

A Synchronization-Based Hybrid-Memory Multi-Core Architecture for Energy-Efficient Biomedical Signal Processing

Rubén Braojos, *Member, IEEE*, Daniele Bortolotti,
Andrea Bartolini, *Member, IEEE*, Giovanni Ansaloni, *Member, IEEE*,
Luca Benini *Fellow, IEEE*, and David Atienza, *Fellow, IEEE*

Abstract—In the last decade, improvements on technology scaling have enabled the design of a novel generation of wearable bio-sensing monitors. These smart Wireless Body Sensor Nodes (WBSNs) are able to acquire and process biological signals, such as electrocardiograms, for periods of time extending from hours to days. The energy required for the on-node digital signal processing (DSP) is a crucial limiting factor in the conception of these devices. To address this design challenge, we introduce a domain-specific ultra-low power (ULP) architecture dedicated to bio-signal processing. The platform features a light-weight strategy to support different operating modes and synchronization among cores. Our approach effectively reduces the power consumption, harnessing the intrinsic parallelism and the workload requirements characterizing the target domain. Operations at low voltage levels are supported by a heterogeneous memory subsystem comprising a standard-cell based ultra-low voltage reliable partition. Experimental results show that, when executing real-world bio-signal DSP applications, a state-of-the-art multi-core architecture can improve its energy efficiency in up to 50% by utilizing our proposed approach, outperforming traditional single-core alternatives.

Index Terms—Biomedical Signal Processing, WBSN, Low-Power Architectures, Reliable Heterogeneous Memory, Code Synchronization.

1 INTRODUCTION

THE increasing social impact of chronic cardiovascular disorders presents a major challenge for healthcare provision [1]. In this context, wearable and miniaturized health monitoring systems, termed Wireless Body Sensor Nodes (WBSNs), offer a large-scale and cost-effective solution [2].

Latest WBSNs are able to perform complex on-node Digital Signal Processing (DSP) routines, such as Electrocardiogram (ECG) compression [3], automated feature extraction [4] and classification [5]. DSP applications embedded in such “smart” WBSNs greatly reduce the required transmission bandwidth, thus increasing the overall energy efficiency. In fact, in this scenario only the retrieved features, as opposed to the acquired samples, have to be sent over

the power-hungry wireless link. This improvement, coupled with the advances in the design of low-voltage and low-rate Analog-to-Digital Converters (ADCs) [6] [7], has led to a change of the dominant contributor to the power consumption of these platforms.

As an illustrative example, Figure 1 (left) reports the energy breakdown of a system performing an ECG features extraction application (named 3L-MMD in [8]). Shown data assumes that samples are acquired by the low-voltage ADC described in [7], processing is performed by the architecture proposed in [8] and a Bluetooth Low Energy protocol is used for wireless transmission [9]. The breakdown highlights that the energy bottleneck resides in the embedded DSP stage, which is often the case in the smart WBSN scenario. To maximize the overall efficiency of smart WBSNs, signal processing must therefore be supported within a tight power budget, while at the same time respecting real time constraints. In this regard, many efforts have been made in the last years, proposing solutions ranging from ad-hoc accelerators [10], [11] to ultra-low-power (ULP) multi-core architectures [8], [12].

In this work, we propose a novel WBSN multi-core architecture for bio-signals processing, which leverages the energy-saving opportunities derived from real-world workloads in this domain. The platform embeds a low-overhead strategy to synchronize computing elements, and allows different execution modes operating at different voltage supplies. It allows the efficient management of Single Instruction-Multiple Data (SIMD) execution and producer/consumer relationships among processors. Moreover, it

- R. Braojos and D. Atienza are with Embedded System Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland - Email: {ruben.braojoslopez, david.atienza}@epfl.ch.
- G. Ansaloni is with Embedded System Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland and University of Lugano, Switzerland - Email: giovanni.ansaloni@usi.ch.
- A. Bartolini and L. Benini are with DEI, University of Bologna, Italy and Integrated Systems Laboratory, ETH Zurich, Switzerland - Email: {barandre, lbenini}@iis.ee.ethz.ch.
- D. Bortolotti is with DEI, University of Bologna, Italy - Email: danielle.bortolotti@umibo.it.

This work has been partially supported by the EC FP7 FET Phidias project (Grant agreement no. 318013) and the BodyPowerSenSE (no. 20NA21_143069) RTD project evaluated by the Swiss NSF and founded by Nano-Tera.ch with Swiss Confederation financing.
Manuscript received Month Day, Year; revised Month Day, Year.

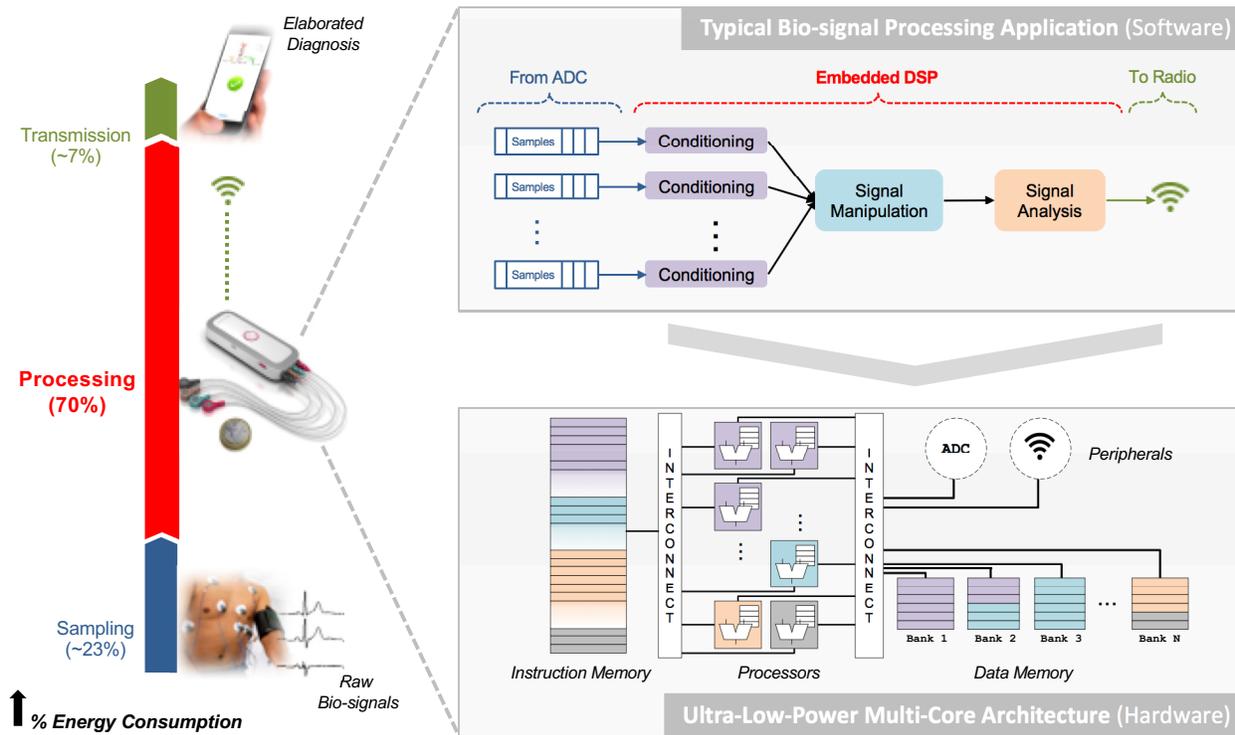


Fig. 1: Left: Energy consumption breakdown of a smart WBSNs executing the 3-input features extraction application described in [8]. Right: scheme of a typical bio-signal DSP application (top) and its mapping on a ULP multi-core WBSN architecture (bottom).

supports an ultra-low voltage sensing state in-between computation phases.

