Hardware/Software Co-Design of Ultra-Low Power Biomedical Monitors

THÈSE N^{O} 7314 (2016)

PRÉSENTÉE LE 9 DÉCEMBRE 2016
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DES SYSTÈMES EMBARQUÉS
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ruben BRAOJOS LOPEZ

acceptée sur proposition du jury:

Prof. J.-Ph. Thiran, président du jury
Prof. D. Atienza Alonso, directeur de thèse
Prof. F. Catthoor, rapporteur
Prof. L. Pozzi, rapporteuse
Prof. A. P. Burg, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

The present is theirs;
the future, for which I really worked, is mine.
— Nikola Tesla

To my grandfather Basilio.
To my parents.
To my family.

-

A mi abuelo Basilio.
A mis padres.
A mi familia.

Acknowledgements

I would not have been able to carry out the work presented in this thesis without the excellent guidance and support given by my advisor, *Prof. David Atienza*. Back in 2009, when I arrived to EPFL as an exchange student who did not know anything about research and almost nothing about the topic of this thesis, he gave me an opportunity to join his group as a trainee. Later, in 2011, he accepted me to start my PhD, which now concludes with this manuscript. During all this time, he has made me understand what research is, how to collaborate in a team and why critical thinking, patience and hard work are the foundations to do a good job. Not only I have learned a lot from him on the professional and academic sides but also we have developed a personal friendship that I hope will last for many years ahead.

I would like to dedicate the next words to thank the members of the jury that evaluated this thesis. I wanted to show my gratitude to *Prof. Jean-Philippe Thiran*, who accepted to be the president of the jury, and who has been a superb director of the EE doctoral school to which I had the honor to belong; to *Prof. Francky Catthoor* for his outstanding feedback in the elaboration of the final version of this manuscript; to *Prof. Laura Pozzi* for her careful and thorough comments on the text; and to *Prof. Andy Burg* for his infinite support not only during the evaluation of the thesis but also throughout all these last years.

In the next lines I want to thank my family for their help, sacrifice and support, which has allowed me to reach this point of my life. I take the liberty to address the next paragraph to them in our native language, Spanish:

Como decía en las líneas anteriores, me gustaría agradecer a mi familia toda la ayuda, el sacrificio y el apoyo incondicional que me han permitido llegar a este punto de mi vida. Gracias especialmente a mis padres, ***José Antonio y Rosa María****, sin los cuales no podría haber llegado a ser quien soy. A mi padre, por ser el ejemplo de superación que he seguido desde pequeño y por enseñarme que casi nada es imposible si se trabaja duro. A mi madre, a quien debo mi forma de ser, quien me ha educado en el respeto y quien, aún estando lejos de ello, me ha hecho sentir capaz de todo. No puedo olvidarme de mi hermano Borja, que aún siendo menor que yo en edad, es mucho más grande en cualquier otro buen aspecto. Debo también recordar a mi hermana Inma, a la que tanto quiero, y a mi sobrino Aitor, que tantas alegrías me ha dado a pesar de su corta edad. Quiero recordar especialmente a mi abuelo Basilio, a cuya memoria está dedicada esta tesis doctoral y quien ha sido para mi otro ejemplo a seguir. También a mi abuela Carmen, que tan querido me hace sentir cada vez que puede con pequeños gestos que para mi representan un mundo. Finalmente, me gustaría acabar este párrafo agradeciendo al resto de mi familia por todo el ánimo y la fuerza que me han dado durante este periodo.*

At this point, I cannot continue without mentioning one of the most important persons in my life, *Marina*. My PhD has been like running a marathon: exciting, difficult, exhausting... Marina arrived to my life just in time, during the last period where the strength started to diminish. She has helped me to look ahead, to believe in myself and she has made me feel what I had never felt before for someone. I want to thank her for her unlimited support, her unquestionable patience and her unbreakable love.

In Spain, it is common to say that one does not choose his own family, but I do not agree with this idea. I think that you choose certain people with whom to share important moments of you life. Those that make you feel good and beloved. Those that you also love. With these words I would like to thank that second family, composed of friends that are there when you need them, that make you feel like at home, even if you are thousands of kilometers away. Special thanks to *Patricia Tovar*, *Kepa* (and *little Daniela*), *Marta Tovar*, *Michelangelo*, *Julia*, *Lara* (and *little Alessia*), *Fran*, *Laura* (and *little Alba*), *Hossein*, *Ivan*, *Miguel*, *Marta Camarena*, *Victor*, *Mateo* and *Patricia F. Varela*.

If one could have one more family, I would say that my third one is formed by all the members and ex-members of the Embedded Systems Laboratory. First, I would like to thank *Homeira* – without whom this lab would not be the same – for her silent but constant support and her selfless help. Then, I would like to thank all my officemates (*Hossein*, *Giovanni*, *Fabio*, *Leila* and *Eli*) for being one of the reasons why I want to come to work everyday. Next, I would like to thank *Ivan*, *Vincenzo* and *Pablo*, because even if we never were officemates, we spent so much time and shared so many good moments together. I would like to thank the first ESL group (*Ahmed, Mohamed, Martino, Alessandro, Arvind, Shivani, Emi and Karim*) for all the good memories that I will keep forever. Also to the new generation (*Dixie, Loris, Soumya, Greg, Tim, Yasir, Amir, Ali and Arman*), for the nice atmosphere they have created that I hope will last for years. Of course I do not want to forget those that temporary passed by the lab (*Alair, Ale Nacci, PRG, Joaquín, Nil, Camilo, Jungsoo, Srini* and many others) who, in some sense or another, collaborated to make this group so special. Finally, I would like to thank *Francine* and our IT master, *Rodolphe*, for their daily assistance.

Special thanks go to those who I collaborated with during my PhD and from whom I have learnt priceless lessons, particularly *Jeremy, Daniele, Andrea, Tony, Simone, Prof. Benini, Prof. Vesin and Prof. Mitra*. Also to *Raphael, Elvira* and all the members of the EDEE Coaching Team with whom I have spent so many good experiences at the EE doctoral school.

Finally, I would not like to finish these acknowledgements without mentioning the friends that I met during my PhD, particularly *Jacopo, Shashi, Camilla, Andrea, Cristina, Matteo and Anna*. Thanks also to those friends I met during my studies in Madrid (*Javi, Hector, David, Jose, Alex* and all the FDI guys). And of course, special thanks to my longtime friends who I enjoy to meet every time I come back to my hometown (*Juan Luís, Tania, Leti, Fati, Ángel, Inma* and many others).

Lausanne, 17 November 2016 Rubén Braojos López

Abstract

Ongoing changes in world demographics and the prevalence of unhealthy lifestyles are imposing a paradigm shift in healthcare delivery. Nowadays, chronic ailments such as cardiovascular diseases, hypertension and diabetes, represent the most common causes of death according to the World Health Organization. It is estimated that 63% of deaths worldwide are directly or indirectly related to these non-communicable diseases (NCDs), and by 2030 it is predicted that the health delivery cost will reach an amount comparable to 75% of the current GDP. In this context, technologies based on Wireless Sensor Nodes (WSNs) effectively alleviate this burden enabling the conception of wearable biomedical monitors composed of one or several devices connected through a Wireless Body Sensor Network (WBSN). These resource-constrained systems allow for long term recording of biological signals and perform embedded advanced digital signal processing (DSP) enabling autonomous diagnosis even outside a hospital environment. Energy efficiency is of paramount importance for these devices, which must operate for prolonged periods of time with a single battery charge. Therefore, in order to minimize power consumption, both the software executing in these platforms and the underlying hardware require a carefully tailored design.

In this thesis I propose a set of hardware/software co-design techniques to drastically increase the energy efficiency of biomedical monitors. To this end, I jointly explore different alternatives to reduce the required computational effort at the software level while optimizing the power consumption of the processing hardware by employing ultra-low power multi-core architectures that exploit DSP application characteristics.

First, at the sensor level, I study the utilization of a heartbeat classifier to perform **selective advanced DSP** on state-of-the-art ECG biomedical monitors. To this end, I developed a framework to design and train real-time, lightweight heartbeat neuro-fuzzy classifiers, detailing the required optimizations to efficiently execute them on a resource-constrained platform. Then, at the network level I propose a more complex transmission-aware WBSN for activity monitoring that provides different tradeoffs between classification accuracy and transmission volume. In this work, I study the combination of a minimal set of WSNs with a smartphone, and propose two classification schemes that trade accuracy for transmission volume. The proposed method can achieve accuracies ranging from 88% to 97% and can save up to 86% of wireless transmissions, outperforming the state-of-the-art alternatives.

Second, I propose a **synchronization-based low-power multi-core architecture** for bio-signal processing. I introduce a hardware/software synchronization mechanism that allows to achieve high energy efficiency while parallelizing the execution of multi-channel DSP applications. Then, I generalize the methodology to support bio-signal processing applications with an arbitrarily high degree of parallelism. The proposed technique includes a dedicated lightweight synchronizer and an instruction set extension (ISE) of the processing cores. Due to the benefits of SIMD execution and software pipelining, the architecture can reduce its power consumption by up 38% when compared to an equivalent low-power single-core alternative. Finally, I focus on the optimization of the multi-core memory subsystem, which is the major contributor to the overall system power consumption. First I considered a **hybrid memory subsystem** featuring a small reliable partition that can operate at ultra-low voltage enabling low-power buffering of data and obtaining up to 50% energy savings. Second, I explore a **two-level memory hierarchy based on non-volatile memories** (NVM) that allows for aggressive fine-grained power gating enabled by emerging low-power NVM technologies and monolithic 3D integration. Experimental results show that, by adopting this memory hierarchy, power consumption can be reduced by 5.42x in the DSP stage.

Key words: Bio-signal processing; Ultra-low Power Architectures; Hardware/Software Co-Design; Biomedical Monitors; Multi-Core Code Synchronization; Energy-Efficient Multi-Core Platforms; Electrocardiogram Embedded Processing; Lightweight Heartbeat Classification;

Résumé

La croissance démographique actuelle et l'augmentation des modes de vie malsains à l'échelle mondiale, imposent des changements significatifs en ce qui concerne les prestations de soins de santé. De nos jours, les maladies chroniques telles que les maladies cardiovasculaires, l'hypertension et le diabète, représentent les causes principales de décès d'après l'Organisation Mondiale de la Santé. Il a été estimé que 63% des décès à l'échelle mondiale sont directement ou indirectement en lien avec ces maladies chroniques. Par ailleurs, d'ici 2030, il a été prévu que le coût des soins atteindra environ 75% du PIB actuel. Dans ce contexte, les technologies basées sur les Nœuds de Capteurs Sans-Fil (*WSN* en anglais), réduisent efficacement les coûts engendrés lors des soins des maladies mentionnées ci-dessus, ce qui à terme pousse à la conception d'équipements biomédicaux portables, composés d'un ou de plusieurs appareils interconnectés au sein d'un Réseau de Capteurs Corporels Sans-Fil (*WBSN* en anglais). Ces systèmes disposant de ressources limitées, permettent un enregistrement long durée de signaux biologiques et effectuent sur ces derniers un traitement numérique avancé, afin de délivrer de façon autonome un diagnostic médical, et cela même en dehors d'un environnement hospitalier. L'efficacité énergétique de ces appareils est d'importance capitale, étant donné qu'ils doivent fonctionner sur de longues périodes de temps, avec pour seule source d'énergie une unique batterie. C'est pourquoi, afin de minimiser la consommation énergétique, les parties logicielles et matérielles de ces appareils doivent être développées avec le plus grand soin.

Dans cette thèse, je propose un ensemble de techniques de conception mixtes logiciel/matériel, permettant d'augmenter de façon significative, l'efficacité énergétique de ces appareils de surveillance biomédicale. Afin d'atteindre cet objectif, j'explore conjointement différentes solutions dans le but de réduire l'effort en terme de calcul au niveau du logiciel, tout en optimisant la consommation énergétique de la partie matérielle, en employant une architecture multi-cœurs ultra-basse consommation, qui exploite les caractéristiques des applications logicielles effectuant le traitement numérique des signaux biologiques.

En premier lieu, au niveau des capteurs, j'effectue une étude de l'utilisation d'un classificateur de battement de cœurs, afin d'effectuer un traitement numérique sélectif sur des signaux cardiaques, à l'aide d'appareils de surveillance biomédicale de dernière génération. Pour ce faire, j'ai développé une plateforme me permettant de concevoir et de tester des classificateurs de battement de cœurs, afin d'établir la liste des optimisations nécessaires, dans le but d'exécuter efficacement ces classificateurs sur des systèmes à ressource limitée. Par la suite, au niveau réseau, je propose un *WBSN* plus élaboré, utilisé pour la surveillance de l'activité des patients et qui propose différents compromis entre précision de la classification des battements de cœurs et le volume de données transmises. Dans ce travail, j’étudie l’association d’un ensemble restreint de *WSN* connecté avec un smartphone, et je propose deux modèles de classifications qui comparent l’évolution de la précision en fonction de la quantité de données transmises. L’intervalle de qualité de la classification évolue entre 88% et 97%, tout en permettant d’économiser jusqu’à 86% des transmissions sans fil, dépassant même, les performances des solutions de dernière génération.

En second lieu, je propose une architecture multi-cœurs basse consommation, reposant sur un mécanisme de synchronisation et conçu pour le traitement des signaux biologiques. J’introduis lors de cette recherche, un mécanisme de synchronisation matériel/logiciel, permettant d’atteindre un haut niveau d’efficacité énergétique, tout en parallélisant l’exécution des applications de traitement des signaux biologiques. Ensuite, je généralise la méthodologie afin de supporter différentes applications disposant d’un degré arbitraire de parallélisme au sein de leur exécution. La technique proposée incorpore un synchroniseur à faible encombrement et une extension du jeu d’instructions pour les différents cœurs de traitement. De par les bénéfices prodigués par le mode d’exécution *Single-Instruction Multiple-Data* (SIMD) et l’exécution logiciel en pipeline, l’architecture atteint une réduction jusqu’à 38% de sa consommation énergétique, en comparaison avec un système basse consommation mono-cœur.

Pour finir, je me suis concentré sur l’optimisation de la hiérarchie mémoire du système multicœur, qui consomme une part majoritaire de l’énergie totale utilisée par le système. En premier lieu, j’ai considéré une hiérarchie mémoire hybride, intégrant une petite partition mémoire protégée et pouvant fonctionner avec une très basse tension d’alimentation, et permettant d’atteindre ainsi, 50% d’économie d’énergie. En second lieu, j’explore une hiérarchie mémoire à deux niveaux, basée sur une technologie à cellules Mémoire Non-Volatile (*NVM* en anglais), permettant d’effectuer un “power gating" avec une granularité très fine, soutenue par les technologies émergentes telles que : NVM basse tension et fabrication monolithique tridimensionnel de circuits. Des résultats expérimentaux montrent qu’en adoptant cette hiérarchie mémoire, la consommation énergétique peut être divisée par 5,42 dans l’étage de traitement numérique des signaux.

Mots clefs : Traitement de Signaux Biologiques, Architectures Ultra-basse Consommation, Conception Mixte Matériel/Logiciel, Systèmes de Surveillance Biomédicale, Synchronisation de Systèmes Multi-Cœurs, Plateforme Multi-Cœurs à Haut Rendement Energétique, Traitement Embarqué d’Electrocardiogramme, Classification de Battements de Cœur

Contents

Acknowledgements **i**

Abstract **iii**

Resumé (Français) **v**

Table of Contents **vii**

List of Figures **xi**

List of Tables **xv**

List of Abbreviations **xvii**

1 Introduction **1**

- 1.1 Embedded Bio-Signal Processing . . . 3
  - 1.1.1 Energy-Saving Opportunities in Bio-signal Processing Applications . . . 4
- 1.2 Low Power Architectures for Bio-Signal Processing . . . 5
  - 1.2.1 Low-power Design Limitations . . . 6
- 1.3 Thesis contributions . . . 7
- 1.4 Thesis Outline . . . 10

2 Optimized Embedded Digital Signal Processing for Health Monitoring **13**

- 2.1 introduction . . . 13
  - 2.1.1 Embedded Processing and Limitations . . . 14
  - 2.1.2 Contributions and Outline of this Chapter . . . 15
- 2.2 Existing Advanced Embedded Signal Processing Methods . . . 16
  - 2.2.1 ECG Conditioning . . . 16
    - 2.2.1.1 Spline Filtering (SF) . . . 17
    - 2.2.1.2 Morphological Filtering (MF) . . . 17
  - 2.2.2 ECG Analysis: Delineation . . . 18
    - 2.2.2.1 Wavelet-based Delineation (DWT) . . . 19
    - 2.2.2.2 Multi-scale Morphological Derivative-based Delineation (MMD) 20
  - 2.2.3 Multi-Channel Signal Fusion . . . 21
  - 2.2.4 ECG Compression: Compressed Sensing . . . 22

2.3 Proposed Selective ECG Processing Based on Embedded Heartbeats Classification 23
- 2.3.1 Target Application 23
- 2.3.2 State of the Art and Problem Challenges 24
- 2.3.3 Classification Method 25
- 2.3.4 Dimensionality Reduction Techniques 26
- 2.3.5 Proposed Training Framework 28
- 2.3.6 Resource-Constrained Optimization Phase 30
- 2.3.7 Experimental Results 32
  - 2.3.7.1 Experimental Set-up 32
  - 2.3.7.2 Classification Accuracy 34
  - 2.3.7.3 Run-time Analysis 35
  - 2.3.7.4 Communication Savings 38
- 2.3.8 Comparison of Classification Methods 39

2.4 Proposed Energy-aware Distributed Wireless Body Sensor Network (WBSN) for Physical Activity Monitoring 40
- 2.4.1 State-of-the-art and Challenges 41
- 2.4.2 Target System Definition 43
  - 2.4.2.1 Device Taxonomy 43
  - 2.4.2.2 Network Topology and Task Assignments 45
- 2.4.3 On-node Feature Extraction 46
- 2.4.4 Classification Framework 47
  - 2.4.4.1 On-Node Classification 48
  - 2.4.4.2 Smartphone-Based Classification 48
- 2.4.5 Experimental Evaluation 50
  - 2.4.5.1 Experimental Set-up 50
  - 2.4.5.2 Data Collection Network 51
  - 2.4.5.3 Classification Accuracy 52
  - 2.4.5.4 Transmission Volume Reduction 53

2.5 Summary and Concluding Remarks 55

3 Synchronization-Based Ultra-Low Power Multi-Core Architectures 57

3.1 Introduction 57
- 3.1.1 Parallelism in Bio-signal Processing Applications 57
- 3.1.2 Contributions and Outline of this Chapter 58

3.2 Low-Power Single- and Multi-Core Architectures for Bio-Signal Processing: State of the Art 59
- 3.2.1 Low-Power Single-Core Architectures for Bio-Signal Analysis 59
  - 3.2.1.1 Firat: A Low Power PIC-compatible ISA 61
  - 3.2.1.2 TamaRISC: a custom low-power ISA for Bio-Signal Processing . 63
- 3.2.2 Low-Power Multi-Core Architecture for Bio-Signal Processing 64
  - 3.2.2.1 Low-Power Multi-core MIMD Parallel Architecture 65

3.3 Proposed Synchronization-Based Multi-Core Architecture for Parallel Bio-Signal Processing Applications 67
3.3.1 Target Multi-Core Processing Architecture 68
3.3.2 Proposed Synchronization Technique 69
3.3.2.1 Software Support: Dedicated Synchronization Instructions . . . 71
3.3.2.2 Hardware Support: Synchronizer 72
3.3.2.3 Synchronization Instructions Insertion 73
3.3.3 Experimental Results 73
3.3.3.1 Experimental Setup 73
3.3.3.2 Performance Results 74
3.3.3.3 Area Footprint 75
3.3.3.4 Power Consumption 75
3.4 Proposed Advanced Synchronization Technique for Arbitrarily Parallel Bio-Signal Processing Applications 77
3.4.1 Target Multi-Core Processing Architecture 78
3.4.2 Synchronization Methodology 79
3.4.2.1 Hardware/Software additional support 79
3.4.2.2 Synchronization Mechanism 80
3.4.2.3 Application Mapping 82
3.4.3 Experimental Set-up 84
3.4.3.1 Evaluated Multi-core and Baseline Single-core Systems 84
3.4.3.2 Simulation Framework 84
3.4.3.3 Benchmark Applications 85
3.4.4 Experimental results 86
3.4.4.1 Performance and Memory Footprint Comparison 87
3.4.4.2 Single- and- Multi-core Energy Consumption 88
3.4.4.3 Synergies Between VFS and Broadcasting 89
3.5 Summary and Concluding Remarks 90

4 Optimized Memory Subsystems for Ultra-Low Power Multi-Core Architectures 91
4.1 Introduction 91
4.1.1 Contributions and Outline of this Chapter 92
4.2 Proposed ULP Multi-Core Architecture Featuring SCM-based Hybrid Memory . 94
4.2.1 Related Work 95
4.2.2 Target Multi-Core Processing Architecture 96
4.2.3 Hybrid Memory Management 97
4.2.3.1 Synchronization Strategy 98
4.2.3.2 Software Adaptations and Mapping 99
4.2.3.3 Hardware Support: Synchronization Unit 99
4.2.4 Experimental Set-up 101
4.2.4.1 Simulation Set-up 101
4.2.4.2 Bio-signal Processing Benchmarks 102

4.2.5 Experimental Results . . . 104
4.2.5.1 Reliable Memory Requirements . . . 104
4.2.5.2 Runtime Performance . . . 105
4.2.5.3 Power Consumption Evaluation . . . 105
4.3 Proposed Nano-engineered ULP Multi-Core Architecture featuring a 2-level NVM-based memory subsystem . . . 108
4.3.1 Technology Background . . . 109
4.3.1.1 Non-Volatile Memories . . . 109
4.3.1.2 3D Integration . . . 111
4.3.2 Proposed Nano-Engineered Architecture . . . 112
4.3.2.1 System Architecture . . . 112
4.3.2.2 Memory Management Approach . . . 114
4.3.2.3 Deep-sleep Sensing . . . 114
4.3.3 Experimental Framework . . . 115
4.3.3.1 Circuit-level Characterization . . . 115
4.3.3.2 Bio-signal Processing Benchmarks . . . 116
4.3.3.3 Explored Nano-engineered Architectures . . . 116
4.3.4 Experimental Evaluation . . . 117
4.3.4.1 Page Buffer Sizing . . . 117
4.3.4.2 NVM Optimization . . . 118
4.3.4.3 Run-time Performance . . . 119
4.3.4.4 Area Footprint . . . 120
4.3.4.5 Power Consumption . . . 121
4.4 Summary and Concluding Remarks . . . 121

5 Conclusions and Future Work 125
5.1 Summary and Contributions . . . 125
5.2 Future Work . . . 128

Bibliography 145

Curriculum Vitae 147

List of Figures

1.1 Proportion of global deaths under the age of 70 years (extracted from [5]) 2
1.2 biomedical monitor example (extracted from [7]) 3
1.3 Bio-signal processing application example 5

2.1 Acquired signal (top) and its baseline estimations using morphological filtering (middle) and spline interpolation (bottom). 17
2.2 ECG heartbeat showing the main characteristic waves and some of the clinically relevant periods of interest 18
2.3 DWT decomposition of an ECG heart beat. Maximum-minimun intervals examined to find the R peak are highlighted in gray. 19
2.4 Acquired signal (top) and its MMD-transformed version (bottom). 21
2.5 Three-leads acquisition and corresponding RMS-combined signal. 21
2.6 Relative improvement in sensitivity of two- and three-leads delineation with respect to single-lead delineation. Black = 2 leads, White = 3 leads. 22
2.7 Target system featuring a classification block that selectively activates a detailed digital signal processing chain. 24
2.8 Multi-layer neuro-fuzzy classifier. 25
2.9 Delineated normal heartbeat [65]. 28
2.10 Classification framework: *Top (PC-side):* NFC training using Random Projections (**a**), Principal Component Analysis (**b**) and Fiducial Points Detection (**c**). Optimization for WSNs (**d**). *Boottom (WSN side):* Real time execution of the optimized implementation (**e**). 29
2.11 Linear approximation of gaussian MFs in the range $[-4.7\sigma, 0]$, compared to a gaussian curve. 31
2.12 Experimental scenario: classification is used to activate an accurate multi-lead morphological delineation only in case of heartbeat abnormality. 32
2.13 Pareto curves of the NDR/ARR relations for various classification methods obtained by averaging the different values extracted from the 4-fold cross validation process. 36
2.14 Overview of the proposed WBSN. The three nodes in yellow are the ones belonging to the final configuration (see Section 2.4.5.3), whereas the remaining nodes have only been used for data collection purposes 44

2.15 The two proposed classification strategies: smartphone-based (a), and on-node (b). In both scenarios, the phone acts as network coordinator. 46
2.16 Structure of the proposed tree for on-node classification 49
2.17 Structure of the proposed neuro-fuzzy classifier 50
2.18 Classification accuracies of the proposed NFC for smartphone-based classification, with multiple combinations of node placements 53

3.1 Firat core architecture [107]. 62
3.2 TamaRISC core architecture [107]. 64
3.3 Low-power multi-core MIMD parallel architecture presented in [84]. 66
3.4 Improved Multi-Core Architecture with Hardware Synchronizer 68
3.5 An Example of Data-Dependent Code Section 70
3.6 Example of dedicated synchronization point stored in data memory 71
3.7 Total Power Consumption of the multi-core architectures while running the different employed benchmarks 77
3.8 Block scheme of a smart WBSN platform. 78
3.9 Hardware architecture of a multi-core WBSN. In red, HW support for the proposed synchronization technique 79
3.10 Examples of synchronization points values. a) cores 0, 1 and 2 should jointly produce data for core 4; data is not yet available. b) cores 0, 1 and 2 have entered a data-dependent branch, core 0 has finished executing it. 80
3.11 Execution flow of a producer-consumer relationship employing the proposed synchronization technique 81
3.12 Partitioning and mapping of the application in Figure 3.8 on a multi-core platform embedding 4 computer units, 4 IM banks and 4 DM banks. 83
3.13 Block schemes of benchmark applications: a) 3L-MF and 3L-MMD; b) RPCLASS Conditionally activated blocks are indicated with grey background. 86
3.14 Power consumption decomposition of the single-core (SC) and multi-core (MC) systems with and without the proposed synchronization approach. 88
3.15 Power consumption (left axis in μW) of the single-core (SC) and multi-core (MC) systems and the respective reduction (right axis, in percentage) when employing the proposed approach in the multi-core platform 89

4.1 Energy consumption breakdown of a biomedical monitor, scheme of a typical bio-signal processing application and generic block diagram of a state-of-the-art ULP multi-core architecture. 92
4.2 Target architecture, featuring hybrid DM banks and HW synchronization unit (SU). 97
4.3 Active/inactive architectural elements in *sensing* and *processing* states with related hybrid memory behavior. 98
4.4 Synchronization in a diverging data-dependent branch (a) and in a producer-consumer relationship processing windows of 2 samples (b). 100

4.5 Flowchart describing the synchronizer's behavior: (a) Clocked logic governing transitions between platform modes. (b) Combinational part orchestrating execution while in processing mode. 101
4.6 Block schemes of benchmark applications: a) 8L-CS; b) 3L-MF and 3L-MMD; c) RPCLASS. Conditionally activated blocks are indicated with grey background. . 103
4.7 Power consumption and area overhead varying the SCM-DM size. The minima are highlighted by a rounded marker. 104
4.8 Power consumption of the target and baseline systems (*no Hybrid* and *no Sync*) for the considered bio-signal processing benchmarks. 107
4.9 Power consumption breakdown on a WBSN-based biomedical monitor executing a multi-channel biosignal processing application. Values are computed based on [124], [139] and [140] 108
4.10 Activity of different contributors of a typical biomedical monitor while performing bio-signal DSP 108
4.11 Block diagrams of the a) multi-core architecture introduced in Chapter 3, featuring volatile SRAM and b) the target NVM-based platform 113
4.12 Average power consumption of 3D_TARGET employing different instruction page buffer (I-PB) and data page buffer (D-PB) sizes (in number of words). Local minima are marked with a red dot 118
4.13 Average page write frequency for the four studied benchmarks 119
4.14 Area breakdown for the studied nano-engineered architectures 121
4.15 Power consumption of the studied architectures. The breakdown on the right half of the figure is in μW 122
4.16 Power consumption breakdown of all considered applications when executed on 3D_TARGET 123

List of Tables

2.1 Composition of the sets of heartbeats employed in the different experiments. . 34
2.2 Average Normal Discard Rate (NDR) in the 4-fold cross validation process for a fixed minimum Abnormal Recognition Rate (ARR) of 95%. 35
2.3 Duty cycle (%) of the sub-systems identified in Figure 2.12. Tests performed on the IcyFlex WSN running at 6 MHz. 37
2.4 Memory footprint (kB) of the sub-systems identified in Figure 2.12. Tests performed on the IcyFlex WSN running at 6 MHz. 37
2.5 Classification performance and number of parameters for different classifiers. 39
2.6 Confusion matrix of the proposed Neuro-Fuzzy Classifier (The values are row-normalized) 54

3.1 Average number of cycles required to process a sample for the different studied benchmarks 75
3.2 Dynamic power distribution of the architectures while running reference benchmarks at 8 MOps/s and 1.2 V 76
3.3 Platform configurations of the single-core (SC) system and the multi-core (MC) one for the different studied benchmarks 87
3.4 Obtained results for the different benchmarks while executing on the multi-core platform featuring the proposed synchronization technique. 87

4.1 Relevant runtime characteristics of the bio-signal processing benchmarks . . . 106
4.2 Leakage and dynamic power of the target platform for the bio-signal processing benchmarks during *sensing* and *processing* phases. 106
4.3 Comparison between different memory technologies. Latency and energy values are observed from literature 110
4.4 Parameters of various key components of the studied nano-engineered architectures 116
4.5 Runtime metrics of the analyzed benchmarks using 8-word program and data page buffers 120

List of Abbreviations

ADC	Analog-to-digital converter
ALU	Arithmetic logic unit
ARR	Abnormal recognition rate
CS	Compressed sensing
COTS	Commercial off-the-self
CPI	Cycles per instruction
D-PB	Data page buffer
D-Xbar	Data crossbar
DM	Data memory
DSP	Digital signal processing
DWT	Digital wavelet transform
ECG	Electrocardiogram
FPD	Fiducial point detection
I-PB	Instruction page buffer
I-Xbar	Instruction crossbar
IM	Instruction memory
ISA	Instruction set architecture
ISE	Instruction set extension

MF Morphological filtering

MIMD Multiple-instruction multiple-data

MMD Multi-scale morphological delineation

MMU Memory management unit

NDR Normal discard rate

NFC Neuro-fuzzy classifier

NVM Non-volatile Memory

PCA Principal component analysis

RMS Root mean square

RP Random projection

RRAM Random-access memory

RTL Register transfer level

SIMD Single-instruction multiple-data

SCM Standard cell memory

SRAM Static random access memory

STTRAM Spin-transfer torque random-access memory

SVM Support vector machine

TSV Through-silicon vias

ULP Ultra-low power

VHDL Very high speed integrated circuit hardware description language

WBSN Wireless body sensor network

WSN Wireless sensor node

1 Introduction

It is estimated that during the past decade, non-communicable diseases (NCDs) have been the responsible of more than 300 million deaths worldwide [1]. NCDs are chronic ailments of long duration and slow progression that are not passed among persons. Most popular NCDs are cardiovascular diseases (like heart attacks and stroke), cancer, chronic respiratory diseases (such as chronic obstructed pulmonary disease and asthma) and diabetes. In general, the proliferation of these diseases is driven by the globalization of unhealthy lifestyles that include habits such as tobacco use, alcohol intake, lack of physical activity and unbalanced diets. In fact, the health impact of such habits can induce rise of blood pressure, increase of blood glucose and lipids levels, and ultimately obesity. All these are known to be intermediate risk factors that can lead to cardiovascular conditions [2].

NCDs are the major cause of global deaths among young to mid-age population. As depicted by Figure 1.1, more than 50% deaths worldwide are directly related to NCDs and the 37% of this fatalities are related to cardiovascular diseases (CVDs). In addition, the world population is growing exponentially and its age composition is changing. According to the United Nations [3], in the period from 2000 to 2015, the amount of people aged 60 years or more increased by 48% up to 901 million, and among those, the group of subjects aged 80 or over grew by 77%. As a result, the incidence of chronic age-related diseases, such as most NCDs, is also expected to suffer a proportional increase. Moreover, it is estimated that already today, CVDs represent the major cause of mortality accounting for one third of worldwide deaths as of 2012 [4], [5].

Nowadays, healthcare delivery is performed on demand and in hospital environments where diseases are treated after symptom-based diagnosis. In order to do a follow-up of these age-related and lifestyle-induced NCDs, continuous supervision becomes mandatory. Given the aforementioned estimations, an unsustainable amount of medical resources (staff, equipment and facilities) will be required, leading to unaffordable healthcare costs [6]. Therefore, there is an urgent need for a paradigm shift towards a more prevention-oriented healthcare delivery model in which early diagnosis and personalized monitoring is prioritized in order to minimize the associated costs.

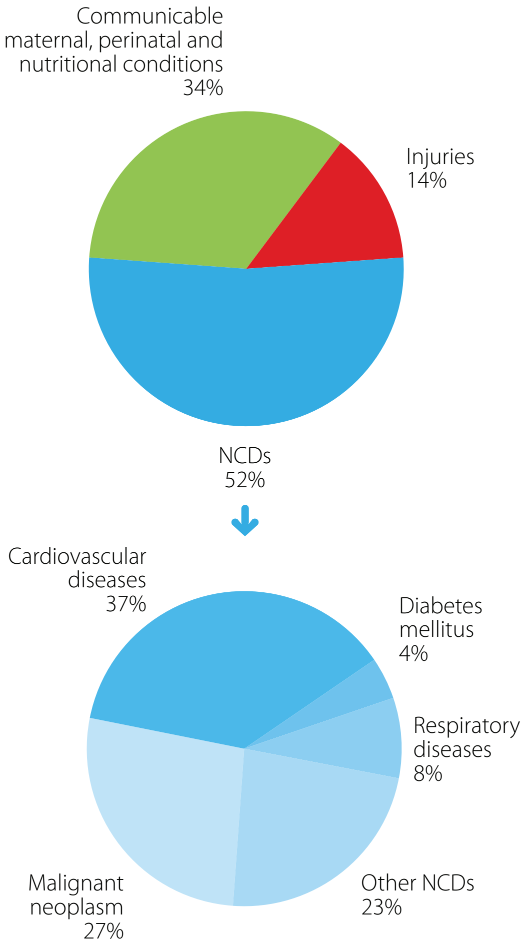


Figure 1.1 – Proportion of global deaths under the age of 70 years (extracted from [5])

In this context, technology breakthroughs in embedded system design and circuit integration have enabled the emergence of Wireless Sensor Nodes (WSN), which are autonomous devices able to sample, process and transmit different types of data. The application of this technology in the field of healthcare and personal monitoring has facilitated the conception of wearable biomedical monitors. These miniaturized battery-powered devices can acquire biological signals (e.g. body temperature, perspiration) behavioral information (e.g. movements) and environmental data (e.g. light, temperature). Several of these WSNs can be deployed thorough the body creating a network, known as Wireless Body Sensor Network (WBSN).

