promising candidate for replacing DRAM memory banks, in order to reduce energy consumption and provide high density [17]. PCM is also being employed to build mass storage memory devices, due to its scalability, high density and endurance [16]. There are already examples of commercial products based on this technology [18].



Fig. 2: Comparison of the access time and cost of volatile and nonvolatile memories, adapted from [16].

Regarding the nonvolatile storage element, the PCM cell is composed of a thin film of chalcogenide material sandwiched between two electrodes. Its operation is based on the phase change between the crystal (low resistance) and amorphous (high resistance) phases of the chalcogenide semiconductor [14]. The different resistance states are used to store the nonvolatile data. The large contrast between the high and low resistances allows the PCM to store more than one bit per cell by taking advantage of intermediate states [14].

The switching mechanism is accomplished by applying electrical current pulses through the memory cell [19]. The set operation uses a moderate current to change the high resistance amorphous state to the low resistance crystal by heating the chalcogenide material above its crystallization, but below its melting temperature for a relatively long period to enhance crystallization. The reset operation does the opposite change by using a high current and heating the material above its melting temperature for a short period to induce the amorphous state. However, the reset current is a concern within the scalability of PCM and is one of the main points of research on this technology [19].

C. RRAM

Resistive random access memory (RRAM) is another potential candidate for storage-class memory besides PCM [20]. It offers many advantages such as low switching voltage and high density. In addition, the RRAM has great compatibility with the manufacturing process of CMOS technology, allowing a fast and low power resumption [21]. The memristor-based RRAM is one of the most promising nonvolatile memories among the emerging nonvolatile technologies [22], [9]. The memristor is a nonvolatile circuit element associated with its memristance, characterized by a non-linear association between magnetic flux and electrical charges [23]. The storage mechanism depends on the polarity and magnitude of the applied voltage. Fig. 3a depicts a memristor structure and Fig. 3b illustrates its operation through the current versus voltage (I-V) loop: ON and OFF represent states of the resistance. When a voltage is applied across the top/bottom electrodes, resistance decreases. Reversing the polarization, the resistance increases.



Fig. 3: a) Memristor structure ; b) Current versus voltage curve of memristor, adapted from [24].

The small size of memristors facilitates memory fabrication in a crossbar array. As a high-density memory, there are many possibilities of integration structures: Cross-point RRAM array, multiple 2-D cross-point arrays and also 3-D vertical RRAM arrays [25]. The main problem in these integrations is the sneak paths: a path that makes undesirable writings on unselected cells [26].

D. MRAM

Magnetoresistive random-access memory (MRAM) is one of the most promising spintronics applications, presenting high endurance, high read/write speed and easy integration with CMOS technology [27]. Its nonvolatile storage mechanism is based on the tunnel magnetoresistance effect observable in structures composed of ferromagnetic layers separated by a thin insulator. This structure is called Magnetic Tunnel Junction (MTJ). Electrical resistance across the MTJ stack may be modified by applying an external magnetic field [8]. The MTJ is the nonvolatile spintronic device that is the key technology in each MRAM memory cell.

The MTJ stack is depicted in Fig. 4. It is basically composed of two ferromagnetic layers: one fixed and the other one free, separated by a very thin tunnel barrier (~ 1 nm) [28]. The fixed layer is used as a reference. The relative orientation of the two ferromagnetic layers determines the variable resistance of the device: low (parallel) or high (antiparallel). Typically, the high resistance represents the logical value '1' and the low resistance represents the '0' logical value.

In addition, MTJ is a device that offers radiation immunity [8]. Over time, it has been extensively explored and there are already several variants that explore switching methods to change the orientation of the free layer. The method of



Fig. 4 MTJ in a) low resistance and b) high resistance.

writing the data directly influences the power, speed and area of hybrid MTJ/MOS circuits [29].

E. STT-MRAM

The spin-transfer-torque random access memory is a special case of the MRAM. This variant is based on the STT-MTJ. It is one of the most promising emerging nonvolatile technologies with a high read/write speed, being considered to replace SRAM in the current memory hierarchy [30]. Compared to MRAM, STT-MRAM offers excellent advantages such as a simpler and smaller structure, reduced write dynamic power and better scalability [19].

The magnetic tunneling junction based on the spintransfer-torque (STT-MTJ) is the most researched MTJ variant and one of the most promising MRAM technologies, presenting low energy consumption, compact design and CMOS compatibility [8, 31]. The operation of the STT-MTJ is illustrated in Fig. 5 [8]. To change the magnetization state of MTJ in this variant, a high-enough switching current is applied through the device. The MTJ state changes from parallel (P) to antiparallel (AP) if the forward biased current is greater than the critical current $I_{P\to AP}$. Once in the anti-parallel state, a reverse-biased current of intensity higher than the critical current $I_{AP\to P}$ must be applied to bring the MTJ back to the parallel state.