Our work is motivated by the limits of conventional Dynamic Voltage Frequency Scaling (DVFS), especially when applied to the memory subsystem. In fact, the failure probability of the conventional 6-Transistors (6T) SRAM cells increases considerably as the supply voltage is reduced [13]. This situation results in 6T-SRAM memories being the limiting factor for aggressive voltage scaling. At the same time, other low-voltage memory implementations such as Standard-Cell Memories (SCM) lead to substantial area overheads, as outlined in [14], due to the relatively large storage requirements of biomedical DSP applications. The authors of [15] presents a comprehensive comparison of SRAM and SCM implementations, showcasing how SCM can operate at ultra-low supply levels.

Stemming from these observations, we propose a hybrid memory scheme, combining dense 6T memories with SCMs, which present an extended reliable voltage range, but are less area-efficient. By adopting this scheme, the target architecture can efficiently support two different operating modes, namely *sensing* and *processing*. These two modes are characterized by different voltage levels and working frequencies.

- In *sensing mode*, the system works in a low-voltage/low-power regime, where only a small memory region, mapped into SCM, can be written in order to store input samples. The vast majority of the memory cells, realized as 6T-SRAM, while *not accessible* at this low-voltage supply level, still reliably *retain* their content.

- In *processing mode*, the system operates at a higher voltage level, so that the whole memory (and the computing elements) are active and reliable.

Our strategy therefore goes beyond DVFS, by trading off the voltage supply level with the memory portion which can be reliably accessed at a given voltage level. State-of-the-art works use multiple voltage islands to achieve low-voltage and low-power operations in the logic, while ensuring reliable access in the memory. Conversely, our approach requires a single voltage domain, avoiding the design overheads of multi- V_{dd} designs [16]. When processing is required on at least one core, the system is supplied at a high V_{dd} level, while when all cores are idle a lower voltage level is selected. By coupling standard 6T-SRAM and SCM regions, it enables reliable operations in the full voltage swing, without requiring complex mechanisms for error detection and/or correction. Moreover, since the architecture conserves its state (the values stored in memories and register files) when idle, it doesn't require memory transfers to and from an external storage across idle periods. Hence, it supports high-frequency switches between sensing and processing with a much smaller timing and energy overhead with respect to an alternative based on power gating.

Our proposed architecture further improves its energy efficiency, when executing in the high-workload processing mode, by adopting fine-grained synchronization among cores allowing for efficient producer-consumer notification and *lock-step* execution of parallel algorithms with data-dependent branches. The first mechanism avoids wasteful active waiting between algorithmic steps executed in different sets of cores, while the second one allows for SIMD-

like execution of algorithms that are applied in parallel over different streams of data. This last feature maximizes the synchrony among cores, which is exploited by coalescing concurrent and identical memory requests, originating from different processors, into a single access. Such strategy greatly reduces the power consumed by the instruction memory. The management of synchronization and the computing modes are concurrently implemented in an integrated hardware/software solution comprising a custom instruction set extension (ISE) and a dedicated hardware component for synchronization.

Our approach is particularly beneficial for bio-signal DSP, where applications usually acquire multiple signals at rather low sample rates (e.g., hundreds of Hz in the case of ECGs [4] [3] [5]). The majority of run-time execution is therefore spent on acquisition alone, as opposed to data processing. As illustrated in Figure 1 (top-right), in most cases multiple input signals are independently processed, before combining them into a single stream for further analysis [4]. In this scenario, a multi-core architecture (such as the one depicted in Figure 1, bottom-right) allows to effectively distribute the DSP over multiple resources, connected to peripherals, instruction and data memory. Nonetheless, the presence of multiple cores mandates an efficient strategies to synchronize the execution of an application at run-time. The synchronization mechanism illustrated in this work allows to efficiently execute in parallel SIMD phases enforcing lock-step execution, and, at the same time, manage the producer-consumer relationships between the different phases avoiding unwanted active waiting periods. The combined benefits of efficient parallel execution and operating modes are more than additive. By effectively distributing the workload over multiple computing units, it is in fact possible to reduce the ratio between processing and sensing time, giving ample opportunity for dynamic voltage scaling.

Summing up, our work presents a novel multi-core architecture, featuring a unified strategy to support both different voltage/accessibility modes and fine-grained synchronization. It employs a hybrid memory organization, dedicated synchronization instructions and a hardware synchronization unit. The main contributions of this paper are therefore the following:

- We propose a ULP multi-core system for bio-signal processing, supporting ultra-low voltage operating modes by featuring a heterogeneous memory architecture.
- We detail a low-complexity synchronization technique, able to effectively manage operating modes, lock-step execution and producer-consumer relationships.
- By exploring different partitionings between the 6T and the SCM portions, we devise an optimal hybrid memory architecture, and we evaluate the performance and the power consumption of the resulting platform.

The rest of the paper is organized as follows. Section 2 acknowledges related efforts in the field, while Section 3 describes the target architecture, its hybrid memory subsystem and the proposed synchronization technique. Next, in Section 4 we detail the experimental setup and the results

in terms of energy efficiency. Finally, the conclusions are presented in Section 5.

2 RELATED WORK

Power consumption is a first-grade optimization goal in the design of digital architectures; as such, it is the focus of a vast body of research, as summarized in [17]. At the architectural level, the support of low-voltage operation modes is a widely used strategy to increase the energy efficiency of processors [18], because of its generality and flexibility. Nonetheless, voltage scaling limits the maximum operating frequency of systems, ultimately penalizing their performance.

To overcome this performance loss, processors can be enriched with application-specific custom instructions or accelerators [19], that efficiently support the most frequent operations of a target domain. In the WBSN context, the authors of [10], [11] and [20] have indeed proposed systems employing dedicated filtering, signal compression and FFT engines. The presence of single-function hardware blocks can nonetheless lead to over-specialized architectures, resulting in a loss of flexibility that can only be partially palliated by adopting reconfigurable accelerators [21].

A more generic approach we adopt in this paper is to employ multiple and homogeneous processing units, able to support a target workload at a low clock frequency. This second strategy, popular in many domains such as multimedia [22], [23], is particularly effective in the WBSN scenario, where multiple signals are usually acquired in parallel and processed within a time window [10], [24].

Dynamic Voltage Frequency Scaling (DVFS), carried out by adjusting the performance and the power consumption at run-time according to the workload [25] [26], is often used in conjunction with a multi-core strategy. For bio-signal analysis applications, such workload is dictated by the acquisition rate of signals, resulting in the presence of both high-activity and idle periods, which can be exploited by the adoption of deep sleep modes to increase the energy efficiency [27]. Nonetheless, the reliability of SRAMs decreases when operating at ultra-low voltages [2], posing a hard limit on the voltage range that can be safely employed. To overcome such a problem, the authors of [24] propose a system where different voltage domains are used for computing and storage resources. As opposed to our work, this choices mandate the use of voltage level shifters, that present a non-negligible area [16].

