Analysis of ReRAM-Based Heterogeneous Memory Hierarchies for Ultra-Low Power Bio-Signal Processing Platforms

ABSTRACT

Energy efficiency and size are critical for Wireless Body Sensor Nodes (WBSN), which are small devices aimed at continuous and long-term health monitoring, typically featuring a limited power source. Aggressive CMOS technology and voltage scaling have been crucial to reducing energy consumption, but in current and future nodes, managing static leakage is becoming a challenge. In this context, emerging non-volatile Resistive RAM (ReRAM) technologies appear as a promising solution for WBSN applications with long idle periods, thanks to their zero standby power and high density. However, ReRAMs suffer from permanent aging-related failures after a relatively small number of writes.

Performing a technologically accurate ReRAM power characterization, this paper shows that a heterogeneous memory hierarchy including ReRAMs instead of SRAMs can achieve significant energy savings of up to 65.4 % for a typical WBSN application. In addition, it proposes a lightweight block replacement strategy to ensure prolonged (1 year) functionality under realistic ReRAM endurance conditions.

CCS CONCEPTS

• **Hardware → Aging of circuits and systems**; *Electronic design automation*;

ACM Reference format:
. 2019. Analysis of ReRAM-Based Heterogeneous Memory Hierarchies for Ultra-Low Power Bio-Signal Processing Platforms. In *Proceedings of Design Automation Conference, Las Vegas, USA, June 2019 (DAC'19),* 6 pages.
https://doi.org/00.000/000_0

1 INTRODUCTION

WHO reports highlight chronic heart diseases as a significant contributor to deaths worldwide [22], a situation compounded by the increasingly unhealthy lifestyle of large parts of the world population. These ailments require continuous and long-term monitoring, which results in high medical costs and patient discomfort. Thus, there is a clear need to shift from traditional hospital-based healthcare toward more personalized approaches.

In this context, Wireless Body Sensor Nodes (WBSN) can provide large-scale cost effective solutions for real-time personalized monitoring of affected patients. WBSNs are autonomous devices able to acquire bio-signals that provide insights on the health of various organs. Since they are typically powered by a small battery, energy efficiency becomes a key design factor for long-term functioning. The new generation of smart WBSNs are a significant step in that direction: in addition to signal acquisition and transmission, they perform signal processing to filter out only the meaningful features for further analysis by qualified medical staff. Therefore, smart WBSNs move the energy bottleneck from the transmission stage to the Bio-Signal Processing (BSP) stage.

BSP applications that deal with slowly changing signals typically present long idle periods (e.g., more than 90 % [4]). In those cases, static leakage dominates the total energy consumption of the system, a phenomenon that intensifies as transistor sizes scale down [1]. In particular, in the base architecture considered in this paper, static leakage can represent up to 88.4 % of the total device power. Unfortunately, SRAMs cannot be power-gated during idle periods as their contents need to be preserved. Furthermore, backing up to non-volatile storage is normally prohibitive in terms of energy and response time.

As a result, emerging Non-Volatile Memory (NVM) technologies, such as Resistive RAM (ReRAM), are promising candidates to curb the energy consumption of the memory subsystem due to static leakage, since they can be turned-off during idle phases. Although they are more tolerant to ultra-low supply voltage scaling than SRAMs, write operations to ReRAM cells are significantly costlier than to SRAMs [21]. In addition, ReRAM cells exhibit permanent failures after a relatively low number of writes [24]. These situations advocate the use of heterogeneous memory hierarchies, where first-level memories reduce the number of writes to second-level ReRAM main emory.

The main contributions of this paper are:

- We characterize ReRAM power consumption using a technologically accurate model, with which we show the important energy savings (up to 65.4 %) achieved with a ReRAM-based heterogeneous memory architecture for BSP applications, in comparison to SRAM-based platforms, under various programming energy ranges.
- We perform the first endurance analysis of a heterogeneous ReRAM-based WBSN platform with a real-world BSP application, observing its effects on device lifetime. Furthermore, we quantify the additional memory capacity required for long-term device operation under different error-probability conditions. Using a ReRAM-cell replacement strategy, we found this overhead to be 47 KiB assuming a cell endurance of 10^8 writes.

This paper is organized as follows. In Section 2, we present the state of the art on resistive memories and WBSN architectures. Then, we analyze the particularities of the design with ReRAMs in Section 3. In Section 4, we describe our WBSN platform and the BSP application used in the experiments. The results of our experiments are presented in Section 5. Finally, we draw our conclusions in Section 6.