Fig. 5: (a) STT-MTJ structure and (b) I-R curve of STT-MTJ device, both published in [8].

III. NONVOLATILE PROCESSORS

NVPs provide zero standby power and instant on/off: the ability to be switched off and return to running in the saved state before power interruption. Based on this feature, a power gating technique can be applied to the circuit to reduce static energy consumption, which makes NVPs ideal for low-power applications such as battery-powered systems [5]. In addition, NVPs also guarantee data security in energy harvesting applications, when intrinsically unstable energy from the environment is used as the circuit supply. To clarify

A. Nonvolatile Flip-Flops

NVFFs are key components in NVP. Their construction usually involves the integration of standard CMOS flip-flop parts with nonvolatile devices (such as MTJs) and their corresponding read/write circuitry. One of its great advantages is the possibility of being disconnected from the power supply on sleep mode without losing the stored data, reducing static energy consumption. This feature allows the implementation of the power gating technique, widely used in systems-on-chip, with zero standby power [32]. In what follows, we will focus on NVFFs based on our chosen target technology, STT-MRAM.

As an essential part of NVPs, NVFFs must be able to store and restore system states. These approaches are possible through the NVFF operating modes: read, write and latch. Reading data is the process of converting the MTJ resistance to a CMOS-compatible voltage level. If a single bit of data is encoded in a pair of MTJs with opposite states, read circuitry can be as simple as an inverter loop [33]. A write driver will set the MTJs state according to the NVFF input. It needs to generate a switching current higher than $I_{AP \to A}/I_{P \to AP}$, whose direction varies according to the input voltage level. In the two-MTJ-per-bit strategy previously mentioned, this current must flow through both devices at the same time, in opposite directions. Flip-flops are expected to be synchronous to the system clock and edge-triggered. This is done by mixing the MTJ-specific circuit with standard CMOS flip-flop parts. Furthermore, if the nonvolatile state was not changed from the previous clock cycle, we may hold the previous state as an electrical charge, rather than going into the reading process again. We call this operation latch mode.

As proposed by Na et al. [31], NVFFs can be presented in two different structures: merged latch and sensing circuit (MLS) and separated latch and sensing circuit (SLS), both depicted in Fig.6. STT-MTJ variant has great scalability, as the write current decreases in proportional to the size of the MTJ [31]. However, with a lower critical current, it also becomes a greater problem to pass a read current that does not cause any switching. The SLS structure emerged so that the latch and the sensing circuit could be optimized simultaneously, in order to circumvent the STT-MTJ problem with the reading current [31].



Fig. 6 (a) MLS structure; and (b) SLS structure, both published in [8].

The hybrid integration of nonvolatile devices, such as MTJs, with CMOS (Complementary metal-oxide-

Moreover, NVFFs can be built from the other emerging nonvolatile technologies presented previously. SR flip-flop storage element are proposed in [35]. Nonvolatile flip-flops that integrate the PCM device are described in [36]. Many other NVFFs are already provided in the literature.

B. Nonvolatile Architectures

Conventional CMOS processors can incorporate on-chip or off-chip nonvolatile memories to keep data while disconnected from the power source. However, the operations of storing and restoring data through the sequential global transfer of data cause undesirable energy and performance overheads due to the data traffic. NVPs integrate nonvolatile flip-flops that perform a localized bit-to-bit transfer, storing the data at the lower levels of the memory hierarchy, as illustrated in Fig. 7. This approach allows quick and energyefficient restoration of the system [4].



Fig. 7: Memory hierarchy of volatile and nonvolatile processors, adapted from [4].

Fig. 8 shows different approaches to implement nonvolatile architectures. In a full NV architecture, data are directly stored in the NVFFs without the need for explicit control signals, making the complexity of the implementation simpler [37]. However, computing is limited to the nonvolatile technology employed, which commonly consumes more time and energy in transferring data than volatile technologies [37]. A hybrid NV architecture, in turn, sacrifices silicon area in order to obtain better performance. The current computing uses volatile flip-flops, storing data in the nonvolatile sub-registers only when the control signal is triggered. This approach is more independent of the nonvolatile device, ensuring faster operation and lower energy cost.



Fig. 8: a) full NV architectures and b) hybrid NV architecture, adapted from [37].