A striking alternative is offered by specialized SRAM implementations that, with larger than standard six-transistors (6T) SRAMs [28] [13], can reliably operate at extremely low V_{dd} . Similarly to [29], [14] and [30], herein we explore the benefits of a hybrid solution, which employs large and low-power SRAM cells only for a small portion of the memory subsystem. As opposed to [14], [30], our solutions do not incur in any error on the computations related to the 6T memory at low voltages. Moreover, compared to [30] and [29] and similarly to [14], [31], we employ only a single and tunable voltage domain for the entire system, resulting in a simpler and leaner implementation. Differently from our previous work [31] which was proposing an hybrid memory design based on 6T and 8T cells, in this work we (i) extend

the original design by using SCM cells instead of the 8T cells which deliver better energy efficiency, while increasing the design flexibility by allowing the integration of smaller memory cuts avoiding additional area overheads; (ii) we propose a novel framework for effective handling of the hybrid memory and transition phases based on the combination of an additional HW synchronization component and a set of programming directives.

The run-time management of multiple resources under tight run-time and memory constraints is a challenging task. To this end, the authors of [32] introduced an approach based on software libraries, that nonetheless incurs in substantial overheads due to busy waitings and system calls. Alternatives relying on hardware locks [33] are also resource-intensive, thus not suited for low-power computing architectures such as WBSNs. As in our previous works [12] [8], we instead support run-time synchronization among different cores with dedicated instruction set extensions, presenting a small area and timing footprint. In [12] and [8], synchronization is only supported to manage parallelism. Herein, we generalize this methodology to orchestrate both parallel execution on multiple resources and dynamically set the operating voltage. In this last respect, leveraging a domain-specific heterogeneous memory system, we go beyond classic DVFS, trading off the accessibility of resources (in addition to the operating frequency) with power consumption.

3 BIO-SIGNAL PROCESSING ARCHITECTURE

The proposed architecture employs a joint synchronization policy to transition between operating modes at different voltage levels, as well as to perform clock-gating of individual computing units. These two strategies target different time granularities: at a coarser granularity, an ultra-low voltage operating point of the entire system is adopted when only data buffering is required (i.e.: during sensing phases), as dictated by the workload of the DSP application and the data acquisition rate. At a finer granularity and while in processing mode, clock-gating enables an efficient synchronization of cores executing code in lock-step or waiting for input data in a producer-consumer relationship, as detailed in Section 3.2.

Both energy-saving strategies are embedded in our multi-core platform, which is composed by an array of *Computing Units* (CUs) interfaced to *Instruction and Data Memories* (IM and DM), as depicted in Figure 2. IM and DM are divided into multiple banks, so that each can be accessed independently and power-gated if they are not required by the application. Each DM bank is itself composed by an area-efficient 6T region (6T-DM) and a highly-reliable SCM region (SC-DM). The communication between cores and memories is based on a high bandwidth logarithmic interconnects, implementing a mesh-of-trees topology and supporting single-cycle communication between cores and memory banks [34]. On the other hand, in case of no banking conflicts, data routing is done in parallel for each core, thus enabling a high sustainable bandwidth for processors-memories communication. In case of multiple conflicting requests, to ensure fair access to memory banks, a round-robin scheduler arbitrates the access to different locations

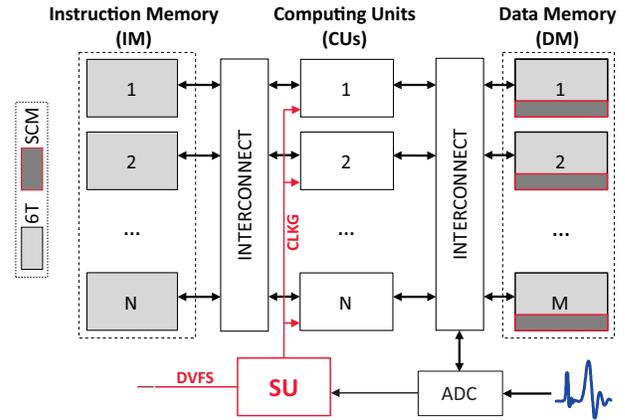


Fig. 2: Target architecture, featuring hybrid DM banks and HW synchronization unit (SU).

of the same bank by different processors. In addition, the interconnect allows to merge simultaneous read requests to the same memory address, reducing the memory accesses and therefore increasing the energy efficiency of the system [8].

A *Synchronization Unit* (SU) is employed to (i) orchestrate the execution of the system, (ii) clock-gate individual cores and (iii) dynamically select the voltage supply level and, therefore, the operating mode. The SU pauses and resumes cores, either after data-dependent branches (to recover lock-step execution) or to manage producer-consumer relationships. Moreover, the SU also dictates the voltage supply of the platform. At the high-V_{dd} *processing* supply level, all computing and storage elements can be reliably employed. Conversely, when all the cores are idle waiting for a window of samples to be acquired, the low-V_{dd} *sensing* mode is enforced, in which only the SC-DM regions are reliably accessible, while the 6T-DM memory are state-retentive. In sensing mode, the analog-to-digital converter (ADC) is in charge of periodically moving the data sampled by the analog front-end to the SC-DM region.

In the following, we detail the implementation of the proposed strategy to support these energy-saving mechanisms, as well as the description of the synchronizer unit which gives the necessary hardware support.

3.1 Hybrid Memory Management Strategy

Considering typical sampling frequencies for biomedical signals (typically around few hundreds of Hertz), the time needed to acquire a window of samples exceeds the time to perform the required computation. Therefore, the workload profile of WBSN application presents periods of low activity, where only data collection is performed. In this *sensing* state, the only requirement for the architecture is to make available enough memory to store locally the data sampled by the ADC. As shown in Figure 3, the only active elements during sensing are the ADC and the reliable SC-DM, where samples are stored for future analysis. In this mode, all the cores, the 6T-DM portion and the IM are inactive. Memory elements beside the SC-DM are not accessible, but their content is reliably retained.

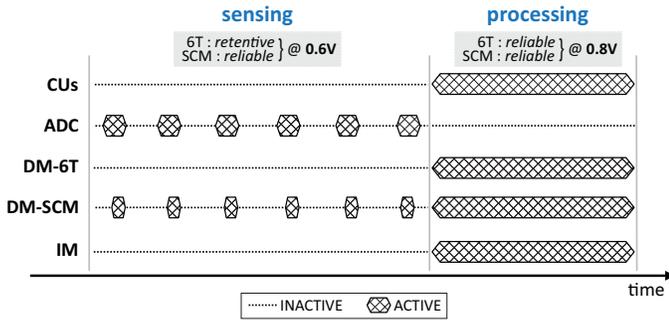


Fig. 3: Active/inactive architectural elements in *sensing* and *processing* states with related hybrid memory behavior.

Once the ADC has transferred the desired number of samples to the data memory, the system switches to *execution* mode (cf. Figure 3), performing a burst of computation on the available data. This operating point is characterized by a high workload, being the required processing elements active and working on the sampled data. The execution mode requires a reliable access to IM and DM banks storing the binary code and application data.

To support this run-time behavior, we consider a hybrid data memory architecture, which overcomes the limitation imposed by classic 6T-SRAM when operating under aggressive voltage scaling. The memory bank structure combines 6T and SCM regions, extending the reliable operating range to low supply voltages. In the case of our target CMOS technology, the SCM portion of the DM is able to reliably operate down to 600mV, while the 6T portion of the DM can be reliably accessed at a minimum level of 800mV (see Section 4.3). Due to the utilization of a single voltage domain, our proposed strategy allows a low-overhead transition between sensing and processing modes, which is only dependent on the rise time of the voltage supply level. This solution extends the one presented in [31] by substituting the 8T reliable memory portion with a SCM design. This lead to a better energy-efficiency, a wider operating range and the possibility of integrating small memory cuts which incurs in large overheads with standard SRAM design.