A WBSN-based biomedical monitor as the one depicted in Figure 1.2 is typically composed of one or several nodes that communicate through a wireless channel. Each of the nodes is normally responsible for acquiring a specific physiological signal and performing the necessary manipulations before transmission. In the example of the figure, different nodes acquire several signals, such as electrocardiogram (ECG), peripheral oxygen saturation (SpO_2) or accelerometer data. The signals are then transmitted to a central coordinator that can make

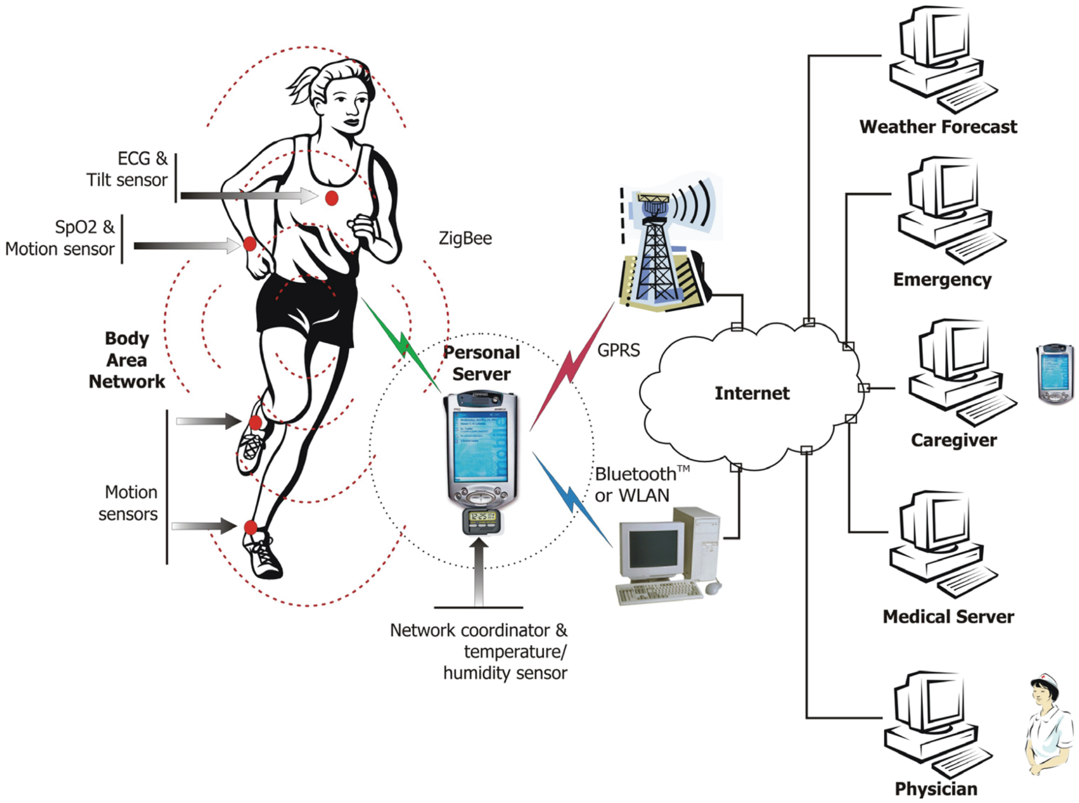


Figure 1.2 – biomedical monitor example (extracted from [7])

further processing or directly transmit them through a gateway possibly connected to the Internet (a PDA in Figure 1.2).

In order to minimize the subject's discomfort while wearing a WBSN-based biomedical monitor for a prolonged time, the amount and dimensions of the nodes must be minimized. This requirement restricts the size and capacity of the batteries used by the WSNs, thus forcing them to limit their computational effort and the volume of data they transmit [8]. Therefore, optimizing the energy efficiency of biomedical monitors is of paramount importance.

In the remaining of this chapter first I briefly discuss about the embedded bio-signal processing performed by biomedical monitors and the potential opportunities for energy savings. Then I describe some of the state-of-the-art approaches to build low-power processing platforms and their main limitations. Afterwards, I detail the contributions of this thesis. Finally, in the last section, I provide a short outline of the document.

1.1 Embedded Bio-Signal Processing

A common approach to reduce the power consumption of WBSN-based biomedical monitors is to perform a preliminary extraction of a set of key parameters (also known as *features*) from the acquired bio-signals [9], which are relevant for the target application. For instance, cardiac monitors extract and report periodically the average heart rhythm instead of the full electrocardiogram signal when they are employed to asses performance while doing sports. These computations are performed by the onboard micro-controller which normally incorporates an embedded processor that typically provides very limited computational power. By transmitting only the computed relevant data, the utilization of the radio link is drastically decreased and therefore the platform power consumption is reduced [10].

Bio-signal processing applications usually implement algorithms to condition acquired signals, extract important parameters or *features* and interpret them to perform an early analysis to detect a target diseases. Even though the extraction of these features greatly increases the battery life of these systems due to the reduction of information to be sent, the performed on-node DSP becomes one of the main contributors to the power consumption of the system [11] requiring a non-negligible amount of computing power. Moreover, thank to the advances in low power sensing technologies and the optimization of transmission schemes in ultra-low power protocols, the DSP stage can even become the dominant part of the overall system consumption.

As a consequence, when the on-node extracted features are not of clinical relevance, the performed computations can be considered as unnecessary and thus, inefficient from a power-consumption viewpoint. For instance, in the case of ECG monitors, the extraction of features such as the heartbeat rate or the duration and exact amplitude of the ECG characteristic waves may only be required during pathological episodes such as an arrhythmia or when there is certain abnormality in the morphology of the signal. In the case of a network of nodes, the energy cost of performing unnecessary processing is even higher since each node is sensing and manipulating its corresponding signals and sending the results through the WBSN.

As a conclusion energy- and transmission-aware embedded software design is of vital importance to avoid situations in which power consumption is unnecessarily increased. Furthermore, the battery life of the biomedical monitors will be greatly influenced by the efficiency of the featured processing hardware. For this reason, *hardware/software co-design* is of vital importance. On the one hand, application- or domain-specific characteristics can be leveraged at the hardware level to obtain high energy efficiency. On the other hand, specific processing hardware capabilities can be exploited at the software level to optimize power consumption.

1.1.1 Energy-Saving Opportunities in Bio-signal Processing Applications

Bio-signal processing applications follow in many cases a feedforward structure in which numerous algorithmic steps are applied sequentially over the input signals. These steps can be moderately complex and can be applied over different streams of data (e.g. filtering of a multi-input signal as in Figure 1.3). As proved by the authors of [10] and [12], these advanced DSP routines can be adapted to be executed in resource-constrained embedded platforms without sacrificing accuracy. Moreover, recent low-power multi-core architectures can benefit from

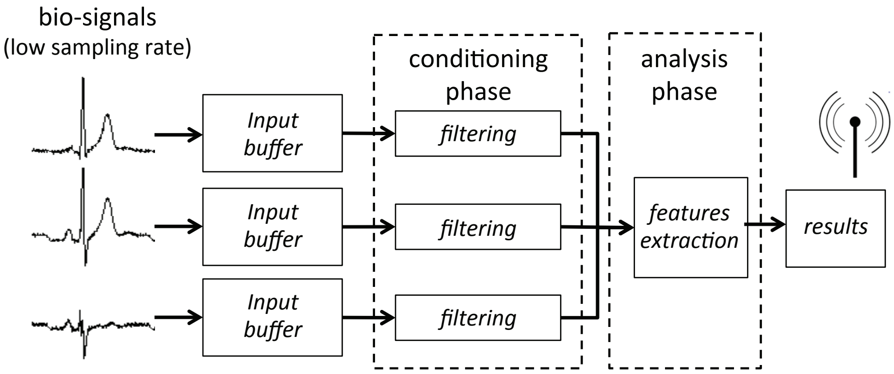


Figure 1.3 – Bio-signal processing application example

the intrinsic parallelism in bio-signal processing applications, allowing for more aggressive voltage scaling than a single-core alternative.

Figure 1.3 represents the application structure of a simple application running on a ECG biomedical monitor. The multi-channel application receives three input signals. It first cleans each of the signals by applying a filter and then analyzes all of them together to extract relevant features. Two levels of parallelism that can be identified in Figure 1.3 are inherent to bio-signal processing applications:

1. As previously introduced, bio-signal processing applications can be naturally partitioned into a set of sequential algorithmic steps. Each of this phases can be executed on a different processing core in a software pipeline. In the example of Figure 1.3, the two phases are *conditioning* and *analysis*. As a result, the workload is spread among different cores.
2. At the same time, each of the phases can work over different datasets. This scheme can be easily parallelized by performing the manipulations of each of the channels concurrently in a different core. Again, the workload is divided among different cores.

Multicore architectures, as shown in [13] and [14] are good candidates for this strategy, because they can distribute the workload over different computing elements, each of them operating at a low frequency in a near-threshold regime. In such a scenario, aggressive voltage scaling leading to considerable energy savings can be applied.

1.2 Low Power Architectures for Bio-Signal Processing

The applicability of WBSN-based biomedical monitors has been investigated in a variety of scenarios [15], ranging from automated analysis of ECGs [10], to respiration rate estimation [16] to the detection of epileptic seizures [17]. Recently, dedicated architectures have been proposed to support these workloads at ultra-low-power levels. Toward this goal, the authors of [18] and [19] advocate the use of custom accelerators (such as FFT and Cordic engines) to efficiently support commonly-used routines. This approach is of limited flexibility, as it assumes the knowledge at design time of the computationally-intensive segments of applications.

A different strategy, illustrated in [20], is instead to aggressively scale the supply voltage to decrease both static and dynamic power. Voltage scaling has been extensively analyzed in the literature, including its limitations and disadvantages [21] [22] [23]. One of the main issues with low-voltage operation is performance degradation, which can limit the degree of achievable voltage scaling for the given processing requirements, as explained later.

1.2.1 Low-power Design Limitations

While reducing the system clock constraint thanks to the parallelization of the applications allows to relax the voltage supply, traditional SRAM memories pose a lower bound on the operating voltage of these platforms, dictated by the minimum level at which data can be reliably accessed [24]. Due to their construction, SRAM cells can suffer from different types of errors if they are not supplied with a minimum voltage level.

Following a different approach, the use of non-volatile memories (NVMs) as main memories, do not present this limitation, hence allowing the complete gating of the power supply of computation and memory elements without losing the stored data. Today's WBSN-based biomedical monitor typically use FLASH memory for non-volatile storage. FLASH memory stores the program and data memory contents when the system is switched off so that they can be restored later. At boot-up, its contents are transferred to the on-chip SRAM and the execution can start/resume normally.

In general, bio-signals are acquired at rather low sampling frequencies (in the order of few hundreds of hertzs) and the workload profile of the embedded digital signal processing stage is dictated by the availability of enough data to process (i.e. a window of samples). As a consequence, the required computational effort follows a cyclic trend combining short periods of intense work (bursts of computations) with intervals of low activity (data buffering). Although power-gating the platform during sensing periods would represent a very interesting saving technique, data needs to be collected and stored periodically at the pace dictated by the sampling frequency.

Power-gating at such a fine grain is not possible with state-of-the-art FLASH memories for two reasons. First, strict real-time deadlines for this application domain can no longer be met due to very long write latencies (the time required to write a word into FLASH, $\approx 120\,\mu$s for small arrays [25], [26]); i.e., the time needed to store the system state would exceed the inter-sample time. Second, the energy cost of shadowing the full data memory several hundreds of times per second can exceed the potential savings obtained from the power-gating.

1.3 Thesis contributions

The main goal of this thesis is to develop a set of hardware/software co-design techniques to improve the energy efficiency of biomedical monitors. To do so, I explore different alternatives to reduce the required computational effort of embedded software while optimizing the underlying processing hardware by employing an enhanced low-power multi-core architecture that exploits the characteristics of bio-signal processing applications. In particular, the contributions of my work can be grouped as follows:

Optimized Embedded Digital Signal Processing

In this field, I focus on reducing the computational requirements of state-of-the-art embedded processing applications by applying two different strategies. First, at the sensor node level, I propose to selectively activate advanced and computationally intensive DSP routines only in case of necessity (e.g. in case of detecting an abnormal or potentially problematic situation). Second, at a sensor network level, I propose a transmission- and energy-aware wireless body sensor network (WBSN) for the energy-efficient monitoring of physical activity. More specifically, the detailed optimizations are:

1. **Selective advanced DSP in ECG biomedical monitors**: In the context of ECG biomedical monitors, I investigate the utilization of a heartbeat classifier that analyzes the signal morphology to detect abnormal situations. In those cases, the typical advanced multi-channel analysis application is activated in order to extract the features of interest. More precisely, the contributions of the proposed work are:
    - An abnormality detector based on a lightweight neuro-fuzzy classifier (NFC) to perform embedded real-time heartbeat classification.
    - A novel dimensionality reduction method based on *Random Projections* (RP) to reduce the classification complexity.
    - An automated two-step training framework, which trains the NFC and concurrently searches for a performant RP matrix. The process is guided by a genetic algorithm that evaluates the performance of the RP-NFC pair.
    - A set of optimizations to reduce the computational complexity and memory footprint of the RP-based NFC. The proposed manipulations are platform-independent and aim at adapting the code to be executed in resource-constrained platforms.
2. **Transmission-aware WBSN for physical activity monitoring**: The system consists of a reduced set of kinetic sensors deployed throughout the subject's body that cooperate with a smartphone to periodically identify the activity that the subject is performing (e.g. walking, sleeping, running). The detailed contributions of this work are:
    - A high precision NFC-based classification scheme that leverages the higher computational resources available in the smartphone.

- An on-node classification scheme that reduces transmission volume trading accuracy using a simpler decision tree, which can be implemented on the target sensors.
- A smart feature extraction strategy for the on-node variant, which only computes and transmits the necessary features within the WBSN instead of the full set required by the NFC option.
- An exhaustive study of the best placement and the optimal number of nodes to obtain the most accurate classification output.

Low-power multi-core architecture for bio-signal analysis

I propose a synchronization-based ultra-low power (ULP) parallel architecture devoted to the execution of bio-signal processing applications. The workload division among cores allows to relax the system clock constraints enabling the possibility to apply voltage scaling and maximize energy savings. First, I develop a synchronization technique that enhances low-power multi-core architectures enabling the efficient execution of massively parallel applications. Then, a generalization of the synchronization technique allows to efficiently run any existing bio-signal application into the proposed ULP multi-core platform. In particular, the details of the proposed architectures and techniques are the following:

3. **Synchronization technique for multi-channel parallel DSP applications**: An ultra-low power architecture able to recover synchronization after data-dependent branches when executing multi-channel bio-signal analysis applications is presented. In particular, the contributions are:
    - A state-of-the-art parallel architecture featuring a set of low-power processing cores interfaced to multi-banked instruction and data memories, which supports aggressive voltage scaling.
    - A synchronization strategy to counteract the two major events leading to cores de-synchronization, namely, data memory access conflicts and data-dependent conditional execution of code.
    - A description of the required hardware/software support including a dedicated lightweight synchronizer and an instruction set extension (ISE) of the processing cores.
    - An experimental evaluation of the benefits of synchronization when executing multi-input DSP applications.
4. **Generalized synchronization technique for bio-signal processing applications**: The synchronization technique is enhanced to additionally support core-to-core notifications. This improvement allows to execute any kind of bio-signal processing application in the ultra-low power platform regardless of its degree of parallelism. The detailed contributions of this work are:

- A modification of the synchronization technique to support producer-consumer relationships among cores by implementing an efficient core-to-core notification mechanism. This approach allows to exploit the benefits of software pipelining by executing different algorithmic steps in parallel employing different cores in a pipeline manner.
- The necessary hardware modifications at the synchronizer level and the extension of the cores ISE to support the new feature.
- A detailed description of the required steps to adapt any existing bio-signal application analysis.

Energy-efficient memory subsystems for ULP multi-core architectures

In this context, I proposed different alternatives to decrease the overall power consumption of the low-power multi-core platforms by optimizing the memory subsystem, which in turn is one of the major contributors at the system level. In particular, I study two different approaches that reduce static power (*leakage*) by reducing the power consumption while the platform is idle (i.e. not processing). First I explore a hybrid memory subsystem based on data memory banks featuring a reliable small partition implemented as standard cell memory (SCM). Second, I completely re-design the memory subsystem to include a non-volatile memory (NVM) partition as new main memory unit. More precisely, I propose two ULP multi-core architectures, each of them featuring one of the following alternatives:

5. **Hybrid SCM-based memory subsystem**: Aggressive voltage scaling below certain levels has shown to be an unsuitable strategy for on-chip SRAM memories. I propose a hybrid memory bank arrangement that includes a minimal reliable partition implemented with SCM. The voltage of these banks can be safely reduced if only the reliable partition is accessed at low-voltage regimes. Following this idea, the multi-core platform is extended to support a *sensing* mode during which data is buffered in the SCM while the platform remains clock-gated. More specifically, the contributions of this work are:
    - A new memory subsystem based on hybrid memory banks featuring a small reliable memory partition able to operate at ultra-low voltage. This new scheme tolerates an aggressive reduction of the supply voltage without compromising the data integrity as long as the non-reliable big partition is not accessed at this level.
    - A new power management strategy that seamlessly transits between *processing* and *sensing* modes without requiring any modification at the application level. To implement this mechanism, a description of the required hardware modifications at the synchronizer level is also provided.
    - A study of the optimal size of the reliable partitions that provides the best tradeoff in terms of power consumption and system area overhead.
6. **NVM-based two-level memory hierarchy**: In this case I propose a completely re-designed

memory subsystem that allows for fine-grained power-gating of the platform when all the processors are idle. To that end, the new memory subsystem employs a low-voltage NVM and a set of volatile small instruction and data page buffers that collectively act as a cache. This work presents a promising solution for next generation ultra-low power architectures for biomedical monitors. The detailed contributions are:

- A fully re-design two-level memory subsystem including a non-volatile main level based on emerging low-voltage NVM technologies such as *Spin-transfer torque RAM* (STTRAM), and a cache-like volatile level implemented as a set of small page buffers.
- A study of the ultra-low power multi-core system integration employing new fabrication processes enabled by monolithic 3D integration.
- A new power management that allows for fast power-gating and recovery of the full platform over short but recurrent idle periods happening between the sampling of consecutive windows of samples.
- Description of the lightweight Memory Management Unit (MMU) that interfaces the NVM unit with the volatile level. This unit cooperates with the hardware synchronizer to properly orchestrate the cores execution.

1.4 Thesis Outline

The remainder of this thesis follows the same structure than the one detailed in the previous section. Each chapter will provide the necessary background and a separate review of the related works. In particular, the content is organized as follows:

Chapter 2 presents the software optimizations that I proposed in this thesis to improve the energy efficiency of biomedical monitors. First, at the WSN level, I introduced the selective processing approach based on the RP-based NFC, which performs on-node heartbeats classification to decide when to perform advanced signal processing. Then, at the network level, I discuss a WBSN for physical activity monitoring, which can be configured with two classification schemes: a highly accurate smartphone-centric classification and a transmission- and energy-aware node-centric alternative.

Chapter 3 describes the synchronization-based ultra-low power multi-core architecture optimized to execute bio-signal processing applications. In the first part of the chapter, I detail the target multi-core system and the proposed synchronization technique to efficiently execute multi-channel bio-signal processing applications. In the second part, I present the generalized technique that allows the mapping of any application with an arbitrarily high degree of parallelism.

Chapter 4 details two approaches that I followed to reduce the overall consumption of ultra-low power multi-core platforms by optimizing the memory subsystem. Firstly, I investigate the utilization of hybrid memory banks provided with a small region of reliable standard cell memory, which is able to operate at ultra-low voltage levels without loosing data integrity. This region is used to buffer data over prolonged periods of time during which the processing platform remains clock-gated and the supply voltage is aggressively reduced. Secondly, I propose a radically different memory subsystem based on a new two-level hierarchy enabled by recent emerging technologies such as 3D monolithic integration and low-voltage on-chip non-volatile memory (NVM). This novel memory arrangement, which incorporates an NVM as the main storage unit, allows to perform fine-grained power-gating drastically improving the system energy efficiency.

Chapter 5 concludes the thesis by summarizing the key contributions and providing pointers for future research in the same direction.

2 Optimized Embedded Digital Signal Processing for Health Monitoring

2.1 introduction

Ongoing changes in world demographics and the prevalence of unhealthy lifestyles are imposing a paradigm shift in the healthcare landscape. Nowadays, chronic ailments such as, cardiovascular diseases, hypertension and diabetes, represent the most common causes of death [4]. These non-communicable diseases (NCD) are today involved in 63% of all deaths worldwide, and are predicted to account for 75% of the current GDP by 2030. Continuous monitoring, needed for the supervision of patients affected by a NCDs, strains the resources of healthcare systems. Technologies based on Wireless Sensor Nodes (WSNs) effectively alleviate this burden, allowing for long term and autonomous recording of biological signals, even outside a hospital environment.

WSNs are miniaturized, wearable embedded devices able to acquire and wirelessly transmit biological bio-signals. The sensing hardware equipped in these devices enables them to sample signals of different nature such as bio-potentials (e.g.: electromyogram, electroencephalogram), body kinetics (e.g.: accelerometer or gyroscopic data) or environmental parameters (e.g.: light, temperature, noise). Body sensor nodes allow long term monitoring of subjects, while producing little discomfort and requiring minimal medical supervision. Several of these devices can be used concurrently to work in a distributed manner recording signals within a low-range *body area network*, known as a Wireless Body Sensor Network (WBSN).

Over the last years, WBSNs have emerged as a leading technology that is poised to drastically change healthcare delivery and the everyday life of subjects. Thanks to a combination of wearable low-power WSNs that communicate through a wireless channel, these networks enable the continuous and unobtrusive monitoring of physiological signals and activities, both for personal and medical purposes. The functionalities and the ease of use of these systems have also received a significant boost thanks to the spread of handheld devices (such as smartphones) [27], which represent the ideal high-performance complement to wearable nodes. In fact, smartphones can provide advanced features such as data logging, transmission to a remote location, and user interface, without affecting the nomadic nature of WBSNs.

A major field of application of WSNs is the ambulatory acquisition of electrocardiograms (ECGs), which represent the electrical activity of the heart. ECGs are the primary instruments for monitoring the heart activity and for early detection of heart pathologies. A breakthrough in the practice of ECGs recording and analysis has been possible thanks to the emergence of smart WSNs [28][29] able to autonomously interpret the ECG data [30][31] by performing on-node digital signal processing (DSP).

While signals like ECG can be recorded using a single wearable device, some other applications require a multi-parametric approach in which several nodes are employed within a more complex WBSN. In particular, in the context of personal monitoring, physical activity recognition attracts a high interest from researchers [9, 32]. Activity monitoring finds application mainly in the healthcare domain, such as in the supervision of patients affected by Parkinson's disease [33], but it is also employed in sports and home monitoring [9]. The activity of a subject can be determined from kinetic data, such as acceleration [34] and orientation, collected by a set of nodes located on parts of the body that convey most of the information, such as limbs and joints.

2.1.1 Embedded Processing and Limitations

In order to minimize the subject's discomfort while wearing a WBSN for a prolonged time, a minimum number of nodes has to be deployed, and their size has to be miniaturized. The latter requirement poses significant limitations on the size of the batteries used by the WSNs, thus forcing them to limit their computational effort and the volume of data they transmit [8].

A common strategy to increase the lifetime of a WBSN [9] is the preliminary extraction of a set of parameters from the sensed data. This operation, performed directly on the wearable node, massively reduces the amount of transmitted data. By transmitting only this relevant information instead of the full raw signals, energy efficiency can be considerably increased by minimizing communication on the power-hungry wireless link as proved by [10].

Smart WSNs applications usually implement algorithms to filter acquired signals, extract important parameters or *features* and interpret them to perform an early analysis to detect a target health condition. Even though the extraction of these features greatly increases the battery life of these systems due to reduction of information to be sent, the performed on-node DSP becomes one of the main contributors to the power consumption of the system [11] requiring a non-negligible amount of computing power.

In this context, the features continuously extracted on-node may not be of clinical relevance for the full monitoring period but only in the cases of abnormal episodes. For instance, cardiac parameters such as heart rate or the duration of the ECG characteristic waves, which are computed onboard in state-of-the-art ECG monitors, may only be necessary during arrhythmia episodes or when there is certain abnormality in the morphology of the signal. As a result, the computation of these features in the absence of abnormalities is unnecessary and therefore inefficient from a power-consumption point of view. This wasting effect is more than multiplicative in the case of a WBSN, where each of the sensing nodes sample and process its corresponding signals transmitting all the computed features through the network up to the gateway.

2.1.2 Contributions and Outline of this Chapter

In this chapter I propose two complementary strategies to improve the energy efficiency of biomedical monitors addressing the aforementioned issues. First, at the WSN level, I propose a new and more complex application scheme for cardiac monitors that performs advance DSP only in the case of detecting abnormalities in the ECG signal. In order to do so, I implemented a lightweight heartbeat classifier that can discern in real time between normal and abnormal heartbeats triggering the costly feature extraction DSP routines only in the latter case. Second, at the network level, I propose a transmission-aware classification scheme to perform physical activity monitoring based on a WBSN. In particular the key contributions of this chapter are:

Selective advanced ECG processing based on lightweight heartbeat classifier:

- I propose to employ a neuro-fuzzy classifier (NFC) to analyze heartbeats in order to identify abnormalities in their morphology. The NFC is incorporated to a state-of-the-art multi-lead ECG processing application in order to activate the advanced DSP only in the cases where the classifier detects an anomalous heartbeat.
- I study the utilization of Random Projections (RP) to reduce the dimensionality of the representation of the heartbeats and propose a hybrid training framework that optimizes the NFC searching for the optimal RP configuration at the same time. The classification strategy is compared to other existing methods such as Principal Component Analysis.
- I describe the required optimization steps to implement and execute the RP-based NFC in a state-of-the-art embedded platform. The adaptations are performed to reduce the computational complexity and memory footprint of the application minimizing the loss of accuracy.
- The experiments show that the lightweight RP-based NFC provides high accuracy (up to 98.9% of sensitivity) when identifying abnormal heartbeats while keeping a low rate of false positives. It is also proved that the proposed selective advanced DSP processing strategy increases the overall energy efficiency of the system.

Transmission-aware WBSN for efficient physical activity monitoring:

- I propose a Wireless Body Sensor Network (WBSN) for activity monitoring in which several sensors cooperate with a smartphone to perform continuous subject monitoring.
- I propose two different classification schemes. In the first one, classification is per-

formed on the smartphone after receiving features extracted in the sensors. In a more energy-efficient version, classification is also performed in the nodes and the smartphone is only used to report the results.

- I performed an study of the best WBSN configuration in order to reduce the amount of deployed nodes while maximizing the accuracy of the classification.
- Both classification schemes are evaluated in terms of accuracy obtaining high classification quality in both cases. The results show that by using a node-based classification scheme, data transmission through the WBSN can be reduced up to 86% while obtaining a misclassification rate of only 12%.

The remaining of this chapter is organized as follows. Section 2.2 gives a brief introduction to some of the existing advanced DSP methods used in the field of electrocardiogram processing. Then, Section 2.3 presents the proposed strategy to perform selective DSP on ECG biomedical monitors and describes the implementation of the employed lightweight heartbeat classifier. After that, Section 2.4 details the proposed transmission-aware WBSN for physical activity monitoring and evaluates the different tradeoffs in terms of classification quality and data transmission. Finally, Section 2.5 concludes the chapter summarizing the main achievements.

2.2 Existing Advanced Embedded Signal Processing Methods

State-of-the-art ECG biomedical monitors perform on-node digital signal processing of the cardiac signals. In particular, after being conditioned, the signals are used to delineate individual heartbeats, retrieve their characteristics and interpret them to detect pathologies. The first two steps, (*conditioning* and *delineation*), present the most challenging real-time constraints, because they deal with the manipulation and analysis of digital signals [35][36][37] and have been the focus of researchers in the last few years. Apart from conditioning and delineation, when the raw signals need to be transmitted through the radio-link, some embedded compression techniques [11] [38] have been proposed in the literature in order to reduce the amount of data to be transmitted. Hereafter, I review the most relevant methods in the field of embedded ECG processing focusing on those that are employed throughout this thesis.

2.2.1 ECG Conditioning

ECG signals are usually corrupted due to several sources of noise. Automatic diagnosis algorithms need to remove these artifacts before interpreting the signals. There are two main sources of signal noise, namely baseline wander and high-frequency muscular noise. The baseline wander consists of a low-frequency component in the ECG signal (from 0.05 to 1Hz), that can be caused by patients' respiration, perspiration or even the misplacement of the sensing electrodes. On the other side, muscular or electromyographic noise can add an extra high-frequency component to the ECG signal (over 50 Hz) and is usually originated by the contraction of body muscles during movements.

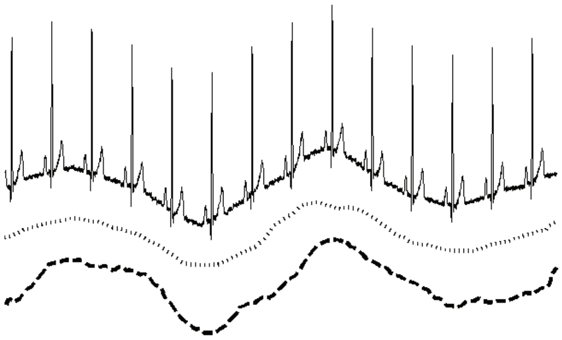


Figure 2.1 – Acquired signal (top) and its baseline estimations using morphological filtering (middle) and spline interpolation (bottom).

While traditional filtering techniques can be applied to clean the ECG signals, the implementation of suitable algorithms to perform this task on embedded platforms have caught the attention of researchers in this domain. In particular, two main techniques have been proposed and widely used for this purpose:

2.2.1.1 Spline Filtering (SF)

This method firstly introduced in [39], is only suitable to remove baseline wander from the ECG signal. It assumes that several time intervals of an ECG stream are silent, i.e., they are devoid of any heart activity [40]. One such segment is the interval between the P and the Q waves (PR segment in Figure 2.2). Stemming from this observation, it is then possible, by placing *knots* on the PQ segment and fitting a cubic polynomial in successive triplets of knots, to estimate the baseline. For each interval between heartbeats $[i, i+1]$, the knots identified for the beats i, $i+1$ and $i+2$ are considered to derive the estimated baseline. Determining the knots position requires an estimation of the QRS complex onset, which can be implemented using a lightweight version of an embedded delineator as the ones described later in Section 2.2.2.

2.2.1.2 Morphological Filtering (MF)

This technique is based in the work of [37] where the authors employ morphological operators to perform filtering of the ECG signal. In particular, the morphological operators *dilation* ($\oplus$) and *erosion* ($\ominus$) are defined as:

$$Dilation: (f \oplus g_s)(x) = \max_{t \in (G_s \cap D_x)} \{f(x-t) + g_s(t)\} \tag{2.1}$$

$$Erosion: (f \ominus g_s)(x) = \min_{t \in (G_s \cap D_x)} \{f(x+t) - g_s(t)\} \tag{2.2}$$

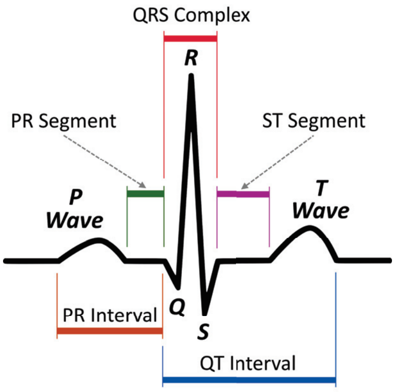


Figure 2.2 – ECG heartbeat showing the main characteristic waves and some of the clinically relevant periods of interest

where $f(x)$ is a discrete ECG signal and D_x is D translated by x. These operators are combined to generate the *opening* and *closing* functions. Opening ($\circ$) of a function f using a structuring element g_s is defined as:

$$f \circ g_s = (f \ominus g_s) \oplus g_s \tag{2.3}$$

If g_s is flat, these manipulations remove from f peaks of length smaller than s. Dually, closing ($\bullet$), defined as:

$$f \bullet g_s = (f \oplus g_s) \ominus g_s \tag{2.4}$$

removes pits of length smaller than s. By employing structuring elements of length greater than the longest ECG wave (typically, the T wave), it is then possible to derive the appropriate baseline, that can then be subtracted from the acquired signal. These morphological methods can be used also to filter high-frequency muscular noise using structuring elements that shorter than the shortest ECG wave.

2.2.2 ECG Analysis: Delineation

ECG delineation consists in accurately delimiting the ECG characteristic waves. In a typical smart WSN scenario, filtered signals are analyzed by a delineation pass to find the fiducial points of each heartbeat, corresponding to the onset, peak and end of its characteristic waves: P, QRS complex and T (Figure 2.2). Abnormalities in the length of ECG waves are important markers of different heart conditions and the output of this delineation phase can be used in a later stage to implement autonomous detectors. Many approaches have been proposed to automate ECG delineation. Some of them (such as methods based on low-pass differentiation [35], neural netwoks [41] or hidden Markov models [42]) present a high computational complexity, so that they cannot be adopted in WSNs. Two possible options compatible with the available resources in WSNs are based on Digital Wavelet Transforms

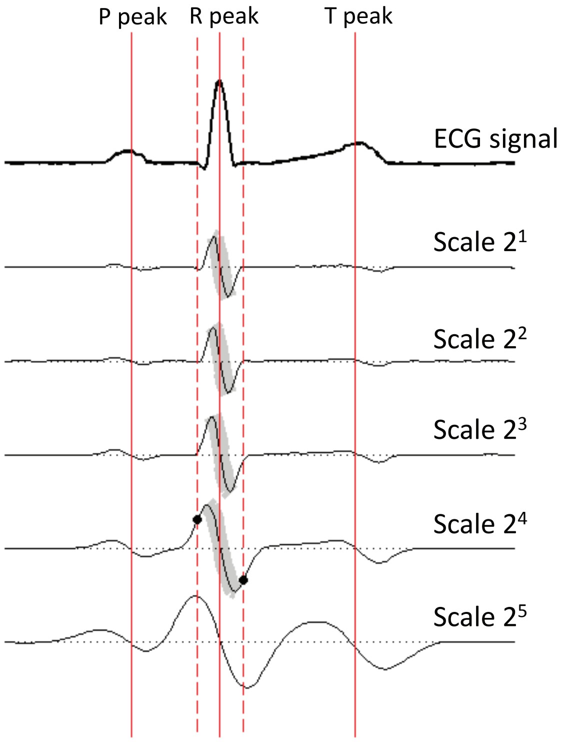


Figure 2.3 – DWT decomposition of an ECG heart beat. Maximum-minimun intervals examined to find the R peak are highlighted in gray.