C. Checkpoint and rollback

Backup and recovery are essential operations for any Backward-Error Recovery (BER) capable processors. In NVPs, the backup is performed on the nonvolatile registers through checkpoints (CKPs): the NVFFs receive a signal that activates writing of the current states in nonvolatile devices. The recovery operation, in turn, consists of triggering the Rollback (RLB) signal, which activates the reading of nonvolatile data, forcing the execution of the processor to return to the last saved states.

One of the main concerns within the context of backup and recovery is to maintain the consistency of memory. In view of this, many approaches are described in the literature and two are illustrated in Fig. 9 due to their simplicity. The method shown in Fig. 9 (a) is described in [8]. It keeps nonvolatile and volatile registers at the same level as the memory hierarchy, preventing loss of performance when storing the checkpoint registers at a higher level. To maintain the consistency of memory, it is possible that CKPs are performed when the execution of the processor encounters an instruction of type Store (STA). However, the STA instruction is very common and this can result in very frequent checkpoints, which is bad due to the power consumed during writing the data on the nonvolatile device. The approach presented in Fig. 9 (b) is described in [38]. It solves this problem by maintaining a duplicate cache: each time a data is modified in the cache its old value is stored in the Dirty Victim Cache (DVC), enabling periodic checkpoints. Of course, this approach requires a larger area.



IV. NEANDER

Neander [7] is a hypothetical multi-cycle processor well known in the literature and the target of this project. It was designed by researchers from the Federal University of Rio Grande do Sul (UFRGS) to introduce undergraduates to computer architecture. In its simplicity, it has a small set of 11 instructions and two status registers: negative (N) and zero (Z) used in conditional bypass operations. Fig. 10 [39] illustrates the Neander architecture.

Neander is formed by a memory, a decoder, an accumulator (AC), an arithmetic logic unit (ALU) that performs basic operations on the accumulator, a multiplexer (MUX) and a program counter (PC) which points to the current memory address. There are also a number of registers: a Memory Address Register (REM), a Memory Data Register (RDM)



Fig. 10 Neander architecture (adapted from [39]).

an Instruction Register (RI) and a State NZ Register (evaluates whether AC is negative or zero). All these components operate together through control signals determined by the Control Unit that guarantee the correct logic functioning.

Data is 8-bit wide and represented in two's complement. There is only one addressing mode: direct. All instructions have at least one byte, where the first four bits indicate the instruction opcode (operation code) and the last ones don't matter. In STA, LDA, ADD, AND, OR statements the second byte indicates the memory address of the operand. In the JMP, JN and JZ instructions, the second byte indicates the memory address to be pointed by the PC. In NOP, NOT and HLT statements there is not a second byte. Table II presents the instructions, respective opcodes and descriptions.

Table II. Instructions of Neander.				
Opcode	Mnemonic	Description		
0000	NOP	Initializes memory		
0001	LDA	AC ← Memory Data		
0010	STA	Memory \leftarrow AC		
0011	ADD	$AC \leftarrow AC + Memory Data$		
0100	OR	$AC \leftarrow AC OR Memory Data$		
0101	AND	AC ← AC AND Memory Data		
0110	NOT	$AC \leftarrow NOT AC$		
1000	JMP	PC ← Memory Data		
1001	JN	If $AC < 0$ then $PC \leftarrow$ Memory Data		
1010	JZ	If $AC = 0$ then $PC \leftarrow$ Memory Data		
1111	HLT	Ends execution		

To illustrate the Neander operation, the data flow associated with the ADD operation is described. When the PC points to the memory address where the ADD instruction is located, the opcode of the instruction will pass from memory to RI. After the operation is identified, the counter will add one more to its output so that the next memory address to be read will indicates the operand location. The memory will read the second byte of the instruction and the MUX will pass the memory out to the REM. After that, the memory will receive the operand address, and then the corresponding data will be transferred to the ALU and added to the accumulator content.

V. RELATED WORKS

Nonvolatile processors have become very valuable and requirements in the industry with the growth of IoT and low power applications. Due to their relevance to the market, NVPs have been widely investigated. The first fabricated nonvolatile processor was presented in 2012 by Wang et al. [6]. This work adopts an NVFF based on ferroelectric capacitors in a SLS structure.

After that, Koike et al. [40] fabricated a microprocessor unit (MPU) based on MRAM with a full nonvolatile architecture. In this work, STT-MRAMs are used for register file, instruction memory and data memory. Compared to FRAM technology, STT-MRAM has better reading and writing times as well a greater endurance. The proposed NVFFs are based on MTJ and MLS structure, providing a short 3microsecond entry/exit delay.

