

# Spintronic Normally-off Heterogeneous System-on-Chip Design

Anteneh Gebregiorgis, Rajendra Bishnoi and Mehdi B. Tahoori

Chair of Dependable Nano Computing (CDNC), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Email: {anteneh.gebregiorgis, rajendra.bishnoi, mehdi.tahoori}@kit.edu

**Abstract**—One of the major challenges in device down-scaling is the increase in the leakage power, which becomes a major component in the overall system power consumption. One way to deal with this problem is to introduce the concept of normally-off instant-on computing architectures, in which the system components are powered off when they are not active. An associated challenge is the back-up and restoration of system states, which in turn can introduce additional costs that erode some of the gains. A promising alternative is the use of non-volatile storage elements in the *System-on-Chip* (SoC) design which can instantly power-down and retain their values. In this work, we show how we can design a normally-off SoC by exploiting non-volatile latches, flip-flops and registers. The idea is to design a hybrid architecture containing conventional CMOS bistables as well as different flavors of spintronic-based non-volatile storage elements, to balance performance, area, and energy efficiency.

## I. INTRODUCTION

With ever-increasing demand for more functionality, higher performance and lower cost for semiconductor products, the *leakage power*, the power consumed statically when devices are not operating, has become one of the most pressing design issues [1]. Until recently, it was considered as a second order effect, however, nowadays it starts to dominate the total power consumption of modern *System-on-Chips* (SoCs) [2]. Therefore, reducing leakage power is extremely important, especially for the design of battery operated hand-held devices for *Internet of Things* (IoT) applications. Non-volatile on-chip storage technologies can play a vital role to deal with this issue by enabling their normally-off computing capabilities.

Several non-volatile processor designs have been already proposed using various non-volatile technologies such as *Flash*, *Ferroelectric Random Access Memory* (FRAM), *Resistive Random Access Memory* (RRAM), *Phase-Change Random Access Memory* and *spintronic* technologies [3–12]. Among all, spintronic technology, in particular *Spin Orbit Torque* (SOT), is the most promising candidate as it has an edge over other non-volatile technologies in terms of fast accesses, high endurance and better scalability [13, 14]. In addition to that, this technology has various other advantageous features such as high density, CMOS compatibility and immunity to radiation induced soft-errors [15]. Thus, SOT is the best suited technology for the non-volatile processor design.

An SOT-based non-volatile microcontroller, comprising of a 16 bit RISC architecture CPU core, is proposed in [16]. In this work, a non-volatile flip-flop is designed using SOT devices for the data backup purpose, which is known as *shadow flip-flop architecture*. In this architecture, the data from the conventional CMOS flip-flop is copied to the non-volatile

component during the standby mode before the power-gating, and the same data is restored back during the wake-up phase. Such architectures are very effective for the leakage reduction because the entire combinational logic part can completely be power-gated unlike the conventional CMOS flip-flop based designs. However, in these designs, there is a considerable delay overhead due to the constant switching between the backup and active components. Therefore, these shadow flip-flop based designs are only applicable for long sleep durations. On the other hand, a *Non-volatile Non-shadow* flip-flop architecture is proposed in [17], in which SOT devices are employed as active components that makes it suitable for shorter sleep durations as well. Nevertheless, the high switching rate in such designs can erode the overall energy gain, since a relatively high amount of current has to pass through the storing cell to make it switch. Therefore, to deal with the leakage power, dynamic energy and performance challenges, a hybrid design approach based on the combination of various volatile and non-volatile storage elements, is required.

In this paper, we propose a novel normally-off SoC architecture using SOT-based non-volatile components. In this design, we have distributed conventional volatile CMOS flip-flops as well as non-volatile shadow and non-shadow flip-flop architectures in such a way that it delivers overall high energy efficiency. This is achieved using two steps: 1) Architecture level exploration to isolate the storing elements in a designs that do not required any backup (i.e., non state holding flip-flops), and such cells will remain in CMOS technology to save the dynamic energy. 2) Replacing those storage elements that need backup (i.e., state holding flip-flops) using either shadow or non-shadow non-volatile flip-flops based on their switching activity rate and power-down duration. Hence, the state holding flip-flops with higher switching activity rates are implemented using shadow flip-flop technique, while the less active once are implemented using non-shadow flip-flops. The state holding elements are identified using architectural level X-analysis of the netlist and are replaced by the corresponding non-volatile shadow/non-shadow flip-flops depending on their switching activities. It should be noted that, this is done in a per core basis within an SoC and hence, only CMOS and shadow or CMOS and non-shadow flip-flops are used in a single core. Simulation result shows that the SoC can be power-gated more than 65% of the time, which leads to more than 2X energy reduction, using the proposed approach.