(DWTs) [43] [44] and Multi-scale Morphological Derivatives (MMDs) [36], which in spite of being conceived as off-line algorithms, have been optimized to execute in embedded platforms [45] [10].

2.2.2.1 Wavelet-based Delineation (DWT)

This delineation method based on the dyadic wavelet transform (hence, DWT delineation) considers a decomposition of acquired signals in five dyadic scales, which can be efficiently computed using a filter bank composed of low- and high- pass FIR filters. Scales represent derivatives of smoothed versions of the input ECG signals, as exemplified in Figure 2.3. To ensure time-invariance among different scales, the filter impulse response is interpolated using the *algorithme á trous* method, illustrated in [46]. Because the different characteristic waves of beats present distinct frequency contents, their fiducial points are retrieved at different scales, the QRS complex being reflected in scales 2^1 to 2^4, while P and T waves presents their major components in scales 2^4 and 2^5.

As scales are computed, the DWT delineator searches for maximum-moduli points at the different scales, reflecting points of maximum slope in the acquired signals. The R peak is identified as the zero-crossing point at scale 2^1 in-between tuples of maximum moduli with different signs across scales from 2^1 to 2^4. Dynamic thresholding is performed to reject maximum moduli with small absolute values.

QRS onset is identified at scale 2^4 by a back-search for the point where its absolute value becomes smaller than one-fourth of the peak associated with the wave. Similarly, the QRS end is retrieved by a forward search for the point where scale 2^4 becomes smaller than three-fourths of its maximum absolute value.

Focusing on search windows before and after the QRS complex, P and T peaks are identified as the zero-crossing points at scale 2^3 between two maximum moduli either at scale 2^4 or, if not such tuple is found, at scale 2^5. Even for P and T waves, dynamic thresholding is employed to filter maximum moduli. Calculation of the onset and end of P and T waves is then similar to the QRS case.

2.2.2.2 Multi-scale Morphological Derivative-based Delineation (MMD)

In this case a multi-scale morphological derivate of the ECG signal is employed to delineate the ECG signal (hence MMD delineation). The morphological derivative M_f^d of a discrete signal $f : D \subset \mathbb{R} \to \mathbb{R}$ at scale s is defined as:

$$M_f^d(x) = \frac{(f \oplus g_s)(x) + (f \ominus g_s)(x) - 2f(x)}{s} \tag{2.5}$$

where $g_s : G_s \subset \mathbb{R} \to \mathbb{R}$ is a structuring element of length s and $\oplus$ and $\ominus$ are the *dilation* and *erosion* morphological operators defined in equations 2.1 and 2.2 respectively.

If a flat structuring element is chosen, computing the morphological derivative of f can be performed by sliding a window of size s over the signal, and calculating the maximum and minimum of the signal as well as its value in the central point of the window:

$$M_f^d(x) = \frac{\max\{f(t)\}_{t \in I} + \min\{f(t)\}_{t \in I} - 2f(x)}{s} \tag{2.6}$$

where $I = [x - s, x + s]$.

As Figure 2.4 shows, the morphological transformation translates peaks on the input signal in pits on the transformed one, while peaks or sudden change in slope of the transformed signal highlights onsets or ends of waves in the input signal. A search on the MMD transform for a negative value exceeding a dynamically-adjusted threshold retrieves the peak of the R wave; peaks (or sudden change in slope) around it retrieve the onset and end of the QRS complex. Before and after the found QRS complex, tuples of zero-crossing points mark the presence of the P and T waves respectively. P and T peaks are identified by the minimum value in-between the crossing points, while their onset and end by the maximum values before and after the crossing points.

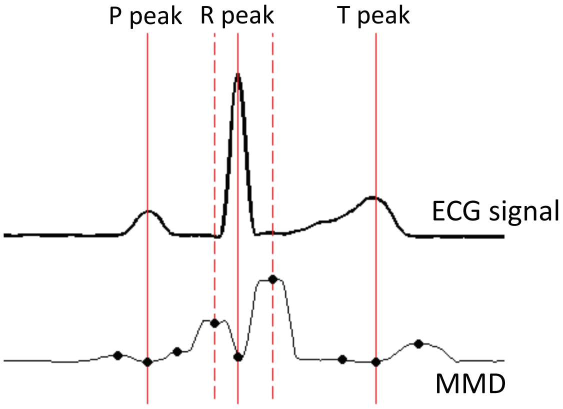


Figure 2.4 – Acquired signal (top) and its MMD-transformed version (bottom).

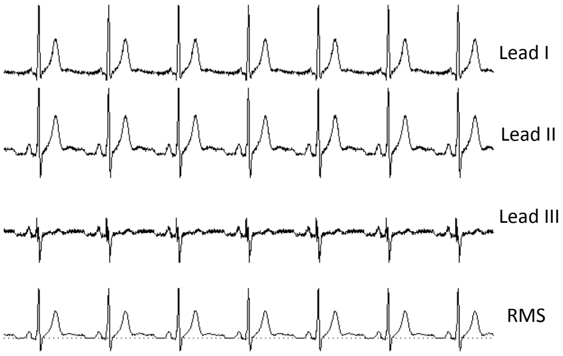


Figure 2.5 – Three-leads acquisition and corresponding RMS-combined signal.

2.2.3 Multi-Channel Signal Fusion

WSN devices usually acquire multiple ECG channels or *leads* concurrently, giving the opportunity to increase the delineation performance by fusing acquired data sampled from different sources. While a traditional arithmetical mean would work for the combination of signals with similar morphologies, a more robust approach is required when the shape of the sensed signal depends on the electrode placement as in the case of ECG. An RMS (Root Mean Square) combination has been proposed and successfully adapted to execute on WSNs [10]. The combination for N channels is performed following the equation:

$$x(t)_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i(t)^2)} \tag{2.7}$$

The example in Figure 2.5 showcases its benefits: as this figure shows, lead I presents a small P wave, while lead III is noisy and has a low T wave. However, their combination has a higher quality compared to each lead in isolation. To properly RMS-combine signals, they must be firstly centered on the iso-electric line, eliminating low-frequency baseline wandering.

Multi-channel data fusion becomes an effective technique to counteract the effect of noise

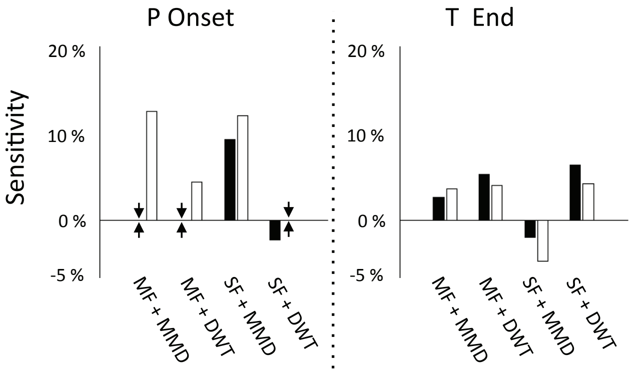


Figure 2.6 – Relative improvement in sensitivity of two- and three-leads delineation with respect to single-lead delineation. Black = 2 leads, White = 3 leads.

when it cannot be completely removed by filtering algorithms. Figure 2.6 reports the relative change in delineation quality (i.e. sensitivity) with respect to single-lead when either two or three leads are RMS-combined after filtering and then used for delineation. For conciseness, only data referring to P onset and T end is reported, being the most challenging points. Even if some outliers are present, in most cases fusing data from different leads resulted in an increased delineation quality. The presented algorithms in Sections 2.2.1 and 2.2.2 have been combined to showcase the different changes in performance thanks to the multi-lead data fusion.

2.2.4 ECG Compression: Compressed Sensing

As previously introduced, a common strategy to reduce the data transmitted through the radio-link in nowadays biomedical monitors consists in performing on-node digital signal processing in order to only send the clinically relevant parameters. However, in some scenarios in which a visual inspection of the ECG is needed, transmitting the raw signal is still strictly required. ECG is usually sampled at a rather high sampling frequency (up to 1 KHz) and as a consequence the amount of data to transmit while streaming the signal is too large leading to a prohibitive energy cost. In this context, compressing the data by doing some pre-processing helps to reduce the utilization of the wireless link and therefore improves energy efficiency. Compressed Sensing (CS) [47] has been proposed recently to be applied in the field of electrocardiography. CS relies on the fact that ECG signals are sparse and can be efficiently compressed by computing:

$$y = \phi x \tag{2.8}$$

where $x \in \mathbb{R}^n$ is the input vector of ECG samples, $\phi \in \mathbb{R}^{k\times n}$ with $k < n$ is the so-called *sensing matrix* and $y \in \mathbb{R}^k$ is the resulting compressed vector. By performing the vector-matrix multiplication, ϕ maps the input vector x into y with a compression ratio of n/k. The amount of data to be sent can be largely reduced by using a big compression ratio. However, the reconstructed (*decompressed*) signal may suffer quality loss [47]. In [11], the authors have proved that ECG signals compressed by 50% can be reconstructed with a very good signal quality, which represents a good tradeoff.

The computational complexity of CS resides in the matrix multiplication expressed in Equation 2.8. However, the main drawback of this algorithm is the high memory footprint due to the large size of the sensing matrix. For instance, assuming a 50% compression ratio over a window of 1024 ECG samples, the necessary random matrix would require up to 2 megabytes of memory to be stored. Nevertheless, in [11] it has been shown that choosing a proper random sparse sensing matrix with few non-zero components per column leads to a low-complexity implementation, which still preserves the compressed data integrity making possible a good signal reconstruction. Moreover, when the non-zero components are forced to be 1 and the amount of ones per column is fixed, the sensing matrix memory footprint can be greatly compacted.

2.3 Proposed Selective ECG Processing Based on Embedded Heartbeats Classification

Early classification of heartbeats has potential benefits both in the clinical practice and in the design of WSNs. On the diagnostic side, it can provide helpful information for speeding up the visual inspections of lengthy ECG recordings by the medical staff, who can focus only on those beats presenting pathological characteristics. From the perspective of system design, the advantages are two-fold: first, if a detailed diagnosis is performed off-node, it can be desirable to transmit or store only pathological beats on the WSN, thus greatly reducing either the energy employed for wireless transmission or the data storage requirements, respectively. Second, if the detailed analysis of heartbeats is executed on the WSN, computation effort can be reduced by activating these advanced algorithms only when abnormal beats are detected, thus drastically decreasing the computational requirements and therefore the power consumption.

2.3.1 Target Application

The target scenario is depicted in Figure 2.7 where a new module is responsible of detecting abnormal morphologies in the heartbeats of the acquired ECG signal. By decoupling early and detailed analysis, and performing the latter only on a small fraction of the acquired bio-signal, this new scheme aims to maximize the energy efficiency of autonomous devices for personal health monitoring. Without loss of generality, this work assumes a context where pathological heartbeats occur less frequently than normal ones, which is the usual case in long-term ECGs acquisitions.

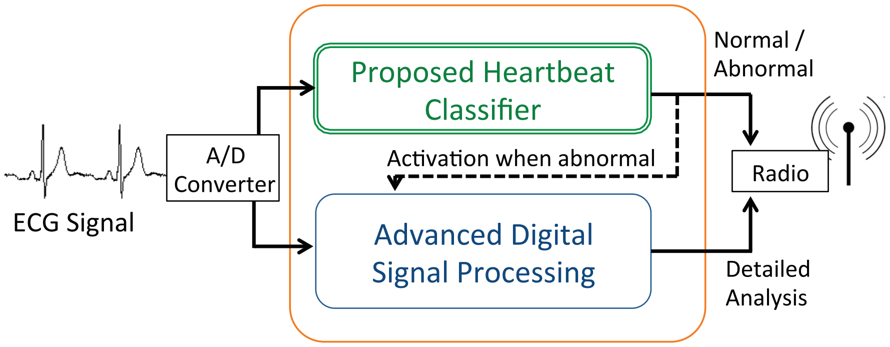


Figure 2.7 – Target system featuring a classification block that selectively activates a detailed digital signal processing chain.

2.3.2 State of the Art and Problem Challenges

In the field of clinical electrocardiography, an important application of smart wireless nodes consists in separating normal and pathological heartbeats, performing an early diagnosis step. For this task, many off-line algorithms have been proposed in the literature based on the morphology of the heartbeat [48, 49, 50]. However, the implementation of these algorithms on WSNs represents an important challenge due to the high run-time demands. One of the most important aspects when devising on-node classification is the high dimensionality of the faced problem which is not only affected by the chosen classification method but also by the size of the input samples to classify (the heartbeats in this case). Using standard off-line techniques, tens of samples before and after the center peak of the heartbeat are required to perform a reliable classification. Dimensionality reduction in these problem is traditionally carried out by extracting a rather small but meaningful enough set of features from the input samples.

Several state-of-the-art strategies for off-line classification of ECGs can be found in the literature. They can be distinguished based on the methodology employed to extract the features of individual heartbeats, which later are the input of the classifier. A first methodology considers the extraction of the most important component in the ECG signal in an appropriate subspace, employing either independent or principal component analysis (ICA [51] or PCA [49], respectively). A different approach, introduced in [52] and [53], relies instead on the detection of morphological features, such as the presence, duration and shape of the heartbeat characteristic waves. A third methodology focuses on (possibly trained) random linear combinations of the input samples, employing Random Projections (RP) [54, 55] for representing heartbeats with a few coefficients. In particular, *Achlioptas* projections [56] adopt matrices consisting only of the elements 0, 1 and -1, thus allowing a compact representation of the projection matrix and requiring low run-time resources. In a preliminary study presented in [57], I showed how *neuro-fuzzy classifiers* (NFCs) [58] can be optimized for and implemented on the constrained resources typically available on WSNs, while still providing high classification accuracy and meeting real-time constraints. To reduce the amount of data and cope with

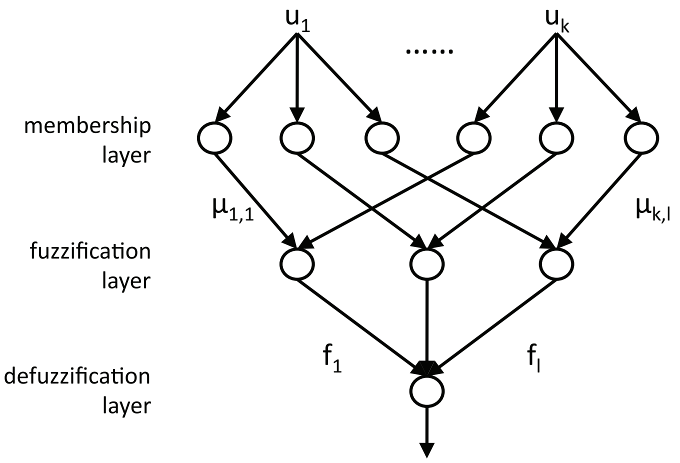


Figure 2.8 – Multi-layer neuro-fuzzy classifier.

the limited resources of WSNs, in that study I chose a dimensionality reduction based on the previously mentioned Random Projections. Herein, I propose and comparatively evaluate different approaches allowing a compact representation of heartbeats: *Random Projections* (RPs) [56], Principal Component Analysis (PCA) [59] and morphological features resulting from an automated Fiducial Points Detection (FPD) [36].

2.3.3 Classification Method

As opposed to the existing approaches for off-line classification of heartbeats based on morphology analysis ([48, 49, 50]), on-node classification has to cope with the limited computation resources that are available on a WSN while providing a comparable accuracy.

Among the classification techniques that could be adapted to on-node execution, neuro-fuzzy classifiers (NFCs) [58] represent a promising option. Their ability to explicitly express uncertainty in classification, given by the employed *fuzzy values*, makes them particularly well-suited to the problem of heartbeat classification [60, 61]. NFCs consist of a simple feed-forward multi-layered structure as the one depicted in Figure 2.8. A first *membership* layer employs Membership Functions (MFs) to compute, for each input value u_k, a membership grade $\mu_{k,l}$ for each of the target classes l. After the training of the NFC, the obtained MFs are gaussian curves, defined by their center c and variance σ:

$$\mu_{k,l}(u_k) = \exp\left(\frac{-(u_k - c_{k,l})^2}{2\sigma_{k,l}^2}\right) \tag{2.9}$$

where l is the corresponding class from the set of target classes. In the subsequent *fuzzification* layer, the membership grades of all the coefficients for each class are combined by means of a weighted product according to the following expression:

$$f_l = \prod_k w_{k,l}\mu_{k,l} \tag{2.10}$$

The resulting *fuzzy values* quantify how likely the examined heartbeat belongs to that specific class. Finally the third *defuzzification* layer of the NFC labels the input sample as one of the target classes based on the fuzzy values: the largest fuzzy value with respect to the values of the other classes dictates the final decision.

NFCs can be effectively trained using established methods, the most common being the gradient descent algorithm described in [61] and the scale conjugate gradient algorithm introduced in [62] and [63], which I employed in the proposed approach. NFCs are computationally simpler and present lower memory requirements than other existing techniques such as gaussian Support Vector Machines (SVMs) [48], while being more accurate than simpler solutions based on linear SVMs and Linear Discriminant Analysis (LDA) [64]. The possible utilization of these techniques is later studied in Section 2.3.8 where the performance and limitations of the different alternatives are discussed.

2.3.4 Dimensionality Reduction Techniques

Reducing the dimensionality of the heartbeat is an effective technique to simplify the complexity of the classification problem. I explored three different solutions to achieve this objective, which are compatible with the limitations of WSN processing architectures.

Random Projections (RPs) Random projections allow to represent the ECG by means of a low number of coefficients, which are obtained by multiplying the input vector of samples by a random projection matrix. In order to improve the run-time performance of the classification, we require the RP matrix to be sparse. This requirement is fulfilled by a $k \times d$ Achlioptas matrix ($\mathbf{P}$) [56], where d is the number of digital samples acquired for each heartbeat and k is the number of desired coefficients in the random projection, with $k \ll d$. The elements of $\mathbf{P}$ are defined as:

$$\mathbf{P}_{k,d} = \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ -1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \end{cases} \tag{2.11}$$

The dimensionality reduction is then achieved by *random-project* the vector v according to the following equation:

$$u_{RP} = \mathbf{P}v \tag{2.12}$$

Because of the structure of the Achlioptas matrix, each row of $\mathbf{P}$ indicates which elements of v have to be added to (or subtracted from) to derive the corresponding vector u_{RP}, without using an actual multiplication. Even though the approximation error introduced by random projections is theoretically bounded [54], in practice it is observed that certain RP matrices perform better than others [57]. As a consequence, the generation of **P** requires a training process (Figure 2.10a). The aim of the training is to derive a matrix **P** which leads to a high-quality classification, resulting in a joint optimization process of the RP matrix and of the NFC.

Principal Component Analysis (PCA) Principal component analysis retrieves the set of linear projections on a set of orthonormal axis, where the variance in the input data is maximized [49]. The columns of the PCA projection matrix **T** of size $k \times d$, with $k \ll d$, are the k leading eigenvectors of the covariance matrix of the input data, which is defined as:

$$S = 1/N \sum_{i=1}^{N} (v_i - \mathrm{E}[v])^T \tag{2.13}$$

where N is the number of elements in an initial training set of heartbeats (*training_set_1*), v_i is the i-th element of the vector containing the digital samples of a heartbeat, and $\mathrm{E}[v]$ is their mean value. As in the case of RPs, dimensionality reduction is performed by a matrix-by-vector multiplication: $u_{PCA} = \mathbf{T}v$.

The extraction of the PCA matrix **T** only depends on the algebraic properties of the input vector v and on the numbers of projection axis k, therefore it can be performed independently from the NFC classifier training (Figure 2.10b). On the other hand, as opposed to the RP case, **T** is not sparse in general, leading to a more complex run-time implementation that in addition involves multiplications to compute u_{PCA} (RPs can be implemented using exclusively additions and subtractions).

Fiducial Points Detection (FPD). Differently from RPs and PCA, the detection of morphological features explicitly *interprets* the input ECG signal. In particular, FPD aims at retrieving the position of the fiducial points (onset, peak and end of the P and T waves, and the beginning and end of the QRS complex) of each heartbeat with respect to the position of the R peak (Figure 2.9). The considered fiducial points (8 in total) constitute the coefficients of the vector u_{FPD}, which is the dimensionally-reduced representation of v, and which is used to subsequently feed the NFC classifier.

In the proposed framework (Figure 2.10c), we perform FPD using the lightweight algorithm described in Section 2.2.2.1. The algorithm is based on the digital wavelet transform (DWT) decomposition, which transforms each characteristic ECG wave into tuples of maxima and minima in the DWT domain. Since the different waves present distinct frequency contents, their fiducial points are retrieved at different scales, the QRS complex having a stronger component at lower scales than the P and T waves. The DWT delineation shows good run-time properties [10] that make it suitable for on-node fiducial point extraction, and its robustness makes it a good choice to deal with pathological beats with abnormal morphologies.

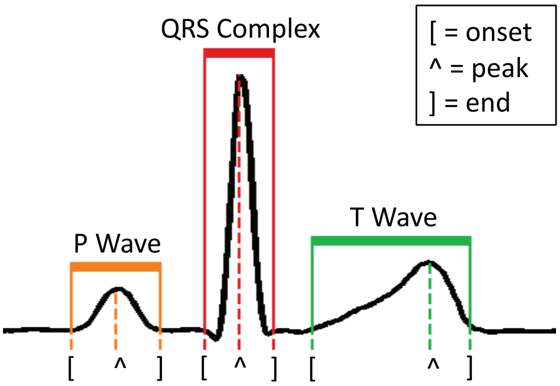


Figure 2.9 – Delineated normal heartbeat [65].

2.3.5 Proposed Training Framework

The high-level scheme of the proposed framework for on-node early classification of normal and pathological heartbeats is shown in Figure 2.10. The framework can be divided into an off-line training phase (Figure 2.10, top), in which the parameters of both the classifier and the dimensionality reduction technique are derived, and a test phase (Figure 2.10, bottom), discerning normal and pathological heartbeats at run-time on the WSN.

Different dimensionality reduction strategies require a different training approach, as illustrated in Figure 2.10a–c. In the case of PCA (Figure 2.10b), principal components are derived from an initial set of heartbeats before the neuro-fuzzy classifier is trained. In particular, the PCA matrix is derived according to the iterative solution presented in [66].

Conversely, when random projections are used (Figure 2.10a), a concurrent optimization of the NFC classifier and the random projection matrix is required. The selection of the best combination is achieved by means of a genetic algorithm [67]. The algorithm starts from an initial population of random matrices and, for each of them, trains the corresponding NFC employing the previously mentioned scale conjugate method [62] using a set of *random-projected* heartbeats (*training_ set_1*). Each of the obtained RP-NFC pairs is then evaluated over a different and larger set of heartbeats (*training_set_2*). According to the result of the evaluation, the genetic algorithm selects the proper chromosomes (i.e., the best **P** matrices) and performs mutation and crossover over them to refine the random projection. According to the realized experiments, an initial population of 20 randomly-generated matrices, and an exploration of 30 generations by the genetic algorithm, are sufficient to converge to a matrix **P** that provides a close-to-optimal performance (results are shown in Section 2.3.7.2).

In the third case, when the heartbeat dimensionality is reduced by using Fiducial Points Detection, no specific optimization is required, as they are extracted from the heartbeats by means of a delineation algorithm, hence in this case only the NFC has to be trained once (Figure 2.10c) using the fiducial points of the initial set of heartbeats (*training_ set_1*).

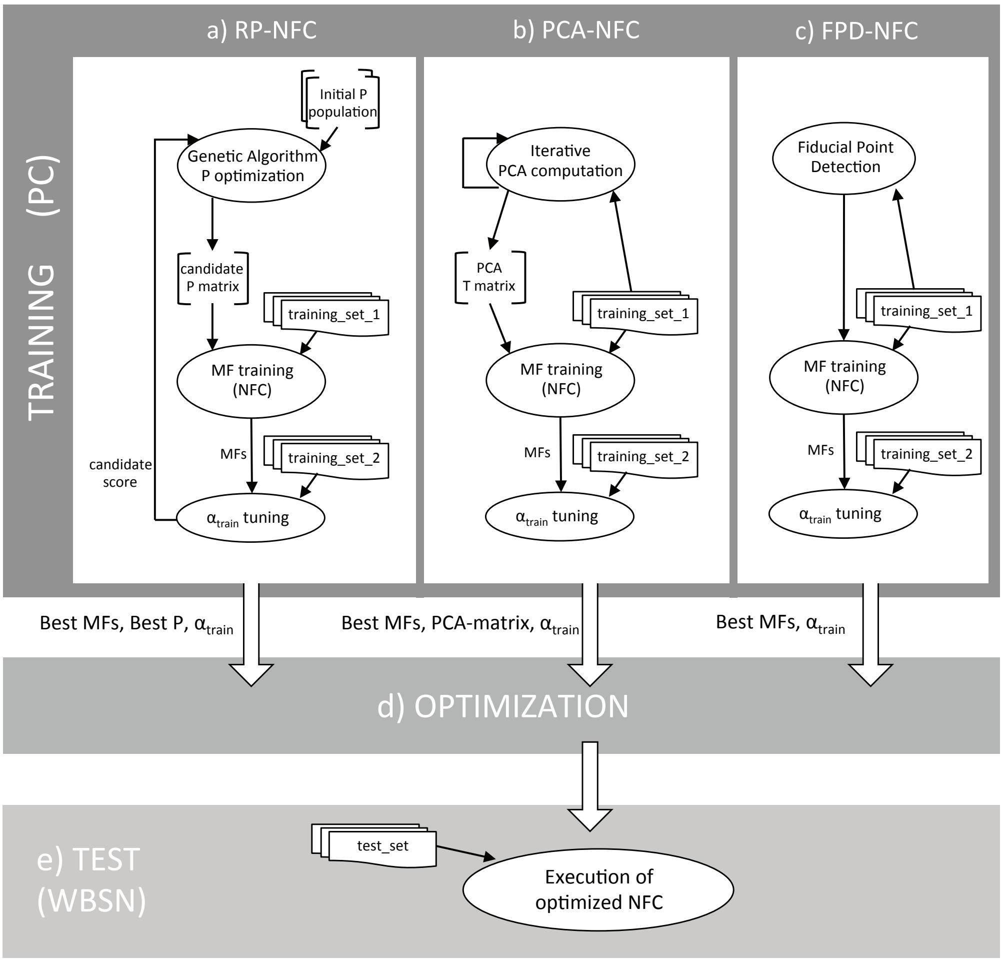


Figure 2.10 – Classification framework: *Top (PC-side):* NFC training using Random Projections (**a**), Principal Component Analysis (**b**) and Fiducial Points Detection (**c**). Optimization for WSNs (**d**). *Boottom (WSN side):* Real time execution of the optimized implementation (**e**).

It is important to mention that the training and test phases of the classification framework have different constraints. On the one hand, the training phase is performed off-line on a host workstation, which employs high-precision floating-point data representation in order to obtain an accurate framework set-up. On the other hand, the test phase and the actual classifier is eventually executed on an embedded WSN, being therefore tightly constrained in terms of memory footprint and computation resources, since only integer arithmetic is admitted and no exponential operations are possible in the architectures devoted for the WSN domain. As a consequence, it is mandatory to transform the classifier after the training (Figure 2.10d) to lower its computational requirements according to the embedded platform capabilities. This step is detailed in Section2.3.6.

Finally, the choice of a proper defuzzification coefficient α_{train} gives the flexibility to unbalance the classifier training process (i.e., it allows to define an upper bound on the number of abnormal beats that are incorrectly classified as normal). Once this percentage is fixed, the performance metric used to train and score the classifier is then the percentage of normal beats correctly detected, and therefore discarded for detailed analysis.

2.3.6 Resource-Constrained Optimization Phase

The dimensionality reduction technique and the trained classifier cannot be employed *as they are* on a WSN platform, due to the available limited resources. In this regard, several considerations have to be taken into account. First, data must be represented in the integer domain, as opposed to the floating-point format used in the training phase. Then, the complex MFs employed in the NFC, which require prohibitive exponential operations for embedded platforms, must be simplified. In addition, the NFC fuzzification layer needs to be analyzed to prevent overflows when performing the product operation. Finally, special care must be taken regarding the memory required to store tables such as the RP matrix, or to represent the different parameters derived during the training phase. In this section, I detail the devised strategies performing these steps (Figure 2.10d), thus enabling the implementation of the proposed classifier on a WSN.

Membership functions linearization: I proposed a linear segmentation of each gaussian MF in the classifier, to avoid the computation of exponentials. Given the centre c and standard deviation σ of a MF, we map it onto the integer range $[0, (2^{16}-1)]$ (i.e., a 16-bit representation) according to the following scheme:

$$MF_{lin}(x) = \begin{cases} 0 & \text{if } |c-x| \geq 4S \\ 1 & \text{if } 4S > |c-x| \geq 2S \\ lin.approx1 & \text{if } 2S > |c-x| \geq S \\ lin.approx2 & \text{if } S > |c-x| \end{cases} \tag{2.14}$$

where MF_{lin} is the linearized MF and $S = 2.35\sigma$. The linear approximation segments are graphically represented in Figure 2.11. As 1 is the smallest non-zero value that can be represented in the chosen integer space, this formulation has the desirable property to be positive in a large range, hence it is rare that a fuzzy value becomes 0 after the defuzzification (product) stage.

Fuzzification: In the defuzzification layer, when all the weighting factors (see Section 2.3.3) are set to 1, only the ratio between the fuzzy coefficients f_l is relevant, as opposed to their absolute values. The realized optimizations of the fuzzification step stem from this observation, and consist in retaining the maximum precision given the 32-bit representation used for the accumulators of the fuzzification products. In the proposed framework, we first force the weighting factors to be 1 (i.e., irrelevant) during the NFC training process. Then, in the WSN implementation of the fuzzification step, the membership grades $\mu_{k,l}$ related to the two first coefficients are multiplied for each of the target classes. The resulting numbers are left-shifted to the maximum amount so that none of them overflows, and then the least significant 16 bits

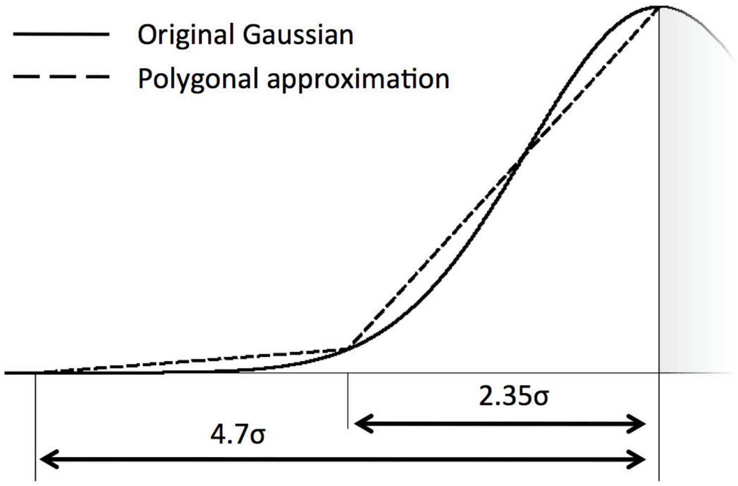


Figure 2.11 – Linear approximation of gaussian MFs in the range $[-4.7\sigma, 0]$, compared to a gaussian curve.

are discarded. All the subsequent membership grades are then processed in a similar way, thus obtaining the fuzzy values of the beats for the different classes with the highest possible precision.

Defuzzification: The defuzzification stage has been adapted in two phases. First, the decision process has been modified to be easily implementable in the resource-limited WSN architecture by avoiding complex manipulations such as divisions. Second, the configuration of the classifier has been unbalanced to guarantee a minimum sensitivity for one of the classes (in this case the pathological class). The proposed implementation of the defuzzification layer marks each beat as either normal or pathological, by considering the two largest fuzzy values $(M1_f,\ M2_f)$ and the sum of all of them $S = \sum_l f_l$. If $(M1_f - M2_f) \geq \alpha_{train} \cdot S$ (with $\alpha_{train} \in [0,1]$), the beat is assigned to the class with the maximum fuzzy value. Otherwise, the beat is marked as *unknown* and considered as potentially pathological. All these manipulations do not employ divisions, and can therefore be efficiently implemented in WSNs. To unbalance the classifier decision to ensure a certain quality, it is possible to choose a defuzzification coefficient α_{test} different from the α_{train} that was obtained during the training phase (described in Section 2.3.5), allowing to adjust the ratio of detected normal and abnormal beats and therefore obtaining the desired sensitivity of a desired class (i.e., pathological heartbeats).

Memory-Aware Representations: As mentioned in Sections 2.3.4, random projection matrices are composed of only three values $(+1, -1$ and $0)$ and are sparse by construction, thus admitting a compact representation where each element can be coded using only two bits. It therefore requires 1/4 of the memory with respect to a corresponding matrix of 8-bits values. On the other hand, this compression is not possible for the PCA matrices which have to be stored employing 16-bit words. While resulting in a memory footprint still compatible with

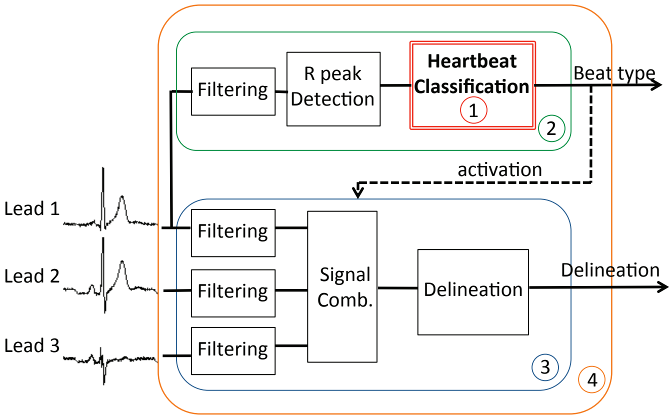


Figure 2.12 – Experimental scenario: classification is used to activate an accurate multi-lead morphological delineation only in case of heartbeat abnormality.

a WSN implementation, the footprint is significantly higher with respect to the compact RP case.

2.3.7 Experimental Results

I evaluate hereafter the effectiveness of the proposed framework in terms of performance and workload. In the next subsections, I detail the employed set-up for the evaluation, detailing the target embedded platform and defining the studied metrics. Then, I discuss the performed study to assess the classification accuracy achieved by coupling the proposed neuro-fuzzy classifier with the different dimensionality reduction techniques presented in Section 2.3.4. In Section 2.3.7.3 I compare their run-time performance in terms of execution time and memory requirements. Finally I prove how the proposed methodology contributes to the system-level reduction of the WSN energy consumption.