Sakimura et al. [41] presented a fully NV microcontroller based on three-terminal SpinRAM technology. Compared to the two-terminal STT-MRAM used in the previous work, the technology with three terminals provides greater data security since the reading and writing paths are different.

A RRAM-based NVP was presented in [42]. In comparison with MRAM, RRAM technology has similar read and write times. Traditionally, NVPs have 4 modes: store, restore, off and normal. This work proposes a fifth mode: retention. When the power failure is short, the processor remains in retention mode with a voltage of 0.4V, lower than the voltage in the normal mode, but enough to keep the system running. This scheme was considered cost-effective and energy efficient. The final results demonstrated that this NVP achieves 20 ns/0.45 nJ restore time/energy.

In a recent study, M. Natsu et al. presented a fabricated MOS/MTJ-hybrid nonvolatile microcontroller with an embedding STT-MRAM dedicated to IoT applications [43]. The manufactured chip achieved ultra-low power consumption and high-speed operation, essential for sensors powered by harvested energy. The results presented showed that this microntroller consumes 47.14 μ W power at an operating frequency of 200 MHz.

Compared to the works reported in the literature, we also use STT-MTJs and an SLS structure. The main contribution of this work is to implement the shadow register proposed in [8], which separates the nonvolatile from the volatile context. This scheme will be described in detail throughout this paper.

VI. WORK PROPOSAL

The initial step towards the nonvolatile implementation of Neander is to describe it in RTL to validate the description of the processor through mixed-signal simulations. This approach allows integrating combinational digital logic to analog flip-flops described in transistor-level. The aim is to replace these volatile FFs with a shadow register that maintains the two states: the volatile data used in normal operation and the nonvolatile data used to restore the system. The chosen approaches are described below.

A. RTL Implementation of Neander

To implement the nonvolatile Neander processor, the first step is to refactor Neander through the RTL design abstraction. RTL code is described as a synchronous circuit in terms of analog registers and combinational logic between them. Hardware description languages (HDLs) were used to create high-level representations of a circuit, making the complexity of the project more manageable. In our work, RTL design requires separate the combinational logic from the sequential logic. This approach is necessary because the registers are described in device-level.

The combinational logic is implemented through digital control signals and described by the Neander finite state machine. The control signals, in turn, define the selector signals of the combinational modules (mux and ALU) and act by deciding the inputs of each register, keeping all states consistent. In this step, the sequential logic comprises type D analog flip-flop (DFF) that act as registers for the system context. In the adopted configuration, the DFF passes the input on the rising clock edge and maintains the same output for the rest of the time, unless the reset receives '0', which will force the output to be '0'. However, the implementation of Neander through the RTL design abstraction uses volatile flip-flops to be validated, since their consistency was already established. As we move towards the nonvolatile implementation of Neander, the volatile flip-flops will be replaced by hybrid volatile/nonvolatile shadow registers, whose model is described below.

B. Proposed Nonvolatile Architecture

The nonvolatile flip-flop is an essential part of the developing processor, being responsible for storing and restoring the processor states. This stage of development aims to validate a functional shadow register to replace the analog DFFs in the RTL design described above, as depicted in Fig. 11.

The hybrid structure takes advantage of the high-speed computing of volatile technology, reserving the nonvolatile circuit for backup and system restoration, prioritizing high performance and low power computing. The REG_NOW is a conventional CMOS volatile flip-flop that stores current data. The REG_CKP is the nonvolatile sub-register composed of a read/write driver connected to a FF. The SV and SNV are respectively the output of REG_NOW and the output of REG_CKP. The REG_NOW copies the value of input D whenever enable (EN) is on. If the RLB operation occurs while the EN is off, REG_NOW must copy the output value from the nonvolatile register. REG_CKP receives the signal CKP, which triggers the checkpoint operation: the nonvolatile register will store input D if the EN is on and, if not, it copies the output value of REG_NOW. The RLB signal is activated to restore the system through the data stored in the checkpoint register.



Fig. 11 a) Volatile register and b) Shadow register from [8].

The proposed read/write driver is shown in Fig. 12. It is a self-referenced cell based on Black & Das circuit that uses the STT-MTJ as nonvolatile device. The functioning of this cell is based on writing complementary data on the two MTJs, represented in the circuit as variable resistors. Regarding the driver variables: MQ is the output and MQN is its complement, MW signal means magnetic write and activates data writing, the D signal is for the input data and the MRE is the magnetic read, which triggers the reading. The activation of the MRE approaches the voltages of the complementary outputs long enough for the signals to stabilize, moving them to a metastable state. After the MRE is turned off, the outputs are updated to the correct result due to the voltage difference caused by the resistor configuration of the complementary STT-MTJs. During the reading, the MW is turned off, activating the virtual ground while the outputs of the tri-state cells remain in a high impedance state.