DAC'19, June 2019, Las Vegas, USA

ACM ISBN 000-0000-00-000/00/00…$00.00
https://doi.org/00.000/000_0

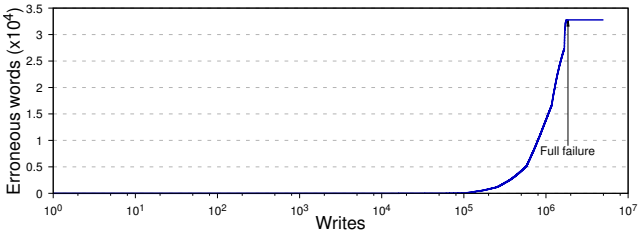

Figure 1: Reliability of a commercial 64 KiB ReRAM module [13]. After $\sim 1.8 \times 10^6$ writes per word, all the words in the chip are damaged.

2 STATE OF THE ART

2.1 Resistive memories

With the growing need for low-power and low-cost autonomous connected devices, emerging memory technologies are promoted by both industries and academics as a future corner stone of integrated low-power technologies. Indeed, thanks to their low-cost fabrication process (Back-End-of-Line integration with common materials [21]), filamentary Resistive Random Access Memories (ReRAM) such as Oxide-based (OxRAM) ReRAM technologies are already included in low-power micro-controllers replacing embedded Flash technologies [13].

ReRAM technologies rely on the controlled creation and destruction of a conductive filament inside an insulating material layer. Write operations are performed by applying a programming voltage (V_{prog}) across the bitcell and controlling the current going through the ReRAM (programming current, I_{prog}) [21, 23] while waiting for the resistance state to switch to a Low or High Resistance State (LRS, HRS), i.e., t_{prog}. As widely shown in the literature, an exponential relationship exists between t_{prog} and V_{prog} [8, 21] while the LRS value is mainly controlled by I_{prog}. On the other hand, read operations are performed by sensing the resistance value (HRS or LRS).

1-Transistor 1-ReRAM (1T1R) bitcells using thin oxide transistors and standard logic layout rules have been demonstrated down to $12F^2$ ($0.0308\mu^2$) [18]. Moreover, compared to high density SRAM bitcells ($0.12\mu^2$) [19], they enable a 4× area reduction with the subsequent die cost reduction. Although ReRAMs have a negligible leakage when compared to SRAMs, they are susceptible to endurance effects. While previous studies assumed an endurance in the order of 10^{10} to 10^{12} writes per cell [23], more recent data [24] show that currently the reality may be closer to 10^6 writes, as illustrated in Fig. 1, calling for schemes to counter that.

2.2 Bio-signal processing systems

WBSNs running BSP applications are instrumental in the monitoring of a number of chronic diseases [9, 20]. Processing of Electrocardiograms (ECG) is vital among them as it is a prime indicator of cardiovascular problems. ECG processing applications typically exhibit moderately complex algorithms that need to execute with high energy efficiency. In this context, domain specific low-power digital processors have gained popularity, as they work efficiently at ultra-low voltage levels [6]. Moreover, multi-core architectures

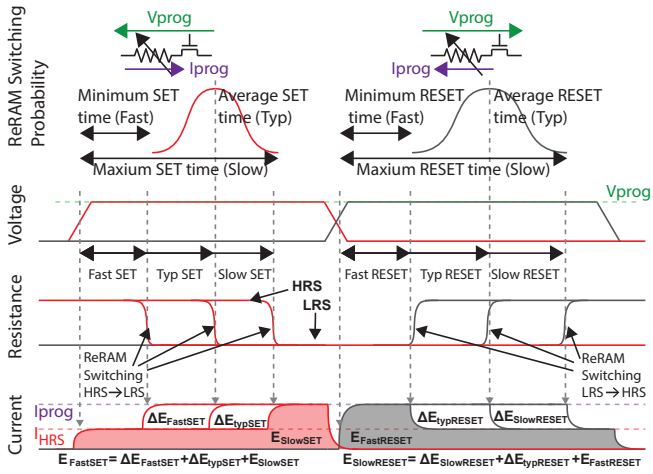

Figure 2: Programming operations in ReRAM technologies (*set* in red and *reset* in gray) with detailed voltage current and energy considerations. ReRAM corners (fast, slow) are also defined.

have been proven to be more efficient than single core equivalents in processing multi-lead input ECG, by exploiting the intrinsic parallelism in these applications, thereby lowering the required working frequency [6]. Those platforms typically implement program (PM) and data memories (DM) with SRAMs, with the main drawback of high leakage power and loss of memory reliability at ultra-low voltages [2]. Recent works have shown that non-volatile ReRAMs can successfully replace SRAMs, thereby bringing down the leakage costs drastically. However, since ReRAMs have limited endurance [24], they are frequently combined with other types of memories in a multi-level hierarchy to reduce the number of accesses to them [4].