2.3.7.1 Experimental Set-up

To comparatively evaluate the proposed classification strategies I investigated their performance when identifying three type of heartbeat morphologies considered of clinical interest in the field of cardiac analysis. In particular, I considered normal heartbeats (hereafter labeled as N) and heartbeats affected by premature ventricular contraction and left bundle branch block (labeled V and L respectively), which present abnormal morphologies. The heartbeats are extracted from the MIT-BIH Arrhythmia Database (publicly available on the PhysioNet website [68]). The considered heartbeats were extracted from the MLII lead of each recording. The ECG recordings in the database are acquired at 360 Hz and we define each heartbeat as the 100 samples preceding the R peak (cf. Figure 2.9), and the 100 samples that follow it.

The real-time performance of the proposed classifier is evaluated by means of its actual implementation on a physical embedded platform. In this work, we have employed the state-of-the-art IcyHeart System-on-Chip (SoC) [69], which integrates a low-power microprocessor featuring a clock frequency of 6 MHz and an embedded RAM of 96 KBs.

As introduced in Section 2.3.1, the goal of the classification is to distinguish between normal and pathological beats, in order to trigger a detailed analysis for abnormal beats only. The general structure shown in Figure 2.7 is embodied by the application whose diagram is detailed in Figure 2.12, where the advance signal analysis is represented by a three-lead morphological delineation (MMD) ([36], Section 2.2.2.2). The employed R-peak detector is a simplified version of the wavelet-based technique proposed in [10] (Section 2.2.2.1).

In order to assess the performance of the system in terms of accuracy, two specific metrics have been defined according to the match between manual annotations from the database and automatic classifications made by the NFC. When a heartbeat that is manually annotated as normal is classified as such, it is considered as a *True Normal* (T_N). In case the heartbeat is misclassified, it is considered as a *False Normal* (F_N). Similarly, both types of matching, *True Abnormal* (T_A) and *False Abnormal* (F_A), are defined for the abnormal class. With this matching criteria, I considered two figures of merit: Normal Discard Rate (NDR) and Abnormal Recognition Rate (ARR). The NDR assesses the rate of normal beats that are correctly identified with respect to the total number of normal heartbeats, and thus they are discarded for further analysis according to the following expression:

$$NDR = \frac{T_N}{T_N + F_N} \tag{2.15}$$

Conversely, ARR reports the percentage of pathological heartbeats that are correctly identified with respect to the total number of abnormal heartbeats:

$$ARR = \frac{T_A}{T_A + F_A} \tag{2.16}$$

It is important to note that both metrics are not complementary and that the best performance is achieved when both figures reach the value 1. Across the experiments a lower bound of 95% on the ARR metric is forced, therefore the training process will tune the defuzzification coefficient α_{train} to meet this requirement. This self-imposed constraint comes from the clinical requirements of an automated system that needs a minimum sensitivity of abnormalities to be considered.

Obtained results are derived from a 4-fold cross-validation process, with heartbeats of the different classes proportionally and randomly divided across folds. Four rounds of experiments have been performed to compute all the results, using a different fold in each round as the test set (*test_set* in Figure 2.10) and the remaining 3 folds as the training set (*train_set_2* in Figure 2.10). A random subset of the training set (*train_set_1* in Figure 2.10) was used to compute the membership functions of the NFC classifier (regardless of the specific dimensionality reduction technique), and to derive the PCA matrix. In *train_set_1*, the classes are equally represented, in order not to overfit the classifier on the largest class (in this case, class

Table 2.1 – Composition of the sets of heartbeats employed in the different experiments.

	N	V	L	Total
MIT-BIH Arrhythmia Database	74,064	6608	8032	88,704
train_set_1	150	150	150	450
train_set_2 (3 folds)	55,548	4956	6024	66,528
test_set (fold size)	18,516	1652	2008	22,176

N). *Train_set_2* is used to adjust α_{train} during the training and, in the case of using RPs, to drive the genetic algorithm that derives the optimal projection matrix. The composition of the different heartbeat sets is detailed in Table 2.1.

2.3.7.2 Classification Accuracy

In this section, we aim to compare the classification accuracy obtained by combining the different dimensionality reduction techniques (RPs, PCA and FPD) with a neuro-fuzzy classifier structure.

When employing RPs and PCA, we considered two different implementations that reduce the dimensionality of the heartbeat to either 8 or 16 coefficients. A larger coefficient set impacts both the size of the RP or PCA matrix and the complexity of the NFC, therefore the decision among the different implementations is a trade-off between classification accuracy and real-time performance (in terms of run-time and required memory). Fiducial points detection (FPD) does not present this flexibility, as each heartbeat is always represented by 8 values, as discussed in Section 2.3.4: the position of the onset, peak and end points of the P and T waves, plus the onset and the end of the QRS complex, relative to the main R peak (Figure 2.9). If a point is not detected, its position is assumed to be the one of the detected neighbor fiducial point that is closer to the R peak.

We also explored two combined solutions, in which 8 coefficients derived from projections (either PCA or RP) are concatenated to the 8 detected fiducial points, resulting in 16 input values for the classifier. In these cases, the fiducial points are added after performing dimensionality reduction over the sample vector of the heartbeat and before training the NFC.

In the first set of experiments, the different studied dimensionality reduction techniques were tested, when α_{train} is trained to obtain a minimum ARR of 95%. The corresponding NDR figures for the different configurations are detailed in Table 2.2. Three main considerations can be derived from these results. First, results using only FPD are considerably poorer than the ones achieved by the other classification techniques, showing that the information contained in the delineation of the ECG characteristic waves is not sufficient to perform accurate classification. Second, it can be observed that for those methods in which different number of coefficients can be employed (i.e., RP and PCA), the performance does not vary significantly when using a larger dimensionality, as the maximum improvement does not exceed two per-

Table 2.2 – Average Normal Discard Rate (NDR) in the 4-fold cross validation process for a fixed minimum Abnormal Recognition Rate (ARR) of 95%.

Dimensionality reduction	**NFC Coefficients** **8**	**16**
FPD	70.25%	-
RP	94.80%	94.92%
RP + FPD	-	94.19%
PCA	88.15%	90.92%
PCA + FPD	-	90.04%

centage points. Third, with the given constraint on the minimum ARR, combining RP or PCA with FPD does not improve the NDR. Although this may seem counterintuitive, the reason for this behavior is that including inputs of different nature tends to increase the number of beats that are classified as *unknown* in the defuzzification stage. This forces the training process to slightly increase the value of α_{train} to meet the minimum ARR requirement, at the cost of negatively affecting the NDR.

In a second set of experiments, I investigated the flexibility of the proposed solutions varying the ARR constraint. In particular, even though α_{train} is still tuned to get a minimum ARR of 95% on the training set, we scaled the coefficient α_{test} to obtain different NDR/ARR trade-offs over the 4 rounds of the cross validation process. Figure 2.13 compares the NDR/ARR Pareto curves obtained for RP (16 coefficients), PCA (16 coefficients), FDP, and the combination of 8 coefficients from RP and PCA with the 8 fiducial points. Two main conclusions can be derived from the results. On the one hand, when the ARR constraint is set below 97%, RP-based solutions outperform the PCA-based methods reaching an NDR of 94.9% and not requiring the utilization of fiducial points. On the other hand, when the accuracy on the recognition of abnormalities is forced to be closer to 100%, the addition of FPD to the standard RP- and PCA-based methods makes the approach more robust, and leads to high values of NDR and ARR (PCA+FPD being the most reliable alternative). Conversely, as we explained above, using FPD alone for the classification does not provide competitive results w when compared to the other methods.

2.3.7.3 Run-time Analysis

As introduced earlier in Section 2.3.1 and depicted in Figure 2.12, the role of the classifier is to activate a detailed analysis for abnormal heartbeats in order to perform selective and computationally intensive advanced processing. In order for the system in Figure 2.12 to be effective, early detection of pathological heartbeats must not be the computation bottleneck during real-time execution. Therefore, classification should require considerably less effort than performing continuous analysis over the acquired signal.

In this section, I analyzed a fully functional and realistic diagnosis application (system (4) in

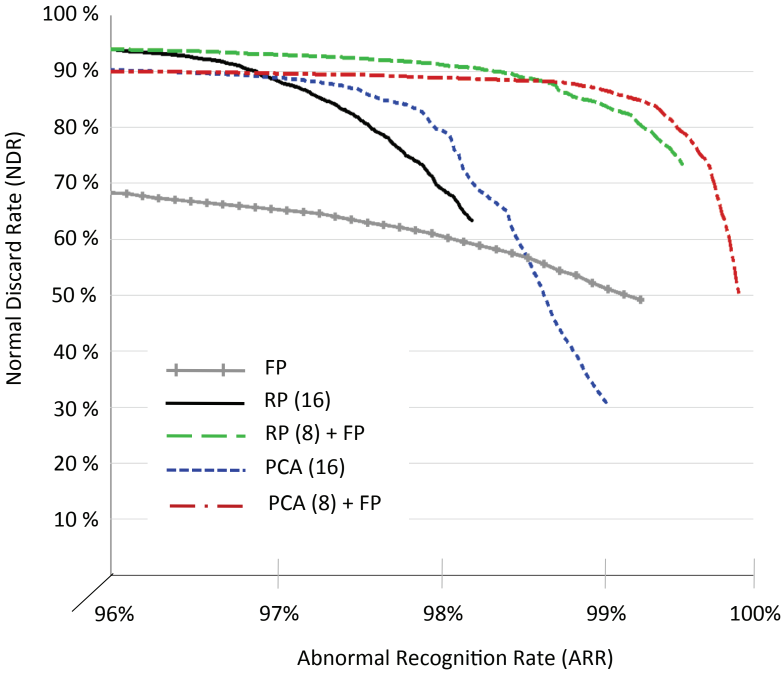


Figure 2.13 – Pareto curves of the NDR/ARR relations for various classification methods obtained by averaging the different values extracted from the 4-fold cross validation process.

Figure 2.12), in which the classification framework is used to trigger the detailed heartbeat analysis. The early heartbeat classification is performed on a single lead (sub-system (2)), whereas the detailed analysis is implemented by a three-lead delineation block (sub-system (3)).

Figure 2.12 shows that, apart from the classifier itself (sub-system (1)), two additional stages need to be incorporated to complete the proposed one-lead early classification (sub-system (2)). Firstly, a filtering stage is required to remove artifacts and baseline wandering caused by respiration and muscle contractions usually corrupting ECG signals. Secondly, a peak detector has to be employed to identify the heartbeats to classify.

We employed state-of-the-art solutions for the filtering stages, the peak detector and the delineation block, proposed by the authors of [10]. Filtering is performed using morphological operators, a wavelet-based algorithm is used for peak detection and a delineation algorithm using multi-scale morphological derivatives (MMDs) is executed over the root mean square (RMS) combination of the three filtered leads in subsystem (3). Their implementation has been optimized for execution on embedded WSNs.

Tables 2.3 and 2.4 report the computational and memory requirements of the different parts of the considered system depicted in Figure 2.12, when executed on the IcyHeart WSN operating at 6 MHz. The first column of the two tables lists the investigated implementations, based on RP, PCA, FPD, and the combined strategies described in Section 2.3.7.2. In order to make a

Table 2.3 – Duty cycle (%) of the sub-systems identified in Figure 2.12. Tests performed on the IcyFlex WSN running at 6 MHz.

Dimensionality reduction	**Sub-system** (1)	(2)	(3)	(4)
FPD	0.63	17.63	83.01	54.59
RP (16)	1.34	18.34		33.52
RP (8) + FPD	1.32	18.32		34.46
PCA (16)	1.24	18.24		34.99
PCA (8) + FPD	1.33	18.32		36.45

Table 2.4 – Memory footprint (kB) of the sub-systems identified in Figure 2.12. Tests performed on the IcyFlex WSN running at 6 MHz.

Dimensionality reduction	**Sub-system** (1)	(2)	(3)	(4)
FPD	7.75	36.43	46.39	82.82
RP (16)	3.04	31.72		78.11
RP (8) + FPD	8.58	37.25		83.65
PCA (16)	5.33	34.01		80.40
PCA (8) + FPD	10.87	39.55		85.94

fair comparison, experiments have been performed using 16 coefficients as input of the NFC, except in the case of FPD, which only employs 8. The second column reports the experimental results obtained for the classifier block (sub-system 1), while the third one also considers the filtering and peak detection (sub-system (2)). In the fourth column, a state-of-the-art implementation of a system performing three-lead MMD delineation (see Section 2.2.2.2) over the full input signal is reported for comparison. This setting reflects the performance of the advanced processing block running continuously, thus analyzing both normal and pathological heartbeats. As it can be seen in the tables, its run-time behavior does not depend on the classification methodology. Finally, in the right-most column of the tables, we provide the values for the complete target system where delineation is performed only on heartbeats marked as abnormal.

By observing the behavior of the classification block, two conclusions can be extracted. On one hand, the memory footprints of the different investigated methods are different, the RP with 16 coefficients being the least demanding one. On the other hand, the necessary additional computational effort for feature extraction and classification is minimal once the heartbeat is filtered and isolated (less than 1.5% of the duty cycle in all cases). Additionally, the FPD method benefits from the wavelet decomposition that is already performed during the R peak detection block.

As an assessment of the run-time efficiency of the proposed classification methods, it can be noticed that the main bottlenecks of the classification chain (sub-system (2)) are represented by the input data filtering plus the peak detector, and not by feature extraction and classification. Among the proposed implementations, the RP with 16 coefficients emerges as the best trade-off, providing a comparable performance with the rest of the methods with the lowest memory requirements.

The final, and most important, result is that the duty cycle of the complete system is always substantially lower than an equivalent one that performs a full detailed analysis on all the beats. Analyzing all the beats of the database described in Table 2.1 and considering numbers on Table 2.2, experimental evidence shows that the run-time of sub-system (4), which employs early classification, is 60% lower than the one of sub-system (3), which performs a detailed analysis of each heartbeat, while presenting a small memory overhead (32 kB in the case of RP with 16 coefficients) still affordable for the target WSN architecture. Moreover, all the dimensionality reduction techniques we proposed achieve savings in terms of duty cycle, even when the classification accuracy is low, i.e., in the case of FPD, showcasing the benefits of early classification.

2.3.7.4 Communication Savings

Computation and wireless communication are two major contributors to the power budget of embedded platforms, accounting for approximately 34% of the total energy consumption in a typical WSN, as shown in [10]. In addition to the reduction in computation requirements discussed in the previous section, the detection of pathological heartbeats contributes to obtain considerable gains in terms of energy efficiency, as the data to be transmitted can be greatly reduced.

In a scenario such as the one illustrated in Figure 2.12, where the WSN reports only the R peak of normal heartbeats and all the fiducial points in case of abnormality detection, the usage of the wireless link can be substantially reduced with respect to the case in which all the fiducial points of all the heartbeats are communicated. In the case of the proposed application (system 4 in Figure 2.12), where RP with 16 coefficients is used to perform feature extraction before classification, and considering all the heartbeats of the employed database (described in Table 2.1) as input signals, we achieve a 63% energy consumption reduction in the wireless module. Moreover, the proposed methodology results in a 60% reduction in the required computational effort and, consequently, energy employed for digital signal processing on an embedded microcontroller.

Overall, the proposed classification-based application in which only pathological heartbeats are analyzed can achieve an estimated 21% total energy reduction on a typical WSN platform, which includes acquisition, processing and wireless transmission, while still being able to report detailed information related to pathological heartbeats and meeting the strict real-time constraints of the domain.

Table 2.5 – Classification performance and number of parameters for different classifiers.

Classifier	NDR	ARR	Required operations
SVM-linear	93.4%	90.14%	Linear combination of the feature vector (16 elements)
SVM-gaussian	96.08%	96.93%	One norm and one gaussian function per SV (210 in total), and their linear combination
LDA	93.68%	90.47%	Linear combination of the feature vector (16 elements)
NFC	94.19%	96.1%	One gaussian function for each class-feature pair (48 in total) and their product aggregation

2.3.8 Comparison of Classification Methods

The flexibility of the proposed framework allows for different classification strategies to be employed after the dimensionality reduction stage. While the previous sections have focused on a neuro-fuzzy implementation, I herein evaluate the performance and computational cost of multiple popular alternatives, either based on Linear Discriminants Analysis (LDA) or on Support Vector Machines (SVMs) with linear or gaussian kernels.

SVMs separate test data into two classes (here, normal and abnormal heartbeats) through a hyperplane, whose coordinates maximize the separation between instances of the two classes in the training set. A linear SVM assumes that the elements belonging to different classes are linearly separable. Under this condition, classification is performed by pre-computing the separating hyperplane parameters, defined by its normal vector w and its offset from the origin b. At run-time, an input vector u is classified with a simple dot-product operation: $sign(w \cdot u + b)$. SVMs can be generalized to non-linear forms by applying the *kernel trick*, i.e., mapping the inputs in a high-dimensional space with a suitable non-linear kernel function, such as a gaussian radial function. Run-time classification using non-linear SVMs requires the evaluation of the input elements against a large number of support vectors (SVs), necessitating therefore a much higher workload to evaluate the kernel functions, and a higher memory footprint to store the support vectors. Finally, LDA classifiers utilize a linear combination of the input feature vector, to assign it to one of two classes. Under the assumption that the probability density functions of the classes are normally distributed with identical co-variance, classification is again performed by executing the dot-product of the feature vector with the normal of the separating hyperplane.

These classifiers are quantitatively compared in Table 2.5, which reports the achieved Normal Discard Rates and Abnormal Recognition Rates when performing heartbeat classification. For all the experiments, a 4-fold cross-validation has been performed, and the input feature vector is composed of an 8-coefficient random projection of the signal, plus the fiducial points. A minimum threshold of 95% is originally set on the ARR, and it is progressively reduced with a step of 1% when the training algorithm, based on the scheme depicted in Figure 2.10a, cannot converge to a solution. Table 2.5 also summarizes the main operations that are required to perform the classification of a given feature vector.

It can be observed that, on the one hand, linear classifiers (SVM-linear and LDA) have limited workload requirements, but on the other hand they incur in more mis-classifications, showing that the feature vectors are not well separated in a linear space. In particular, none of them is able to meet the minimum threshold of 95% set on the ARR. The SVM-gaussian strategy has better classification performance, but at a price of a memory footprint and a computational effort that is not compatible with a WSN implementation. In fact, 6KB of memory must be employed only to store the 210 support vectors, a computationally expensive square root operation is required to compute one norm for each SV, and the number of (linearized) gaussian functions for each heartbeat is an order of magnitude higher with respect to a neuro-fuzzy classifier. Experimental evidence previously detailed in Sections 2.3.7.2 and 2.3.7.3 shows that the proposed NFC-based classification scheme offers the best tradeoff between accuracy and complexity, achieving similar performance with respect to a SVM-gaussian, while having a small computation and memory overhead with respect to simpler linear strategies.

2.4 Proposed Energy-aware Distributed Wireless Body Sensor Network (WBSN) for Physical Activity Monitoring

Activity monitoring has been an active research field over the past years, and it finds application in a large variety of domains. These include medical applications, to assist patients affected by chronic conditions [70] [33], as well as personal monitoring during home and sport activities [9]. Recently proposed WBSNs have emerged as a promising technology to perform a more personal and fine-grained health monitoring. Thanks to a combination of wearable low-power WSNs that communicate through a wireless channel, these networks enable nonstop and unobtrusive analysis of subjects bio-signals, habits and environment, both for personal and medical purposes.

In the last few years, the versatility of these WBSNs have suffered a boost thanks to the ever-growing utilization of smartphones in our everyday life [27], which can provide advanced features such as data logging, transmission to a remote location, and user interface. In addition, the smartphone can become a dedicated high-performance node of the WBSN where to offload heavy processing workloads from the weaker WSNs. Stemming from this observation, herein I propose and study a Wireless Body Sensor Network (WBSN) for activity monitoring that combines wireless sensor nodes and a smartphone, in order to provide different tradeoffs between classification accuracy and transmission volume (which, in turn, has a major impact on energy consumption).

A large range of classification techniques for activity monitoring has been proposed in the literature [71] [9]. Among them, several techniques show low complexity – including *Naïve Bayes* classifiers, while more complex approaches employ algorithms with floating point operations, such as *nearest neighbor* and *support vector machines* (SVM). Other approaches based on simple feed-forward artificial neural networks [72] have also shown competitive accuracies on activity monitoring applications. The adaptation of such complex algorithms for resource constrained platforms has shown clear benefits for some applications such as heartbeat classification (see Section 2.3 and [57]) but typically lead to a non-negligible classification accuracy loss when multi-modal signals are combined, thus making the porting to wearable devices impractical. Therefore, in this work I propose two different approaches (later detailed in Section 2.4.2):

- In an accuracy-oriented configuration, the nodes of the proposed network send their extracted features to the smartphone, which performs a precise and complex classification thanks to its high computational capabilities.
- In a transmission-aware configuration, one of the wearable nodes is in charge of performing the classification and sending the result to the smartphone, which acts only as a gateway toward the user.

2.4.1 State-of-the-art and Challenges

An extensive survey of the literature related to activity monitoring is available in [9], which also provides a quantitative comparison of the accuracy of the existing solutions. In particular, the work in [72] reaches the highest classification accuracy (95%) by employing an artificial neural network that processes acceleration data coming from the wrist. Although the authors successfully manage to employ only accelerometer data, the classification is performed offline, and no indication regarding how the proposed technique would behave on portable devices with limited resources is provided. Conversely, in [73], the authors propose an online monitoring technique using a watch-like sensor on the wrist, which acquires data from accelerometers, a microphone and light sensors. The system employs feature extraction both in the time and in the frequency domain and a nearest neighbors classifier reaching an accuracy of approximately 91% but only on arm-related movements.

The work in [32] proposes a network to monitor general-case activities (similar to the ones targeted in this work) using a hidden Markov model, which acts on data coming from sensors on shoulder, chest and wrist. The proposed classification achieves accuracies of up to 90%, but the confusion matrices show pronounced difficulties in discriminating activities where the upper body is static (e.g., sitting and standing). In [34], the authors propose a system for online monitoring focused on activity changes. The network is able to perform most of the computations on a node located on the chest, and to this end it employs a computationally simpler classifier based on a decision tree. The proposed methodology, combined with a set of custom features, leads to an overall accuracy of 90.8%. However, although the system achieves peaks of accuracy when detecting transitions, the performance while monitoring ongoing activity shows high classification degradations.

One of the most recent works in the field of activity monitoring is discussed in [33], where a system for monitoring patients affected by Parkinson's disease is discussed. The authors propose a power saving technique that reduces sampling frequency during static activities, such as lying and sitting. They reduce the sampling rate down to 30 Hz during sleep periods while requiring high sampling frequency (up to 200 Hz) during most of the other activities. Although this policy proves to be effective in the target scenario, where the patients are spending the majority of their time on static activities, in a general case it is not optimal. In addition it requires a subject-dependent training.

Multiple conclusions can be derived from this analysis, regarding both architectural and methodological aspects. First, a large variety of node placements has been proposed over the years, including wrist, upper arm, chest, hip, thigh, crus, ankle, and several combinations of them. However, not all the existing solutions aim at minimizing the patient's discomfort, often purposely increasing the number of sensors to collect a larger amount of data. Second, common patterns can be identified to increase the system lifetime: on-node feature extraction – which heavily reduces data transmission– is applied in most circumstances, and efforts to reduce the sampling frequency can also be found. However, most of the existing works still employ sampling rates of more than 100 Hz because of the dynamic nature of activity recognition, and this has a negative impact on the system power consumption due to the sensing circuit. Finally, most of the existing classification techniques fail to reach accuracies above 90%, and the best results are achieved by sensors on the chest and on highly-dynamic joints (such as wrists and ankles), the latter creating major discomfort during everyday activities. In this work, we aim at minimizing the invasiveness of the network. In fact, while some requirements of the proposed system are related to its confort and usage,the implementation of such a WBSN entails several design challenges:

- The distribution of sensors over the body imposes the necessity of transmitting the sampled data throughout a network. A proper selection of the data to be sent requires an exhaustive study of the amount of processing that can be performed at each node and to determine the most relevant parameters that need to be employed instead of the raw signal.

- In addition, the studied kinetic signals (e.g. accelerometers or gyroscope data) usually need to capture body movements that typically impose a high sampling frequency. The nature of these signals will directly impact the computational and memory requirements of the algorithm which usually become the main obstacle to perform embedded advanced processing on WBSN nodes.

- As a consequence, the accuracy of the activity recognition can be affected not only by the chosen classification technique but also by the sensed signal and the extracted parameters on node. Moreover, as previously introduced, state-of-the-art classifiers are not suitable to be executed on-node without loss of quality in the classification. Thus, a proper selection and optimization of the classification scheme is needed possibly leveraging the computational power offered by the smartphone at the cost of high data transmission from the WBSN.

2.4.2 Target System Definition

The proposed WBSN for activity monitoring is illustrated in Figure 2.14 and it aims at detecting seven different activities of clinical relevance:

- Walking (*Wa*): moving in any direction at a slow to normal speed.
- Sitting (*Si*): resting in a sitting position with or without moving the upper half of the body.
- Standing (*St*): resting in a vertical orientation without moving in any direction.
- Laying (*La*): resting in a horizontal orientation.
- Running (*Ru*): moving in any direction at a moderate to high speed.
- Walking upstairs (*Up*).
- Walking downstairs (*Dw*).

The network is composed of several wearable devices deployed throughout the body of the subject, in order to sense data related to acceleration, orientation, and optionally other bio-signals of general interest, such as the ECG. In addition, a smartphone is also incorporated in the network to provide higher computation capabilities that can be exploited either during the activity detection, or for enhanced high-level functionalities. The workload balance among the devices has an impact on the quality of the classification and on the amount of transmitted data, which in turn plays a major role for the overall energy consumption. Two tradeoffs between accuracy and transmission are identified for the proposed system, and they are discussed in depth in the following.

2.4.2.1 Device Taxonomy

As previously introduced, the proposed WBSN includes two classes of devices, each one characterized by its different computation capabilities, size and energy budget. In particular, the network is composed of:

- A set of **wearable sensor nodes**, which are in charge of sampling the signals of interest by incorporating accelerometers, gyroscopes, and optionally other sensors. These devices are battery-powered and are required to be small to minimize the subject's discomfort: the combination of these requirements leads to a limited energy budget, which translates into low-power systems with limited computation capabilities. In fact, wearable nodes typically feature very basic microcontroller architectures with integer arithmetic modules and modest memory resources, thus imposing significant limits on the complexity of the algorithms they can execute. It is important to note that, the embedded radio component which enables wireless communication within the WBSN

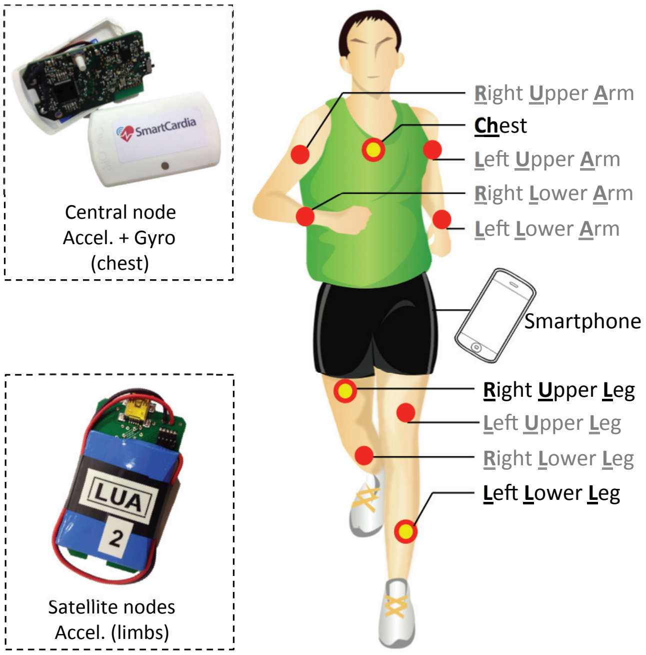


Figure 2.14 – Overview of the proposed WBSN. The three nodes in yellow are the ones belonging to the final configuration (see Section 2.4.5.3), whereas the remaining nodes have only been used for data collection purposes

typically represents the most energy-hungry hardware of the system [8] in systems as the targeted one where a high volume of traffic is required.

- A **smartphone**, which provides higher computational capabilities due to more complex processing cores, and larger batteries that are generally recharged on a daily basis. These devices typically include hardware support for floating point operations, larger volatile and non-volatile memories, and multiple wireless interfaces. As a consequence, these devices are ideal for executing complex and accurate classification algorithms, as well as high-level operations, e.g., interfacing with the user, logging the recorded data, and possibly communicating this data to a remote service over the internet.

The two kinds of devices have different roles in the WBSN. In particular, the classification can be performed either on a selected wearable node, which collects the data from the other sensors and then executes the classification algorithm, or by the smartphone, which receives the data from all the sensor nodes. These two strategies generate different traffic volumes through the WBSN and, while the former approach is only applicable when the classification algorithm is relatively simple, the latter provides a more general solution that allows to implement complex classifiers with higher precision.

2.4.2.2 Network Topology and Task Assignments

In the proposed WBSN, several wearable nodes are placed on the body, as shown in Figure 2.14: two nodes including accelerometers are located in each limb, and a more complex node that also includes gyroscopes is located on the chest. The node on the chest can be optionally used to sense and analyze the ECG signal [74, 45], thus extending the functionality of the network.

The aim of this work is obtaining a high classification accuracy deploying a reasonably small network. The size of the WBSN is reduced by identifying a subset of nodes that allows to obtain minimal loss of accuracy and discomfort. In particular, the position of the nodes in the final WBSN is selected after a detailed analysis of multiple possible combinations, which is reported in Section 2.4.5.3.

The structure of the communication among the nodes, as well as the workload distribution throughout the network, is dictated by a tradeoff between the volume of transmitted data and the final classification accuracy, and can possibly be changed dynamically. A network with localized workload would send a lower amount of data through the wireless channel, but most of the computation would be performed with lower precision on microcontrollers with limited resources. In order to tackle this tradeoff, we propose two alternative methods, which are illustrated in Figure 2.15. The two strategies are defined as follows:

- **Smartphone-based classification** (Figure 2.15a): in this configuration, each wearable node performs a local feature extraction on the data, and then sends the resulting information to the smartphone, which acts as the center of a star topology. Then, the smartphone performs a complex real-time classification on its CPU, enabled by the floating point support and the larger amount of resources available on the device. This configuration allows a more accurate classification, but also requires all the nodes to transmit their data, thus generating a high traffic volume;

- **On-node classification** (Figure 2.15b): in this configuration, a selected wearable node is in charge of collecting data from the other devices, performing a local classification, and then sending the result to the smartphone only when a change of activity is detected (periodic packets are also exchanged between the node and the smartphone to probe the connectivity). However, the on-node classification is less accurate than the classification performed on the smartphone, because of the limited computational resources. In the proposed network, the node on the chest is selected as the center of the network, as it can be assumed to have a clearer transmission path to the smartphone with respect to the devices on the leg. Moreover this node is responsible of sensing a higher number of channels (i.e., accelerometers and gyroscope) and selecting it as the center of classification reduces the amount of data to be transmitted within the network. Finally, as the node on the chest is worn over a cardio belt, its size and its battery can be enlarged to ensure a sufficient lifetime in spite of the higher workload.

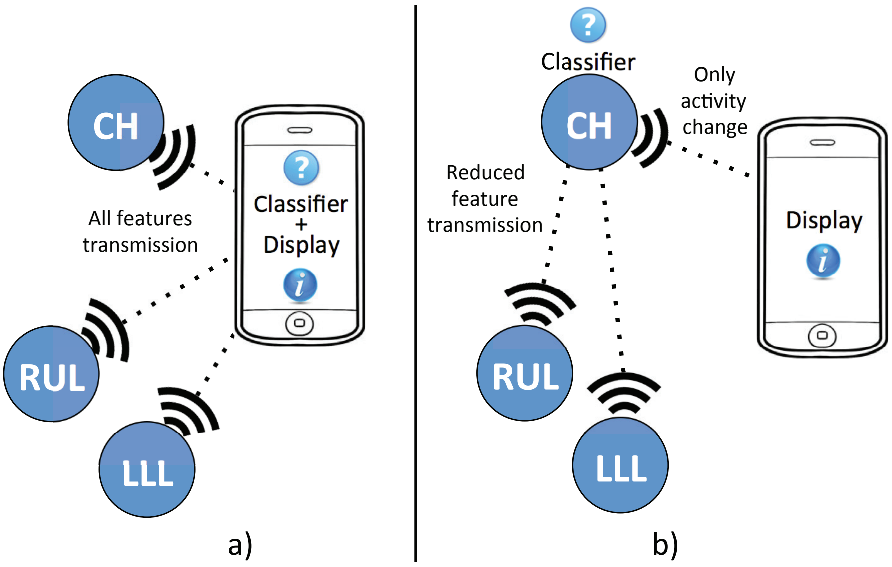


Figure 2.15 – The two proposed classification strategies: smartphone-based (a), and on-node (b). In both scenarios, the phone acts as network coordinator.

2.4.3 On-node Feature Extraction

In both strategies defined in the previous section, the wearable nodes are in charge of performing feature extraction on the sensed signal. This is a common technique [9] to avoid the transmission of the entire signal, which would not be feasible for small battery-powered devices. After acquiring the samples from the sensor, each wearable node extracts relevant information from the raw data stream using the embedded microcontroller. This strategy also has a positive effect on the classification because of two reasons. First, the classifier can act on data with a low degree of redundancy and second, the size of the classification problem is effectively reduced following the same rationale as in the case of heartbeat classification introduced in Section 2.3.

Multiple features can be computed on a wearable node, each one conveying different information, and requiring different computational efforts to be extracted. In the literature [9], features are typically classified into four groups:

- *Time-domain features*, which include mean, standard deviation, median, percentiles, derivatives, zero crossings, and many others. They are extracted directly from the signal, and provide relevant information about its waveform and its statistical behavior. Their computation complexity is linear with respect to the signal size, thus making them ideal for low-power on-node extraction;
- *Frequency-domain features*, which can be derived from the fast Fourier transform (FFT) of the signal, and include the spectral energy and entropy, the principal frequency energy, and a selection of the first n coefficients of the FFT. Their complexity is superlinear;

- *Wavelet features*, which include the coefficients of the wavelet transform of the signal, and can detect variations in the frequency components over time;
- *Custom features*, which are derived *ad hoc* from domain-specific considerations about the target problem. In the context of activity monitoring, these features include signal magnitude area [34], inter-axis correlation [75], and time-domain gait detection [76].

A quantitative comparison of different feature sets is provided in [71], though the reference classifier is different from the ones employed in the proposed WBSN. Results show that, while frequency-domain features lead to the best classification accuracy, time-domain features provide the most intriguing trade-off between accuracy and complexity. Conversely, wavelet features provide good results when detecting transitions among different activities [9], but they do not perform well overall [71]. In the proposed WBSN, we extract two time-domain features for each axis of the accelerometers and the gyroscope: mean and standard deviation. These features, along with the aforementioned low computational effort, lead to the high classification accuracy as later shown in Section 2.4.5.3.