Fig. 12 Black & Das circuit published in [8].

However, the STT-MTJ could easily be replaced by another nonvolatile technology, such as any of those presented in Section 2. The structure of the flip-flop could also be changed to prioritize other advantageous characteristics.

VII. EXPERIMENTAL SETUP

This section presents the techniques and technologies used within the scope of this work.

A. RTL Implementation

The combinational implementation of the Neander was based on assign statements. The equation for assign statement is presented below: whenever the expression on the right undergoes any change, the expression on the left is immediately changed.

dimgrayrgb0.41, 0.41, 0.41 airforcebluergb0.36, 0.54, 0.66 cadetrgb0.33, 0.41, 0.47 darkslategrayrgb0.18, 0.31, 0.31

darkslategray

assign < net_expression > = < expression of different signals or constant value >

The assign statements makes it possible to determine the load signals and the inputs of the registers using the ternary conditional operator. A simple example is the operation of the Instructions Register (RI) described in combinational logic presented below, where: loadRI represents the load signal that decides the input of the register; state corresponds to one of the eight states implemented that guarantee the correct operation of the Neander; smem is the output of memory; e_ri is the input of the RI and sri is the output. In the second logic state of the Neander, the loadIR receives a high signal so the output of the memory is loaded in this register.

This output corresponds to the opcode of the instruction that must be executed. When the loadRI is off, the input of the DFF must be kept the same. Assign statements were also used in decisions corresponding to ALU and MUX.

darkslategray

assign loadRI = (state == 2) ? 1 : 0; assign e_ri = (loadRI == 1) ? smem : sri;

The analog DFFS cells were implemented using NCSU FreePDK 45nm synthetic CMOS technology [44]. Analog to digital converter (ADC) and digital to analog converter (DAC) were used as bridges between the analog modules of the sequential part and the digital modules of the combinational part.

For verification, Cadence Design Systems Incisive and Spectre toolsets [45] were used. Due to the mix of analog and digital signals, all modules except ALU, MUX and memory were implemented in Verilog-AMS, which includes analog and mixed-signal extensions (AMS).

B. Shadow Register Description

The description of the shadow register scheme was performed at device level, employing a SPICE netlist [45]. We used the NCSU FreePDK 45nm CMOS technology, and the behavior of the nonvolatile STT-MTJ device was simulated using the SPICE model from Harms et al. [28]. The signals CKP, RLB, EN and MRE were implemented as external pins to test the behavior of the circuit.

Given the delicate nature of the analog behavior of the circuit and the STT-MTJ model, several simulations were observed in order to arrive at timing and sizing constraints imposed by the applied technology. For that, the Cadence Design Systems Spectre tool was used.

VIII. RESULTS

The initial steps for the nonvolatile implementation of Neander were duly simulated and validated. The results of the RTL description and the behavior of the shadow register are shown below.

A. RTL Design

All the 11 operations of the combinational Neander implementation were successfully validated through the RTL design. As the ADD operation was briefly explained in Section IV, it was chosen for a representative simulation.

Table III presents the part of the memory used in this simulation and a brief explanation of each line. Fig. 13 shows the representative simulation: the operation of adding the value 248 (-8 in signed decimal) to the zeroed accumulator. Relevant signals that facilitate the reader's understanding were considered. The signals are inputs (e_ac, e_pc and e_ri) and outputs (sac, snz, spc, sri) of the registers, load signals (loadAC, loadNZ, loadPC and loadRI), the output of the memory (smem), the State and the incPC signal which is triggered when the PC output must be increased by one. The data are represented in unsigned decimal in the simulation. In Table III the data are represented in binary.

Fig. 14 represents the sequential logic of the Neander through a mixed-signal simulation, showing the correctness of the DFF and ADC executions. In this simulation, a_clock



Fig. 13 Mixed-signal simulation of ADD operation.

is the analog clock signal and the $affd_ac[7]$ is the first bit in the analog input of the AC that appears on the output in $affo_ac[7]$ and converted to a digital signal in ffo_ac[7] after a clock cycle.



Fig. 14 Mixed-signal simulation validating the DFF and ADC executions.