3 SYSTEM DESIGN WITH RERAMS

3.1 Characterization of ReRAM programming energy

In this work, we consider a 50 ns cycle time for the full system (i.e., 20 MHz). As shown in [21], a few tens of ns programming time (t_{prog}) can be achieved for both *set* (HRS to LRS) and *reset* (LRS to HRS) operations while considering 1 V to 1.5 V programming voltage (V_{prog}). In this context we assume that 100 % of the bitcells can be programmed in one clock cycle (to reduce distribution tails and satisfy this assumption, specific programming strategies such as adaptive programming voltage [16] can be considered). Finally, we consider $I_{prog} = 100\,\mu A$ to achieve a sufficient HRS/LRS ratio, a low variability in the LRS state and a several-years retention [21].

As shown in Fig. 2, we model the programming operation in two distinct phases separated by a sharp switching: *(i)* the pre-programming phase, in which the ReRAM is in its previous resistive state; *(ii)* the post-programming phase, in which the ReRAM is in the aimed resistive state. As a margin we consider that no state switching happens during the first and the last 5 ns. We then sum

the programming energy corresponding to phases (i) and (ii) to determine the energy consumed during the full programming pulse. From this model, four corner cases are identified: fast *set*, slow *set*, fast *reset* and slow *reset*. Then, assuming balanced programming operations (i.e., $I_{set} = I_{reset}$), we can reduce it to two corners: the Best Case (BC) corresponding to a slow *set* and fast *reset* and the Worst Case (WC) corresponding to fast *set* and slow *reset*. When considering a $t_{prog} = 50\,\text{ns}$, $V_{prog} = 1.2\,\text{V}$ and $I_{prog} = 100\,\mu\text{A}$, we use (1) and (2) to determine the energy consumed in each corner as $E_{BC} = 1.248\,\text{pJ/bit}$ and $E_{WC} = 5.472\,\text{pJ/bit}$.

At the end of the programming operation, if any bit in the word did not successfully switch, the write amplifier notifies the memory controller and the word is marked as permanently defective.

$$E_{set} = V_{prog} * (t_{prog} * I_{HRS} + (t_{cycle} - t_{prog}) * I_{prog}) \quad (1)$$

$$E_{Reset} = V_{prog} * (t_{prog} * I_{prog} + (t_{cycle} - t_{prog}) * I_{HRS}) \quad (2)$$

3.2 ReRAM block substitution mechanism for extended system lifetime

The main drawback of ReRAMs is their relatively low endurance (from 10^4 to 10^8 cycles [12] in academic foundries and 10^5 cycles qualification in commercially available products [13]). This is less important when they are employed for storage, but becomes critical if ReRAMs are used as main memory, especially in WBSNs.

Fortunately, ReRAMs age at the level of individual words instead of bigger blocks — indeed, they age at the bit-cell level, but we will work at the word level for the sake of simplicity. Therefore, if we know the access pattern of the application, we can propose solutions that work at a small granularity. For example, in the analyzed BSP application, we can identify individual memory words that are written approximately 18 times per second. If we assume a worst case in which a ReRAM word fails after 10 000 writes, then some words in the ReRAM will start to fail after roughly 555 s (less than 10 minutes). This is clearly an insufficient lifetime for a WBSN of any practical purpose.

In Section 5.3 we explore how much additional ReRAM is needed to provision, to achieve uninterrupted system operation for extended periods of time. Previous works have explored complex mechanisms to protect against permanent failures in high capacity ReRAMs, used as a DRAM replacement solution for main memory [15, 17], whose study is out of the scope of this work. Herein, we use a lightweight energy-efficient mechanism to substitute failed memory words transparently for software applications.