The rest of the paper is organized as follows: basics of the spintronic SOT technology and related non-volatile processor architectures are explained in Section II. Section III describes the non-volatile storing components and the architectural de-

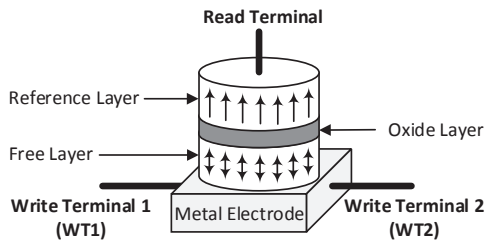


Fig. 1. Spin Orbit Torque based storage cell

tails of the proposed non-volatile SoC design. The simulation results are explained in Section IV. Finally, Section V concludes the paper.

## II. BACKGROUND

### A. Spin Orbit Torque Technology

In the *Spin Orbit Torque* (SOT) technology, a *Magnetic Tunnel Junction* (MTJ) cell, as shown in Figure 1, is the storing device where the data is stored as resistance states. The SOT based MTJ cell comprises of two ferromagnetic layers separated by a thin oxide layer and a metal electrode. Out of these two ferromagnetic layers, one has always a fixed magnetic orientation, known as *Reference Layer* (RL). In contrast, the magnetic orientation of the other layer can be freely rotated, that is referred to as *Free Layer* (FL). When the magnetic orientation of the two ferromagnetic layers are parallel to each other ('P' configuration), it exhibits a low resistance value. On the other hand, when the magnetic orientation of those two layers are anti-parallel to each other ('AP' configuration), it exhibits a high resistance value. The MTJ device has total three terminals, out of which two are write terminals and one is read terminal, as shown in the figure. The write current flows between the two write terminals through the metal electrode, changing the magnetic orientation of the FL. The *Rasbha effect* [18] or the *Spin Hall Effect* [19] are responsible for such switching phenomenon. Furthermore, a low read current, that flows through the MTJ stack, is sensed using a *sense amplifier* to read the stored value. Compared to well known spintronic *Spin Transfer Torque* (STT) devices, SOT devices switch faster and consume less energy [20]. Unlike STT, the read and write current paths in SOT are isolated, reducing the possibility of the read disturb to virtually zero. Furthermore, it possesses higher endurance and better reliability as the write current does not pass through the oxide layer.

### B. Related work

The non-volatile processing capability enables normally-off computation as opposed to conventional SoC designs, using which the system states can be maintained in the absence of the power supply. In the traditional *save and restore* technique [21, 22], a mature high density flash is employed for the data backup where the data is stored separately in the memory array. But it incurs huge delay for the data transfer. Therefore, non-volatile solutions where the data backup can be done locally within the flip-flop or register, have been gaining popularity. In this regard, several non-volatile processors are proposed using various non-volatile technologies. For instance, an FRAM based processor is proposed in [3] with zero standby power with the feature of the automatic system backup during power failures, using a reconfigurable voltage detection

system. Other FRAM based solutions are also proposed for the storage at the SoC level [4, 5]. However, the store latency in FRAM technology is very high [23]. A few RRAM based processor designs are proposed [6, 7], but these suffer from very limited endurance [23].

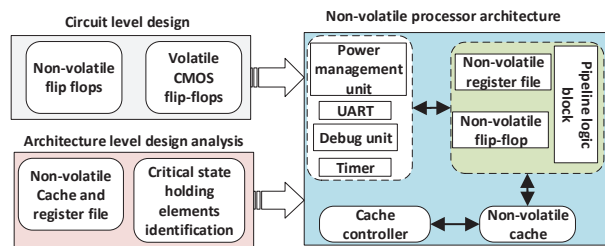
STT technology has almost infinite endurance and relatively better storage latency [14, 24, 25]. Many STT-based non-volatile processor designs have been demonstrated recently [8–12]. In [8], a power-gated microprocessor unit using STT technology is proposed in which non-volatile flip-flop circuits are employed to store the internal pipeline and program counter registers. Moreover, a compare and compress recovery architecture using STT is proposed in [9] to reduce the area of non-volatile registers. Various checkpointing schemes are also proposed using this technology in which non-volatile components are used to store the state of the system periodically during the execution of the application, so that the system can rollback in case of any system failure [10–12]. Similarly, an SOT-based non-volatile microcontroller design is proposed for standby-power-critical applications [16]. However, all these works have employed the non-volatile elements for the backup purpose only and hence, the data transfer for backup and restore will induce significant delay overhead.