B. Shadow Register Behavior

First, the read / write driver was tested separately from the shadow register logic as it has the most delicate behavior of the circuit. Its validation is shown in the purely analog simulation presented in Fig. 15, which performed all possible transactions to ensure the correct operation. It should be noted that the STT-MTJs were initially connected and read with the configuration to store the logical '0' data. The MW magnetic writing is connected to write input D in the circuit and the MRE triggers the reading, respecting the times necessary for the MQ and MQN nodes to equalize. Although the MW signal interferes with the MQ and MQN output signals, it turns out that this does not impair the expected behavior.

The simulations presented in Fig. 16 and Fig. 17 validate the expected behavior of the sub-registers REG_NOW and REG_CKP, respectively. In Fig. 16, REG_NOW internal signals are shown: EV is the internal enable signal, which decides between copying the input DV to IV – updating the stored value – or holding the current QV, which is the volatile output. In Fig. 17, DNV is the internal input. The remaining variables were previously discussed. Although the MRE interferes with the MQ output signal, it does not affect the



operation, as the internal reading is not performed simultaneously with the RLB.



After that, we coupled the volatile and nonvolatile subregisters to the remaining modules and simulated the ensemble. An example of the results obtained through simulation is shown in Fig 18. The goal of this particular simulation is to verify that the volatile sub-register performs as expected, while the nonvolatile sub-register is storing a checkpoint. Later on, a rollback signal is issued, replacing the volatile state with the checkpoint content and carrying on the execution. In this simulation, the volatile sub-register initially stores logic '0', as seen by its output SV, and so does the nonvolatile register (output SNV). Input data remains '1' through the first half of the period. While the register is not enabled, the volatile sub-register keeps storing zero, and the global output Q remains zero. Once EN is turned on, SV follows D after the clock's next rising edge, and so does Q, which remains connected to SV. A checkpoint (input CKP) issued near the 10 ns triggers the nonvolatile writing process. This process is completed around the 20 ns mark, and SNV now corresponds to the input D at the time CKP was issued. Around 32 ns, D changes to a logic '0', and so does SV and Q after the clock's next rising edge. The register is disabled near the 35 ns, thus preventing D from affecting the output.



Another checkpoint (input CKP) issued near the 40 ns mark, and the previously stored checkpoint is replaced. Finally, at the 55 ns mark, RLB is activated. SNV is copied to the volatile sub-register, completing the rollback operation. A logic '0' remains in the global output Q, as expected.



Fig. 18 Simulation of the shadow register proposed in Fig. 11.

IX. FINAL CONSIDERATIONS

Nonvolatile processors offer zero standby power and instant on/off. Due to these features, NPVs have the potential to assist the growing demand for reduced power consumption which has grown exponentially in the market over the past few years. In this work, a combinational implementation based on RTL was presented in order to validate a mixedsignal simulation of Neander with analog volatile flip-flops described at transistor-level. This approach sets the stage for receiving the hybrid volatile/nonvolatile shadow register scheme described in this paper. In the future, the shadow register, already verified, will replace the DFFs in the implementation of Neander. In addition, a review of emerging nonvolatile memories and basic concepts about nonvolatile processors was presented. Furthermore, related works have been reported for comparison purposes.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the National Council for Scientific and Technological Development (CNPq) and the State of Rio Grande do Sul Research Foundation (FAPERGS).