We propose to divide the application data space into small blocks, which are replaced when any of their words fails. Then, we use the number of accesses to the words in each block to compute their replacement time and, thus, the required ReRAM capacity along time. Hence, we create a trade-off between management overhead and underexploited storage. Interestingly, if the data words correspond to small data objects, they can be reordered according to the number of write accesses to achieve more homogeneous blocks and better storage exploitation — unfortunately, this is not the case for the BSP application chosen in our study, which works over a single data array.

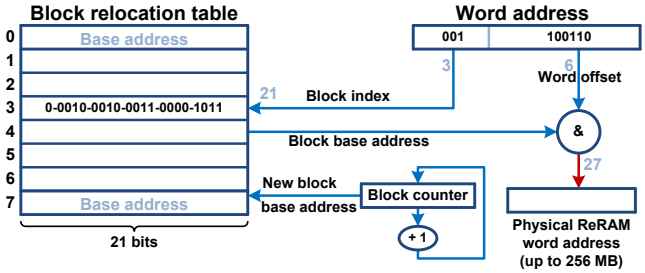

Figure 3: ReRAM block relocation mechanism (maximum 256 MiB addressable).

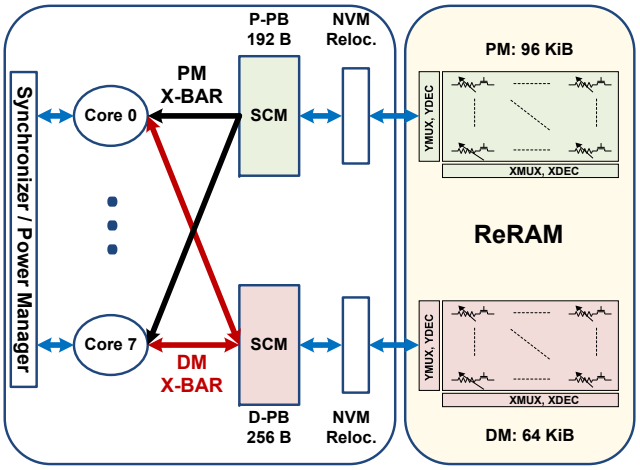

Figure 4: Schematic representation of the heterogeneous architecture.

We propose the mechanism shown in Fig. 3 to implement ReRAM block replacement. A small Block Relocation Table (BRT) with 8 entries holds the base address in ReRAM of each block. For every processor access to main DM, the address is decomposed into a block index part and a word (offset) part. The block index is used to select an entry in the BRT that supplies the current base address of the block in the ReRAM, which is then concatenated with the word offset to calculate the final ReRAM address.

When a word in a block fails, the complete block is relocated to a different position in the ReRAM. Consequently, the block counter is increased by one and its value is used as the new starting address for that block. We implement the BRT and the block counter as 21-bit registers (with a total size of 189 bits) and add their read energy for every access to the ReRAM. We assume that the BRT remains always active (to reduce the number of writes to ReRAM) and account for its leakage power as well.

4 EXPERIMENTAL SETUP

4.1 System architecture

We consider the heterogeneous BSP architecture shown in Fig. 4, which consists of eight cores featuring a three-stage Harvard architecture generated with Synopsys ASIP designer. The memory

Table 1: Energy consumption per access and leakage power for each type of considered WBSN memory.

	SRAM		ReRAM		PB	
Size	64 KiB	96 KiB	64 KiB	96 KiB	256 B	192 B
Read (pJ/bit)	0.229	0.287	0.264	0.266	0.028	0.024
Write (pJ/bit)	0.194	X	5.472	X	0.032	0.028
Leakage (µW)	1.476	2.160	0.033	0.053	0.004	0.003

architecture features a hybrid system partitioned into a first level with Standard Cell Memory (SCM) page buffers (PBs) and a second ReRAM level that acts as main PM and DM. The size of the PBs is 192 B for instructions (P-PB) and 256 B for data (D-PB). The PBs act as fully-associative memories with eight 8-word lines (P-PBs) or sixteen 8-word lines (D-PBs), respectively. The sizes of the main PM and DM are 96 KiB and 64 KiB, respectively.

A Memory Management Unit (MMU) controls the exchange of data between the page buffers and the main memory, adopting a least-recently-used strategy. The MMU can be turned off to bypass the page buffers, obtaining the equivalent of a system with only main memory. Finally, a Synchronizer module coordinates the execution of cores, and is also in charge of power-gating the entire system when the cores are in an idle-state (deep sleep mode). The page buffers are never powered off to decrease the number of writes to the ReRAM required to back them up when power-gated and increase its endurance.