To address this issue and improve the overall energy-efficiency, we propose a data transfer free non-volatile heterogeneous SoC architecture using combination of conventional CMOS and different flavors of non-volatile flip-flop designs.

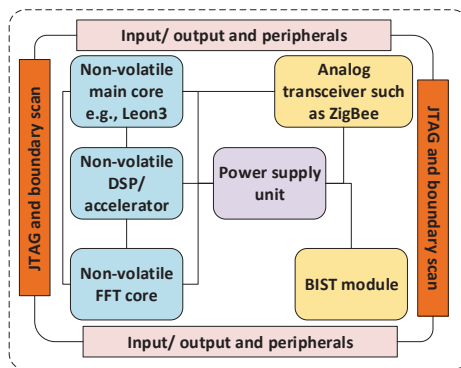
## III. NON-VOLATILE SOC ARCHITECTURE

### A. Overview

Compared to the volatile processors, non-volatile processors using SOT technology have the following main advantages: 1) They are more energy-efficient as they can be shut



(a) Non-volatile processor architecture



(b) Non-volatile SoC architecture

Fig. 2. Cross-layer non-volatile processor design approach for normally-off heterogeneous SoC architecture

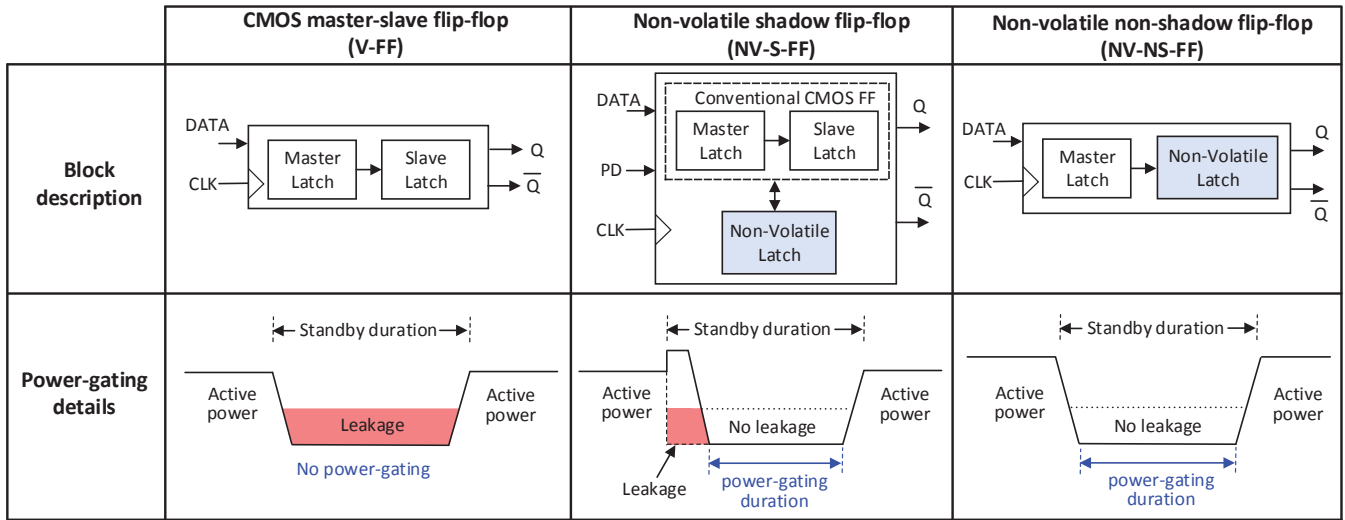


Fig. 3. Illustration of flip-flop architectures employed in our framework and their behavior in the standby mode.

down to achieve zero leakage without losing their state. 2) Better endurance and resiliency to power failures, and 3) Fast backup and restore time. These advantages make SOT-based non-volatile processors as a promising alternative for highly energy-constrained systems. In this work, we present a cross-layer non-volatile processor design schemes for energy-efficient heterogeneous SoC design. In the proposed non-volatile SoC architecture, the digital components (processing elements) are designed to be non-volatile by using different non-volatile storage elements. For this purpose, different non-volatile flip-flop designs are explored at the circuit level and a selective state preservation mechanism is investigated at the architecture level.