3.2 Synchronization Strategy

To determine the dynamic voltage supply level of the platform, as well as to properly clock-gate individual cores, we propose a hybrid hardware/software (HW/SW) synchronization mechanism. The approach extends the one described in [8] by also including the management of multiple operation modes. Its hardware support is provided by the above-mentioned SU, which orchestrates the execution of the multi-core system based on the received interrupts from the ADC and the synchronization instructions issued by the cores. Software support consists of a set of dedicated instructions (*SINC*, *SDEC* and *SNOP*), which modify a number of reserved locations (*synchronization points*) in the data memory. Synchronization points, implemented as single data words, store the information regarding (i) which cores have started and ended the execution of a data-dependent branch, and (ii) which consumer cores are clock-gated while waiting for data from producer cores or from

the ADC. One synchronization point is therefore required for each data-dependent branch and each producer/consumer relationship. Each of these words consist of a 1-bit flag per core which indicates if the core is registered for the corresponding event and a core counter which keeps track of how many of them have not arrived to the end of the event. In addition, a *SLEEP* instruction requests the synchronizer to clock-gate the issuing core until the next synchronization event happens (e.g. new data to process is available).

Code Excerpt 1 Example of lock-step code

```

1: function lock_step_example()
2: {
3:     ...in lock-step
4:     SINC(<synch_point_A>)
5:     if(<some condition>) {
6:         conditional_code_B()
7:     }
8:     else {
9:         conditional_cod_C()
10:    }
11:    SDEC(<synch_point_A>)
12:    SLEEP();
13:    ...continue in lock-step
14: }
    
```

3.2.1 Software Adaptations and Mapping

To enforce lock-step execution after data-dependent blocks of code, each core executes a *SINC* instruction before conditional branches, to notify the synchronizer about a possible desynchronization. When the core finishes executing the branch, it issues a *SDEC* and enables clock-gating with a *SLEEP* instruction. After all cores that diverged finish executing the conditional section, the synchronizer wakes them up to resume their execution in lock-step. A simple example showcasing a typical de-synchronization due to a data-dependent branch is presented in Code Excerpt 1. A graphic representation and a time diagram of this run-time behavior is also depicted in Figure 4-a.

Code Excerpt 2 Example of producer code

```

1: function producer_example()
2: {
3:     SINC(<synch_point_B>)
4:     produce_new_data()
5:     SDEC(<synch_point_B>)
6: }
    
```

Code Excerpt 3 Example of consumer code

```

1: function consumer_example()
2: {
3:     while(<no data to consume>) {
4:         SNOP(<synch_point_B>)
5:         SLEEP()
6:     }
7:     consume_data()
8: }
    
```

Producer-consumer relationships require the consumer cores waiting for data to execute a *SNOP* instruction, registering themselves in the corresponding synchronization point. Afterwards, such cores request to be clock-gated by

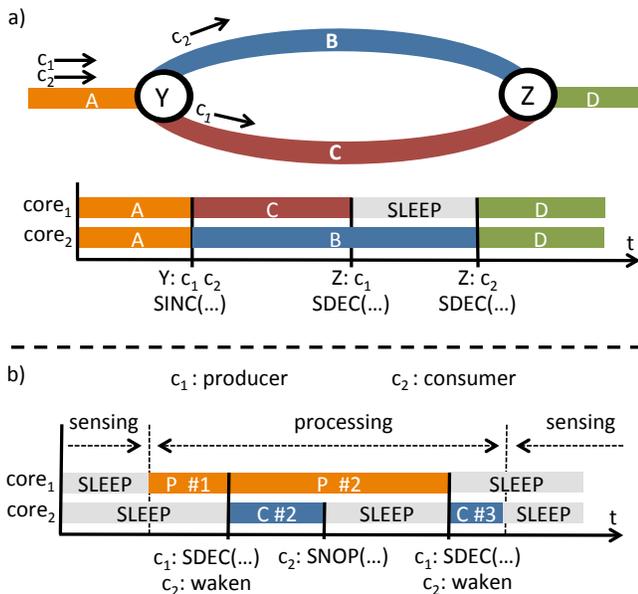


Fig. 4: Synchronization in a diverging data-dependent branch (a) and in a producer-consumer relationship processing windows of 2 samples (b).

issuing a SLEEP instruction, thus avoiding active waiting. Producers, instead, use SINC to register in the synchronization point when starting to compute data for the consumer cores, and SDEC when data is ready. The synchronizer detects when all the necessary input data from the producers is available (i.e. all the producers have issued the SDEC instruction), and resumes execution of all the registered cores. Pseudo-code excerpts presented in Code Excerpts 2 and 3 showcase a generic example of a producer-consumer relationship, which is also represented on the diagram depicted in Figure 4-b.

To map an application to the proposed platform starting from an equivalent single-core implementation, the code (written in C) must be partitioned into phases that can be executed in parallel in a pipelined manner. Phases should correspond to a non-negligible workload, but a finely-tuned load balancing is not required. Subsequently, the custom synchronization instructions are properly placed to manage data-dependent branches and producer/consumer relationships. While these two steps must be manually performed by the application programmer at present, they can be automated. Each phase is then assigned to a number of computing units corresponding to the number of parallel computing streams within a phase (e.g., three CUs are assigned for the “conditioning” phase in Figure 1) and the IM and DM content referring to different phases are mapped in different banks of IM and DM in order to reduce access conflicts. The assignment of computing units and memory banks is performed semi-automatically through linking directives.

3.2.2 Hardware Support: Synchronization Unit

The aforementioned Synchronization Unit (SU) is interfaced between the read-write ports of the cores and the interconnect networks, to monitor the state of each computing unit and orchestrate their execution. In addition, this module

is also connected to the ADC interrupt line and the stall, sleep and wake-up pins of each of the processors. The SU is composed by a sequential and a combinational part, which are detailed in the following and whose behaviours are depicted from a high level of abstraction in the flowcharts in Figure 5.

On one hand, the sequential (clocked) logic is responsible for controlling the transitions between sensing and processing modes (cf. Figure 5-a). A lack of activity while in processing mode (i.e. when all cores have issued a SLEEP instruction as showcased in the producer-consumer relationship of Figure 4-b) triggers a transition towards the low-power state. This condition is detected by the synchronizer, which lowers the clock frequency and the voltage supply to the low-V_{dd} level, setting the system to sensing mode. When a new window of data becomes available, the ADC makes the system transit to the processing mode. In such a case, the synchronizer raises the platform voltage, waits for a stabilization period, increases the clock frequency and wakes up the corresponding cores.

On the other hand, the combinational circuitry coordinates the execution among cores while in processing mode (cf. Figure 5-b). First, explicit stalls due to memory conflicts coming from the interconnect are handled and forwarded to the corresponding cores. Second, lock-step execution is ensured among all those cores issuing the same instruction during the same clock cycle by stalling all of them if one is explicitly stalled due to a memory conflict. Third, in the case of issuing a synchronization instruction, the value to be written into the corresponding synchronization point is derived by setting the necessary flags, modifying the core counter and merging into a single write requests the results of possibly concurrent manipulations of the same point. Moreover, the synchronizer is also in charge of waking-up the registered cores when the core counter to be stored reaches zero, as described in Section 3.2.

4 EXPERIMENTAL RESULTS

In this section we first present the chosen set-up and simulation framework. Then, we make a detailed exploration to choose an optimal balance between the SCM and the 6T memory regions composing the data memory sub-system. Finally, we comparatively evaluate the energy efficiency of the studied architecture featuring the proposed techniques.