In order to derive the features from the stream of samples, a segmentation based on the common sliding window technique [77] is included on each node. The selected time-domain features are then extracted over a set of L consecutive samples, which form a window, before moving to the next set of samples, which may partially overlap with the previous window. The overlapping between two consecutive windows determines the time that elapses between the production of two features, and hence the responsiveness of the WBSN.

Finally, it is important to point out that no filtering is applied on the sensed data. Even though the samples are affected by high-frequency noise, the hardware-level low-pass filtering inherently applied by the sensor has been shown to be sufficient for classification purposes [78], without explicit data filtering via digital signal processing on the resource-constrained microcontroller.

2.4.4 Classification Framework

The classification algorithm is in charge of estimating the current activity, starting from the features transmitted by the nodes. As mentioned in Section 2.4.2, the complexity and the accuracy of the algorithm is influenced by the execution environment, namely, by the availability of hardware support for floating point arithmetic and by the memory size of the target device.

In the proposed WBSN, we advocate two separate classifications techniques, depending on the device that performs the classification. In the context of smartphone-based classification, a *neuro-fuzzy classifier* (NFC) is employed. Thanks to a set of well-established training techniques [57], an NFC can handle the number of classes required for activity monitoring applications with high accuracy, even when subject-independent training is employed. In addition, an NFC has a simpler structure with respect to most of the mentioned classifiers, which translates into lower computational complexity and memory requirements, making it suitable for mobile applications.

2.4.4.1 On-Node Classification

Similarly to many floating point classifiers, NFCs also show a perceivable loss of accuracy when integer approximations are used [57]. For this reason, in the case of on-node classification, we introduce a simplified algorithm based on a *decision tree*. The decision tree is a set of thresholding rules that discriminate activities based on the value of one feature at a time. If the tree is properly balanced, the complexity of this technique is logarithmic in the number of classes, it has limited memory requirements, and it can be efficiently implemented in a microcontroller without floating point support. In addition, the rules that we derive for activity monitoring are sufficiently general to ensure that the same decision tree can maintain its performance across different subjects.

In particular, in the proposed on-node classification configuration, a simplified decision tree is used to classify the current activity on the microcontroller of the node located on the chest. The node then sends a notification to the mobile phone only when the activity changes, thus greatly reducing data transmission. The proposed decision tree is based on an analysis of the parameters that are expected to discriminate different activities, and its structure is summarized in Figure 2.16. For example, the variance of the accelerometer data coming from the x-axis on the crus can discriminate static activities (sitting, standing and laying) from dynamic ones: a high value of the standard deviation indicates a movement on that axis, thus indicating a dynamic activity. It can be observed that, excluding the features that are directly sensed on the chest, the nodes on the legs are only required to transmit three features (the standard deviation of the x-axis of the crus, the mean of the z-axis of the crus, and the mean of the x-axis of the thigh), thus greatly reducing the transmission workload.

The selection of these features and the thresholds have been determined by analyzing the data of three different subjects and after studying the best configuration of the WBSN as later shown in Section 2.4.5.3. Moreover, because of the general nature of the rules included in the tree, this on-node classification has proven to be suitable for cross-subject utilization.

2.4.4.2 Smartphone-Based Classification

In the proposed smartphone-based classification, the handheld device is in charge of receiving the features from all the wearable nodes, and assign the current activity to one of the 7 target classes. The decision is taken using a neuro-fuzzy classifier similar to the one used for heartbeat classification in Section 2.3, which follows the feed-forward structure shown in Figure 2.17.

As shown in the figure, this NFC structure is composed of the standard three layers. The first

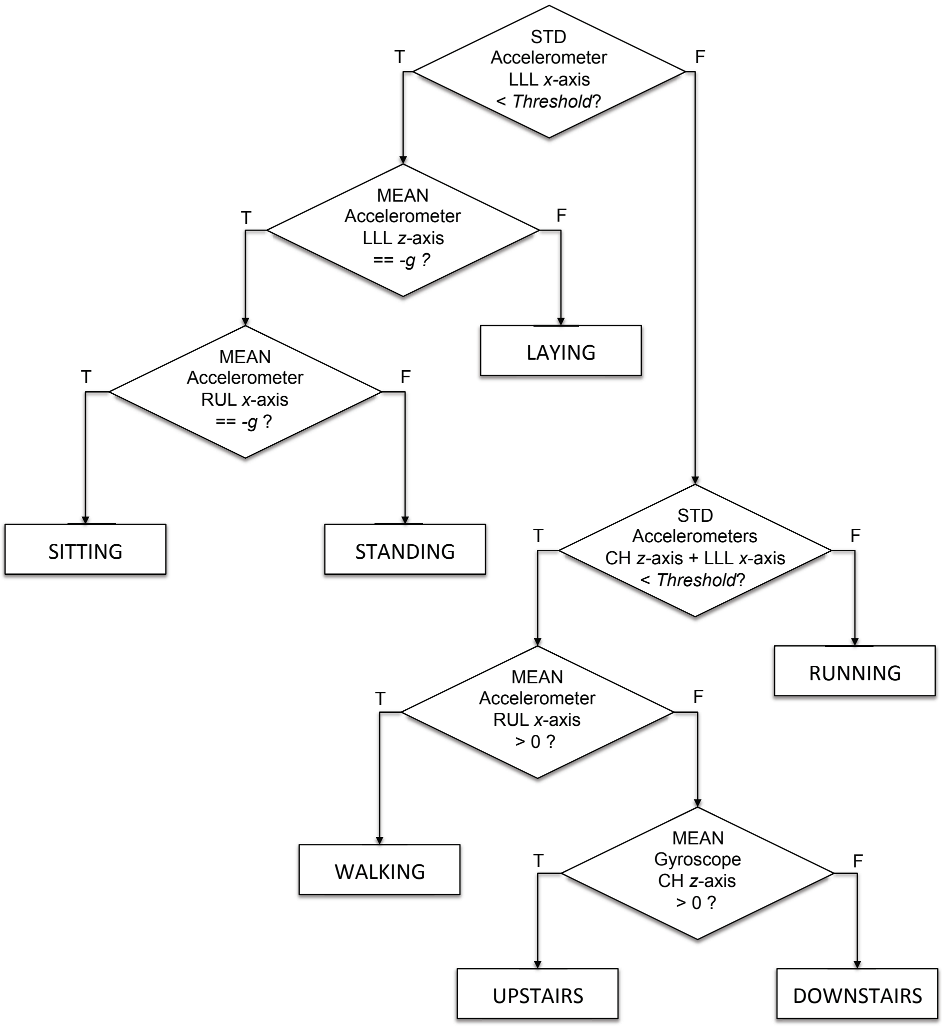

STD Accelerometer LLL *x*-axis < *Threshold*?
T
F
MEAN Accelerometer LLL *z*-axis == -*g* ?
MEAN Accelerometer RUL *x*-axis == -*g* ?
LAYING
SITTING
STANDING
STD Accelerometers CH *z*-axis + LLL *x*-axis < *Threshold*?
RUNNING
MEAN Accelerometer RUL *x*-axis > 0 ?
WALKING
MEAN Gyroscope CH *z*-axis > 0 ?
UPSTAIRS
DOWNSTAIRS

Figure 2.16 – Structure of the proposed tree for on-node classification

membership layer takes as input the features extracted by the nodes, and for each feature computes a membership grade for each of the activities. In particular, the grade $\gamma_{i,k}$ for the ith feature and class k is defined according to a membership function $MF_{i,k}$, i.e., a gaussian distribution with a mean value $\mu_{i,k}$ and a standard deviation $\sigma_{i,k}$. These statistical parameters are determined during the training phase, and at run-time these values are employed to compute the membership grade $\gamma_{i,k}$

$$\gamma_{i,k}(x_i) = \mathrm{MF}_{i,k}(x_i) = \exp\left(\frac{-(x_i - \mu_{i,k})^2}{2\sigma_{i,k}^2}\right), \tag{2.17}$$

where x_i denotes the value of the ith feature. In the subsequent *fuzzification layer,* the membership grades of the same class are multiplied as:

$$f_k = \prod_i \gamma_{i,k} \tag{2.18}$$

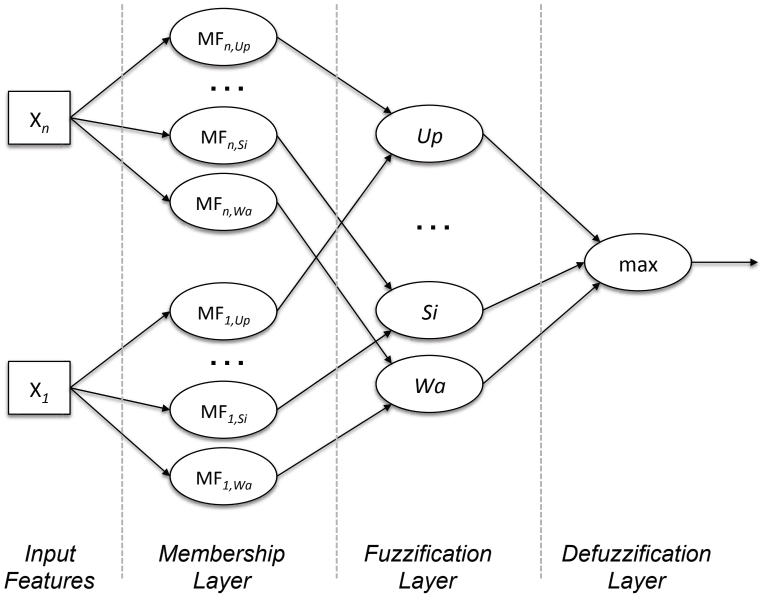


Figure 2.17 – Structure of the proposed neuro-fuzzy classifier

The resulting value represents how likely the activity belongs to class k. In the third *defuzzification layer*, the maximum among the different values of f_k is selected, and the activity is classified accordingly.

2.4.5 Experimental Evaluation

The proposed WBSN has been implemented and tested on three subjects for training and testing purposes. As discussed previously, the proposed network is able to generate a new activity classification every 2 s, and able to display the output on the smartphone thanks to an application developed as user interface.

2.4.5.1 Experimental Set-up

The proposed WBSN is implemented using two different kinds of custom nodes: one targeting the chest, and one targeting the limbs. All the nodes in the network embed a *Jenic JN5148* microcontroller [79] specifically designed for *ZigBee* applications, and comprising the IEEE 802.14.4 transceiver [80], 128 kB of ROM, and 128 kB of RAM. The node on the chest additionally includes an *LSM330DLC* inertial module by STMicroelectronics [81], which features a triaxial accelerometer and gyroscope. The nodes targeting the limbs, on the other hand, include an *LIS3LV02DQ* inertial sensor by STMicroelectronics [82], which only features a triaxial accelerometer. Finally, the mobile phone is a mid-range Sony Xperia U running Android 4.1.2, which interfaces towards the ZigBee network by means of a custom dongle, which also embeds a Jenic chip and a serial converter by FTDI.

The accelerometer and gyroscope sensors were configured to sample at a frequency of 50 Hz for any activity being monitored, which is a major improvement with respect to the state of the art [33]. In order to perform segmentation with the sliding window technique, the nodes use part of the available RAM to store windows of 4 s, with an overlap of 2 s. The window length was selected to store a large number of samples, thus avoiding outliers and inconsistencies, as the target feature is expected to have a smooth behavior over such a long time interval. This overlap determines the responsiveness of the network: in the proposed implementation, a new estimation of the activity is produced every 2 s, which fits the slow dynamics of human movements.

In order to perform the offline training of the NFC, a traditional scaled conjugate gradient method has been employed [62]. The training has been performed on a high-performance workstation machine using data collected from three different subjects. This strategy ensures a high accuracy without requiring a personalized training phase, and makes the WBSN robust when operating in conditions that differ from the training setup.

2.4.5.2 Data Collection Network

The aforementioned offline training of the classification algorithms requires the collection of a large amount of data from different subjects, while they perform multiple activities. This step is also needed to evaluate the goodness of the selected feature set, and the most suitable node placement on the subject's body. In order to perform these analyses, data needs to be collected with no on-node manipulation, and many different node positions have to be evaluated at the same time.

The data collection network, which is more invasive than the final system, includes 9 wearable nodes: two for each arm (upper arm and forearm), two for each leg (thigh and crus), and one on the chest, as shown in Figure 2.14. Each node collects the data from the available sensors – i.e., accelerometers on the limbs, accelerometer and gyroscope on the chest–, and sends it to an external network coordinator connected to a workstation.

The main design challenge in the data collection network is handling the high throughput generated by the nodes. In fact, ZigBee natively supports moderate data rates of up to 250 kbps, but in practice the overall network throughput is heavily reduced by contention, interference and data framing [83]. As a consequence, in order to avoid dropped packets, each node is allowed to buffer the data up to its maximum memory capacity, and then is synchronized by the central coordinator to be the only transmitting node while flushing its buffer contents, thus emulating a collision-free burst transmission. As an additional measure to avoid bottlenecks and data losses, the data gathering was performed in a controlled environment and during short sessions (approximately of one minute), in which one activity at a time is monitored.

The data collection phase has been iterated on multiple subjects to gather the samples used for training and cross-validation of the proposed classifiers. The corresponding results are reported in the following section.

2.4.5.3 Classification Accuracy

We first investigate the accuracy of the proposed classification methodologies, and we motivate the selected node placement, in order to assess the performance gap between the wearable nodes and the smartphone, which is due to the higher computational capabilities of the latter. The accuracy results of the proposed neuro-fuzzy classifier, implemented on the smartphone using floating point precision, are reported in Figure 2.18. The accuracy was determined using an 8-fold cross validation process over the data coming from three different subjects, gathered using the collection network discussed in Section 2.4.5.2. Thanks to the availability of data from the limbs and the chest, we have performed an exhaustive comparison among the possible node placements. As explained in Section 2.4.5.2, 9 nodes have been used to perform the data collection. As a result, up to 512 configurations, including one or several nodes, are possible. However, given the nature of the target activities, which are equally influenced by movements in both right and left sides of the body, this design space can be drastically reduced by analyzing the characteristics of the acquired signals and discarding those configurations that present symmetrical sources (i.e. similar data from a node and the equivalent one on the opposite side). For instance, a configuration presenting nodes placed on the right and left upper arms (RUA+LUA) provides a similar classification quality that the configuration presenting one of them. In addition, mirrored configurations (e.g., LLL+RUL and RLL+LUL) provide similar performances and therefore considering only one of them is sufficient.

Figure 2.18 shows the classification quality of the considered configurations that achieve more than 85% accuracy. According to the depicted data, the configuration that achieves the best results with a limited number of nodes is the one comprising the chest, and two sensors located on the right thigh and on the left crus. This result reflects the ability of the nodes on the legs to discriminate among static activities, and the ability of the node on the chest to distinguish among dynamic ones. Overall, the proposed WBSN effectively exploits the information coming from the three sensors to achieve an accuracy of 97.2%, thus improving the results of [72] (95%), which is based on a similar classification method and is currently the most accurate work in the literature.

Table 2.6 shows the row-normalized confusion matrix for the NFC. As expected, most of the activities are correctly predicted. The main source of error is the distinction between walking upstairs and downstairs: this can be explained by the similar mechanics of the two activities, which are discriminated by the classifier relying only on the orientation of the chest.

On the other hand, the node-based classification (using the decision tree discussed in Section 2.4.4.1) achieves lower accuracy, which is caused by the limited computational capabilities. In particular, the proposed classifier achieves an accuracy of 88% with the three selected nodes, a result that is comparable to the existing approaches based on decision trees [9]. Although the on-node classification is 9.2% less accurate than the NFC running on the smartphone, it

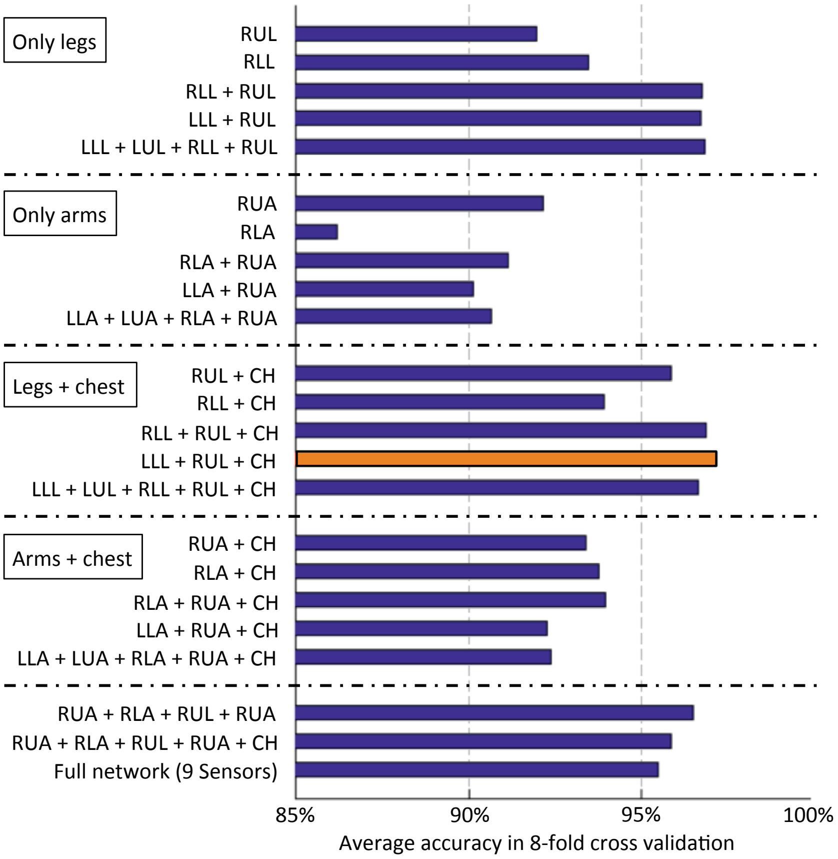


Figure 2.18 – Classification accuracies of the proposed NFC for smartphone-based classification, with multiple combinations of node placements

requires a considerably lower overhead in terms of power consumption and data transmission, as discussed in the following section.

2.4.5.4 Transmission Volume Reduction

While the smartphone-based classification is the best option from the accuracy point of view, the required transmission bandwidth of the on-node classification alternative makes it more competitive when analyzed from an energy efficiency perspective. In order to study the transmission-accuracy tradeoff, a quantitative analysis of the bandwidth required by the on-node and the smartphone-based classification strategies is presented in this section. In both configurations the traditional feature extraction provides a massive reduction of data transmission with respect to raw data streaming, which can be estimated to be approximately 98.4% with the selected sampling frequency (50 Hz). On top of this, significant savings can be achieved by employing on-node classification, and by accepting its lower accuracy. For the sake of illustration, the reported bandwidth estimations do not account for the packet overhead introduced by the underlying transmission protocol.

Table 2.6 – Confusion matrix of the proposed Neuro-Fuzzy Classifier (The values are row-normalized)

	Dw	Up	Si	La	St	Wa	Ru
Downstairs (*Dw*)	88.7	4.1	1.0	4.1	0.0	2.1	0.0
Upstairs (*Up*)	3.4	95.5	0.0	0.0	0.0	1.1	0.0
Sitting (*Si*)	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Laying (*La*)	0.0	0.0	0.0	100.0	0.0	0.0	0.0
Standing (*St*)	0.0	0.0	0.0	0.0	100.0	0.0	0.0
Walking (*Wa*)	3.3	0.0	0.0	0.0	0.0	96.7	0.0
Running (*Ru*)	0.0	0.0	0.0	0.0	0.0	0.0	100.0

On the one hand, in the proposed smartphone-based classification, each of the nodes sends a new set of features to the smartphone with a rate of ρ transmissions per second (in the proposed WBSN, $\rho = 1/2$, i.e., a new set every 2 s). Each of the features is encoded with 2 Bytes. Hence, a total of 48 Bytes is required to encode all the 24 transmitted features, (i.e., the mean and the variance of the three axes of all the accelerometers, plus the mean and variance of three axes of the gyroscope on the chest). As a consequence, the required bandwidth for the proposed network is 48 Bytes$\cdot\rho$ =24 Bytes/s.

In the proposed on-node configuration, on the other hand, only the two sensors on the legs are sending features to the node on the chest, and only three of these features are required (see Figure 2.16): the standard deviation of the x-axis of the crus, the mean of the z-axis of the crus, and the mean of the x-axis of the thigh, totaling to 6 Bytes. The node on the chest is in charge of transmitting the following data to the smartphone:

- One byte to communicate a new activity, every time a change is detected. This contribution to the overall bandwidth can be further divided into two parts: the first one is due to the actual number of times the subject changes activity during the observation period (let us denote this number by Δ). The second contribution is due to the situations when the classifier incorrectly predicts the current activity: in the worst case, this requires a transmission to communicate the erroneous activity, and a second one when the prediction is corrected. Let us denote the accuracy of the on-node classifier by α_{node} (in the proposed network, it is equal to 88%). The traffic generated due to erroneous classification can thus be estimated as $2(1-\alpha_{node})\rho$ Bytes/s;
- A periodic probe message (1 Byte) to check whether the connection is alive. We denote this overhead traffic by Ω (measured in Bytes/s).

Overall, the number of packets transmitted in the on-node classification scenario is equal to:

$$B_{on-node} = 6\cdot\rho + \Delta + 2\cdot(1-\alpha_{node})\cdot\rho + \Omega \text{ [Bytes/s]} \tag{2.19}$$

Assuming that the actual activity changes (Δ) are in the order of ten times per hour (which is realistic, considering that sleep periods are also included in the average), and that the overhead (Ω) was unfavorably assumed to be in the order of 1 packet per minute, the contribution of the mispredicted activities dominates the traffic between the node on the chest and the smartphone. Still, according to this worst-case estimation, the global network traffic is equal to approximately 3.14 Bytes/s, i.e., 86.9% savings with respect to the more accurate smartphone-based classification.

2.5 Summary and Concluding Remarks

In this chapter I have introduced two complementary strategies to improve energy efficiency of state-of-the-art applications executing in WBSN-based biomedical monitors. On the one hand, at the sensor node level, I have proposed the utilization of a heartbeat classifier to perform selective advanced DSP in ECG single-node biomedical monitors. The classifier only activates the costly DSP analysis routines in case of detecting an abnormality in the heartbeat morphologies. The proposed implementation consists of a lightweight, heartbeat neuro-fuzzy classifier coupled with a feature extraction technique based on *Random Projections*. Experimental results show that the accuracy of the proposed classifier when identifying abnormalities can reach up to 98.9% keeping a low rate of mis-classifications. With respect to a typical system that is continuously performing DSP analysis, my proposed approach can reduce the duty cycle and transmission volume of an ECG biomedical monitor by up to 60% and 64%, respectively.

On the other hand, at the sensor network level, I have proposed an energy- and transmission-aware WBSN devoted to the identification of physical activities. The system is composed of several nodes deployed throughout the body of the subject and interfaced with a smartphone. In particular, two classification schemes, which trade accuracy for transmission volume, are proposed. First, the highly accurate smartphone-centric alternative is based on an NFC that exploits the high computational resources available on the mobile phone. Second, the transmission-aware scheme performs on-node classification employing a decision tree and minimizing the data transmission. According to the experimental results, the high-precision classification reaches 97.2% accuracy while the on-node option reduces the transmission volume by up to 86% with a small classification degradation, leading to a 88% accuracy.

3 Synchronization-Based Ultra-Low Power Multi-Core Architectures

3.1 Introduction

Recent advances in embedded bio-signal analysis have changed the landscape of health monitoring applications, allowing for continuous digital signal processing (DSP) directly on low-power Wireless Body Sensor Nodes (WBSNs) [15]. In addition to acquisition and wireless transmission of sampled data, state-of-the-art WBSNs embed advanced real-time applications, able to automatically retrieve relevant diagnostic data such as the analysis of respiration or heart rhythm [16] and the detection of epileptic seizures [17].

Energy efficiency is a fundamental aspect of this portable autonomous systems devoted to perform bio-signal analysis, where a considerable amount of processing is performed with limited energy supplies. An effective technique to achieve computational power savings is supply voltage scaling, all the way to the sub-threshold region. In the literature, voltage scaling has been extensively analyzed, including its limitations and disadvantages [21] [22] [23].

One of the main issues with low-voltage operation is performance degradation, which can limit the degree of achievable voltage scaling for a given processing requirement. Parallel computing using multiple cores can alleviate this issue, provided that applications can be parallelized. To this end, in [84] near threshold low-power multi- and single-core architectures are compared in terms of power and performance ability for several multi-channel bio-signal processing applications. It has been shown that the multi-core approach achieves better energy efficiency compared to the single-core approach for medium and high workloads [85].

3.1.1 Parallelism in Bio-signal Processing Applications

Bio-signal analysis applications consist mostly of moderately complex sequences of arithmetic manipulations on single- or multi-input biological signals. As shown in [10] and [12], this advanced signal processing can be carefully optimized to run in real-time on typical embedded low-power micro-controllers. The analysis of these multiple-input signals highlights considerable parallel computation opportunities, which can be exploited on multi-core processing platforms in conjunction with low voltage operation to achieve energy savings.

First, DSP algorithms applied over multiple streams of data can be parallelized to execute in low-power multi-core platforms by performing the processing of each of the streams in a different core. This strategy follows the single-instruction multiple-data (SIMD) paradigm and has been previously studied in the literature [84], [86]. However, this technique offers good energy savings only when executing purely parallel applications (i.e. without conditional segments of code).

Second, software pipelining of different algorithmic phases can be exploited by executing each of the phases in one or many cores. As in the SIMD case, this arrangement divides the application workload into different cores, thus reducing the required system clock frequency. Nevertheless, this workload division requires efficient core-to-core notification mechanisms to properly manage producer-consumer relationships in order to avoid costly active waitings or imprecise periodic polling.

3.1.2 Contributions and Outline of this Chapter

In this chapter I first present a low-power multi-core architecture featuring a hybrid hardware/-software synchronization technique that allows maximizing SIMD execution while executing multi-channel parallel applications. This architecture is the result of a research collaboration between Jeremy Constantin from the Telecommunications Systems Laboratory (EPFL, Switzerland), Ahmed Dogan from the Embedded Systems Laboratory (EPFL, Switzerland) and the author of this thesis, Rubén Braojos. More precisely, the proposed architecture is able to recover synchronization after data-dependent branches by forcing the cores to wait for others in order to continue executing in lock-step. In the second part of this chapter, I generalize the synchronization technique in order to support any existing bio-signal processing application. To this end, the first proposed architecture is minimally modified and a new synchronization mechanism to efficiently support producer-consumer relationships is implemented. In particular the key contributions of this chapter are:

Synchronization-based multi-core architecture for parallel bio-signal processing:

- I present a low-power architecture featuring a novel synchronization methodology based on the insertion of dedicated instructions to recover lock-step execution among cores after diverging in data-dependent segments of code.

- I describe the required hardware and software support, namely a full-custom lightweight hardware synchronizer and a dedicated instruction set extension of the processing cores.

- I detail the mechanism employed to achieve the re-synchronization after execution of conditional segments of code.

- The obtained results show that the target low-power multi-core architecture achieves high energy savings of up to 38% while only increasing its area footprint by 2%.

Advanced synchronization technique for arbitrarily parallel bio-signal DSP applications:

- I propose a generalization of the previously mentioned synchronization technique in order to efficiently support the management of producer-consumer relationships among cores.
- I describe the necessary steps to adapt any existing bio-signal processing application to adopt the proposed synchronization strategy.
- I study the performance and power consumption of three different state-of-the-art benchmarks from the field of embedded ECG processing, which represent real-world applications with different workloads and runtime characteristics.
- The experimental evaluation shows that the target low-power multi-core architecture employing the proposed advanced synchronization technique can obtain up to 40% energy savings while executing the studied benchmarks.

The remaining of this chapter is organized as follows. First, Section 3.2 reviews the main efforts in the field of low-power embedded singe-core and parallel multi-core architectures. It gives a detailed description of the MIMD parallel architecture proposed by [84] and the TamaRISC core, which are the base of the work presented in this chapter. Then in Section 3.3, I propose a low-power multi-core platform featuring a synchronization technique that provides high energy efficiency while executing multi-channel bio-signal applications. Afterwards, in Section 3.4, I present a generalization of the synchronization technique that allows efficient execution of any DSP application regardless off its degree of parallelism. Finally, Section 3.5, summarizes the content of this chapter and discusses the possible improvements that can be explored to make the proposed architectures more energy efficient.

3.2 Low-Power Single- and Multi-Core Architectures for Bio-Signal Processing: State of the Art

In this section I provide a summary of the most relevant works in the field of low-power architectures devoted to embedded processing, highlighting those contributions that focus on performing energy-efficient bio-signal processing.

3.2.1 Low-Power Single-Core Architectures for Bio-Signal Analysis

In the field of low-power embedded architectures, several different strategies have been proposed to reduce the consumption of the proposed systems. These adopted strategies can be classified according to the flexibility of the platforms, which in general is traded for a higher energy efficiency. For instance, application-specific integrated circuits (ASICs) have been widely proposed for some specific bio-signal processing applications providing a very low energy consumption [18] at the price of a very limited applicability. On the opposite side, general purpose processors for embedded processing (e.g. ARM-based architectures [87]) are commonly used to implement portable and handheld monitoring devices due to their flexibility. However, due to the low energy efficiency of these architectures, such devices are frequently employed during short periods of time.

While ASICs have been integrated in many proposed platforms for health monitoring and specially in the field of ECG processing [88] [89] [90] [91], these options do not provide any programability or configurability, reducing the field of application to the specific task for which they were designed, and therefore making them not suitable for more general biomedical monitors, which need to cover an extensive diversity of bio-signal processing applications. Some of the routines present in these applications have been optimized to be executed in accelerators that are integrated in hybrid systems in which a more general processing unit is employed [92] [93] [94] [38], providing a higher versatility. Such systems can be composed of several of these accelerators interfaced with a programable processing unit that can execute the tasks that cannot be performed by the dedicated hardware. For instance, in [18], the proposed heterogenous system can achieve a high energy efficiency thank to the utilization of ad-hoc accelerators for bio-signal processing subroutines such as the ones involved in Finite Impuluse Resopnse (FIR) filters and Fast Fourier Transform (FFT). In practical terms, this type of systems can be employed in a wider application domain but their energy efficiency is influenced by several factors. First, the algorithmic steps to be accelerated need to be known at design time and the corresponding dedicated hardware needs to be specifically designed and integrated in the system, which has a non-negligible impact in chip area and design difficulty. Second, those tasks that are not performed by the *ad-hoc* modules are executed in the general purpose processor of the system, which is typically much less energy-efficient, impacting negatively in the overall system consumption.

In an effort to increase the versatility of bio-signal processing platforms, many studies [95] [96] [97], have employed commercial off-the-shelf (COTS) general purpose low-power processors. One of the most important one is the MSP430 System-on-chip (SoC) from Texas Instruments [98], which is employed in research and commercial platforms [99] [30] [100]. Apart from the MSP430 family, other alternatives such as system-based on the ARM processor [87], have gained a lot of popularity in the last decade in the field of wearable sensor nodes and handheld devices due to the availability of basic low-power configurations. In general, COTS are very constrained, featuring few kilobytes of memory, and running at very low clock frequencies (e.g. MSP430 is clocked at 8MHz) in order to obtain maximum savings. Even though these platforms offer different sleeping or inactive regimes to achieve energy efficiency, some low-power techniques such as voltage scaling cannot be exploited due to performance degradation that would not allow meeting the computational requirements of bio-signal processing applications.

In this context, several works have adopted a different approach by proposing custom implementations of existing general purpose Instruction Set Architectures (ISAs) to achieve a superior energy efficiency by applying low-power techniques when compared to the corresponding COTS alternative. For instance, in [101] the authors show how a custom implementation of PIC16 ISA, one of the well established ISAs from Microchip Technology, can achieve up to 100x energy reduction per instruction with respect to an equivalent MSP430-based COTS micro-controller [102]. Several of these works target low-power systems devoted to perform long-term monitoring at very low sampling frequency without requiring high computational power. These options are not suitable for low-power wearable monitors, which perform complex and continuous bio-signal processing in real time. In this direction, Domain Specific Instruction-Set Processors (DSIPs [103]) are designed for a specific application domain, such as the bio-signal processing one, but not for a specific functionality. Recent works described in the following subsections have proposed specific ISAs that provide the necessary performance at low power thank to their energy-aware design. The first, named *Firat* [13], is a PIC-compatible ISA for bio-signal processing that showcases how near-threshold voltage scaling can be applied to obtain energy efficiency in well established general purpose architectures. The second, *TamaRISC* [104], is a simpler and custom reduced-instruction-set-computing (RISC) architecture, which is a cornerstone of the proposed ultra-low power multi-core architectures described in the remaining of this thesis.

3.2.1.1 Firat: A Low Power PIC-compatible ISA

Firat is a RISC architecture based on the instruction set of the PIC24 [105] featuring a Harvard memory model. In particular, Firat implements a subset (66 instructions) of the PIC24 standard excluding those that are intended to access big amounts of data stored in the program memory through the data-path. The instruction types include arithmetic logic unit (ALU) operations, program flow and control operations, bit-oriented operations, single- or multi-bit shifts and data-move operations. The ALU instructions comprise addition and subtraction with and without carry, logic operations (XOR, AND, OR) and 16-bit signed and unsigned multiplication. The ALU features in addition hardware support for integer division, based on an iterative non-restoring algorithm. Both arithmetical and logic shift operations are supported employing a typical barrel shifter. The program flow and control operations (CALL and RETURN) can address the instruction memory with a direct mode or relative to the program counter. Furthermore, branching is possible in direct or indirect modes, with different condition modes dependent on the ALU status register consisting on the commonly used carry, zero, negative, and overflow flags. Move instructions perform single or double data transfers between the register file and the data memory. Finally, the ISA also has support to manage interrupt sources and to clock-gate the entire architecture (SLEEP) by means of specialized instructions.