4.1.1 Baseline architectures. In order to study and compare the power efficiency of the hybrid system, we have considered four configurations, all of which work at 1 V supply and a frequency of 20 MHz (50 ns), which respects the read and write latencies of the considered ReRAMs:

- *SRAM-only:* Baseline architecture similar to [3]. The main memory is composed of 6T SRAMs and is directly interfaced with the processing cores, without page buffers.
- *ReRAM-only:* Baseline architecture with a single level of memory implemented entirely with ReRAMs and directly interfaced with the processing cores, without page buffers.
- *Hybrid-BC:* This architecture corresponds to Fig. 4, with the main memory implemented with ReRAMs, the page buffers with SCMs and considering the best case for the ReRAM write energy of $E_{BC} = 1.248$ pJ/bit (cf. 3.1).
- *Hybrid-WC:* This configuration is the same as Hybrid-BC, but considering the worst-case ReRAM write energy of $E_{WC} = 5.472$ pJ/bit (cf. 3.1).

4.1.2 Power characterization. The values to estimate the power consumption of the processing elements are derived from a post-place-and-route net-list, considering a 28 nm low-power, high-k metal gate, Process Design Kit (PDK) technology under nominal supply conditions ($V_{DD} = 1$ V) at 300 K. These values are used in conjunction with usage statistics for the considered application obtained from a SystemC model.

Table 1 shows the values for energy consumption and leakage power for the different types of memories in each platform configuration. The parameters for the ReRAM memory periphery were derived using NVsim [7] with the ReRAM programming conditions detailed in Section 3.1. For the SRAM main memory, they are calculated with the CACTI memory modeling tool [10]. We assumed that energy consumption per operation for the SCM page buffers is similar to that of an SRAM of equivalent size. When the block replacement strategy is considered for the ReRAMs, the energy overhead of the additional circuitry is also accounted for.

4.1.3 Memory hierarchy optimization for WBSN applications. Caching schemes, such as the page buffers, work adequately for applications with high access locality. However, several key applications in the WBSN domain, such as compressed sensing, exhibit markedly random access patterns. In those cases, caching may result in write amplification because whole lines are replaced after only a few words are used. In the proposed architecture, each page buffer eviction writes 8 words (128 bits) to memory, even if just one word was modified. With SRAM or DRAM-based architectures, this is mainly an issue of wasted energy (and performance). However, due to their higher write current, in ReRAM-based memories writing 128 bits in parallel may be unfeasible. For example, with a 100 μA programming current per bit, a 128-bit parallel write requires a current of 12.8 mA. Even if in the typical case only part of the bits are modified, guaranteeing the required current for the worst case would require strong modifications to the macro memory design to avoid reductions in both chip and battery reliability.

To better cope with such complex access patterns, we extend the page buffers' "dirty bits" to cover individual words, at the cost of 128 extra bits in total. During write-backs, the MMU checks the state of the dirty bits for word masking before sending requests to the ReRAM main memory.

4.2 Target WBSN application

We have used Compressed Sensing (CS) of ECG over 8-leads to evaluate the platform behavior with a real-life application [11]. CS performs the lossy encoding (50 %) of the input ECG signals, with the computation corresponding to each lead mapped on a different processing core. The compression of input signals is achieved by a random sensing matrix (with k rows and n columns). As explained in [5], the generation of the sensing matrix can be characterized with four main parameters: an LFSR polynomial, the LFSR seed, the number of index bits (depending on k), and the number of non-zero elements (ones) per column. These configuration parameters enable the generation of a large set of different sensing matrices. In essence, CS performs 12 memory accesses over the output buffer for each sample received at the input. Since the indexes of the accesses are calculated using the LFSR, CS exhibits a memory access pattern with low locality that stresses traditional caching solutions.

The input ECG signals were derived from the MIT-BIH Normal Sinus database [14]. In this study, we have considered ECG windows of 1024 samples, acquired with a frequency of 500 Hz. The entire execution time (acquisition and processing) of CS for a window of samples takes 2.16 s. Although the input samples are processed as they arrive (and not stored), the LFSR-dependent access pattern of CS requires keeping in memory the complete output buffer of 512 samples. Therefore, we have divided these 512 data words into 64 blocks with 8 words each, as explained in Section 3.2. The number of accesses to each word, as shown in Fig. 5, can be used to compute the required ReRAM capacity along time.