Figure 2 shows the proposed non-volatile processor as well as the SoC design methodology. As it can be seen from the figure, a non-volatile processor is designed by performing circuit level non-volatile flip-flop design and architectural level analysis to determine the critical state holding elements for non-volatility (see Figure 2(a)). The non-volatile design scheme given in Figure 2(a) is then used to design a non-volatile heterogeneous SoC as shown in Figure 2(b). The heterogeneity of the SoC in our work is not only defined by component variation, but also by the usage of different non-volatile flip-flop designs among the non-volatile cores within the SoC.

### B. Flip-flop designs

Flip-flops are the basic building blocks for any SoC designs that are primarily used for state holding as well as fixing timing boundaries. The non-volatile technologies are introduced for these flip-flop designs in order to reduce the leakage consumption. In our framework, we have classified flip-flops into three categories, namely, *conventional CMOS-based flip-flop*, *non-volatile shadow flip-flop* and *non-shadow flip-flop*, depending on the design requirements. For non-volatile component design, we have employed SOT-based storage devices as they are reliable, fast and consumes less access energy compared to spintronic STT devices [14, 26]. The details of these flip-flop architectures (as shown in Figure 3) are explained as follows:

**CMOS master-slave flip-flop architecture (V-FF):** This is

the conventional master-slave flip-flop design and depending on the level of the clock signal, the input data is either stored or propagated to the output. Such flip-flop design uses cross coupled inverters to store the data, hence always require a supply voltage to retain the information. Therefore, in our framework, we can employ these flip-flops in the parts of the design where the data backup is not required, e.g. branch history tables, and zero registers. This architecture has lower area, less active energy and faster access compare to the non-volatile flip-flop architectures.

**Non-volatile shadow flip-flop architecture (NV-S-FF):** In this architecture, a non-volatile component is connected to the aforementioned conventional CMOS-based flip-flop architecture, so that the data backup is performed during the standby mode. The data backup is activated based on the *Power-Down* (PD) pin, and once it is done, the entire logic block can be power-gated completely. During the wake-up mode, the data is first restored back to the conventional CMOS flip-flop design, so that the normal operation can be safely resumed. Using this design, there is a significant reduction in leakage, nevertheless this design is not applicable for short power-down durations because of the large delay of the data movement between backup and active components.

**Non-volatile non-shadow flip-flop architecture (NV-NS-FF)** [26]: In this architecture, the non-volatile element is the active component of the flip-flop, as shown in Figure 3. This design can perform power-gating instantly, which means it enables efficient normally-off computing by allowing very aggressive power-gating, makes it applicable for short standby periods as well. Moreover, the switching delay in this architecture is not contributing to the critical path delay because the writing into non-volatile storing devices is done in parallel to the propagation of the data to the output ports. Therefore, it has similar timing characteristics as that of the conventional CMOS-based flip-flop design. Since the storing cells are active components, significant amount of current has to flow through these cells in order to switch the stored value, resulting in an increase in the overall dynamic energy of the cell.

To address the endurance and high dynamic energy prob-

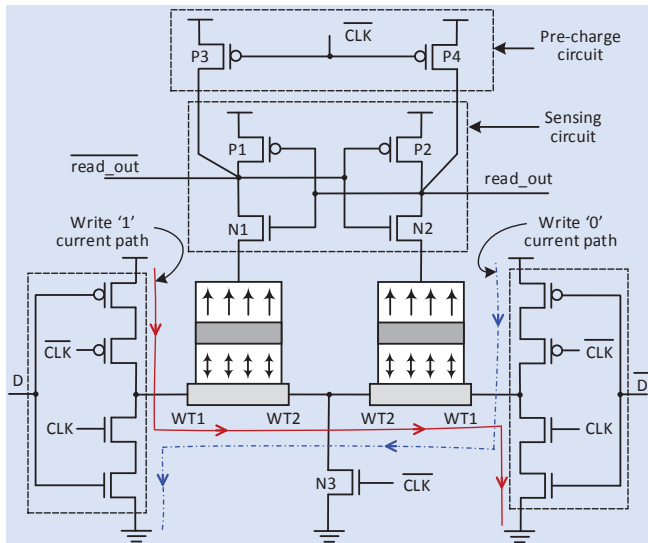


Fig. 4. The non-volatile latch design employed in the shadow and non-shadow flip-flop architectures.

lems associated with frequent writing into the SOT devices, we introduce a *write avoidance scheme* for these non-volatile flip-flop designs as proposed in [26]. Using this method, we can avoid rewriting the same value into the non-volatile component to save the dynamic energy of the system. This means that the redundant operation can be avoided by comparing (using XOR) the value to be written with the already stored value.