4.1 Setup and Evaluation Framework

We consider a target system composed by 8 ULP TamaRISC cores [12], featuring a three-stage pipeline, 16-bit data width and 24-bit instructions. The energy consumption of this core is comparable to commercial low-power processors such as ARM Cortex-M0 [35]. Nonetheless, different cores can be embedded in the proposed system, as long as they allow extensions to incorporate the custom synchronization instructions described in Section 3.2.

Cores are interfaced with a 96 KByte Instruction Memory (32 KWords of 24 bits width) divided into 8 banks and a 64 KByte Data Memory (32 KWords of 16 bits width) divided into 16 banks. Each DM bank presents a reliable region of SCM cells and an area-efficient one implemented as 6T cells.

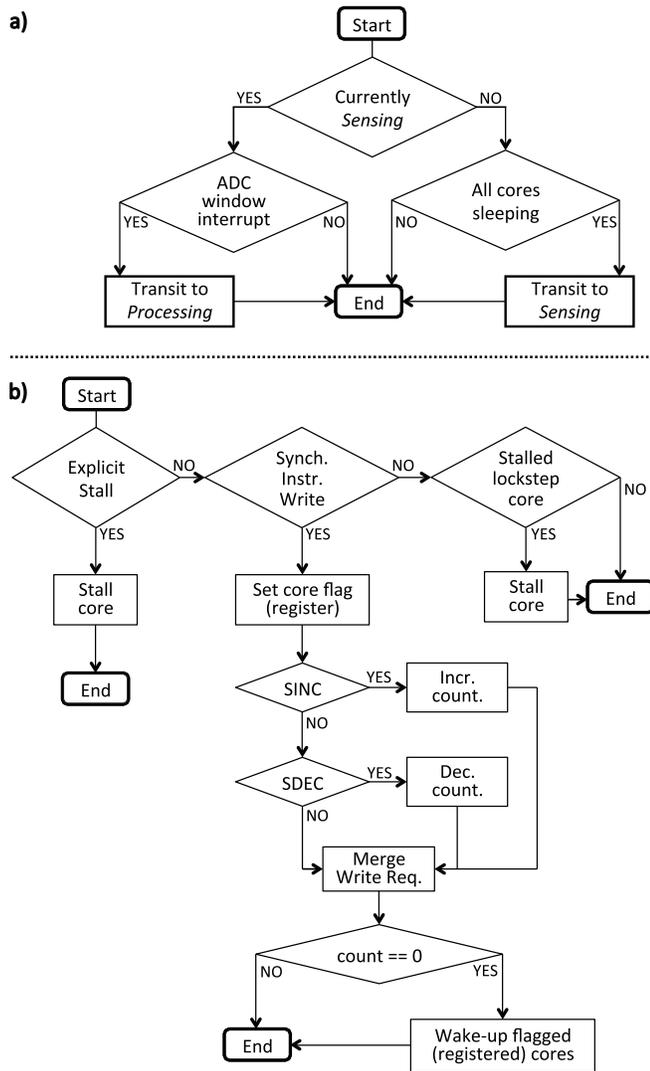


Fig. 5: Flowchart describing the synchronizer's behavior: (a) Clocked logic governing transitions between platform modes. (b) Combinational part orchestrating execution while in processing mode.

The system clock frequency in the processing mode is 20MHz (the maximum possible in the chosen technology for the target system with 800mV supply voltage), while in sensing mode the clock is set at 10KHz to lower the dynamic power. The resources that are not required to be active by the application, such as unnecessary cores and IM banks, can be powered down at boot time.

Similarly to [8], the developed experimental framework combines detailed post-layout characterization of the system with faster cycle-accurate simulations of complex biosignal analysis applications. Processors and their C compiler are designed using ASIP Designer from Synopsys. The resulting RTL and SystemC implementations are then embedded as components of the multi-core virtual platform. The lower-level RTL description is used to characterize each of the employed architectural blocks at a 40nm technology node through an EDA toolchain: Design Compiler from Synopsys and Encounter from Cadence are used in the synthesis and place- and-route step, while Modelsim from

Mentor Graphics is employed to retrieve switching activity of the platform when executing synthetic benchmarks. In a second step, the obtained energy values are imported in the higher-level SystemC platform simulator, allowing the evaluation of energy consumptions when executing complete real-world applications under different architectural configurations.

4.2 Bio-signal Processing Benchmarks

We have considered four bio-signal analysis benchmarks, which are widely used in the field of electrocardiogram embedded processing [8] [3] and present different levels of complexity and parallelism. They also present different tradeoffs in terms of results elaboration and runtime requirements (i.e. computational and memory resources). Their characteristics are summarized next.

Compressed Sensing (8L-CS): This signal compression algorithm has been extensively investigated in different domains, including low-power sensing and image processing. CS assumes that the input data has a sparse representation in a transformed domain, so that the data dimensionality can be dramatically reduced. Mamaghanian et al, [3] used CS to implement a low-complexity ECG compression algorithm based on the multiplication of the input vector of samples by a sparse *sensing matrix* resulting in a much smaller set of measurements which can be later used to reconstruct the original signal. The algorithm used in this benchmark utilizes a software version of the energy-efficient pseudo-random number generator introduced in [36] to generate the sensing matrix used to achieve a 50% compression. The resulting 8L-CS does not present any data-dependent branch nor code divergence leading to an almost full lockstep execution of code among cores. In our implementation, eight ECG leads (8L) are processed in parallel employing all the cores of the platform.

Morphological Filtering (3L-MF): ECG acquisitions are normally corrupted by different sources of noise and artifacts (including human perspiration, muscular activity or small displacements of the employed electrodes), which must be filtered to retrieve a high-quality signal. Morphological filtering [37] performs this task by employing structuring elements to unwanted components from input streams. Herein, we consider an optimized version of this algorithm [4], which removes both low and high frequency noise components using flat and peak-shaped structuring elements. This benchmark filters in parallel ECG signals from a standard three-channel acquisition using three computing cores. In contrast to 8L-CS, the presence of numerous data-dependent branches in the 3L-MF code highlights the ability of the platform to recover lockstep execution after diverging sections of code.

ECG Delineation (3L-MMD): This application perform the automated identification of the fiducial points of ECGs, i.e.: the starting, peak and end of the three ECG main waves (QRS complex, P and T waves). This process is known as delineation. On top of the filtering stage of 3L-MF, this benchmark performs a Root Mean Square (RMS) fusion of the filtered signals resulting in a single ECG stream, that is later delineated employing an algorithm based on Multi-scale Morphological Derivatives as in [8]. This benchmark

TABLE 1: Relevant runtime characteristics of the bio-signal processing benchmarks

	8L-CS	3L-MF	3L-MMD	RP-CLASS
Active cores	8	3	5	6
Active IM banks	1	1	4	6
Parallelism (IPC)	7.99	2.98	2.24	3.65
Code overhead (%)	0	2.55	0.91	0.71
Sensing time (%)	94.71	95.11	90.97	92.06
Processing time (%)	5.29	4.89	9.03	7.94
<i>Synch. cycles (%)</i>	0	1.67	0.96	0.62

requires synchronization for both lockstep execution and producer-consumer notifications to transfer data among the three processing stages, namely filtering, combination and delineation. 3L-MMD employs five cores of the platform, three of them executing code in lockstep.

Selective ECG processing (RP-CLASS): This benchmark, detailed in [5], embeds a neuro-fuzzy classifier that detects abnormal heartbeats. When a detection occurs, a further analysis is executed on the abnormal heartbeat. By default a single ECG channel is filtered and analyzed by the classifier. Only for abnormal heartbeat, three-channels filtering and delineation is performed, as described above for 3L-MMD. This benchmark presents a complex structure, requiring lockstep execution in some cases and a sophisticated control flow across cores. RP-CLASS utilizes 6 cores of the platform and benefits from both proposed synchronization mechanisms.