REFERENCES

- A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutori*als, vol. 17, no. 4, pp. 2347–2376, 2015.
- [2] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.
- [3] U. Dudko and L. Overmeyer, "Optical wake-up from power-off state for autonomous sensor nodes," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3225–3232, 2021.
- [4] Y. Liu, Z. Li, H. Li, Y. Wang, X. Li, K. Ma, S. Li, M.-F. Chang, S. John, Y. Xie, J. Shu, and H. Yang, "Ambient energy harvesting nonvolatile processors: From circuit to system," in 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015, pp. 1–6.
- [5] F. Su, K. Ma, X. Li, T. Wu, Y. Liu, and V. Narayanan, "Nonvolatile processors: Why is it trending?" in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017.* IEEE, 2017, pp. 966–971.
- [6] 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," in 2012 Proceedings of the ESSCIRC (ES-SCIRC). IEEE, 2012, pp. 149–152.
- [7] R. F. Weber, Fundamentos de arquitetura de computadores. Sagra Luzzatto, 2000.
- [8] R. M. Brum, "On the design of hybrid cmos and magnetic memories, with applications to reconfigurable architectures," Ph.D. dissertation, University of Montpellier, Montpellier, France, 2014.
- [9] J. Meena, S. Sze, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale Research Letters*, vol. 9, pp. 1–33, 09 2014.
- [10] S. Yu and P. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, 2016.
- [11] J. Portal, M. Bocquet, M. Moreau, H. Aziza, D. Deleruyelle, Y. Zhang, W. Kang, J.-O. Klein, Y. Zhang, C. Chappert, and W. ZHAO, "An overview of non-volatile flip-flops based on emerging memory technologies," *Journal of Electronics Science and Technology*, vol. 12, pp. 173–181, 06 2014.
- [12] T. Eshita, W. Wang, K. Nomura, K. Nakamura, H. Saito, H. Yamaguchi, S. Mihara, Y. Hikosaka, Y. Kataoka, and M. Kojima, "Development of highly reliable ferroelectric random access memory and its internet of things applications," *Japanese Journal of Applied Physics*, vol. 57, p. 11UA01, 11 2018.
- [13] Xiao-Hong Du and Bing Sheu, "Modeling ferroelectric capacitors for memory applications," *IEEE Circuits and Devices Magazine*, vol. 18, no. 6, pp. 10–16, 2002.
- [14] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.
- [15] R. F. Freitas and W. W. Wilcke, "Storage-class memory: The next storage system technology," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 439–447, 2008.

- [16] S. W. Fong, C. M. Neumann, and H. P. Wong, "Phase-change memory—towards a storage-class memory," *IEEE Transactions on Electron Devices*, vol. 64, no. 11, pp. 4374–4385, 2017.
- [17] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, p. 24–33, Jun. 2009.
- [18] K. Bourzac, "Has intel created a universal memory technology? [news]," *IEEE Spectrum*, vol. 54, no. 5, pp. 9–10, 2017.
- [19] Y. Fujisaki, "Review of emerging new solid-state non-volatile memories," *Japanese Journal of Applied Physics*, vol. 52, p. 040001, 04 2013.
- [20] Z. Wang, S. Ambrogio, S. Balatti, S. Sills, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Postcycling degradation in metal-oxide bipolar resistive switching memory," *IEEE Transactions on Electron Devices*, vol. 63, no. 11, pp. 4279–4287, 2016.
- [21] Meng-Fan Chang, Albert Lee, Chien-Chen Lin, Mon-Shu Ho, Ping-Cheng Chen, Chia-Chen Kuo, Ming-Pin Chen, Pei-Ling Tseng, Tzu-Kun Ku, Chien-Fu Chen, Kai-Shin Li, and Jia-Min Shieh, "Read circuits for resistive memory (reram) and memristor-based nonvolatile logics," in *The 20th Asia and South Pacific Design Automation Conference*, 2015, pp. 569–574.
- [22] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, "Design implications of memristor-based rram cross-point structures," in 2011 Design, Automation Test in Europe, 2011, pp. 1–6.
- [23] S. P. Mohanty, "Memristor: From basics to deployment," *IEEE Potentials*, vol. 32, no. 3, pp. 34–39, 2013.
- [24] S. Pal, V. Gupta, W. Ki, and A. Islam, "Design and development of memristor-based rram," *IET Circuits Devices & Systems*, vol. 13, pp. 548 – 557, 07 2019.
- [25] P.-Y. Chen, Z. Li, and S. Yu, "Design tradeoffs of vertical rram-based 3-d cross-point array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 12, pp. 3460–3467, 2016.
- [26] M. Zidan, H. Fahmy, M. Hussain, and K. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, vol. 44, p. 176–183, 02 2013.
- [27] W. J. Gallagher and S. S. P. Parkin, "Development of the magnetic tunnel junction mram at ibm: From first junctions to a 16-mb mram demonstrator chip," *IBM Journal of Research and Development*, vol. 50, no. 1, pp. 5–23, 2006.
- [28] J. D. Harms, F. Ebrahimi, X. Yao, and J.-P. Wang, "Spice macromodel of spin-torque-transfer-operated magnetic tunnel junctions," *IEEE transactions on electron devices*, vol. 57, no. 6, pp. 1425–1430, 2010.
- [29] L.-B. Faber, W. Zhao, J.-O. Klein, T. Devolder, and C. Chappert, "Dynamic compact model of spin-transfer torque based magnetic tunnel junction (mtj)," in 2009 4th International Conference on Design & Technology of Integrated Systems in Nanoscal Era. IEEE, 2009, pp. 130–135.
- [30] H. Noguchi, K. Ikegami, S. Takaya, E. Arima, K. Kushida, A. Kawasumi, H. Hara, K. Abe, N. Shimomura, J. Ito, S. Fujita, T. Nakada, and H. Nakamura, "7.2 4mb stt-mram-based cache with memory-accessaware power optimization and write-verify-write / read-modify-write scheme," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), 2016, pp. 132–133.
- [31] T. Na, K. Ryu, J. Kim, S. H. Kang, and S.-O. Jung, "A comparative study of stt-mtj based non-volatile flip-flops," in 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013). IEEE, 2013, pp. 109–112.