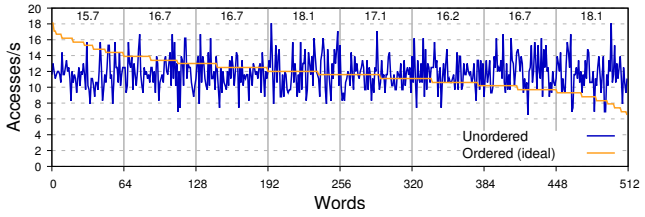

Figure 5: Write accesses distribution over data words for CS. Vertical bars show the grouping of words into 64-word blocks, with the maximum number of writes to any word in each block. Also, as shown in orange in the figure, the distribution of accesses if the application variables can be reordered.

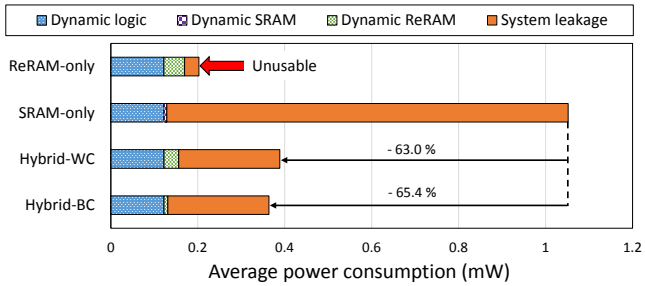

Figure 6: Average power of the WBSN system components at 1 V supply.

5 EXPERIMENTAL RESULTS

5.1 Power and energy consumption analysis

Fig. 6 showcases the system average power requirements in the different considered configurations. *Dynamic logic* accounts for the dynamic power of processing components like cores, crossbars, clock tree, etc., along with that of the page buffers and the BRT, when applicable. *Dynamic SRAM* and *Dynamic ReRAM* are the dynamic power of the SRAM and ReRAM main memories, respectively. Finally, *System leakage* depicts the combined leakage power of the entire system.

The SRAM-only configuration has the highest power requirement, with leakage accounting for the majority of the power (Fig. 6). Although the proposed ReRAM-only platform requires more dynamic power compared to the SRAM-only system, it decreases the leakage power by over 95 % by power-gating the entire platform when the cores are idle. Even if these results make the ReRAM-only configuration desirable, writing directly to the ReRAM without a memory hierarchy may result in premature stuck-at failures, that make this option unusable at the current ReRAM endurance rates. In turn, the heterogeneous architecture diminishes substantially the leakage power even when the page buffers and the BRT are taken into account: although dynamic power is marginally higher than in the SRAM-only configuration, significant overall power reductions of 63.0 % and 65.4 % are achieved, in the worst and best cases, respectively.

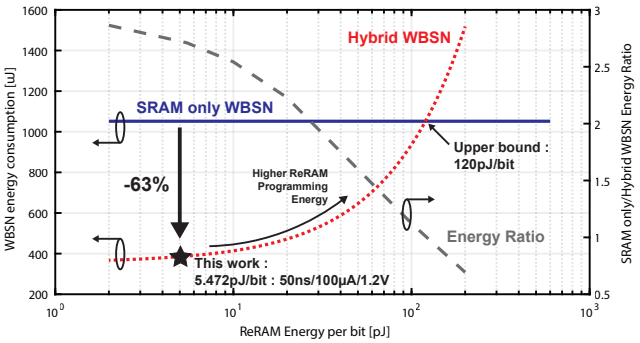

Figure 7: Energy consumption of WBSN systems embedding SRAM (blue) or ReRAM (red) versus the ReRAM programming energy. In gray, energy ratio between SRAM and ReRAM based WBSNs

Fig. 7 shows the evolution of the energy consumed in two WBSN configurations: SRAM in blue and ReRAM with page buffers (Hybrid) in red. To ease the reading, the energy ratio between these two configurations is also shown (gray dotted line). With the assumptions taken in this work (i.e., the worst case programming energy is 5.472 pJ/bit), the proposed system consumes 63 % less energy than the SRAM-based one. However, for several reasons (e.g., process variability, temperature, retention constraints or use of complex iterative write strategies), the average programming energy of ReRAM may be higher or lower than the considered values. To generalize this study, we determine the energy consumption of the proposed architecture for a wide range of ReRAM programming energies. We show that the proposed architecture is advantageous compared to the SRAM-based approach until the ReRAM programming energy reaches 120 pJ/bit, which corresponds to the WC corner (as defined in Section 3.1) of a 1 µs programming pulse with a 100 µA current at 1.2 V. Thereby, the proposed solution is suitable even for high programming energy ReRAM technologies and shows the important reductions in energy consumption that can be attained incorporating ReRAMs in the design of WBSNs.