Table 1 reports the most relevant workload characteristics of the four bio-signal processing benchmarks considered in this work, when mapped on the target multi-core platform. It highlights the small overhead caused by the insertion of synchronization instructions, in terms of runtime as well as code size. Moreover, the table shows that the sensing periods, where the processing cores stay idle, are dominant, accounting in all the cases for more than 90% of the time.

4.3 Reliable Memory Requirements

In the first round of experiments we have explored the energy efficiency of the multi-core platform when different

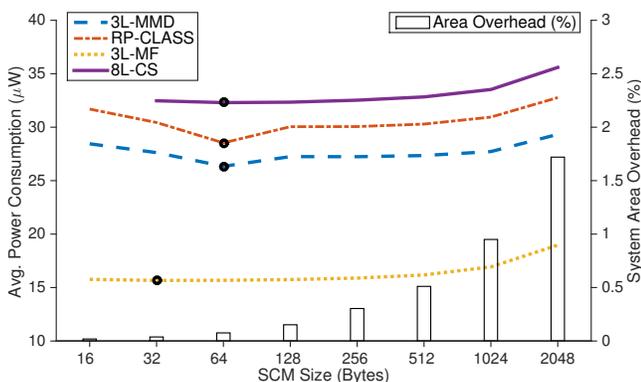


Fig. 6: Power consumption and area overhead varying the SCM-DM size. The minima are highlighted by a rounded marker.

TABLE 2: Leakage and dynamic power of the target platform for the bio-signal processing benchmarks during *sensing* and *processing* phases.

	Avg. Power Consumption (uW)			
	<i>processing</i>		<i>sensing</i>	
	Leakage	Dynamic	Leakage	Dynamic
3L-MMD	2.09	276.52	1.24	0.06
RP-CLASS	2.39	339.92	1.41	0.06
8L-CS	1.81	588.47	1.12	0.06
3L-MF	1.62	298.77	0.98	0.06

sizes of highly-reliable SCMs are employed (Figure 6). The considered SCM design uses a cross-coupled pair of AND-OR-INV (AOI) as the storage element, which is more energy efficient than 6T-SRAM. The choice of this memory element, combined with the use of regular place and route, results in more than 3x area saving [14] compared to the SCM design in [38] that uses a latch as the storage element.

In applications with multiple producer-consumer computation phases, the availability of a large SCM region enables the acquisition of wider samples windows and thus maximizes the pipelined execution of different phases. For what concerns the supply voltage levels for *sensing* and *processing*, such values were determined considering the measurements results presented in [14]. The minimum operating voltage point was measured by the authors of [14] over nine chips and the results show that for the majority of the chips, the SCMEM operated correctly at voltages below 0.4V and on average it has 400mV lower minimum operating voltage point than the 6T memory. However, we considered the worst case scenario, i.e. the highest minimum voltage for both SCMEM and 6T among the different measured chips, which conservatively lead to 600mV for sensing and 800mV for processing.

As shown in Figure 6, the illustrated trade-off results in an optimal size of the SCM region of 64 bytes for three out of four of the considered benchmarks. Thus, we used this size in the experiments of the following sections. This choice increases the area of the data memory by 0.2% and leads to a negligible system area overhead ($\approx 0.1\%$) w.r.t. a design including only 6T-SRAM. Since irregular 6T memory banks cannot be generated with standard memory compilers, the addition of small SCM regions does not imply a reduction of the 6T part but a superposition. As expected, the benefits of employing wider SCM regions are most evident in the 3L-MMD and RP-CLASS benchmarks, which expose producer-consumer relationships. For these cases, the ability to process a larger window of data in a pipelined fashion across multiple processors is leveraged to increase parallelism and reduce the time spent in processing mode. For the other two benchmarks, only modest gains can be achieved by employing bigger SCMs due to the reduced number of transitions between sensing and processing modes. Such time overhead, due to transitioning between processing and sensing modes, has been conservatively modeled as 100ns in our experiments, taking into account wide margins with respect to silicon implementations [39], [40].

4.4 Power Consumption Evaluation

Table 2 reports the obtained values for leakage and dynamic power for both processing and sensing phases. In all cases,

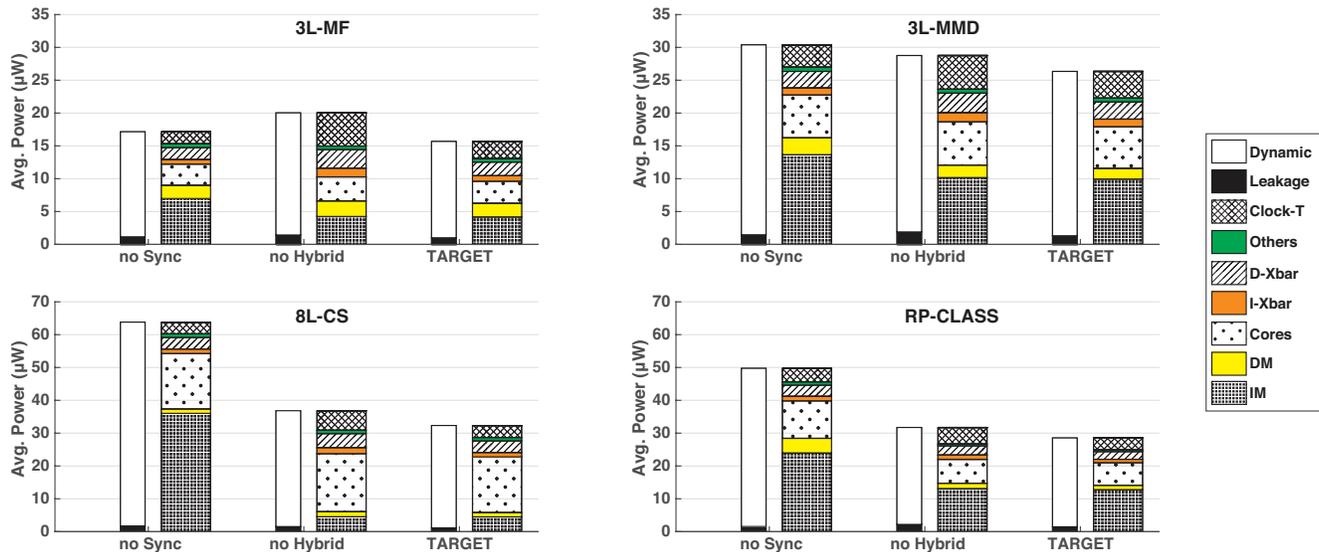


Fig. 7: Power consumption of the target and baseline systems (no Hybrid, no Sync) for the considered bio-signal processing benchmarks. The bars present the power breakdown for the architectural components beside its leakage/dynamic ratio.

leakage power is effectively reduced by $\approx 40\%$ when transitioning to the state-retentive sensing mode. In the context of WBSN applications, this aspect is particularly relevant, since the benchmarks spend in this state up to 95% of their execution time. As expected, dynamic power is negligible during the sensing periods where most of the system is clock-gated and the voltage is reduced.