### C. Non-volatile latch design

The schematic of the SOT-based latch design that is used for both NV-S-FF and NV-NS-FF architectures is shown in Figure 4. As shown in the figure, the non-volatile latch consists of two MTJs, a read and a write control circuitry. Those two MTJs should connect in a way that they always hold opposite magnetic orientations and hence opposite resistance states. The read operation is performed using a pre-charged based sensing mechanism, which is fast and energy efficient [27, 28]. On the other hand, the write operation is performed by using a tristate inverter. The current path directions to store-1 and store-0 are shown in the figure. The read and write mechanisms are controlled using the CLK signal for non-shadow flip-flop architecture whereas, for shadow designs, CLK signal is ANDed to the PD signal to control the read and write mechanisms.

### D. Architectural level exploration

A straightforward way of designing a non-volatile processor is by replacing all CMOS-based flip-flops and register files with their SOT-based non-volatile variants [3]. However, since only a fraction of flip-flops and registers are the actual state holding elements, replacing every flip-flop and register by the non-volatile variant is highly energy-inefficient approach. Moreover, we have observed that some of the non state holding flip-flops and registers have higher switching activity. Hence, making them non-volatile results in a higher energy consumption due to the higher write energy and relatively longer write latency of non-volatile technologies. Therefore, we perform an architectural exploration to determine the state holding elements of a processor and replace them with non-volatile variants. In order to reduce the dynamic energy, the non

**Algorithm 1:** X-analysis using gate-level simulation to identify state holding flip-flops

---

```

1 function: X-analysis;
   Input : gate-level netlist, checkpoint interval, workload
           and random input
   Output: State holding flip-flops
2 while !end_of_sim_time do
3   for Each checkpoint do
4     old_ff_states ← extract_state(flip-flop list);
5     flip-flops with X-value ← non state holding;
6     Change_ff_state ← Random(NOP, Pipeline
7       flush, random value);
8     new_ff_states ← extract_state(flip-flop list);
9     if Simulation fails then
10      candidate_ff ←
11        value_changed_ff(new_ff_states,old_ff_states);
12      new_ff_states ← old_ff_states;
13      continue to next checkpoint;
14    end
15    state_holding_ff ← candidate_ff;
16 end
17 return state_holding_ff;

```

---

state holding elements are implemented using the conventional CMOS V-FF. For this purpose, first we develop a backup policy to determine the crucial elements and X-analysis is performed afterwards to identify those crucial elements, and finally, they are replaced with non-volatile variants.

1) *Backup policy:* The most important part of designing a non-volatile processor is the decision of what to backup and how to backup. If one uses trivial decision at this stage, then the resulting design will be highly energy inefficient. For different micro-architectures such as pipelined, non-pipelined, in-order and out-of-order cores, the backup policy (what to backup) should cover all the important state defining elements such as program counter and special purpose registers. This phase must guarantee that no state and computation progress is lost during the power-down.

Once the crucial state elements are identified, different backup mechanisms can be adopted to store the processor state during power-down period. For instance, authors in [3] adopted a mirrored backup mechanism to store the state of the processor during power-down. In their approach, the content of the regular registers and flip-flops are copied into the non-volatile flip-flops before power-down and the content is copied back during the power-up. However, this approach has significant area overhead and several cycles are wasted in transferring data to and from the non-volatile flip-flops.

2) *X-Analysis for selective state preservation:* In order to identify the state holding flip-flops of a processor, we conduct an extensive X-analysis (gate-level simulation) in combination with various techniques such as pipeline flushing, No Operation (NOP) insertion and randomized initialization of flip-flops. These techniques are useful to identify the state holding flip-flops of a processor as they enable us to change, the processor state and control values of all flip-flops.

Algorithm 1 presents the gate-level simulation based state holding flip-flop analysis (X-analysis) used in this work.