While the bit-width of the built-in data-path is 16 bits, Firat supports 32-bit operations. Each instruction however is encoded in 24 bits and the access to memory can be performed at a

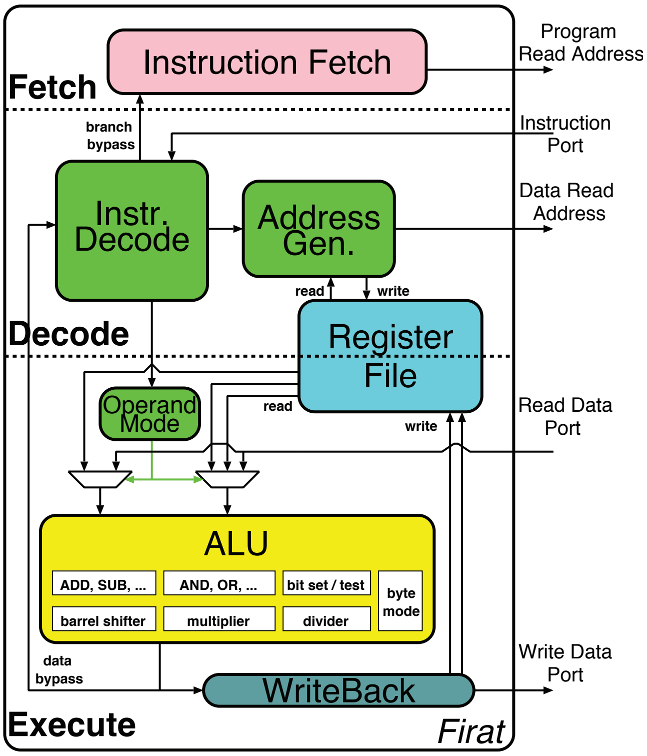


Figure 3.1 – Firat core architecture [107].

byte level. Firat provides a low CPI (cycles per instruction), since all the instructions, except CALL and RETURN, are executed in a single clock cycle, which, as shown by the authors of [106] leads to high energy efficiency. The internal core architecture depicted in Figure 3.1, is implemented as a traditional three-stage pipeline including fetch, decode and execution stages. On the storage side, Firat features an internal register file including 16 working registers and can access to data memory through two dedicated read and write ports. To minimize the CPI, Firat includes a single-instruction prefetch mechanism and allows for data bypassing from the execution stage to the register file in order to avoid read-after-write hazards. When supplied with the nominal voltage of the library used to implement it (90nm CMOS technology, 1.2V), the average energy spent by each instructions reaches 30 pJ. In comparison to an equivalent PIC-based COTS, the architecture con obtain up to 890x energy savings [107] providing even better performance for bio-signal analysis applications due to the feature data-path optimizations. Even though this improved energy efficiency outperforms the state-of-the-art architectures, the DSP stage on bio-signal processing systems remains one of the main contributors to the overall power consumption highlighting the necessity of even more energy-efficient implementations.

3.2.1.2 TamaRISC: a custom low-power ISA for Bio-Signal Processing

TamaRISC was proposed in [104] as an alternative simplified RISC architecture to perform bio-signal processing. Current RISC ISAs, as the previously discussed Firat, can obtain great energy efficiency and reduced CPI but are still very complex due to the availability of specific hardware that a-priori may not add extra benefits for bio-signal processing applications (e.g. advanced addressing mechanisms and decoding logic). Instead, TamaRISC offers an extremely reduced set of instructions (11 against the 66 available in Firat), that can cope with all the typical computational requirements of bio-signal processing reducing a lot the hardware complexity of the architecture.

TamaRISC implements a Harvard-like architecture featuring a three-stage pipeline including fetch, decode and execution stages. The diagram of the core architecture is depicted in Figure 3.2. The data-path width is 16 bits and it incorporates a local register file of 16 registers (16 bits each). In addition, the core is interfaced to the instruction memory (IM) through a read port and to the data memory (DM) through two ports, one for reading and another one for writing respectively.

As mentioned before, the core simplification relies on the aggressive reduction of the instruction set that includes 11 base instructions and the possibility to be extended with custom instructions. The instruction types include 8 arithmetic-logic (ALU) operations, 2 program flow instructions and 1 general data move operation. The ALU instructions comprise addition and subtraction with and without carry, logic operations (XOR, AND, OR), right and left shift (arithmetic and logic) and 16-bit signed and unsigned multiplication (with 32 bit results).The general branching instruction supports direct and register indirect mode, as well as by offset with respect to the program counter, and it can be conditionally executed according to the commonly used carry, zero, negative and overflow flags. In addition, an instruction to perform function calls (CALL) is supported implemented as a "branch and link" to a fixed address.

Conversely to Firat, all instructions take a single cycle to be executed in TamaRISC. This is possible thank to the bypassing of data from the execution to the decode stage that allow single-cycle memory-to-memory instructions. Moreover, the architecture complexity is considerably reduced thanks to the regular 24-bit instruction encoding that allows for simplified decoding hardware. In addition, all the ALU operations in the execution stage receive 3 operands that are obtained by using the same addressing modes, greatly lowering the complexity of the required logic devoted to this task.

Regarding the energy efficiency of TamaRISC, the ISA outperforms state-of-the-art equivalent architectures obtaining an average 17.1 pJ per instruction when supplied with the nominal voltage of 1.2V. In addition, TamaRISC supports voltage scaling as shown in [13] allowing for a further energy efficiency. For instance, at 1.0V and for the same workload, TamaRISC outperforms by 4x the work of [18]. The simplified architecture and high CPI allows obtaining very good performance at a low power consumption.

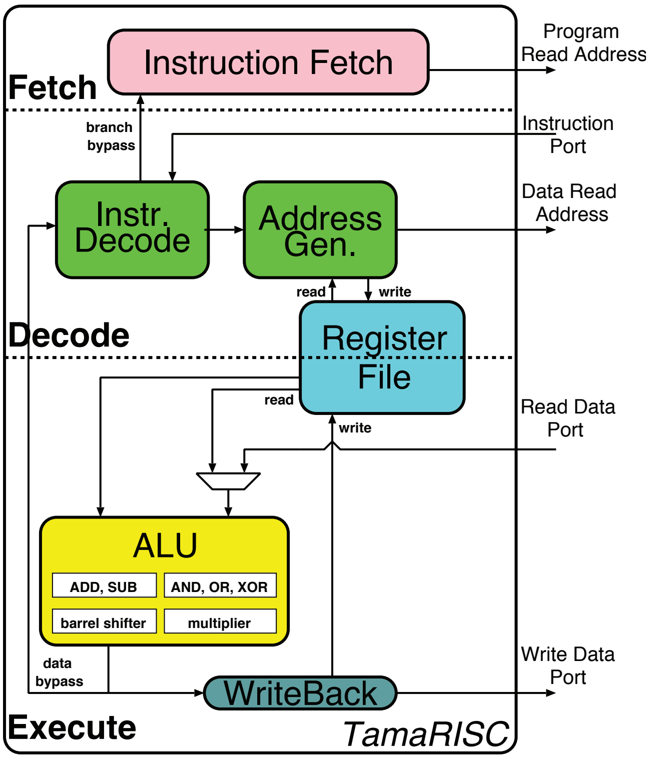


Figure 3.2 – TamaRISC core architecture [107].

Finally, TamaRISC allows for instruction set extensions (ISEs) that can be leveraged to further improve the performance and efficiency of the platform. For example, the authors of [38] proved that by adding a minimal hardware support and an extra instruction to the ISA, TamaRISC could improve its energy efficiency by a factor of 11.6x obtaining a 62x execution speed-up while performing ECG Compression (see Section 2.2.4 in Chapter 2). TamaRISC and an ISE are later used in this thesis to implement an efficient synchronization mechanism when multiple instances of the processor are combined in a ultra-low power multi-core architecture.

3.2.2 Low-Power Multi-Core Architecture for Bio-Signal Processing

Aggressive voltage scaling has been proposed as an energy-saving strategy to reduce power consumption of low-power systems. However, the degradation suffered by the transistors when approaching the near-threshold regime reduces drastically the computational capabilities of these platforms, leading to a performance loss that is incompatible with the requirements of bio-signal processing. In this context, low-power parallel architectures, such as multi-core systems, have been proposed as an alternative to first improve performance, and second counteract the degradation effect while still exploiting aggressive voltage scaling.

Many parallel architectures have been proposed in the literature to perform digital signal processing at low-power regimes. In particular, near-threshold memories [108] [109] have raised an important interest in the last few years since they provide low power consumption at the cost of suffering from reliability issues that can be tolerated by employing low-overhead mitigation techniques. Following the path of near-threshold computing, the work of [110] proposes a parallel architecture based on a cluster of multi-core tiles that shared locally a fast cache memory. However, the multi-level memory hierarchy is complex and voltage scaling is only applied over the processing cores requiring the utilization of voltage level converters and extra logic to manage the different voltage regions. On a more specific domain, the inclusion of dedicated hardware that exploits parallelism while working at near-threshold technique has been also proposed in [111], where a JPEG compression application is executed by means of a co-processor able to counteract the performance loss suffered by reduction of the voltage supply.

Following a different approach, the authors of [112] and [113] proposed architectures that exploit parallelism at the application level by featuring several execution lanes in their datapaths. The first one consists of a custom-design architecture featuring hardware support to perform single-instruction-multiple-data (SIMD) operations at the micro-architecture level. The second one, instead is a VLIW architecture to perform multi-signal parallel processing. Nevertheless, both architectures target computing-intensive signal processing tasks and minimize energy consumption delivering the necessary throughput at high clock frequencies, exploiting aggressive power-gating during long idle periods.

Finally, in the field of low-power embedded platforms, [84] show a good evaluation of existing single- and multi-core configurations for bio-signal processing. In particular, the authors propose a parallel architecture that compensates the performance degradation due to voltage scaling by exploiting the intrinsic parallelism of bio-signal processing applications. This multiple-instruction multiple-data (MIMD) architecture is described in detail in the following section and can be considered the precursor architecture of the ultra-low power architecture proposed in this thesis.

3.2.2.1 Low-Power Multi-core MIMD Parallel Architecture

The parallel platform proposed by [84] depicted in Figure 3.3 consists of a multi-core Harvard architecture in which several processing units (i.e. cores) are interfaced to shared but separated instruction and data memories (IM and DM respectively). In particular, the architecture features 8 TamaRISC processing cores (c.f. Section 3.2.1.2), which offer one of the best tradeoffs in terms of performance and energy per instruction providing a very reduced instruction set with enough hardware support to efficiently perform bio-signal processing. As dictated by these processors, the base architecture data bit-width is 16 bits while the instructions are encoded in 24 bits. Both instruction (32K words of 24 bits) and data memory (32K words or 16 bits) are implemented as multi-banked separated memories interfaced to the cores by instruction and data crossbars (*I-Xbar* and *D-Xbar* respectively).

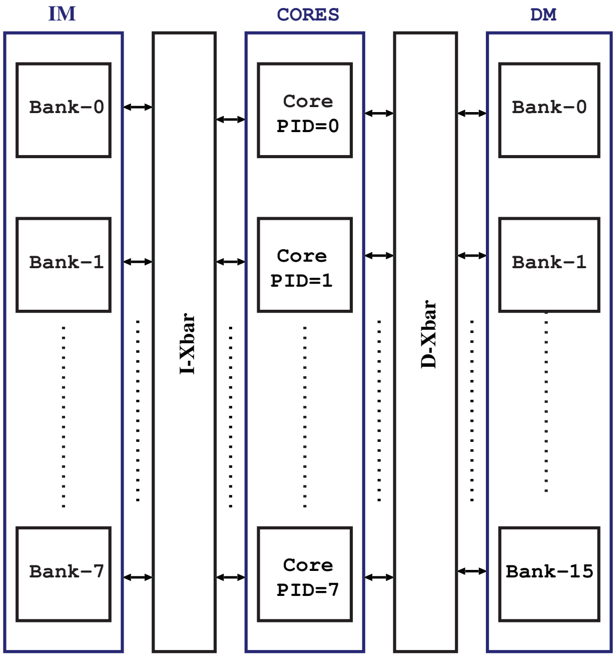


Figure 3.3 – Low-power multi-core MIMD parallel architecture presented in [84].

The design of the interconnects between cores and memories is a key aspect carefully optimized in this platform. The D-Xbar and I-Xbar are implemented as separate Mesh-of-Trees (MoT) networks to support high-speed combinational access from the cores to the memories at a low-power regime [114]. While the I-Xbar supports only read access, the D-Xbar is implemented as two independent MoTs that support concurrently write and read accesses. In all cases the latency of the memory is one clock cycle unless there is a conflicting request in which two or more cores request different words from the same memory bank during the same clock cycle. In those cases, the interconnect is able to serialize the access serving them in order following a round-robin-like strategy. In addition, I-Xbar is provided with a novel *broadcasting* mechanism that merges several read requests to the same word during the same clock cycle into a single memory access that is then forwarded to all the requesting cores. As shown by the authors of [84], this design choice is specially beneficial in parallel bio-signal processing applications where algorithmic steps are performed concurrently over different streams of data. However, the exploration of further optimized interconnects is an extensive field of research that falls out of the main scope of this thesis.

The energy-aware memories are evenly divided into different banks that can be individually power-gated at boot time if not needed by the desired parallel application. More precisely, the instruction memory is divided into 8 banks while the data memory is divided into 16. This configuration, justified in [107], may result intuitive since a core can only access an instruction memory location at a time, thus, requiring 8 IM banks if, in the best possible case, each core accesses one bank. Similarly, a core can access two data locations within a clock cycle (one read request and one write request), thus, justifying the division in 16 banks. At the logic level, the address space of the data memory is split in a *shared* region common for all cores and a private one that is equally divided into subspaces (one per core) managed by a lightweight address translator equipped in the TamaRISC micro-architecture.

The platform targets energy efficient execution of multi-channel bio-signal analysis applications maximizing the benefits of a single-instruction multiple-data (SIMD) strategy while still providing MIMD support. Therefore, only purely parallel applications lacking from any data-dependent code can fully exploit the benefits offered by this architecture. While this type of applications may be present in the bio-signal DSP domain, it is not representative as, first, not all applications perform parallel processing over multiple streams of data, and, second, conditional code execution is normally required in the involved algorithms. However, the authors showed that for a multi-lead ECG Compress Sensing application (c.f. Section 2.2.4 of Chapter 2) the proposed platform can achieve up to 38% extra energy savings with respect to an equivalent multi-core architecture. The energy efficiency improvement is dominated by the great reduction (up to 40.6%) in the active power consumption of the instruction memory thanks to the syncrhonuous access and the implemented broadcasting mechanism. However, these gains are diminished when cores lose synchronization leading to a minimal reduction in power consumption and low possibilities to exploit the intrinsic parallelism of bio-signal processing applications.

3.3 Proposed Synchronization-Based Multi-Core Architecture for Parallel Bio-Signal Processing Applications

As previously highlighted, one of the key contributors to the overall power consumption that can be improved by employing a parallel architecture is the active power dissipated by the instruction memory due to the high amount of accesses. In fact, a first attempt to alleviate this issue, previously described in Section 3.2.2.1, was presented in [84] where the authors propose to reduce accesses to the instruction memory by employing a crossbar interconnect that supports broadcasting. However, the benefit of broadcasting (up to 40.6% active power savings [84]) relies on lockstep execution of algorithm parts that can be performed using the single instruction multiple data (SIMD) processing paradigm. As a consequence, substantial power savings can only be achieved by synchronous instruction execution, which even for many embarrassingly parallel applications is not guaranteed, due to data dependent program flow as well as data memory access conflicts, which bring the processing cores out of lockstep.

Barrier insertion techniques are widely used in parallel computing architectures to achieve synchronization [115]. Cores are synchronized at certain barrier points of execution, i.e., a core is stalled until all other cores reach the same point. In the literature, many software-only [116] and software-hardware hybrid implementations of barriers are proposed [117]. However, these techniques are rather complex for an embedded lower-power platform, where both energy efficiency and low complexity, due to real-time applications, are critical. More-

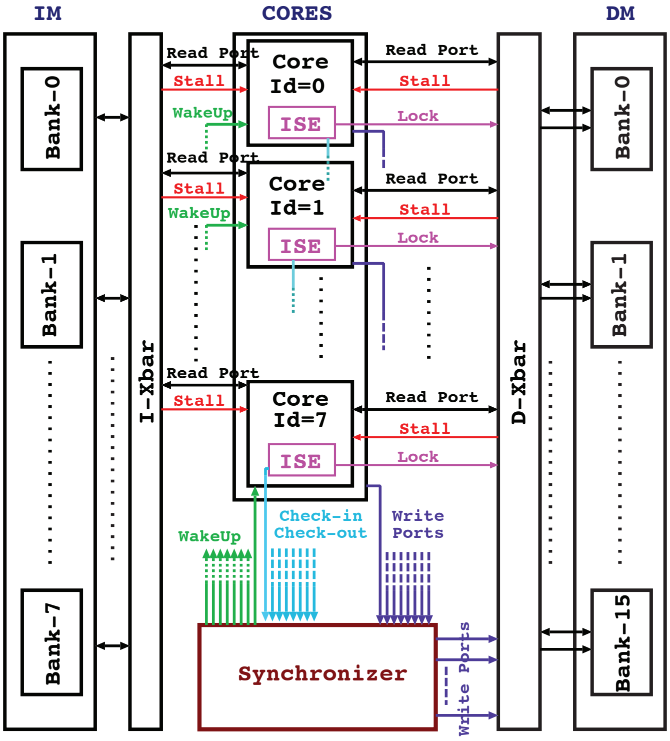


Figure 3.4 – Improved Multi-Core Architecture with Hardware Synchronizer

over, mainstream SIMD architectures (e.g., GPUs) lack the flexibility needed for dynamically managing lockstep execution of cores during data-dependent program flows. Multi-core synchronization has been mostly proposed for high-performance computing (HPC) [118]. However, software-based HPC protocols cannot be applied in a scenario such as the targeted embedded multi-core processors. First, these platforms typically do not embed an operating system capable of providing the necessary runtime support for such protocols. Second, the resources of these platforms are very limited, usually delivering a throughput of few mega-operations per second and featuring some tens of kilobytes of memory. As a consequence, the energy overhead due to the utilization of complex software-based protocols would not compensate for the potential savings. For all these reasons, in this thesis I propose a light-weight hardware/software synchronization mechanism, which provides the necessary support to achieve reduced power consumption on low-power multi-core architectures.

3.3.1 Target Multi-Core Processing Architecture

The target multi-core architecture (c.f. Fig. 3.4) is an extension of the one proposed in [84], which is described in Section 3.2.2.1. The resulting architecture was developed and implemented with the help of Jeremy Constantin, who defined the model of the processing cores, and Ahmed Dogan, who made the HDL implementation of the multi-core platform and the power/area characterization. It consists of 8 processing TamaRISC cores (c.f. Section 3.2.1.2), a shared data memory (DM, 64 kByte in total, divided into 16 banks) and a shared instruction memory (IM, 96 kByte in total, divided into 8 banks). Central data and instruction crossbars (hereafter *D-Xbar* and *I-Xbar*, respectively) interconnect the shared memories and the processing cores. In case of multiple conflicting memory access requests (occurring when a memory bank is accessed by more than one core at different memory locations), the cores are served in sequence and the waiting cores are clock gated. Each TamaRISC core consists of a custom 16-bit reduced instruction set computing (RISC) architecture [85], providing a complete RISC instruction set including instructions for interrupt and sleep mode support. The sleep mode allows external clock gating of the entire core, until a wakeup event occurs.

The platform aims at supporting SIMD operation to exploit data level parallelism typical in bio-signal processing applications. To this end, this new multi-core architecture improves the one in [84] by featuring a light-weight hardware synchronizer that cooperates with the cores to coordinate lockstep execution of code.

3.3.2 Proposed Synchronization Technique

The substantial power savings achieved through SIMD operation highly rely on synchronous code execution among the cores. A loss of synchronization between the cores can occur mainly due to two reasons: data access conflicts and data-dependent program flow.

A data access conflict occurs when a DM bank is accessed by more than one core at different memory locations during the same clock cycle. In this case, the cores that have been served continue their code execution while the rest of the cores wait for data to be served. The proposed technique aims at addressing this issue by enforcing lockstep execution. This enhancement stalls synchronous cores until all of them have been served successfully. When a data access conflict occurs, to detect whether the cores are synchronous, their program counters are compared and monitored in combination to the stall signals provided by the crossbar interconnect when conflict arbitration is needed.

The second reason leading to a loss of synchronization is related to the conditional change of the program flow. Many applications involve data-dependent code sections which lead to conditional execution of different parts of the code and, consequently, a natural desynchronization. we propose to address this problem with a lightweight synchronization technique characterized by the following steps:

1. The target application needs to be analyzed in order to find data-dependent code sections. For instance, for the Code Excerpt 3.1, Figure 3.5 depicts the program flow. In the figure, A, B and C (herein referred as check-in points) are the beginning of the data-dependent code sections, whereas A', B' and C' (referred as check-out points) are the points where the corresponding data-dependent code sections end. As shown by the figure, the conditional segments of code can contain more data-dependent segments that also must be identified.

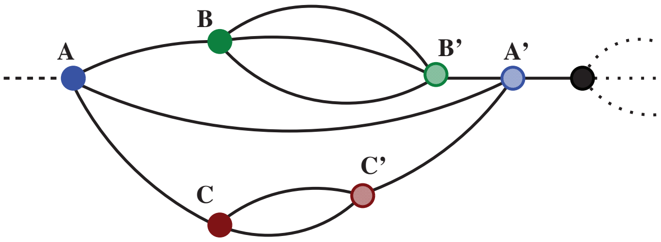


Figure 3.5 – An Example of Data-Dependent Code Section

Code Excerpt 3.1 Example of data-dependent code excerpt

```
function data_dependent_example()
{
  ... in lock-step
  switch(condition_1)               //...check-in A
  {
    case 1:
        switch(condition_2) {       //... check-in B
            case 1:
                conditional_code_1();
            break;
            case 2:
                conditional_code_2();
            break;
            case 3:
                conditional_code_3();
            break;
        }                           //...check-out B'
        break;
    case 2:
                conditional_code_4();
        break;
    default:
        if(condition_3)             //... check-in C
                conditional_code_5();
        else
                conditional_code_6();
        end
        break;                      //... check-out C'
    }                               //... check-out A'
    ...continue in lock-step
}
```

2. For each data-dependent section, a memory position (*synchronization point*) is reserved in memory to annotate the execution status when a core arrives to the check-in and check-out points. In these memory positions (see Figure 3.6), 1-bit core identity flags and total number of cores currently running the corresponding data-dependent code

section are stored.

3. Once a core arrives to a check-in point, the corresponding synchronization point is modified by setting the identity flag and incrementing the core counter indicating that a core has started but not finished the data-dependent segment of code.
4. When a core arrives to a check-out point, the core counter is decremented leaving the identity flag untouched and indicating that the core has finished the conditional piece of code. The arriving core is forced (clock-gated) to wait for the other cores, expected to arrive at the same check-out point, to resynchronize. Once all the expected cores reach the check-out point, the core counter becomes zero which indicates that all the cores can continue their execution in lockstep (clock-gating can be disabled).

3.3.2.1 Software Support: Dedicated Synchronization Instructions

To support the above-described strategy, we implemented a custom instruction set extension (ISE) of the target TamaRISC architecture and added two dedicated instructions (*SINC* and *SDEC*) and a core output (lock signal) to support the check-in and check-out processes. More specifically, *SINC* and *SDEC* are dedicated to the check-in and check-out processes, respectively, whereas the lock output signal is used to ensure atomicity in the read-modify-write operation serializing concurrent check-in/check-out processes.

Apart from the ISE, the microarchitecture of the core has suffered small changes to support the new functionality. First, a specific core register (R_{sync}) is used to store the base address of the reserved DM region where the synchronization points are stored. Second, based on this register and the only input operand of the synchronization instruction (a literal indicating which synchronization point to modify), the core address generator of the decode stages has been accordingly modified. Third, the new lock signal has been implemented to be activated during the realization of the synchronization instruction (2 cycles). The details of the ISE are the following:

- *SINC*: The assembler semantic is: *SINC #literal.* The literal stands for the synchronization point index which addresses the position of the synchronization point in the assigned DM reserved region. The instruction reads data from the memory address, calculated by adding the literal value to the R_{sync} base address register. This read data is forwarded to the write port (without any manipulation since the corresponding modifi-

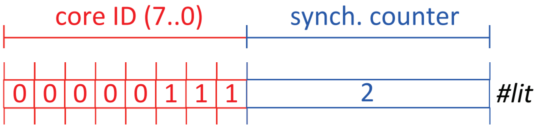


Figure 3.6 – Example of dedicated synchronization point stored in data memory. The values show that 3 cores have checked-in and only one has reached the check-out point.

cations are performed in the synchronizer), and the core output indicating a check-in request to the synchronizer is activated (c.f. Fig 3.4).

- *SDEC*: The assembler semantic is: *SDEC #literal*. It is similar to SINC, but the output indicates a check-out request. In addition, after requesting the check-out the core goes into sleep mode (self clock-gating) until a wake up occurs (i.e., when all expected cores reach to the check-out point).

- *Lock Output Signal*: This output is activated when *SINC* and *SDEC* instructions are executed. This signal locks the memory position, accessed via the instructions, until it has been modified with the new value for serializing not synchronous memory accesses among the cores in sequential check-in/check-out processes.

3.3.2.2 Hardware Support: Synchronizer

In order to follow the strategy previously defined, the ISE is not enough to orchestrate the execution of code. For instance, self-clock-gated cores, after performing a SDEC instruction, need to be waken-up to resume execution when all involved cores in the data-dependent segment of code have reached the corresponding check-out point. To this end, as depicted in Figure 3.4, a new synchronization unit responsible for providing this functionality has been interfaced to the system.

Usually, several cores reach a check-in point together and then the cores may branch to different conditional code path. Depending on the taken conditional code sections, the cores can reach the corresponding check-out point together, separately, or some of them together while the others separately. To check-in/check-out a core needs two clock cycles, since a memory read and then a memory write are needed. The synchronizer merges multiple check-in/check-out requests for a synchronization point and modifies the assigned memory position, accordingly. The synchronization point status is updated with the new identity flags and the core counter (incremented for check-in and decremented for check-out). Merged check-in/check-out requests are also executed in two clock cycles.

As previously mentioned, when all the expected cores reach a check-out point, the core counter becomes zero. In this case, the synchronizer wakes up all cores waiting to be resynchronized (indicated by the core identity flags), and the corresponding memory word is reset to zero.

In addition to managing re-synchronization after data-dependent branches, this unit also enforces lock-step execution when one or more synchronous cores suffer a memory conflict while accessing data memory. In such cases, the synchronization unit clock-gates all the synchronous cores (identified by their equal program counter value) until all memory conflicts are solved and cores can continue in lockstep.

3.3.2.3 Synchronization Instructions Insertion

The proposed synchronization strategy requires the insertion of the *SINC/SDEC* instructions between and after data-dependent segments of code. To this end, a small compiler modification allows for inserting assembly instructions in the required locations by means of pragma codes added directly to the application C code. For a given code the check-in and check-out instructions are inserted as shown in the Code Excerpt 3.2.

Code Excerpt 3.2 Synchronization Points Insertion

```
function instruction_insertion_example()
{
  ...in lock-step
  SINC(<synch_point_X>)
  if(<some condition>) {
    conditional_code_B()
  }
  else {
    conditional_code_C()
  }
  SDEC(<synch_point_X>)
  ...continue in lock-step
}
```

These check-in and check-out instructions are inserted on each data-dependent conditional statement (while/for loops, case/if-else statements, etc.). While manually implemented in the current status of the work, this step can in principle be automated during the compilation process by adding the proper rules in the compilation toolchain. However, this improvement needs a deep knowledge of the compiler technology as well as a formal verification which is out of the scope of this thesis, and therefore remains as an open topic for future work.

3.3.3 Experimental Results

I evaluate hereafter the effectiveness of the proposed synchronization strategy when used in a low-power multi-core architecture. In the next subsections, I detail the employed set-up for the experimental evaluation. Then I discuss the obtained performance metrics. Finally, I evaluate the runtime performance, energy efficiency and area footprint of the target multi-core architecture featuring the proposed synchronization technique.

3.3.3.1 Experimental Setup

To assess the proposed synchronization paradigm in terms of power and performance, we implemented two multi-core designs, with and without the synchronization feature. Both designs are synthesized in a 90 nm low-leakage process technology. For an accurate power analysis of the designs, toggling information while running the reference benchmarks is obtained by simulating a fully routed design (including the clock tree) with back-annotated timing information. The power values at scaled voltages are calculated considering that the power decreases with the square of the supply voltage as shown in [107]. The scaling of the operating voltages is limited to the transistor threshold voltage level to avoid performance variability and functional failures occurring mainly at sub-threshold voltages.

In order to explore the power and performance of the synchronization strategy, three different multi-channel benchmarks have been employed:

- **Morphological Filtering (MF)**: This benchmark removes baseline wander and high-frequency noise from the ECG signals employing the algorithm introduced by [37] as described in Section 2.2.1.2 of Chapter 2. The algorithmic steps in this benchmark present numerous blocks of conditional segments of code.
- **Multi-scale Morphological Delineation (MMD)**: This application finds the so-called ECG fiducial points, i.e. the onset, peak and end of the main ECG characteristic waves (c.f. Figure 2.9). To this end, the algorithm decomposes and interpret the input ECG signal by means of a morphological derivate [36] as explained in Section 2.2.2.2 of Chapter 2. The underlying code presents few data-dependant branches of usually large blocks of code.
- **RMS Multi-channel combination (RMS)**: This benchmark is the smallest but more computing intensive one. For a given set of multi-channel ECG samples acquired at time t, the algorithm performs a root mean square (RMS) combination of the N input channels as described in Section 2.2.3 of Chapter 2. The only conditional block of code is located in the root square computation which is implemented following the iterative algorithm described in [119].

The achieved minimum critical path delay at the nominal voltage (1.2 V) is 8.9 ns and 9.6 ns for the architectures without and with the synchronization feature, respectively. The targeted applications do not require such high clock frequencies, thus no vital timing issue is present. A relaxed constraint of 12 ns gives good power results for both designs with and without the synchronization feature [84], while still allowing for considerable voltage scaling.

3.3.3.2 Performance Results

In order to evaluate the improvement in performance obtained thanks to the proposed synchronization technique, I report in table 3.1 the average amount of active cycles executed in each of the studied benchmarks when processing a fixed window of samples. It is important to note that an effective active cycle is counted as such when the architecture is not fully clock-gated (i.e. at least one core is not clock-gated). In addition, all benchmarks have been compiled using the same optimization flags and the instance of the benchmarks running

Table 3.1 – Average number of cycles required to process a sample for the different studied benchmarks

	Baseline multi-core architecture without synchronizer	Target multi-core architecture with synchronizer
MF	779	328
RMS	652	352
MMD	1224	609

on the target system have been modified to include the specialized synchronization instructions. As a consequence, the runtime metrics reported for this system include the overhead of executing the synchronization instructions.

As reported in Table 3.1, thanks to the resynchronization process, the average number of active cycles per sample have been greatly reduced due to the synchronous access to IM that avoid memory conflicts. As a result, considerable speed-up (up to 2.4x) have been achieved on all the benchmarks despite the synchronization overhead. The target architecture featuring the synchronization achieves between 2.5 and 5.2 instructions per clock cycle, whereas the baseline multi-core architecture can only reach 2.0 for the best of the studied benchmarks.

3.3.3.3 Area Footprint

The total area footprint of the target system featuring the proposed synchronization technique is 1154.8 kilo-Gates-Equivalent (kGE, where 1 GE≈3.136 μm^2), which represents only a 2% increase with respect to the multi-core baseline [84] (1128.8 kGE). In fact, the area footprint of the chip is dominated by the instruction and data memory banks, which account for more than 85% of the total area and remain unmodified.

At the processors level, for the baseline architecture the 8 TamaRISC instances were occupying an area equivalent to 87.3 kGE, whereas in the new implementation the cores have been modified to adopt the proposed ISE leading to a 16.1% increase (101.7 kGE in total).

Finally, the new synchronization logic adds an almost negligible amount of area to the total account requiring only 4.6 kGE for the new block managing the check-in and check-out points in data-dependent branches and 6.6 kGE for the necessary circuitry that forces lockstep execution in case of memory conflicts.

3.3.3.4 Power Consumption

Table 3.2 reports the breakdown of the average power consumption of the multi-core baseline and target platforms. The architecture featuring the proposed synchronization strategy achieves up to 60% savings for the total number of IM bank accesses when running the employed bio-signal processing benchmarks. This reduction in memory accesses greatly impacts

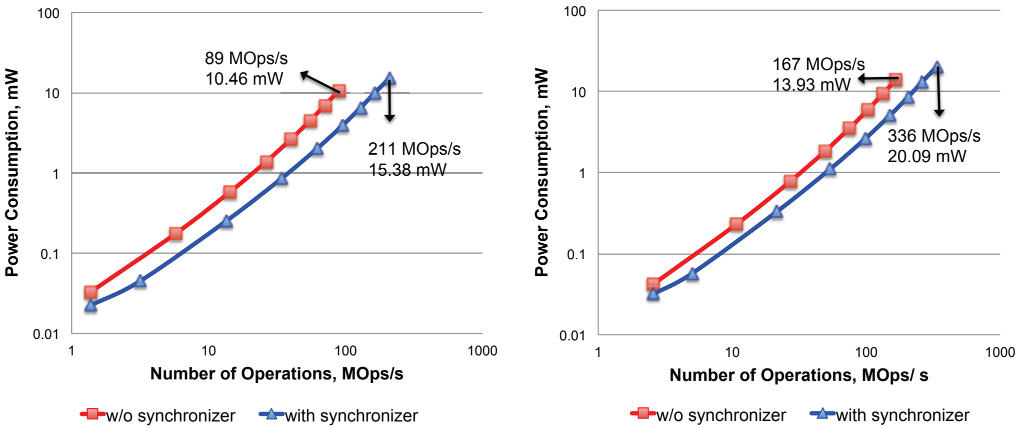
Table 3.2 – Dynamic power distribution of the architectures while running reference benchmarks at 8 MOps/s and 1.2 V

	Baseline multi-core architecture without synchronizer	Target multi-core architecture with synchronizer
Total	$0.64\,\text{mW} < P < 0.94\,\text{mW}$	$0.47\,\text{mW} < P < 0.58\,\text{mW}$
Cores	0.14 mW	0.16 mW
IM	$0.20\,\text{mW} < P < 0.36\,\text{mW}$	$0.09\,\text{mW} < P < 0.15\,\text{mW}$
DM	$0.05\,\text{mW} < P < 0.08\,\text{mW}$	$0.06\,\text{mW} < P < 0.08\,\text{mW}$
D-Xbar	0.06 mW	0.05 mW
I-Xbar	0.03 mW	0.02 mW
Syncronizer	-	0.01 mW
Clock Tree	$0.09\,\text{mW} < P < 0.16\,\text{mW}$	$0.05\,\text{mW} < P < 0.08\,\text{mW}$

the power consumed by the IM, as seen in Table 3.2. However, on the other hand, due to the synchronization overhead that requires to read and write the reserved positions for the synchronization points in every check-in and check-out point, the total number of DM accesses is incremented leading to a small increase in the power consumed by the DM. In addition, the synchronizer unit adds up to a 2% overhead to the overall consumption. Nevertheless, the amount of additional DM accesses is affordable since it remains below 10% and the incurred power consumption overheads are more than compensated by the savings obtained in the IM. The cores in the improved architecture consume on average 15% more power than those of the baseline architecture due to the hardware requirements for the introduced ISE. This effect is counteracted by savings in the interconnects due to increased SIMD operation, which effectively reduces the signal activity in the crossbars. Moreover, the improved architecture achieves 2x power savings in the clock tree, since it requires lower clock frequency for a given workload compared to the architecture without the synchronization feature. In total and without exploiting voltage scaling, the proposed synchronization strategy provides up to 38% dynamic power savings for a fixed workload.