To highlight the efficiency of our proposed solution, we compared it with two different baseline systems. The first baseline system (*no Hybrid* in Figure 7) does not implement the hybrid memory subsystem and it is always running at the higher voltage level of 800mV, while still employing synchronization for managing lock-step execution and efficient producer-consumer waiting. We set the working frequency of this baseline at 2.5MHz (which allows it to barely meet the real-time constraints of the considered benchmarks) in order to minimize the power consumption of elements which are not clock-gated, such as the clock tree itself. The energy profile of the *no Hybrid* architecture has been investigated in detail in our previous work [8], which showcases how synchronization alone leads to tangible efficiency gains with respect to a single-core alternative (40% less energy) and with respect to a multi-core which does not support synchronization (32% less energy).

In the second case (*no Sync* in Figure 7), we employ active waiting instead of clock-gating to manage producer-consumer relationships and lock-step execution is disabled, but we still allow the system to transit to the low-power sensing mode when all the cores are idle. To make a fair comparison, in this setting access conflicts are reduced by assigning different IM and DM banks to each processor, even when they execute the same computing phase. As in the target system, *no Sync* adopts a 20MHz clock when in processing mode, with the aim of increasing as much as possible the time spent in the energy-efficient sensing mode.

Figure 7 shows the breakdown of the average power

consumption for 60s of activity for all the three architectures considering the time spent in sensing and processing modes. Two main conclusions can be drawn from this comparison. First, energy savings are consistently achieved in all benchmarks by employing different operation modes supported by a hybrid data memory. Savings derive from a reduction of up to 32% in leakage power of all system components, as well as from the dynamic power of the clock tree (reaching 60% reduction in 3L-MF) due to the lower frequency employed in sensing mode. Efficiency gains grow linearly with the ratio between the time spent in sensing mode and the total run-time. The overhead deriving from the use of SCMs in hybrid banks is instead negligible, due to their small required size of just few bytes. Therefore, a high-workload application, always residing in processing mode, would require the same energy in our target system and in the *No hybrid* one.

Second, synchronization can effectively increase the system efficiency. In fact, synchronization allows merging memory requests of data and instruction words, thus minimizing the accesses to memories and the number of active banks, leading to a reduction in leakage and dynamic energy. These two aspects are especially beneficial when multiple cores execute the same processing phase, as in the case of 8L-CS where memory consumption is reduced by 83%.

5 CONCLUSIONS

Nowadays, very promising opportunities for increasing the energy efficiency of digital platforms gains reside at the architectural and system design level. Such solutions require the specialization of computing resources for a target application domain. In this work, we have presented a dedicated computing architecture for bio-medical signal processing, which harnesses the high-level features of applications in this domain. The proposed system adapts to the varying

workload requirements typical of bio-signal analysis applications, which are leveraged as a energy-saving opportunity by a hybrid memory scheme. Moreover, the employed multi-core structure exploits parallel and pipelined execution, matching application-level characteristics, while allowing SIMD execution and avoiding active waiting.

The platform supports two operating modes, which determine the accessibility of resources and the system energy consumption: a high-performance processing mode and a low-power sensing mode. Such dual-mode operation is supported by employing specialized data memory banks, which include an area-efficient 6T SRAM partition and a low-voltage reliable SCM partition. At the same time, our strategy includes a light-weight mechanism to perform the transitions between modes and allow synchronization of cores for broadcasting of data and instructions and clock-gating of individual cores. Experimental results showcased that, by using our proposed methodology, overall energy gains of up to 50% can be achieved while requiring a negligible 0.1% area increase in a multi-core platform devoted to bio-medical DSP applications.

REFERENCES

- [1] World Health Organization (WHO), "Cardiovascular diseases." <http://www.who.int/mediacentre/factsheets/fs317/en>, 2015. [Online].
- [2] Y. Hao and R. Foster, "Wireless body sensor networks for health-monitoring applications," *Physiological measurement*, vol. 29, no. 11, p. R27, 2008.
- [3] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vanderghyest, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 9, pp. 2456–2466, 2011.
- [4] F. Rincón, J. Recas, N. Khaled, and D. Atienza, "Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 6, pp. 854–863, 2011.
- [5] R. Braojos, G. Ansaloni, and D. Atienza, "A methodology for embedded classification of heartbeats using random projections," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 899–904, IEEE, 2013.
- [6] D. C. Daly and A. P. Chandrakasan, "A 6b 0.2-to-0.9v highly digital flash ADC with comparator redundancy," in *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, pp. 554–635, Feb 2008.
- [7] F. Zhang, J. Holleman, and B. P. Otis, "Design of ultra-low power biopotential amplifiers for biosignal acquisition applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, pp. 344–355, Aug 2012.
- [8] R. Braojos, A. Dogan, I. Beretta, G. Ansaloni, and D. Atienza, "Hardware/software approach for code synchronization in low-power multi-core sensor nodes," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2014, pp. 1–6, IEEE, 2014.
- [9] Texas Instruments, "Measuring Bluetooth Low Energy Power Consumption." <http://www.ti.com/lit/an/swra347a/swra347a.pdf>, 2016. [Online].
- [10] H. Kim, S. Kim, N. Van Helleputte, A. Artes, M. Konijnenburg, J. Huisken, C. Van Hoof, and R. F. Yazicioglu, "A configurable and low-power mixed signal SoC for portable ECG monitoring applications," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 8, no. 2, pp. 257–267, 2014.
- [11] H. Kim, Y. Kim, and H.-J. Yoo, "A low energy bio sensor node processor for continuous healthcare monitoring system," in *Solid-State Circuits Conference, 2008. A-SSCC'08. IEEE Asian*, pp. 317–320, IEEE, 2008.
- [12] A. Y. Dogan, J. Constantin, M. Ruggiero, A. Burg, and D. Atienza, "Multi-core architecture design for ultra-low-power wearable health monitoring systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 988–993, EDA Consortium, 2012.
- [13] B. H. Calhoun and A. Chandrakasan, "Analyzing static noise margin for sub-threshold SRAM in 65nm CMOS," in *Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European*, pp. 363–366, IEEE, 2005.
- [14] D. Bortolotti, H. Mamaghanian, A. Bartolini, M. Ashouei, J. Stuijt, D. Atienza, P. Vanderghyest, and L. Benini, "Approximate compressed sensing: ultra-low power biosignal processing via aggressive voltage scaling on a hybrid memory multi-core processor," in *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, pp. 45–50, ACM, 2014.
- [15] A. Teman, D. Rossi, P. Meinerzhagen, L. Benini, and A. Burg, "Power, area, and performance optimization of standard cell memory arrays through controlled placement," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 4, p. 59, 2016.
- [16] W.-K. Mak and J.-W. Chen, "Voltage island generation under performance requirement for SoC designs," in *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, pp. 798–803, IEEE Computer Society, 2007.
- [17] W. Nebel and J. Mermet, *Low power design in deep submicron electronics*, vol. 337. Springer Science & Business Media, 2013.
- [18] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 251–259, ACM, 2001.
- [19] J. Cong, Y. Fan, G. Han, and Z. Zhang, "Application-specific instruction generation for configurable processor architectures," in *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, pp. 183–189, ACM, 2004.
- [20] J. Kwong and A. P. Chandrakasan, "An energy-efficient biomedical signal processing platform," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 7, pp. 1742–1753, 2011.
- [21] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 7, pp. 1625–1637, 2013.
- [22] Y. He, Y. Pu, R. Kleihorst, Z. Ye, A. A. Abbo, S. M. Londono, and H. Corporaal, "Xetal-pro: An ultra-low energy and high throughput simd processor," in *Proceedings of the 47th Design Automation Conference*, pp. 543–548, ACM, 2010.
- [23] Y. Pu, D. Gyvez, J. Pineda, H. Corporaal, and Y. Ha, "An ultra-low-energy/frame multi-standard jpeg co-processor in 65nm CMOS with sub/near-threshold power supply," in *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pp. 146–147, IEEE, 2009.
- [24] J. Hulzink, M. Konijnenburg, M. Ashouei, A. Breeschoten, T. Berset, J. Huisken, J. Stuyt, H. de Groot, F. Barat, J. David, et al., "An ultra low energy biomedical signal processing system operating at near-threshold," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 5, no. 6, pp. 546–554, 2011.
- [25] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Proceedings of the 1998 international symposium on Low power electronics and design*, pp. 76–81, ACM, 1998.
- [26] M. Ashouei, J. Hulzink, M. Konijnenburg, J. Zhou, F. Duarte, A. Breeschoten, J. Huisken, J. Stuyt, H. De Groot, F. Barat, et al., "A voltage-scalable biomedical signal processor running ECG using 13pj/cycle at 1MHz and 0.4v," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2011 *IEEE International*, pp. 332–334, IEEE, 2011.
- [27] M. Seok, S. Hanson, Y.-S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "The Phoenix processor: A 30pw platform for sensor applications," in *VLSI Circuits, 2008 IEEE Symposium on*, pp. 188–189, IEEE, 2008.
- [28] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T sub-threshold SRAM employing sense-amplifier redundancy," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 141–149, 2008.
- [29] R. G. Dreslinski, G. K. Chen, T. Mudge, D. Blaauw, D. Sylvester, and K. Flautner, "Reconfigurable energy efficient near threshold cache architectures," in *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, pp. 459–470, IEEE, 2008.
- [30] I. J. Chang, D. Mohapatra, and K. Roy, "A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 2, pp. 101–112, 2011.