TABLE I. SIMULATION SETUP DETAILS

| Parameters                | Value   |
|---------------------------|---|
| VDD, Temperature, Process | 1.0V, 27°C, Typical   |
| CMOS Technology           | TSMC 65 nm GP   |
| Damping factor            | 0.5   |
| Thermal stability factor  | 104   |
| Bias magnetic field       | 0.1 T   |
| Saturation Magnetization  | $1.1 \times 10^6$ A/m                                       |
| Metal Electrode           | $100 \text{ nm} \times 50 \text{ nm} \times 2.5 \text{ nm}$ |
| Critical current          | 59 $\mu$ A  |
| 'AP'/P' resistance        | 10 K $\Omega$ /5 K $\Omega$                                 |

SoC design configuration

| Core    | # gates | # flip-flops |
|---------|---------|--------------|
| Leon3mp | 37030   | 2364         |
| FFT     | 10414   | 705          |
| AES     | 21725   | 530          |
| ZigBee  | 6079    | 480          |

Inputs to the algorithm are gate-level netlist of the design, selected workloads, checkpoint intervals and random inputs. The checkpoint interval is a set of time points in which the state of the processor flip-flops are checked by applying the above mentioned techniques. The output of the algorithm is the set of state holding flip-flops. These flip-flops are defined as flip-flops in which the controlled change in their state can affect the processor state and the gate-level simulation flow. The algorithm first simulates the workload application using the gate-level netlist. Then, for each checkpoint, it extracts the current state (i.e., `old_ff_state`) of the flip-flops. Flip-flops with X values are excluded from the analysis by treating them as non state holding elements (Lines 3-5). Afterwards, it applies one of the architectural techniques (pipeline flushing, NOP or random initialization) and extracts the new states of the flip-flops (Lines 6-7). Finally, it checks the simulation flow status and if it fails then, flip-flops which changed their state are considered as a candidate for state holding flip-flops and their values are restored to the `old_ff_value` in order to continue to the next simulation checkpoint (lines 8-12). The candidate flip-flop list is updated on every checkpoint and the last candidates (output of the algorithm) are considered as the state holding flip-flops and their switching activity is studied to decide whether to implement them using NV-NS-FF or NV-S-FF.

#### IV. EXPERIMENTAL SETUP AND RESULTS

##### A. Experimental setup

To evaluate the proposed non-volatile SoC design scheme, we use a SoC consisting of four different components, namely, 1) Leon3mp processor core, 2) a security co-processor (AES), 3) an accelerator for Fast Fourier Transform (FFT core), and 4) the digital part of a communication interface (ZigBee). Synopsys Design Compiler is employed to synthesize the components using TSMC 65 nm library. We use ModelSim and post-synthesis gate-level simulations to conduct the architectural level X-analysis and identify the important state holding elements. The circuit level details are extracted by SPICE simulation using Cadence spectre tool. For that purpose, we have employed the MTJ model that is proposed in [29]. The circuit and architectural level setups used in this work are given in Table I.

##### B. Circuit level results

We developed netlists for conventional CMOS-based master-slave V-FF as well as NV-S-FF and NV-NS-FF archi-

TABLE II. DESIGN PARAMETERS FOR CONVENTIONAL MASTER-SLAVE, NON-VOLATILE SHADOW AND NON-SHADOW FLIP-FLOP ARCHITECTURES. WRITE AVOIDANCE SCHEME FOR BOTH SHADOW AND NON-SHADOW FLIP-FLOPS.

| Parameters           | V-FF | NV-S-FF        | NV-NS-FF [26] |
|----------------------|------|----------------|---------------|
| Active latency (ps)  | 83   | 126            | 106           |
| Active energy (fJ)   | 13   | 19             | 68            |
| Backup latency (ps)  | –    | 404            | –             |
| Backup energy (fJ)   | –    | 59             | –             |
| Wake-up latency (ps) | –    | 117 + 2 cycles | 103           |
| Wake-up energy (pJ)  | –    | 12.2           | 9.8           |
| Transistor count     | 26   | 83             | 67            |
| MTJ count            | 0    | 2              | 2             |

tures (similar to as proposed in [17]) and their circuit level design parameters were extracted using SPICE simulation, that are depicted in Table II. Since the store energy for the non-volatile component is more than CMOS-based designs, the active and backup energies are higher. Therefore, to take the system-level advantage, we have integrated the write avoidance scheme for both shadow and non-shadow flip-flops. This significantly improves the active energy at the system level, however, at the cost of 30 extra transistors per flip-flop. On the other hand, the active latency is similar for all three architectures. Moreover, the wake-up latency for shadow and non-shadow architectures is almost similar as well, but shadow flip-flop needs extra two cycles to restore the value to the conventional V-FF. Note that, backup and wake-up parameters for V-FF and backup parameter for NV-NS-FF are not applicable.