5.2 Low access locality performance evaluation

The architectural optimizations introduced in Section 3 to improve page buffer performance in cases of low access locality are successful for the considered CS application, which would otherwise suffer a 6.2 × write amplification to main (ReRAM) memory. The reduction in the number of words written to ReRAM (from 874 312 to 140 729 writes) can be used to optimize the structure of the bus between the page buffers and the ReRAM, and to subsequently reduce the energy needed by the ReRAM itself to test the previous values of the affected array cells. Finally, out of the words sent to the ReRAM for writing, the percentage of bits actually changed over the course of one sampling window is only 39.5 %.

5.3 Analysis of device lifetime based on ReRAM endurance

This section evaluates the additional ReRAM needed to be included in the system to ensure long-term operation. Since the reliability

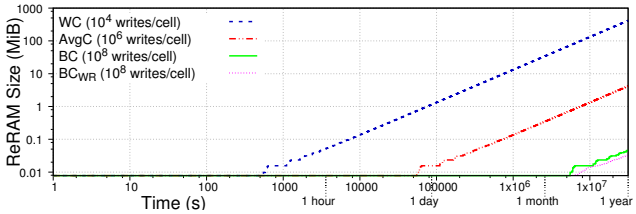

Figure 8: ReRAM overprovisioning requirements to achieve up to one year of system lifetime, depending on the number of writes to a word before it fails permanently.

conditions of ReRAMs depend on several factors such as manufacturing process, physical size or programming current, we select three possible endurance ranges: Memory cells fail permanently after 10^4 writes (*worst case*, WC), after 10^6 writes (*average case*, AvgC), or after 10^8 writes (*best case*, BC).

Using the average number of writes per second for each block obtained in Section 3, we can use (3) to calculate the number of blocks needed along time (hence, the required ReRAM capacity) per core, where A_i is the number of accesses per second to block i, and W is the estimated number of writes to a cell before failure:

$$Blocks(t) = \sum_{i=1}^{Blocks} \left\lceil \frac{t \times A_i}{W} \right\rceil \quad (3)$$

According to this, Fig. 8 shows the minimum ReRAM capacity required to enable up to one year of system operation. In the BC, the system can work for one year using 47 KiB of ReRAM (considering only the CS application). As we decrease the endurance of the ReRAM, the system requires progressively more memory to remain operative during the same amount of time. Thus, in the AvgC, it will require 4.2 MiB of total ReRAM capacity to operate during one year whereas, in the WC, it will require up to 417 MiB. The figure also shows that the capacity required with a hypothetical optimal block-replacement strategy working on individual words is 34 KiB (*best case, word-replacement*, BC_{WR}), providing a comparison for our lightweight solution.

As a conclusion, further optimization on the construction and programming of ReRAMs to enhance their durability are crucial to profit from their benefits on leakage reduction in the design of reliable embedded devices. Moreover, such improvements would also entail a potential reduction in costs as ReRAMs typically present an area reduction factor of 4 × compared to SRAMs.

6 CONCLUSION

In this work, we have shown that significant system power reduction over state-of-the-art WBSNs is achievable by employing a hybrid memory structure featuring ReRAM NVMs as main memory. When the worst case of ReRAM write energy is considered, savings of 63.0 % are obtained for the energy consumption of the complete system, which ascends to 65.4 % in the best case. Additionally, we have analyzed the hybrid WBSN energy consumption at various levels of possible ReRAM write energy per bit, showcasing the viability of such an architecture compared to traditional SRAM-based equivalents, proving its efficiency up to 120 pJ/bit.

Furthermore, we have proposed for the first time an endurance-related failure analysis of a heterogeneous memory-based WBSN running a BSP application. In that regard, we have shown how much additional ReRAM capacity is required to extend the device lifetime for long-term operation, considering various endurance ratings. Under favorable conditions, our system could remain operative using a lightweight block replacement mechanism for one year with only 47 KiB of ReRAM.

Finally, in this study we have considered operation at nominal voltages, because although ReRAMs promise additional improvements thanks to their higher resilience in comparison to SRAMs when operating at NTV ranges, WBSN systems require medical compliance and certification, which can only be achieved reliably at nominal voltages. However, in the near future it is likely to be able to reach even larger benefits if medical device standards start accepting NTV values.