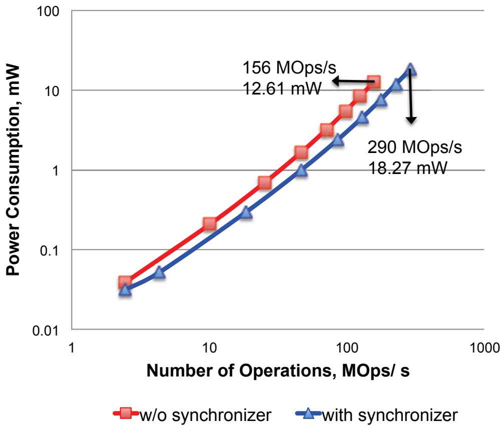
The benefits of the proposed synchronization technique goes beyond energy savings when high performance is required. For instance, Figure 3.7 compares the power consumption of the multi-core platform with and without implementing the proposed synchronization technique while executing the three used benchmarks. It can be observed that, at maximum voltage (i.e. 1.2 V), the highest possible throughput is boosted by 2.4x for a workload profile like MF (211 MOps/s vs 89 MOps/s), 1.86x for RMS (290 MOps/s vs 156 MOps/s) and 2.0x for MMD (336 MOps/s vs 167 MOps/s).

The target multi-core architecture featuring synchronization further improves its energy efficiency when voltage scaling is considered. When the average workload of a given application is bounded, the target system, which is more performant, can deliver the necessary throughput with a much lower system clock frequency. This allows for reducing the supply voltage and therefore obtain larger energy savings. Given the studied benchmarks and the curves



(a) MF

(b) MMD



(c) RMS

Figure 3.7 – Total Power Consumption of the multi-core architectures while running the different employed benchmarks

of Figure 3.7, when the sampling frequency of the input signal is fixed to the standard high-resolution ECG rate of 1KHz, the target system can obtain energy savings of more than 50% for all the studied benchmarks. However, even though synchronization always makes the multi-core architecture more energy efficient, for low workloads (e.g. few KOps/s), a single-core alternative would be more appropriate since the gains in dynamic power consumption would not compensate the extra leakage consumed by the bigger and more complex system.

3.4 Proposed Advanced Synchronization Technique for Arbitrarily Parallel Bio-Signal Processing Applications

Energy efficiency is of paramount importance for battery-supplied biomedical monitors, which must operate autonomously for prolonged periods of time. To minimize energy consumption, processing on these devices requires a carefully tailored computing architecture. As

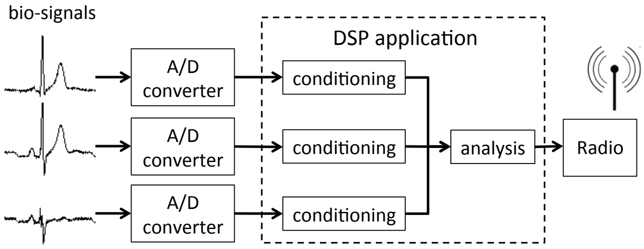


Figure 3.8 – Block scheme of a smart WBSN platform.

previously shown, an effective method to decrease power consumption is voltage-frequency scaling (VFS), which trades-off the voltage supply (and, consequently, energy consumption) for peak clock frequency [21] [22] leading to a performance loss below the requirements bio-signal processing applications.

In this context, low-power multi-core architectures can improve energy efficiency, exploiting the benefits of single-instructions-multiple-data (SIMD) architectures when executing code synchronously thanks to the technique detailed in Section 3.3. However, the maximum energy savings of these architectures are only achieved for a subset of the application domain where multiple cores execute the same phases of an application (e.g., conditioning in Figure 3.8) on multiple acquired inputs. However, these synchronization approaches do not provide the necessary flexibility to perform parallel processing of streams on multiple cores, while also supporting efficient producer-consumer relationships among cores.

Herein, I propose a generalized synchronization technique for the proposed low-power multi-core platform, enabling the efficient parallel execution of embedded bio-signal processing applications presenting an arbitrarily degree of parallelism. The solution stems from the observation that applications in this field [16] [120] [10] [12] are divided into several consecutive phases. In the illustrative example of Figure 3.8, multiple signals are acquired in parallel and independently processed, and outputs are subsequently combined and transformed into a single data stream or set of features that are later analyzed. Similar schemes are found in most bio-signal processing applications [15].

In particular, with this technique I generalize the concept of low-power multi-core execution to more complex applications presenting multiple internal phases, with an arbitrarily large degree of parallelism and producer-consumer relationships among phases. To achieve this goal, I propose a novel synchronization mechanism, allowing an efficient mapping of advanced bio-signal processing applications.

3.4.1 Target Multi-Core Processing Architecture

The proposed technique has been designed for the previously employed low-power multi-core architecture [84] (c.f. Section 3.3). Nevertheless, the methodology could be applied to any low-power multi-core system as the one depicted in Figure 3.9. Three properties, common in

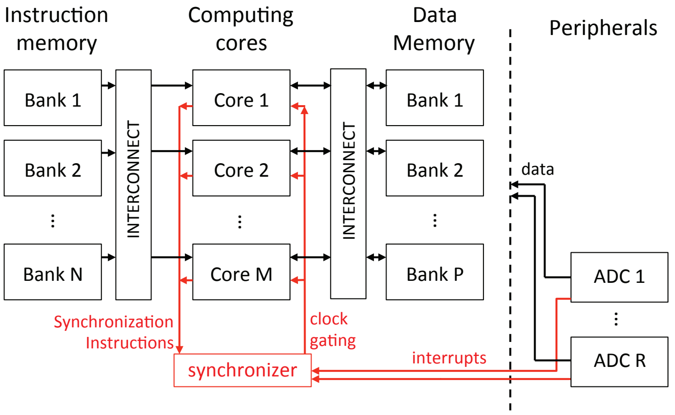


Figure 3.9 – Hardware architecture of a multi-core WBSN. In red, HW support for the proposed synchronization technique

this family of systems, must be satisfied by the platform to obtain the maximum benefits from the proposed synchronization method:

1. Instruction and data memories (IM and DM respectively) must be divided into several banks so that they can be read/written independently and power-gated if not used in order to save energy. Alternatively, a multi-port implementation would also suffice but the unused memory could not be power-gated leading to less energy savings.
2. The logical address space of the data memory needs to be divided into shared and private sections, each core having its dedicated region.
3. In order to maximize the savings, the interconnect network between the memories and the processing units need to implement a *broadcasting* mechanism similar to the one described in Section 3.2.2.1. This mechanism merges multiple read requests from the same location in memory and in the same clock cycle into a single memory access.

3.4.2 Synchronization Methodology

In this section I describe in detail the proposed synchronization mechanism, the required modifications at the hardware and software levels and how any existing bio-signal processing application can be adapted to obtain maximum energy savings when mapped onto a low-power multi-core platform.

3.4.2.1 Hardware/Software additional support

The proposed approach is an extension of the previously introduced synchronization technique (Section 3.3) and consists of a hybrid hardware/software synchronization mechanism enabled by an instruction set extension (ISE) of the cores that cooperate with a dedicated lightweight synchronization unit.

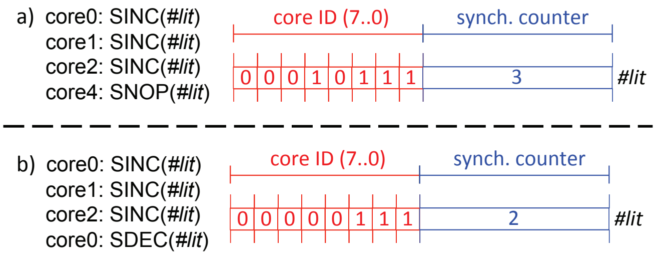


Figure 3.10 – Examples of synchronization points values. a) cores 0, 1 and 2 should jointly produce data for core 4; data is not yet available. b) cores 0, 1 and 2 have entered a data-dependent branch, core 0 has finished executing it.

On one side, software support is provided by a set of dedicated instructions (*SINC*, *SDEC* and *SNOP*), employed to synchronize code execution. These instructions use reserved locations (*synchronization points*) in the shared data memory, which store information about the execution flow. Synchronization instructions modify synchronization points, which are divided in two fields: their most significant bits contain 1-bit identity flags corresponding to the identifiers of each core, while the least significant bits are used as an up/down counter (as illustrated in Figure 3.10). The newly introduced *SNOP(#lit)* instruction appends the identification flag of the issuing core to the *#lit* synchronization point, without modifying the core counter. *SINC(#lit)* also sets the core identification flag, but in addition increases by one the counter. Finally, the *SDEC(#lit)* instruction, without modifying the identification flags, decreases the counter. Conversely to the *SDEC* implementation described in Section 3.3, in this extension, *SDEC* does not clock-gate the issuing core after execution. Instead, the execution continues normally and a *SLEEP* instruction after *SDEC* is needed to intentionally requests the synchronizer to clock-gate the issuing core until the next synchronization event happens.

On the other side, hardware support for synchronization is provided by a lightweight synchronizer (Section 3.3.2.2) unit that manages the interaction among cores, ensuring lock-step execution when possible, avoiding de-synchronization due memory access conflicts and keeping track of the execution flow. To this end, the unit can clock-gate (pause) cores and resume them, according to the received interrupts and the synchronization instructions issued by the cores. This unit is able to merge several concurrent synchronization instructions so that the synchronization point is updated correctly.

3.4.2.2 Synchronization Mechanism

The dedicated synchronization instructions are used both to manage producer-consumer relationships and to enforce lock-step execution after data-dependent branches.

In the first case, producer-consumer relationships require the consumer cores waiting for data to execute a `SNOP` instruction, registering themselves in the corresponding synchronization

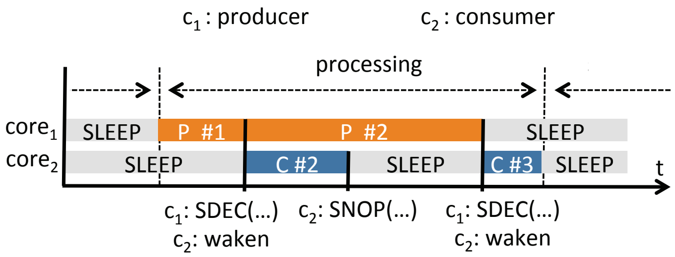


Figure 3.11 – Execution flow of a producer-consumer relationship employing the proposed synchronization technique

point. Afterwards, such cores request to be clock-gated by 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) when the value of the counter in the synchronization point reaches zero (Figure 3.10-a), and resumes execution of all the registered cores. Pseudo-code excerpts presented in Code Excerpts 3.3 and 3.4 showcase a generic example of a producer-consumer relationship, which is also represented on the diagram depicted in Figure 3.11.

Code Excerpt 3.3 Example of producer code

```
function producer_example()
{
  SINC(<synch_point_B>)
  produce_new_data()
  SDEC(<synch_point_B>)
}
```

Code Excerpt 3.4 Example of consumer code

```
function consumer_example()
{
  while(<no data to consume>) {
    SNOP(<synch_point_B>)
    SLEEP()
  }
  consume_data()
 }
```

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. When all cores that initially entered the branch finish executing it, the core counter of the synchronization point becomes zero, and cores are notified by the synchronizer to resume their execution in lock-step. A example showcasing a simple code excerpt causing a potential de-synchronization due to a data-dependent branch is presented in Code Excerpt 3.5.

Code Excerpt 3.5 Example of lock-step code

```
function lock_step_example()
{
  ...in lock-step
  SINC(<synch_point_A>)
  if(<some condition>) {
    conditional_code_B()
  }
  else {
    conditional_cod_C()
  }
  SDEC(<synch_point_A>)
  SLEEP();
  ...continue in lock-step
}
```

In the common case when more than one synchronization instruction are issued on the same memory location, the synchronizer merges the requests to perform a single and consistent memory modification. Apart from another core, the data producer can be an external peripheral, such as an analog-to-digital converter (ADC) sampling a bio-signal at a constant frequency and providing a data-ready interrupt that will be connected to the synchronizer (as shown in Figure 3.9. When processing cores need new data but this is not available, cores subscribe to the interrupt line through a memory-mapped register, execute a *SLEEP* instruction and remain clock-gated by the synchronizer until the arrival of an interrupt from the registered source is forwarded by the synchronizer.

3.4.2.3 Application Mapping

The proposed synchronization technique enables to theoretically adapt any existing bio-signal processing application to be efficiently executed on the target low-power multi-core architecture. Starting from the C source code of a standard single-core implementation, three steps have to be performed to parallelize and execute an application using the proposed synchronization method, namely:

1. **Partitioning**: The different algorithmic phases (or tasks) of the application have to be identified so that workload is divided among cores. As one or more of these phases may be applied over several streams of data, to exploit lock-step execution, the processing of each data stream should be assigned to different cores. Partitioning naturally follows

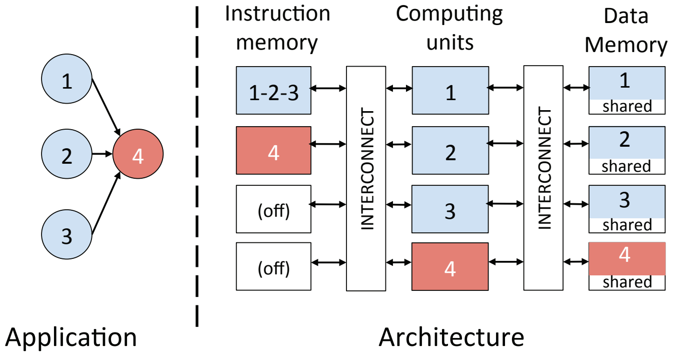


Figure 3.12 – Partitioning and mapping of the application in Figure 3.8 on a multi-core platform embedding 4 computer units, 4 IM banks and 4 DM banks.

the structure of bio-signal processing algorithms in which one or several bio-signals are taken as input and processed in sequential steps.

2. **Insertion of synchronization instructions**: First, for producer-consumer relationships among the identified phases, *SNOP* instructions are added to consumer cores, while *SINC* and *SDEC* to producers as explained previously in Section 3.4.2.2. Similarly, *SINC* and *SDEC* pairs are also inserted before and after data-dependent code segments executing on cores assigned to parallel computation streams. For each data-dependent branch and producer-consumer relationship, a dedicated synchronization point is reserved in memory. The insertion of synchronization instructions is performed manually in the source code by placing the assembly statements in the corresponding points of the code.

3. **Mapping**: Binary code of the different phases is placed in different IM banks in order to avoid access conflicts and benefit from broadcasting. Moreover, the threshold between shared and private sections in memory and the number of synchronization points must be configured. This last process is handled automatically by a set of scripts that receive a set of compilation directives to specify the different code regions and parameters. The compilation and linking process of the final application is then performed.

Figure 3.12 graphically shows the result of applying these steps to the application introduced in Figure 3.8. First, the application is divided into two phases: conditioning and processing. Because conditioning is performed in three different inputs, it is assigned to three different cores that execute the task in parallel exploiting the SIMD capability, each of them processing one input. The processing phase is assigned to a fourth core that consumes the data produced by the first three. *SNOP* and pairs of *SINC* and *SDEC* are placed properly to manage the producer-consumer relationship and ensuring lock-step execution. In the mapping step, code dedicated for the different phases is placed in different IM banks, with cores executing the same application phase sharing the same bank.

3.4.3 Experimental Set-up

Before reporting the results of the evaluation performed to assess the effectiveness of the proposed technique, in this section I detail the main characteristics of the experimental set-up. First, I define the systems taken under consideration, then I detail the simulation framework developed to evaluate the proposed strategy and finally I described the employed benchmark suite.

3.4.3.1 Evaluated Multi-core and Baseline Single-core Systems

The target multi-core system (Section 3.3) employs 8 TamaRISC cores (c.f. Section 3.2.1.2), 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. Crossbars are sized accordingly and a three-channels analog-to-digital converter (ADC) module is interfaced to the system using memory mapped registers located in shared DM. Data-ready interrupt lines coming from the ADC were connected to the synchronizer, which forwards them to cores when required.

In the experiments, it is considered as baseline configuration a single core connected to the same memory subsystem as in the previous case, so that unused memory banks can be powered-off. To manage the memory interface in this system, simpler decoders can be used instead of crossbars allowing higher clock frequencies at the same voltage level. In addition, for the power consumption evaluation a multi-core system that does not feature the proposed synchronization technique is also considered.

3.4.3.2 Simulation Framework

The simulation framework developed to evaluate the proposed method is composed by the programming tool-chain (compiler, builder and linker) and the simulation environment. The former allows for the compilation of code to be loaded and executed on the platform and requires a set of building directives, which guide the automatic linking process (c.f. Section 3.4.2.3). The latter includes a synthesizable RTL description and a System-C architectural simulator of the target platform, which encapsulates the model of the employed TamaRISC cores.

The baseline and target architectures were characterized executing bio-signal processing applications, at two levels of abstraction. At the lower level, post-layout RTL simulations (using a $90nm$ low-leakage process) are employed, measuring the average energy consumption of each architectural element when executing small code sections. Data gathered from the simulations is then used to annotate a System-C model of the system, from which application-wide energy consumption figures are extracted in different settings.

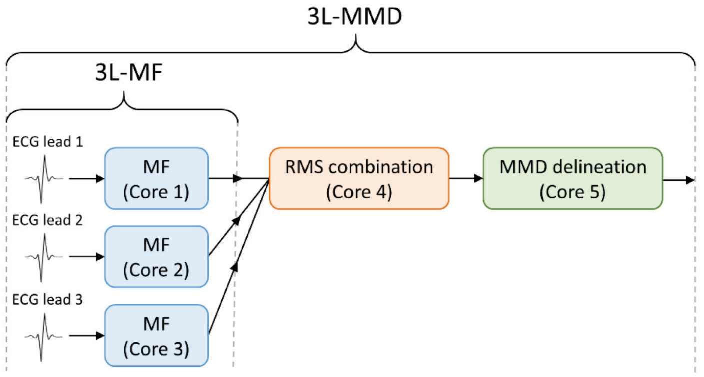
Output of the framework is then the average power consumption obtained from an extended period of simulated time (60 seconds for all the performed experiments), which would not be possible to obtain with time- and resource-consuming post-place-and-route simulations. This aspect is of great relevance for target bio-signal processing applications, as the input bio-signals have slow dynamics (e.g., in the example of heart monitoring, the normal heart rate ranges from 60 to 100 beats-per-minute), requiring extended simulations to capture a good measure of the average power consumption of the different architectural configurations.

3.4.3.3 Benchmark Applications

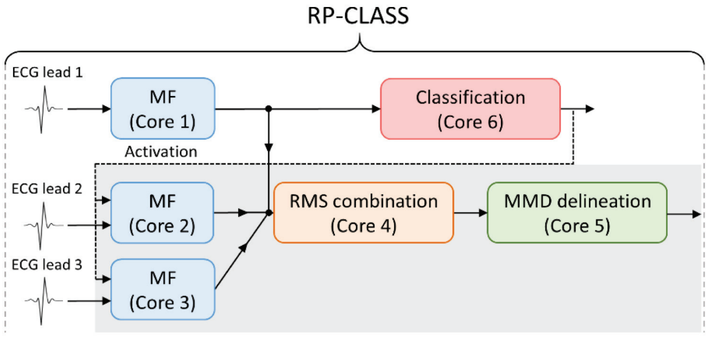
I considered three highly optimized applications, from the field of embedded electrocardiogram (ECG) signal processing that represent different types of applications with different workloads and runtime characteristics.

- **3-lead morphological filtering (3L-MF)**: This benchmark (Figure 3.13a) performs three-lead morphological filtering [37], which removes unwanted components from acquired ECG signals, and operates in parallel on three different input streams (c.f. Section 2.2.1.2 in Chapter 2). When mapped on three cores, the application does not employ producer-consumer relationships, so that synchronization primitives are only used to recover lock-step execution among cores.
- **3-lead multi-scale morphological-based delineation (3L-MMD)**: This benchmark (Figure 3.13a), performs a three-lead delineation using a multi-scale morphological derivatives (c.f. Section 2.2.2.2 in Chapter 2). In addition to filtering the three input signals using a similar scheme to 3L-MF, 3L-MMD also aggregates using an RMS Combination algorithm as the one described in Section 2.2.3 of Chapter 2. Them, the application analyses the resulting combined streams to automatically detect the ECG fiducial points. Consequently, as opposed to 3L-MF, it cannot be mapped using the technique described in Section 3.3. The application is mapped onto five cores, of which three perform filtering in parallel and two are employed to combine the signals and identify the fiducial points, respectively.
- **Selective ECG delineation based on a heartbeat classifier (RP-CLASS)**: This application employs a heartbeat classifier operating on a single lead to discern normal from pathological heartbeats as described in Section 2.3 of Chapter 2. When an abnormal situation is detected, a three-lead delineation is activated only for the pathological heartbeat. RP-CLASS is mapped onto six cores (Figure 3.13b), and showcases the ability of our proposed synchronization technique to manage both control and data flows among cores. It also exemplifies a case where workload is not uniform: as abnormal heartbeats are rare, in fact, the four cores in the delineation chain are seldom activated.

Across experimental tests, standard multi-lead ECG inputs have been used. To evaluate the 3L-MF and 3L-MMD, a multi-lead signal from a healthy subject of the CSE Database [121]



(a)



(b)

Figure 3.13 – Block schemes of benchmark applications: a) 3L-MF and 3L-MMD; b) RPCLASS Conditionally activated blocks are indicated with grey background.

has been employed. For the RP-CLASS application, 20% of pathological beats were inserted, representing the average presence of abnormalities in the CSE database.

The configurations of the considered single-core and multi-core platforms for each of the benchmarks are detailed in Table 3.3.

3.4.4 Experimental results

Three aspects are investigated in this section. First, the run-time requirements of the described benchmarks are analyzed while executed in the baseline and target architectures. Then, the power consumption of the building components of both systems is shown and the obtained numbers are discussed. Finally, the most complex of the evaluated benchmarks, RP-CLASS, is further employed to demonstrate the effectiveness of the proposed synchronization technique in reducing the power consumption of the multi-core system even in the case of unbalanced workload and not lock-step code execution.

	3L-MF		3L-MMD		RP-CLASS	
	SC	MC	SC	MC	SC	MC
Active Cores	1	3	1	5	1	6
Active IM banks	1	1	3	4	4	6
Active DM banks	3	16	3	16	11	16
Min. Clock (MHz)	2,3	1,0	3,4	1,0	3,3	1,0
Min. Voltage (V)	0,6	0,5	0,6	0,5	0,6	0,5

Table 3.3 – Platform configurations of the single-core (SC) system and the multi-core (MC) one for the different studied benchmarks

	3L-MF	3L-MMD	RP-CLASS
IM Broadcast (%)	40,36	23,44	10,30
DM Broadcast (%)	3,74	2,82	1,07
Code Overhead (%)	2,57	0,92	0,69
Run-tim Overhead (%)	1,65	0,96	0,60
Avg. Power (μW)	**31,8**	**50,3**	**56,9**

Table 3.4 – Obtained results for the different benchmarks while executing on the multi-core platform featuring the proposed synchronization technique.

3.4.4.1 Performance and Memory Footprint Comparison

The evaluated benchmarks were optimized to be executed in both the single- and multi-core architectures, considering in each case the least possible amount of memory and computational requirements while meeting real-time constraints. In particular, the unused memory banks are powered-off and the system clock frequency is reduced to the minimum in order to exploit the benefits of voltage-frequency scaling (VFS). Details of the executed experiments are shown in Tables 3.4 and 3.3. Three main conclusions can be drawn from these numbers. First, all applications can run in real-time and at a lower clock frequency in the case of the multi-core architecture, which allows performing aggressive voltage scaling (see Table 3.3). Indeed, higher demands of computing power are solved using a larger number of cores instead of increasing the system clock frequency.

Second, the single-core architecture presents lower memory requirements. In fact, the mapping of code in the IM is less constrained, whereas in the multi-core platform instructions need to be placed in different memory banks to avoid access conflicts. In addition, in order to support the division of the data memory into shared and private sections, all the data memory banks of the multi-core platform need to be active due to the design of the ATU unit.

Third, the introduced overhead due to the proposed methodology is very low. In the worst case (3L-MF), the inserted special instructions add-up less than 3% of the total code while, at

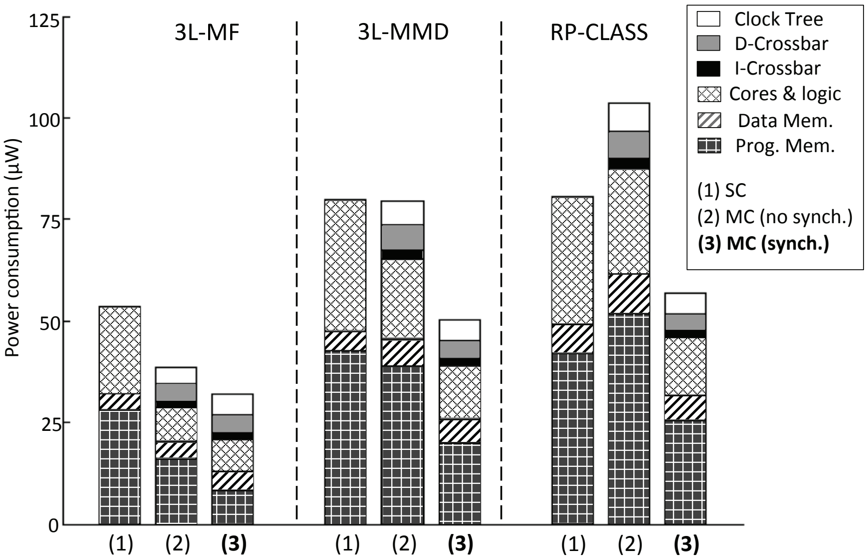


Figure 3.14 – Power consumption decomposition of the single-core (SC) and multi-core (MC) systems with and without the proposed synchronization approach.

run-time, the issued synchronization primitives represent 1,65% of the active cycles .

3.4.4.2 Single- and- Multi-core Energy Consumption

Power consumption numbers from Table 3.4 show a considerable reduction of up to 40% when employing the proposed approach in the multi-core platform. Figure 3.14 presents a decomposition of the power consumption of the building components of both architectures. Moreover, it shows the power consumption of a multi-core system that does not employ the proposed synchronization approach as the one introduced by [84], in which producer-consumer relationships need to be implemented performing inefficient active waiting.

The experiments show that the multi-core system adds a non-negligible overhead (e.g. up to 34% of the total energy in 3L-MF) due to the extra necessary components (crossbars, logic and a more complex clock tree). In addition, when the synchronization technique is not employed, the total power consumption of the multi-core platform can be lower, comparable or higher (e.g. 3L-MF, 3L-MMD and RP-CLASS respectively) than the consumption of the single-core architecture, depending on the workload balance among cores. However, if our proposed approach is used, the energy requirements are drastically reduced in all the cases, achieving important savings thanks to the benefits of VFS.

As Figure 3.14 shows, one of the advantages of our technique is the reduction of the program memory consumption due to instruction broadcasting. In addition, although the synchronization technique slightly increases memory usage, the DM consumption is not incremented significantly. In fact, when the application memory footprint is large, like in RP-CLASS, the multi-core DM becomes more energy-efficient, since it operates at a lower voltage level and

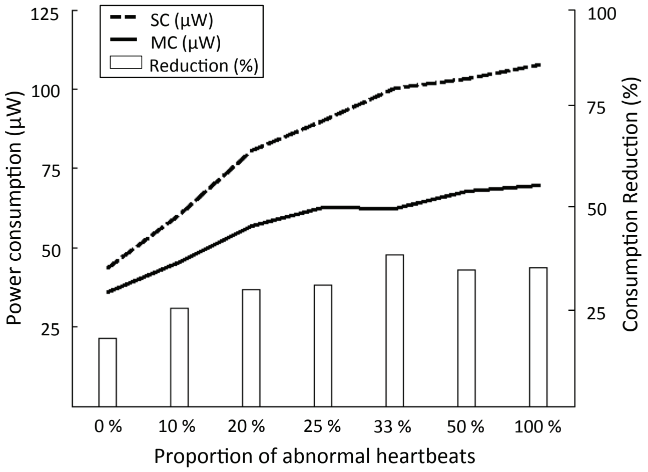


Figure 3.15 – Power consumption (left axis in μW) of the single-core (SC) and multi-core (MC) systems and the respective reduction (right axis, in percentage) when employing the proposed approach in the multi-core platform

only few banks can be powered-off in the baseline system.

Finally, although the multi-core system adds an overhead (e.g. 34% of the total energy in 3L-MF) due to the extra necessary components (crossbars, synchronizer and a more complex clock tree), the total power consumption is drastically reduced thanks to the benefits of voltage-frequency scaling and lock-step execution, achieving savings of up to 40%.

3.4.4.3 Synergies Between VFS and Broadcasting

The proposed synchronization methodology allows exploiting the benefits of voltage-frequency scaling and broadcasting. These two features, on their own, improve the energy efficiency of low-power multi-core systems ([21] [20] [86] [122]), and in combination lead to even greater savings. Figure 3.15 shows the energy consumption of the baseline and the target architectures and the percentage reduction while executing the RP-CLASS applications with different inputs, varying the amount of pathological heartbeats. For all tests the abnormal heartbeats have been distributed uniformly.

When there are no pathological heartbeats, the analysis chain (4 cores) is never activated and no parallel computation is carried out. However, energy savings of 17% are still obtained due to voltage-frequency scaling since the workload is divided and pipelined among cores in the multi-core system. In addition, when abnormalities are present, broadcasting reduces the consumption when the analysis chain is activated due to the lock-step execution of code. In that case, the benefits of both features combined allow for improvements in the energy efficiency of up to 38% in the best case.

3.5 Summary and Concluding Remarks

In this chapter I have presented an ultra-low power multi-core architecture featuring a synchronization technique that allows efficient execution of bio-signal processing applications. The platform is composed of 8 processing cores interfaced with multi-banked memories through combinational crossbar interconnects. The architecture aims at minimizing power consumption by relaxing the system clock constraint and applying voltage scaling. To that end, the applications are parallelized dividing the workload among the different cores.

First, I have proposed a hardware/software technique devoted to maximize lock-step execution in multi-channel parallel applications following a single-instruction multiple-data (SIMD) paradigm. It consists of a dedicated hardware synchronizer and an instruction set extension, which jointly enable to recover synchronization after the execution of data-dependent segments of code. The obtained results show that, when compared with a state-of-the-art multi-core equivalent, the proposed synchronization-based architecture provides up to 38% energy savings while only increasing the area footprint by 2%.

Second, I have generalized the technique to any bio-signal processing application presenting an arbitrarily high degree of parallelism. I have proposed mechanisms to concurrently manage lock-step execution of code and producer-consumer relationships among cores. The methodology describes the necessary hardware and software support as well as the steps to adapt an existing application in order to adopt the proposed technique. According to the experimental results, the proposed architecture can obtain up to 40% energy savings while running real-world ECG processing benchmarks.

4 Optimized Memory Subsystems for Ultra-Low Power Multi-Core Architectures

4.1 Introduction

The increasing social impact of chronic cardiovascular disorders presents a major challenge for healthcare provision [4]. In this context, wearable and miniaturized health monitoring systems, termed Wireless Body Sensor Nodes (WBSNs), offer a large-scale and cost-effective solution [15] that have enabled the conception of smart biomedical monitors.

These devices are able to perform complex on-node Digital Signal Processing (DSP) routines, such as Electrocardiogram (ECG) compression [11], automated feature extraction [10] and classification [57]. DSP applications embedded in biomedical monitors greatly reduce the required transmission bandwidth, thus increasing the overall energy efficiency of the system. 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 has lead to a change of the dominant contributor to the power consumption of these platforms, which now resides in the embedded DSP stage as shown in Figure 4.1. To maximize the efficiency of biomedical monitors, signal processing must be supported within a tight power budget, while at the same time respecting real time constraints. Bio-signal processing architectures must therefore be carefully designed targeting ultra-low power (ULP) consumption. In this regard, many efforts have been made in the last years, proposing solutions ranging from ad-hoc accelerators [112, 123] to ultra-low-power synchronization-based multi-core architectures [84, 124] as the ones presented in Chapter 3. Low-power multi-core architectures rely on the inherent parallelism of bio-signal processing applications to divide the workload among different cores and reduce the system clock frequency. This timing constraint relaxation allows for lowering the supply voltage. This technique has been widely exploited in the literature [125] [86] but its application is reaching a reliability limit. In fact, SRAMs typically embedded in bio-signal processing architectures start to suffer from errors when operating at ultra-low voltages [126], posing a hard limit on the voltage range that can be safely employed.

In addition, the workload profile of bio-signal processing applications is dictated by the sampling frequency of the sensed data, which is typically in the order of few hundreds of

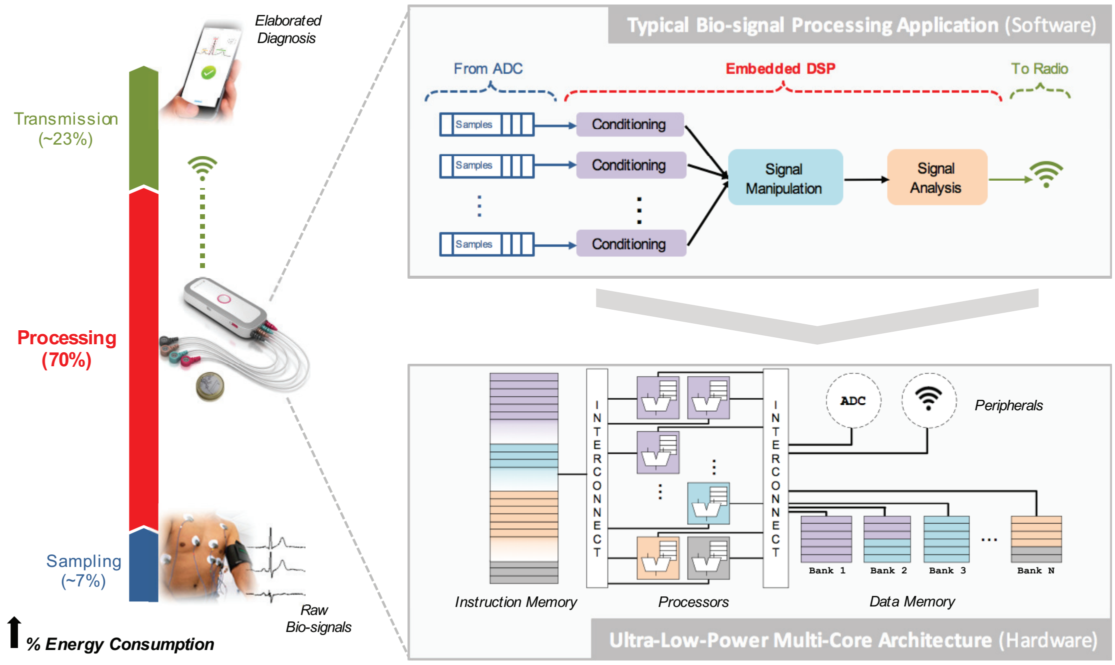


Figure 4.1 – Energy consumption breakdown of a biomedical monitor, scheme of a typical bio-signal processing application and generic block diagram of a state-of-the-art ULP multi-core architecture.

hertzs. Even when supplied with the minimum reliable voltage, highly optimized multi-core architectures can deliver the necessary throughput within a shorter time than the sampling period, leading to short but recurrent intervals of inactivity during which the processing platform remains idle but leaking. This power consumption could be reduced by employing advanced power management policies allowing for low-power sensing modes. To this end, different solutions have been proposed in the literature based on the utilization of different voltage domains or aggressive power gating. Nevertheless, the former entails an increase of the design complexity necessitating the integration of voltage level shifters while the latter is unfeasible in the bio-signal processing domain due to the short length of the idle periods, which would not allow to store and recover the system state in persistent memories.