##### C. System level results

In our SoC benchmark, the digital part of ZigBee is utilized only during sending and receiving data. Leon3 is considered as the main processing core (mostly active) and it offloads tasks to the accelerators AES and FFT. Hence, the sleep and wake-up times of these two components are controlled by the main core. To illustrate the energy benefits of the proposed heterogeneous SoC design, first we extract the power-gating intervals of the SoC components based on their utilizations during workload execution. Then, the non-utilized components are power-gated to save leakage energy. The non-volatile flip-flop designs adopted in this work allow us to power-gate the design in a single cycle for NV-NS-FF and two cycles for NV-S-FF designs.

Table III shows the architecture level configuration and energy per cycle of the SoC elements used in this work. The energy consumption given in the table represents the average energy consumption of all flip-flops in a core. For each core, first the state holding flip-flops are identified using architectural

TABLE III. ACTIVE TIME ENERGY CONSUMPTION OF THE NON-VOLATILE FLIP-FLOP BASED SOC ELEMENTS

| Benchmark design | # flip-flop | % of non volatile FFs | Used flip-flop combinations | Energy per cycle (in pJ) |
|------------------|-------------|-----------------------|-----------------------------|--------------------------|
| Leon3mp          | 2364        | 10.78%                | V-FF and NV-S-FF            | 5.64                     |
| FFT              | 705         | 17.02%                | V-FF and NV-NS-FF           | 1.61                     |
| AES              | 530         | 21.13%                | V-FF and NV-NS-FF           | 1.09                     |
| ZigBee           | 480         | 19.37%                | V-FF and NV-NS-FF           | 0.35                     |
| <b>Total</b>     | <b>4079</b> | <b>14.21%</b>         | –                           | <b>8.69</b>              |

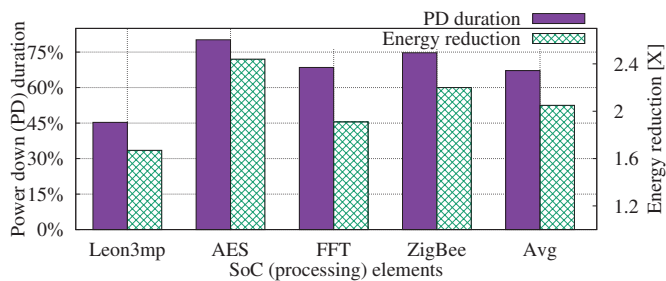


Fig. 5. Power-gating duration and energy improvement of the SoC elements (Leon3mp, FFT, AES and ZigBee) using NV-NS-FF and NV-S-FF designs.

level X-analysis. Then, the non-volatile flip-flop implementation of the state holding flip-flops is determined based on their average switching activities. If the state holding flip-flops have high average switching activity, NV-S-FF implementation is used to avoid frequent write accesses to the energy intensive non-volatile part. Hence, the Leon3mp core is implemented using V-FF for non state holding and NV-S-FF for state holding elements. Since the average switching activities of FFT, AES and ZigBee are relatively low, they are implemented using V-FF and NV-NS-FF to avoid copying of data during power-down and power-up periods.

Figure 5 shows the power-down duration and energy improvement of the SoC elements. As depicted in the figure, the power-gating duration of the components depends on their utilization frequency. In comparison to others, Leon3 has the smallest power-gating duration. Hence, on average the SoC components can be power-gated more than 65% of the time. This power-gating duration improvement results in more than 2X energy saving of the SoC as shown in the figure.

## V. CONCLUSIONS

Leakage power reduction has become an important issue in the design of energy-constrained devices for Internet of Things (IoT) applications. Emerging non-volatile memory technology can play a vital role in reducing leakage power consumption based on normally-off computing concept. In this work, we have exploited the non-volatile memory technology to design a normally-off SoC using SOT-based latches, flip-flops and registers. This paper demonstrated a heterogeneous SoC architecture containing conventional CMOS flip-flops as well as different flavors of spintronic-based non-volatile storage elements, to improve the energy-efficiency by exploiting power-gating technique.

## VI. ACKNOWLEDGEMENT

This work was partly supported by the European Commission under the Horizon-2020 Program with the grant agreement number 687973 as part of the GREAT project (<http://www.great-research.eu/>) and by ANR/DFG as part of the MASTA project. We are thankful to our colleague Mohammad Saber Golanbari for providing us the SoC benchmark circuits used in this work.