- [32] N. Sakimura, T. Sugibayashi, R. Nebashi, and N. Kasai, "Nonvolatile magnetic flip-flop for standby-power-free socs," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 8, pp. 2244–2250, 2009.
- [33] W. Black and B. Das, "Programmable logic using giantmagnetoresistance and spin-dependent tunneling devices," *Journal of Applied Physics*, vol. 87, pp. 6674–6679, 05 2000.
- [34] T. Hanyu, T. Endoh, D. Suzuki, H. Koike, Y. Ma, N. Onizawa, M. Natsui, S. Ikeda, and H. Ohno, "Standby-power-free integrated circuits using mtj-based vlsi computing," *Proceedings of the IEEE*, vol. 104, no. 10, pp. 1844–1863, 2016.
- [35] J. Zheng, Z. Zeng, and Y. Zhu, "Memristor-based nonvolatile synchronous flip-flop circuits," in 2017 Seventh International Conference on Information Science and Technology (ICIST), April 2017, pp. 504– 508.
- [36] J.-M. Choi, C.-M. Jung, and K.-S. Min, "Pcram flip-flop circuits with sequential sleep-in control scheme and selective write latch," *JSTS: Journal of Semiconductor Technology and Science*, vol. 13, 02 2013.
- [37] W. Yu, S. Rajwade, S. Wang, B. Lian, G. E. Suh, and E. Kan, "A non-volatile microcontroller with integrated floating-gate transistors," in 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), 2011, pp. 75–80.
- [38] P. Michele and L. Regis, "Effective checkpoint and rollback using hardware/os collaboration," in 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007), 2007, pp. 370–378.
- [39] F. R. Schneider, R. E. Poli, D. Barden, D. K. Leonel, D. A. Fiorentin, F. M. Trindade, F. S. de Vasconcellos, M. C. d. B. O. Neto, and R. P. Ribas, "Design of a 4-bit processor for evaluating of the e/d nmos technology from ccs/unicamp," in 4th Microelectronics Students Forum, 2004.
- [40] H. Koike, T. Ohsawa, S. Ikeda, T. Hanyu, H. Ohno, T. Endoh, N. Sakimura, R. Nebashi, Y. Tsuji, A. Morioka *et al.*, "A power-gated mpu with 3-microsecond entry/exit delay using mtj-based nonvolatile flip-flop," in 2013 IEEE Asian Solid-State Circuits Conference (A-SSCC). IEEE, 2013, pp. 317–320.
- [41] 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," in 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014, pp. 184–185.
- [42] Z. Wang, Y. Liu, A. Lee, F. Su, C. Lo, Z. Yuan, J. Li, C. Lin, W. Chen, H. Chiu, W. Lin, Y. King, C. Lin, P. Khalili Amiri, K. Wang, M. Chang, and H. Yang, "A 65-nm reram-enabled nonvolatile processor with time-space domain adaption and self-write-termination achieving > 4× faster clock frequency and > 6× higher restore speed," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 10, pp. 2769–2785, 2017.
- [43] M. Natsui, D. Suzuki, A. Tamakoshi, T. Watanabe, H. Honjo, H. Koike, T. Nasuno, Y. Ma, T. Tanigawa, Y. Noguchi, M. Yasuhira, H. Sato, S. Ikeda, H. Ohno, T. Endoh, and T. Hanyu, "A 47.14- μW 200-mhz mos/mtj-hybrid nonvolatile microcontroller unit embedding stt-mram and fpga for iot applications," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 11, pp. 2991–3004, 2019.
- [44] J. E. Stine, I. Castellanos, M. Wood, J. Henson, F. Love, W. R. Davis, P. D. Franzon, M. Bucher, S. Basavarajaiah, J. Oh, and R. Jenkal, "Freepdk: An open-source variation-aware design kit," in 2007 IEEE International Conference on Microelectronic Systems Education (MSE'07), 2007, pp. 173–174.
- [45] K. S. Kundert and P. Gray, *The Designer's Guide to Spice and Spectre*. USA: Kluwer Academic Publishers, 1995.