- [31] D. Bortolotti, A. Bartolini, C. Weis, D. Rossi, and L. Benini, "Hybrid memory architecture for voltage scaling in ultra-low power multi-core biomedical processors," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6, IEEE, 2014.
- [32] C. Ferri, R. I. Bahar, M. Loghi, and M. Poncino, "Energy-optimal synchronization primitives for single-chip multi-processors," in *Proceedings of the 19th ACM Great Lakes symposium on VLSI*, pp. 141–144, ACM, 2009.
- [33] C. Stoif, M. Schoeberl, B. Liccardi, and J. Haase, "Hardware synchronization for embedded multi-core processors," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pp. 2557–2560, IEEE, 2011.
- [34] A. Rahimi, I. Loi, M. R. Kakoe, and L. Benini, "A fully-synthesizable single-cycle interconnection network for shared-L1 processor clusters," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pp. 1–6, March 2011.
- [35] ARM Ltd., "Cortex-M0 Processor." <http://www.arm.com/products/processors/cortex-m/cortex-m0.php>, 2015. [Online].
- [36] J. Constantin, A. Dogan, O. Andersson, P. Meinerzhagen, J. Rodrigues, D. Atienza, and A. Burg, "TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing," in *VLSI and System-on-Chip (VLSI-SoC), 2012 IEEE/IFIP 20th International Conference on*, pp. 159–164, Oct 2012.
- [37] Y. Sun, K. L. Chan, and S. M. Krishnan, "ECG signal conditioning by morphological filtering," *Computers in Biology and Medicine*, vol. 32, pp. 465–479, Sept. 2002.
- [38] O. Andersson, B. Mohammadi, P. Meinerzhagen, A. Burg, and J. N. Rodrigues, "Dual-v t 4kb sub-v t memories with < 1 pw/bit leakage in 65 nm CMOS," in *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, pp. 197–200, IEEE, 2013.
- [39] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pp. 123–134, IEEE, 2008.
- [40] W. Kim, D. Brooks, and G.-Y. Wei, "A fully-integrated 3-level DC-DC converter for nanosecond-scale DVFS," *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 1, pp. 206–219, 2012.



Andrea Bartolini received a Ph.D. degree in Electrical Engineering from the University of Bologna, Italy, in 2011. He is currently a post-doctoral researcher in the Department of Electrical, Electronic and Information Engineering Guglielmo Marconi (DEI) at the University of Bologna. He also holds a postdoc position in the Integrated Systems Laboratory at ETH Zurich. His research interests concern dynamic resource management ranging from embedded to large scale HPC systems with special emphasis on software-level thermal and power-aware techniques. His research interest also includes ultra-low power design strategies for bio-sensors nodes operating in near-threshold.



Giovanni Ansaloni is currently a post-doctoral researcher at the Faculty of Informatics of Università della Svizzera Italiana (USI-Lugano, Switzerland). From 2011 to 2015, he was a researcher at EPFL (Lausanne, Switzerland). He received the MS degree in Electronic Engineering from University of Ferrara (Italy) in 2003, the MAS degree from the ALaRI institute (Switzerland) in 2005 and the PhD Degree from University of Lugano (Switzerland) in 2011. His research efforts focus on smart Wireless Body Sensor Nodes systems and applications, including software optimizations of processing algorithms for bio-signal analysis and architectural explorations of ultra-low-power WBSN platforms.



Luca Benini is full professor at the University of Bologna and he is the chair of Digital Circuits and Systems at ETHZ. He has served as chief architect for the Platform2012/STHORM project in STmicroelectronics, Grenoble in the period 2009-2013. He has held visiting and consulting researcher positions at EPFL, IMEC, Hewlett-Packard Laboratories and Stanford University. Dr. Benini's research interests are in energy-efficient system design and multi-core SoC design. He is also active in the area of energy-efficient smart sensors and sensor networks for biomedical and ambient intelligence applications. He has published more than 700 papers in peer-reviewed international journals and conferences, four books and several book chapters. He is a fellow of the IEEE and a member of the Academia Europaea.



Rubén Braojos received his M.Sc. degree in computer science and engineering from Complutense University of Madrid (UCM), Spain, in 2010. He is currently a Ph.D candidate at the Embedded Systems Laboratory (ESL) at the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. His research interests include embedded bio-signal processing, ultra-low power embedded systems and WBSN applied to the field of healthcare.



Daniele Bortolotti received the M.S. degree in Electronic Engineering and the Ph.D. degree in Electronics, Computer Science, and Telecommunications from the University of Bologna, Italy in 2010 and 2014, respectively. He is currently a Post-Doctoral Researcher in the Department of Electrical, Electronic and Information Engineering Guglielmo Marconi (DEI) at the University of Bologna. The focus of his research has initially been on virtual platforms, architectural aspects for multi-processors systems-on-chip. Recently

his focus comprises HW/SW design strategies for ultra-low power bio-sensors nodes operating in near-threshold for WBSN applications and low-level power management techniques for many-cores HPC nodes.



David Atienza (M'05-SM'13-F'16) is associate professor of electrical and computer engineering, and director of the Embedded Systems Laboratory (ESL) at the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland. He received his MSc and PhD degrees in computer science and engineering from UCM, Spain, and IMEC, Belgium, in 2001 and 2005, respectively. His research interests include system-level hardware-software co-design methodologies for high-performance

multi-processor system-on-chip (MPSoC) and ulow-power embedded systems, including especially new 2-D/3-D thermal-aware design for MPSoCs and ultra-low power system architectures for wireless body sensor nodes. He is a co-author of more than 250 papers in peer-reviewed international journals and conferences, several book chapters, and five U.S. patents. Dr. Atienza received the IEEE CEDA Early Career Award in 2013, the ACM SIGDA Outstanding New Faculty Award in 2012, a Faculty Award from Sun Labs at Oracle in 2011, and was Distinguished Lecturer of IEEE CASS in 2014-2015. He is an IEEE Fellow and Senior Member of ACM.