## REFERENCES

- [1] K.-S. Yeo and K. Roy. *Low Voltage, Low Power VLSI Subsystems*. 2005.
- [2] C. Singh and R. Tangirala. As nodes advance, so must power analysis. Available: <http://semiengineering.com/as-nodes-advance-so-must-power-analysis/>, 2014.

- [3] Y. Wang, et al. A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops. In *ESSCIRC*, 2012.
- [4] M. Qazi, et al. A 3.4-pJ FeRAM-Enabled D Flip-Flop in 0.13-um CMOS for Nonvolatile Processing in Digital Systems. *JSSC*, 2014.
- [5] S. Bartling, et al. An 8mhz 75 ua/mhz zero-leakage non-volatile logicbased cortex-m0 mcu soc exhibiting 100lt; 400ns wakeup and sleep transitions. In *ISSCC*, 2013.
- [6] A. Lee, et al. A reram-based nonvolatile flip-flop with self-write-termination scheme for frequent-off fast-wake-up nonvolatile processors. *JSSC*, 2017.
- [7] Y. Liu, et al. 4.7 A 65nm ReRAM-enabled nonvolatile processor with 6X reduction in restore time and 4X higher clock frequency using adaptive data retention and self-write-termination nonvolatile logic. In *ISSCC*, 2016.
- [8] H. Koike, et al. A power-gated mpu with 3-microsecond entry/exit delay using mtj-based nonvolatile flip-flop. In *ASSCC*, 2013.
- [9] Y. Wang, et al. A compression-based area-efficient recovery architecture for nonvolatile processors. In *DATE*, 2012.
- [10] M. Xie, et al. Fixing the broken time machine: Consistency-aware checkpointing for energy harvesting powered non-volatile processor. In *DAC*, 2015.
- [11] S. Senni, et al. Non-volatile processor based on mram for ultra-low-power iot devices. *JETC*, 2016.
- [12] D. Chabi, et al. Ultra low power magnetic flip-flop based on checkpointing/power gating and self-enable mechanisms. *TCS*, 2014.
- [13] International Technology Roadmap for Semiconductors. <http://www.itrs.net>, 2013.
- [14] F. Oboril, et al. Evaluation of hybrid memory technologies using sot-mram for on-chip cache hierarchy. *TCAD*, 2015.
- [15] G. Prenat, et al. Ultra-fast and high-reliability sot-mram: From cache replacement to normally-off computing. *TMSCS*, 2016.
- [16] N. Sakimura, et al. A 90nm 20mhz fully nonvolatile microcontroller for standby-power-critical applications. In *ISSCC*, 2014.
- [17] R. Bishnoi, et al. Non-volatile non-shadow flip-flop using spin orbit torque for efficient normally-off computing. In *ASP-DAC*, 2016.
- [18] K. Ishizaka, et al. Giant rashba-type spin splitting in bulk bitei. *Nature materials*, 2011.
- [19] M. Cubukcu, et al. Spin-orbit torque magnetization switching of a three-terminal perpendicular magnetic tunnel junction. *APL*, 2014.
- [20] Garello, et al. Ultrafast magnetization switching by spin-orbit torques. *APL*, 2014.
- [21] W.-k. Yu, et al. A non-volatile microcontroller with integrated floating-gate transistors. In *DSN-W*, 2011.
- [22] M. Padhye and D. Gross. *Freescal: Wireless Low-Power Design and Verification with CPF*.
- [23] J. S. Meena, et al. Overview of emerging nonvolatile memory technologies. *Nanoscale research letters*, 9(1):526, 2014.
- [24] R. Bishnoi, et al. Improving write performance for STT-MRAM. *TMAG*, 2016.
- [25] R. Bishnoi, et al. Design of defect and fault-tolerant nonvolatile spintronic flip-flops. *TVLSI*, 2017.
- [26] R. Bishnoi, et al. Low-power multi-port memory architecture based on spin orbit torque magnetic devices. In *GLSVLSI*, 2016.
- [27] W. Zhao, et al. Design considerations and strategies for high-reliable stt-mram. *Microelectronics Reliability*, 2011.
- [28] R. Bishnoi, et al. Self-timed read and write operations in stt-mram. *TVLSI*, 2016.
- [29] K. Jabeur, et al. Compact model of a three-terminal mram device based on spin orbit torque switching. In *ISCDG*, 2013.