REFERENCES

[1] Zia Abbas and Mauro Olivieri. 2014. Impact of Technology Scaling on Leakage Power in Nano-Scale Bulk CMOS Digital Standard Cells. *Microelectron. J.* 45, 2 (Feb. 2014), 179–195.

[2] D. Bortolotti et al. 2014. Approximate Compressed Sensing: Ultra-Low Power Biosignal Processing via Aggressive Voltage Scaling on a Hybrid Memory Multi-Core Processor. In *IEEE/ACM ISLPED*. 45–50.

[3] R. Braojos et al. 2014. Hardware/software approach for code synchronization in low-power multi-core sensor nodes. In *DATE*. 1–6.

[4] R. Braojos et al. 2016. Nano-Engineered Architectures for Ultra-Low Power Wireless Body Sensor Nodes. In *CODES+ISSS*. 1–10.

[5] J. Constantin et al. 2012. TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing. In *IEEE/IFIP VLSI-SoC*. 159–164.

[6] A. Y. Dogan et al. 2012. Multi-Core Architecture Design for Ultra-Low-Power Wearable Health Monitoring Systems. In *DATE*. 988–993.

[7] Xiangyu Dong et al. 2012. NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory. *IEEE TCADICS* 31, 7 (July 2012), 994–1007.

[8] R. Fackenthal et al. 2014. A 16Gb ReRAM with 200MB/s Write and 1GB/s Read in 27nm Technology. In *IEEE ISSCC*. 2.

[9] R. Fensli et al. 2005. A Wearable ECG-Recording System for Continuous Arrhythmia Monitoring in a Wireless Tele-Home-Care Situation. In *IEEE CBMS*. 407–412.

[10] HP Labs 2018. CACTI. (2018). http://www.hpl.hp.com/research/cacti/.

[11] Hossein Mamaghanian et al. 2011. Structured Sparsity Models for Compressively Sensed Electrocardiogram Signals: A Comparative Study. In *IEEE BioCAS*. 125–128.

[12] C. Nail et al. 2016. Understanding RRAM Endurance, Retention and Window Margin Trade-Off Using Experimental Results and Simulations. In *IEEE IEDM*. 4.

[13] Panasonic 2018. ReRAM Embedded Super Low-Power Consumption MCU MN101L. (2018). https://industrial.panasonic.com/ww/products/semiconductors/microcomputers/mn101l

[14] PhysioBank. 2012. MIT-BIH Normal Sinus Rhythm Database. (Feb. 2012). https://www.physionet.org/physiobank/database/nsrdb/

[15] M. K. Qureshi et al. 2009. Enhancing Lifetime and Security of PCM-Based Main Memory with Start-Gap Wear Leveling. In *IEEE/ACM MICRO*. 10.

[16] G. Sassine et al. 2018. Sub-pJ Consumption and Short Latency Time in RRAM Arrays for High Endurance Applications. In *IEEE IRPS*. 5.

[17] Stuart Schechter et al. 2010. Use ECP, Not ECC, for Hard Failures in Resistive Memories. In *Proc. of ISCA*. ACM, 141–152.

[18] W. C. Shen et al. 2012. High-K Metal Gate Contact RRAM (CRRAM) in Pure 28nm CMOS Logic Process. In *International Electron Devices Meeting*. 31.6.1–31.6.4.

[19] M. E. Sinangil et al. 2011. A 28nm High-Density 6T SRAM with Optimized Peripheral-Assist Circuits for Operation down to 0.6V. In *IEEE ISSCC*. 260–262.

[20] Michael Sung et al. 2005. Wearable Feedback Systems for Rehabilitation. *Journal of NeuroEngineering and Rehabilitation (JNER)* 2, 1 (June 2005), 17.

[21] E. Vianello et al. 2013. Back-End 3D Integration of HfO2-Based RRAMs for Low-Voltage Advanced IC Digital Design. In *IEEE ICICDT*. 4.

[22] WHO 2018. The Top 10 Causes of Death. (May 2018). http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[23] H.-S. P. Wong et al. 2012. Metal–Oxide RRAM. *Proc. of the IEEE* 100, 6 (June 2012), 1951–1970.

[24] J. Yang-Scharlotta et al. 2014. Reliability Characterization of a Commercial TaOx-Based ReRAM. In *IEEE IIRW*. 131–134.