4.1.1 Contributions and Outline of this Chapter

In this chapter I present two enhanced ULP multi-core architectures that allow to obtain further energy savings by exploiting application-specific opportunities. More precisely, I propose important changes at the memory subsystem level in order to support more energy efficient regimes during idle periods. In the first half of the chapter, I study the utilization of hybrid memory banks, which combine Standard Cell Memories (SCM) with traditional SRAM memories in order to support ultra-low system-level voltage regimes. In the second part of the chapter, I introduce a completely redesigned memory subsystem consisting on a 2-level hierarchy that includes a low latency Non-Volatile Memory (NVM), which enables fine-grained power-gaiting during idle periods. In particular, the most relevant contributions of this chapter are the following:

ULP multi-core architecture featuring SCM-based hybrid-memory

- I propose the utilization of a new hybrid memory subsystem in order to support low-voltage regimes. The banks of the new memory subsystem are composed of a small reliable memory partition implemented as SCM and a large region realized as regular SRAM.
- I extend the synchronization technique presented in Chapter 3 in order to support the new power management policy. The synchronizer unit is modified accordingly to seamlessly orchestrate the transitions between different working regimes.
- I study the best partitioning of the hybrid banks in terms of power consumption and area footprint. Obtained results show that, for the target bio-signal processing applications, a small amount of only 64 Bytes of SCM (2 words per bank) suffices to obtain large energy gains with an area footprint below 0.1%.
- Energy-wise, the new ULP architecture can improve its efficiency by up to 50% while spending more than 90% of the time in low-power mode.

Nano-engineered ULP multi-core architecture featuring a 2-level NVM-based memory subsystem

- I propose a fully re-designed 2-level memory subsystem. On the persistent level, I employed emerging low-latency low-voltage NVM based on *Spin-Transfer Torque RAM* (STTRAM) and Resistive RAM (RRAM). On the volatile level, I include a full-custom set of small page buffers that collectively act as a cache.
- I study and characterize the resulting ULP multi-core platform which is modeled to be fabricated employing emerging 3D monolithic integration.
- I describe the necessary architectural changes such as the inclusion of a lightweight Memory Management Unit (MMU) that manages page transfers between the NVM and the volatile buffers, or the modification at the synchronizer level to orchestrate the cores execution.
- I detail a new power management policy that allows seamless fine-grained power-gating of the entire architecture during periods of inactivity.
- The obtained simulated results show that the resulting architecture would reduce its area footprint by up to 5x and improve its energy efficiency by up to 5.42x.

The remaining of the chapter is organized as follows. First, Section 4.2 describes the ULP architecture featuring the SCM-based hybrid memory subsystem. Then, in Section 4.3, the nano-engineered architecture featuring the NVM-based 2-level memory subsystem is presented. Finally, Section 4.4 concludes the chapter summarizing the main achievements.

4.2 Proposed ULP Multi-Core Architecture Featuring SCM-based Hybrid Memory

Current most optimized platforms for bio-signal processing, such as the ones previously presented in Chapter 3, usually require a small fraction of the time spent in data acquisition to make all the necessary computations. As a consequence, the workload profile of bio-signal processing application presents periods of low activity, where only data collection is performed. System clock frequency scaling is sometimes proposed to reduce the dissipated dynamic energy but in general, voltage cannot be reduced under certain levels.

In this work, I propose a novel ULP multi-core architecture for bio-signal 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. The 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 [126]. 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 [24], due to the relatively large storage requirements of biomedical DSP applications.

Stemming from these observations, I 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, implemented as SCM, can be accessed 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 can be reliably utilized.

This strategy goes beyond DVFS, by trading off the voltage supply level with the memory portion which can be reliable accessed at the given operation mode. 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, the presented approach requires a single voltage domain, avoiding the design overheads of multi-V_{dd} designs [127]. Moreover, 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.

The 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 thanks to the strategies described in Chapter 3.

4.2.1 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 [128]. At the architectural level, the support of low-voltage operation modes is a widely used strategy to increase the energy efficiency of processors [129], 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 [130], that efficiently support the most frequent operations of a target domain. In the WBSN context, the authors of [112], [123] and [18] 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 [131].

A more generic approach adopted in the proposed architecture 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 [14, 132], is particularly effective in the bio-signal DSP scenario, where multiple signals are usually acquired in parallel and processed within a time window [112, 133].

Dynamic Voltage Frequency Scaling (DVFS), carried out by adjusting the performance and the power consumption at run-time according to the workload [125] [86], 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 [20]. Nonetheless, the reliability of SRAMs decreases when operating at ultra-low voltages [126], posing a hard limit on the voltage range that can be safely employed. To overcome such a problem, the authors of [133] propose a system where different voltage domains are used for computing and storage resources. As opposed to the this work, these choices mandate the use of voltage level shifters, that present a non-negligible area [127].

A striking alternative is offered by specialized SRAM implementations that, while larger than standard six-transistors (6T) SRAMs [134] [126], can reliably operate at extremely low V_{dd}. Similarly to [135], [24] and [136], herein I 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 [136] and [24], the proposed solutions do not incur in any error on the computations related to the 6T memory at low voltages. Moreover, compared to [136] and [135], I employ only a single and tunable voltage domain for the entire system, resulting in a simpler and leaner implementation.

The run-time management of multiple resources under tight run-time and memory constraints is a challenging task. To this end, the authors of [116] 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 [117] are also resource-intensive, thus not suited for low-power computing architectures such as the ones embedded in biomedical monitors. As exposed in Chapter 3, run-time synchronization among different cores is supported with dedicated instruction set extensions, presenting a small area and timing footprint. In [84] and[124], synchronization is only supported to manage parallelism. Herein, I extend this methodology to orchestrate both parallel execution on multiple resources and dynamically set the working operating voltage. In this last respect, leveraging a domain-specific heterogeneous memory system, I aim at going beyond classic DVFS, trading off the accessibility of resources (in addition to the operating frequency) with power consumption.

4.2.2 Target Multi-Core 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 already described in Chapter 3. Both energy-saving strategies are embedded in the target 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 4.2. 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 [114]. On the other hand, in

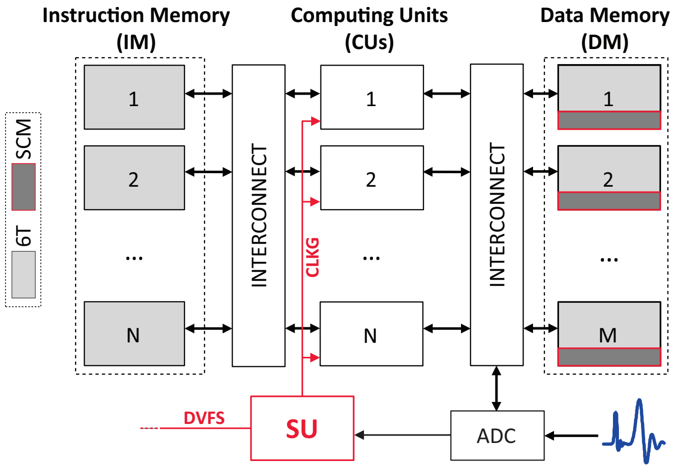


Figure 4.2 – Target architecture, featuring hybrid DM banks and HW synchronization unit (SU).

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 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 [124].

A *Synchronization Unit* (SU in Figure 4.2) as the one described in Section 3.3.2.2 of Chapter 3 is employed. This synchronizer has been extended to, in addition to orchestrate the execution of the system, dynamically select the voltage supply level and, therefore, the operating mode. The synchronizer pauses and resumes cores, either after data-dependent branches (to recover lock-step execution) or to manage producer-consumer relationships. Moreover, the it also dictates the voltage supply of the platform. At the high-Vdd *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-Vdd *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.

4.2.3 Hybrid Memory Management

Considering typical sampling frequencies for biomedical signals (typically around few hundreds of Hertzs), the time needed to acquire a window of samples exceeds the time to perform the required computation. Therefore, the workload profile of bio-signal processing 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 4.3, 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 do not perform any activity. Memory elements

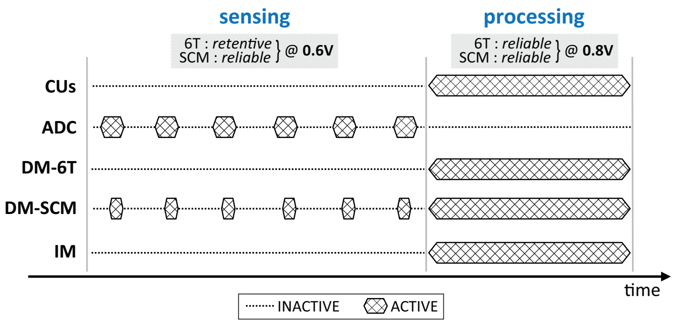


Figure 4.3 – Active/inactive architectural elements in *sensing* and *processing* states with related hybrid memory behavior.

beside the SC-DM (i.e. 6T-DM, cores register files an system registers) are not accessible, but their content is reliably retained. Once the ADC has transferred the desired number of samples to the data memory, the system switches to *execution* mode (cf. Figure 4.3), 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. It also presents a larger data memory footprint with respect to the sensing mode, due to the intermediate data being generated during processing. 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, in this work I considered 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 the target CMOS technology, the SCM portion of the DM is able to reliably operate down to 600 mV, while the 6T portion of the DM can be reliably accessed at a minimum level of 800 mV. Due to the utilization of a single voltage domain, the 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.

4.2.3.1 Synchronization Strategy

To determine the dynamic voltage supply level of the platform, as well as to properly clock-gate individual cores, I propose a hybrid hardware/software (HW/SW) synchronization mechanism extending the one described in Chapter 3 by also including the management of multiple operation modes. Its hardware support is provided by the above-mentioned synchronizer unit, 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. One synchronization point is therefore required for each data-dependent branch and each producer/consumer relationship. In addition to the synchronization instructions, 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).

4.2.3.2 Software Adaptations and Mapping

As described in Section 3.4.2.2 of Chatper 3, 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 graphic representation and a time diagram of this run-time behavior is also depicted in Figure 4.4-a.

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 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. Figure 4.4-b, represents graphically a producer-consumer relationship, the run-time sequence of issued instructions and the differentiation between sensing and processing phases.

To map an application into the proposed platform starting from an equivalent single-core implementation, the program flow must be partitioned into phases that can be executed in parallel in a pipelined manner (c.f. Section 3.4.2.3 of Chapter 3). 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 cores are assigned for the “conditioning" phase in Figure 4.1). Subsequently, the custom synchronization instructions are properly placed to manage data-dependent branches and producer/consumer relationships. Finally, linking directives indicating the manually performed application partition are provided to place the IM content referring to different phases in disjoint IM banks, which subsequently reduces access conflicts.

4.2.3.3 Hardware Support: Synchronization Unit

The aforementioned Synchronization Unit 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 receives an ADC interrupt line for each of the sensed channels and the stall, sleep and wake-up pins from each of the cores. The

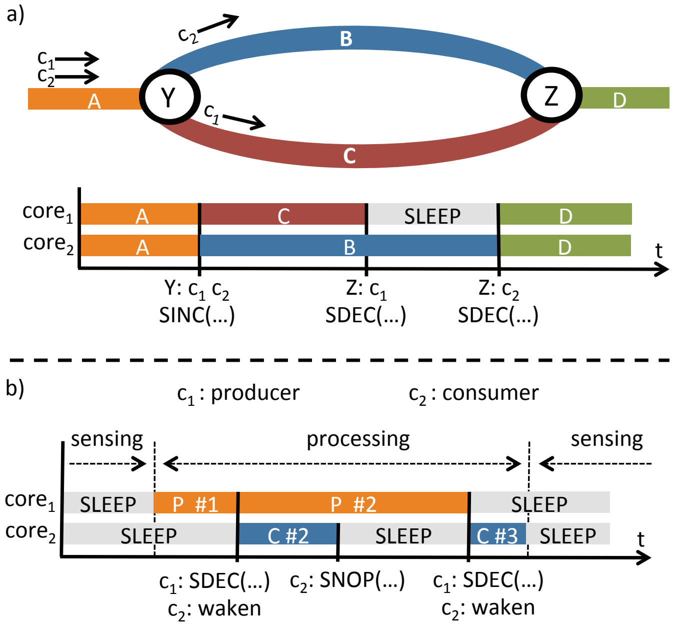


Figure 4.4 – Synchronization in a diverging data-dependent branch (a) and in a producer-consumer relationship processing windows of 2 samples (b).

synchronizer is composed by a sequential and a combinational part, which are detailed in the following and whose behaviors are depicted from a high level of abstraction in the flowcharts in Figure 4.5. On one hand, the sequential (clocked) logic is responsible for controlling the transitions between sensing and processing modes (cf. Figure 4.5-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 4.4-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-Vdd 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 4.5-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.

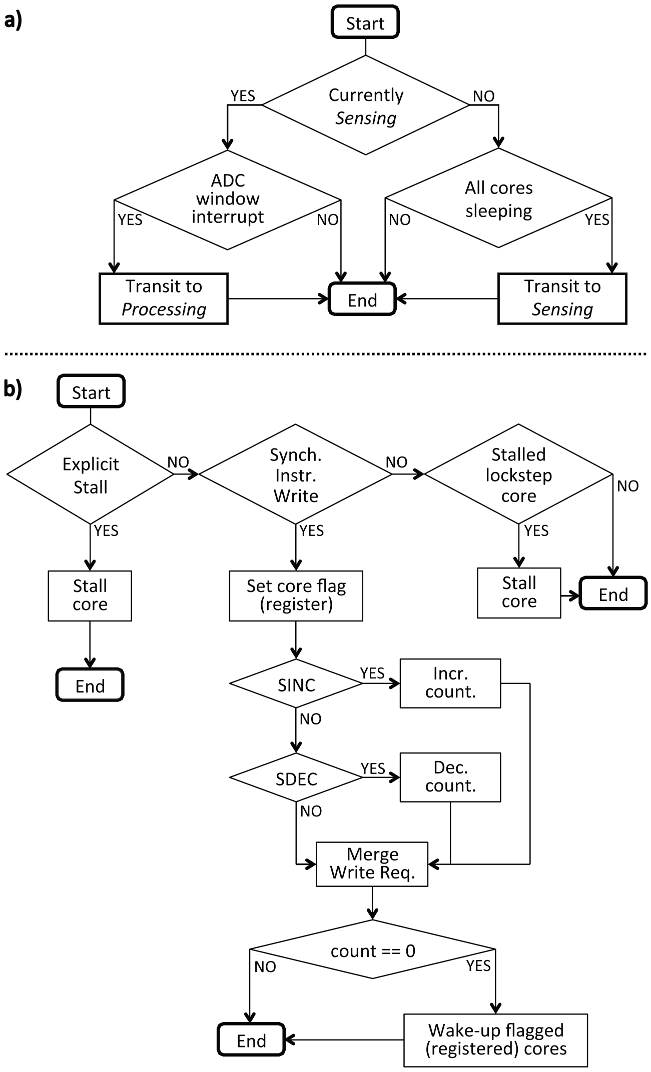


Figure 4.5 – Flowchart describing the synchronizer's behavior: (a) Clocked logic governing transitions between platform modes. (b) Combinational part orchestrating execution while in processing mode.

4.2.4 Experimental Set-up

In this section I present the chosen set-up and simulation framework. Then, I briefly described the bio-signal processing benchmark suite employed to assess the performance of the target platform.

4.2.4.1 Simulation Set-up

I considered a target system composed by 8 ULP TamaRISC cores (c.f. Sectio 3.2.1.2) as the ones employed in the architectures described in Chapter 3. Nonetheless, different cores can be embedded in the proposed system, as long as they allow extensions to incorporate the custom synchronization instructions described in previous sections. 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.

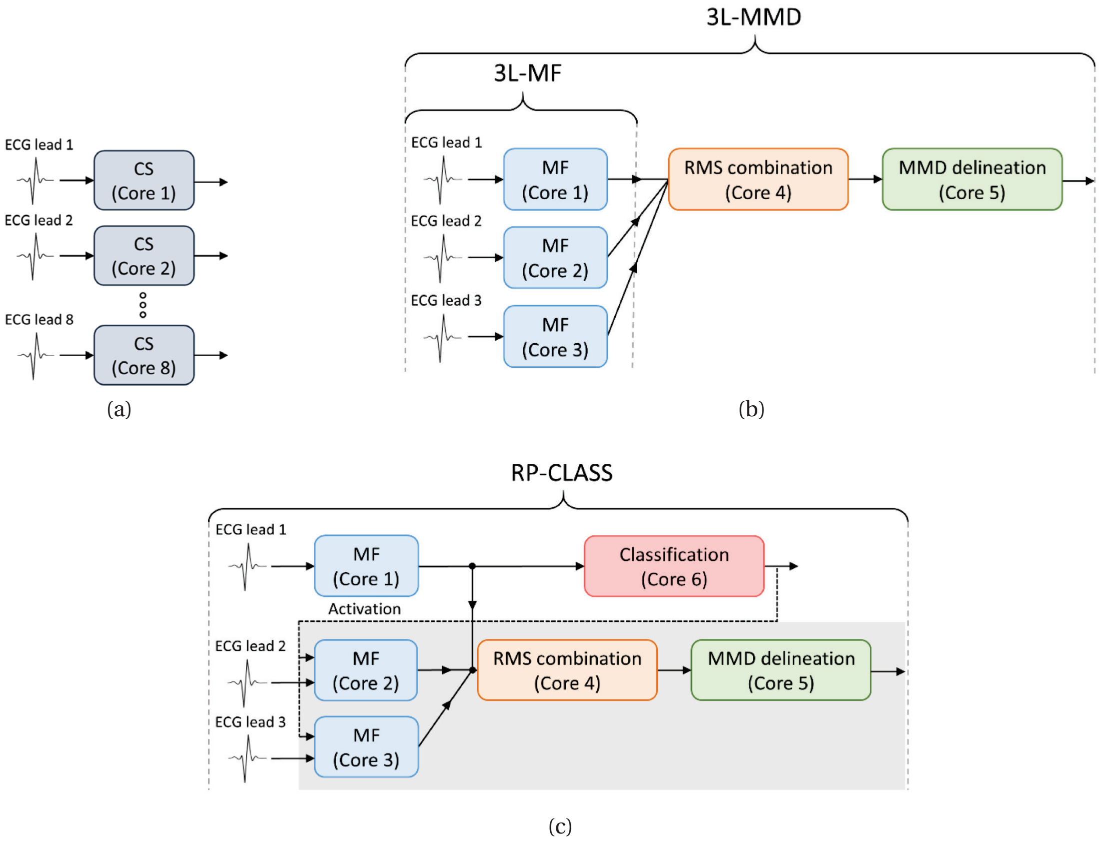
The system clock frequency in the processing mode is 20 MHz (the maximum possible in the chosen technology for the target system with 800 mV supply voltage), while in sensing mode the clock is set at 10 KHz to minimize the dissipated dynamic power due to the clock-tree. The resources that are not required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

Similarly to [124], the developed experimental framework combines detailed post-layout characterization of the system with faster cycle-accurate simulations of complex bio-signal analysis applications. First, the energy characterization of the architectural components of the multi-core system is performed at a 40 nm technology node through an EDA toolchain: Design Compiler from Synopsys and Encounter from Cadence are used in the synthesis and place-and-route steps, while Modelsim from Menthor Graphics is employed to retrieve switching activity of the platform when executing synthetic benchmarks. In a later step, the obtained energy values are used to parametrize a cycle-accurate SystemC simulator of the platform, allowing the evaluation of its energy efficiency when executing real-world applications under different architectural configurations.

4.2.4.2 Bio-signal Processing Benchmarks

Four bio-signal processing benchmarks, which are widely used in the field of electrocardiogram embedded analysis [124], [38], [122] [11] are considered. These applications present different levels of complexity and parallelism as well as diverse 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 ECG processing (c.f. Section 2.2.4 of Chapter 2). The algorithm used in this benchmark utilizes a software version of the energy-efficient pseudo-random number generator introduced in [38] 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 the proposed implementation, eight ECG leads (8L) are processed in parallel employing all the cores of the platform as depicted in Figure 4.6a.
- **Morphological Filtering (3L-MF)**: Morphological filtering removes noise from ECG signals as described in Section 2.2.1.2 of Chapter 2. Herein, I considered an optimized



(a) (b)

(c)

Figure 4.6 – Block schemes of benchmark applications: a) 8L-CS; b) 3L-MF and 3L-MMD; c) RPCLASS. Conditionally activated blocks are indicated with grey background.

version of this algorithm [10], which removes both low and high frequency noise components. This benchmark filters in parallel ECG signals from a standard three-channel acquisition using three computing cores as depicted in Figure 4.6b. 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 delimits the starting, and a peak points of the three ECG main waves (c.f. Section 2.2.2.2 of Chapterch:sw). 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 processed to search for the desired points. This benchmark 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 as depicted in Figure 4.6b.
- **Selective ECG processing (RP-CLASS)**: This benchmark, detailed in Section 2.3 of Chapter 2, embeds a neuro-fuzzy classifier that detects abnormal heartbeats. When a detec-

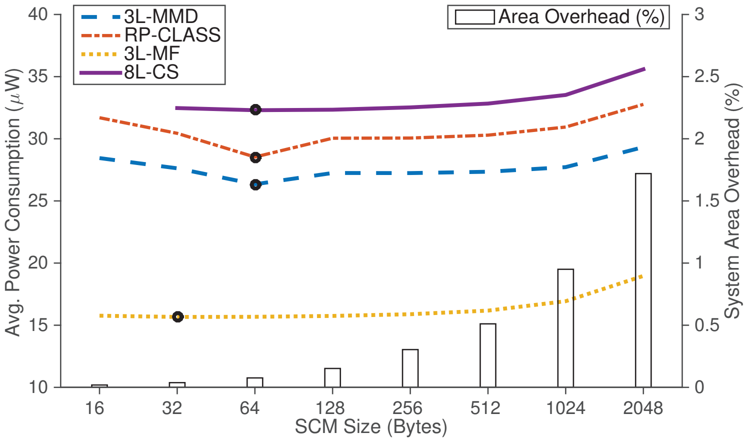


Figure 4.7 – Power consumption and area overhead varying the SCM-DM size. The minima are highlighted by a rounded marker.

tion 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 heartbeats, three-channels filtering and delineation is performed. This benchmark presents a complex structure, requiring lockstep execution only in some situations and a sophisticated control flow across cores. As depicted in Figure 4.6c, RP-CLASS utilizes 6 cores of the platform and benefits from both proposed synchronization mechanisms.

4.2.5 Experimental Results

In this section I first make a detailed exploration to choose an optimal balance between the SCM and the 6T memory regions composing the data memory sub-system. Then, I briefly discuss on the run-time performance obtained by the architecture for the studied benchmarks. Finally, I comparatively evaluate the energy efficiency of the studied architecture featuring the proposed hybrid memory hierarchy with synchronization support.

4.2.5.1 Reliable Memory Requirements

In the first round of experiments I explored the energy efficiency of the multi-core platform when different sizes of highly-reliable SCMs are employed (Figure 4.7). 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 [24] compared to the SCM design in [137] 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 [24]. The minimum operating voltage point was measured by the authors of [24] over nine chips and the results show that for the majority of the chips, the SCMEM operated correctly at voltages below 0.4 V and on average it has 400 mV lower minimum operating voltage point than the 6T memory. However, the worst case scenario have been considered, i.e. the highest minimum voltage for both SCMEM and 6T among the different measured chips, which conservatively lead to 600 mV for sensing and 800 mV for processing.

As shown in Figure 4.7, 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, I 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 (≈ 0.1%) with respect to 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 100 ns in the experiments, taking into account wide margins with respect to silicon implementations [123, 138].

4.2.5.2 Runtime Performance

Table 4.1 reports the most relevant workload characteristics obtained while executing the four bio-signal processing benchmarks considered in this work on the target multi-core platform. Three main conclusions can be drawn from these numbers.

First, it can be observed the small overhead caused by the insertion of synchronization instructions, in terms of run-time as well as code size (Synch. Cycles and Code Overhead in Table 4.1, respectively). Second, the obtained instructions per cycle (IPC) is always over 1, highlighting the exploited parallelism by the synchronization technique. Finally, 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.

4.2.5.3 Power Consumption Evaluation

Table 4.2 reports the obtained values for leakage and dynamic power for both processing and sensing phases. In all cases, leakage power is effectively reduced by ≈ 40% when transitioning to the state-retentive sensing mode. In the context of biomedical monitors applications, this

Table 4.1 – 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
Synch. cycles (%)	0	1.67	0.96	0.62

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 the proposed proposed solution, I compared it with two different baseline systems. The first baseline system (*no Hybrid* in Figure 4.8) does not implement the hybrid memory subsystem and it is always running at the higher voltage level of 800 mV, while still employing synchronization for managing lock-step execution and efficient producer-consumer waiting. In the second case (*no Sync* in Figure 4.8), I employ active waiting instead of clock gating to manage producer-consumer relationships and lock-step execution is disabled, but I 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 last setting I reduce access conflicts by assigning different IM and DM banks to each processor, even when they execute the same computing phase.

Table 4.2 – Leakage and dynamic power of the target platform for the bio-signal processing benchmarks during *sensing* and *processing* phases.

	Avg. Power Consumption (uW)			
	Processing		*Sensing*	
	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

Figure 4.8 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. Second, synchronization can

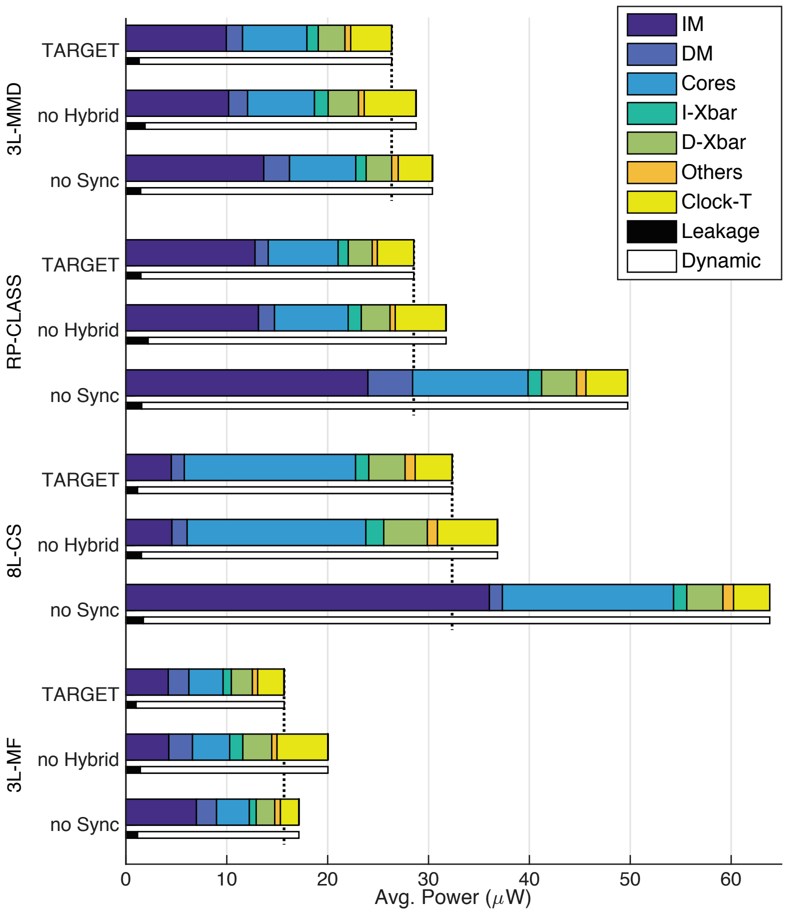


Figure 4.8 – Power consumption of the target and baseline systems (*no Hybrid* and *no Sync*) for the considered bio-signal processing benchmarks.

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%.

The combined benefits of efficient parallel execution and operation modes (shown in Figure 4.8) 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.

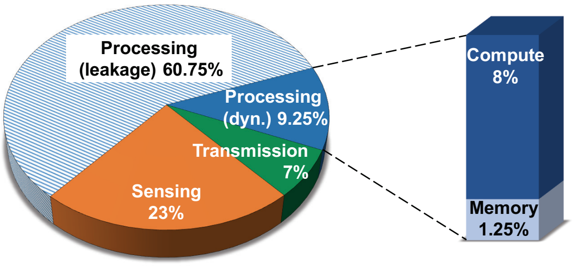


Figure 4.9 – Power consumption breakdown on a WBSN-based biomedical monitor executing a multi-channel biosignal processing application. Values are computed based on [124], [139] and [140]

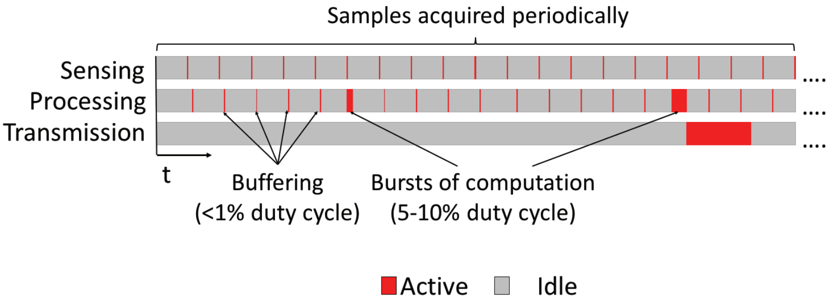


Figure 4.10 – Activity of different contributors of a typical biomedical monitor while performing bio-signal DSP

4.3 Proposed Nano-engineered ULP Multi-Core Architecture featuring a 2-level NVM-based memory subsystem

With the reduction of the energy required by signal transmission and data sampling, the efficient implementation of the digital signal processing (DSP) stage is a key aspect in order to minimize the power consumption of biomedical monitors. As shown in Figure 4.9, most of the power dissipated by these devices is due to the processing of the acquired samples. To perform complex bio-signal processing routines within an ultra-low-power envelope, embedded digital platforms must be carefully tailored to the specific domain and its workload characteristics. For instance, the state-of-the-art electrocardiogram (ECG) compression and filtering algorithms experience extended idle periods (>90% of the inter-signal arrival time), when executed on a common bio-signal processing platform [11] [124]. This is primarily due to the low sampling frequency of the acquired signals (with low processing requirements). The long idle periods (cf. Figure 4.10) increase the leakage power of the overall system, which reaches up to 86% of the total power consumption.

Stemming from these observations, herein I propose a nano-engineered bio-signal processing architecture that addresses the previously mentioned limitations. The platform leverages the benefits of emerging nanotechnologies, focusing on low-voltage, non-volatile memory structures (STTRAM [141], RRAM [142]), in conjunction with ultra-dense, fine-grained 3D integration (termed monolithic 3D [143], [144]).

In particular, I introduce a new ultra-low-power and domain-specific platform, aggressively exploiting the characteristic of upcoming technological breakthroughs to fully exploit application-level energy saving opportunities. The key aspect of this new architecture is the fully redesigned memory subsystem enabled by the utilization of emerging technologies such as the ones previously mentioned. In addition, I present a novel system management policy, which allows the low-overhead power gating of computational as well as storage elements to substantially decrease both active and leakage power consumption.

The achieved power savings derive from the adopted synergistic approach. NVM enables power gating of the system at idle times. The proposed architecture considers the domain application nature, and the NVM limitations by building a 2-level memory hierarchy (latch-based cache-like level 1 and NVM-based level 2). The ultra-dense monolithic 3D integration enables the efficient transfer (1-cycle transfer) between the memory levels for a quick power gating application. The required interconnection density is provided by Monolithic 3D integration resulting in negligible area overheads with respect to the ones achievable by state-of-the-art Through Silicon Vias (TSVs) [145].

4.3.1 Technology Background

Ultra-low-power bio-signal analysis platforms have to abide to conflicting requirements, as complex applications must be supported within real-time constraints, at the same time retaining a high flexibility, which allows to execute a wide range of bio-signal processing algorithms. In this section I describe how non-volatile memories and 3D-integration fabrication processes become technological enablers to achieve this goal at high degree of energy efficiency, targeting multi-core WBSNs architectures.

4.3.1.1 Non-Volatile Memories

Emerging low-voltage non-volatile memories (NVM) have caught significant attention in the last years, and have been explored in various computing domains (from embedded to high-performance) [146] [147] [148] [149] [150] [151] [152] [153] [154]. In this work, I focus on STTRAM and RRAM, but the same approach can be extended to other low-voltage NVMs such as Conductive-Bridge RAM (CBRAM) [142]. FLASH memory is also examined to illustrate the efficacy of emerging memory technologies at the system level. Table 4.3 shows a qualitative comparison of emerging memory technologies versus key representatives of current embedded memory (volatile and nonvolatile) technologies. Three important considerations can be derived from it. First, non-volatile memories provide significant area savings over traditional SRAM designs, due to their small cell size. Second, their non-volatility provides massively lower leakage power on a system level. The third, and most important, observation

	SRAM	STTRAM	RRAM	FLASH
Cell Size (F^2)	Big (~ 120)	Small (~ 6)	Small (~ 4)	Small (~ 4)
Read Latency (ns)	Low (< 10)	Low (< 10)	Low (< 10)	High (> 100)
Write Latency (ns)	Low (< 10)	Med (10's)	Med (10's)	High ($> 10^4$)
Read Energy (pJ/bit)	Low (< 1)	Low (< 2)	Low (< 2)	High (> 100)
Write Energy (pJ/bit)	Low (< 1)	Med ($1 - 10$)	Ned ($5 - 20$)	High (> 1000)
Leakage	High	Low	Low	Low
Volatility	Yes	No	No	No
Endurance (writes/bit)	High ($> 10^{15}$)	High (10^{15})	Med (10^{12})	Low (10^6)
Availability	Yes	Mostly Experimental	Mostly Experimental	Yes

Table 4.3 – Comparison between different memory technologies. Latency and energy values are observed from literature

is that they enable aggressive energy-saving run-time strategies. In fact, they provide the ability to completely power down a digital circuit, while still retaining data integrity. This characteristic can potentially result in significant system energy reduction, especially when the total execution time is dominated by idle periods, as in the considered case. Although cycling endurances of STTRAMs and RRAMs are significantly lower than those figures for SRAM memories (particularly RRAM), architectural changes to the system design can still be made to exploit their area and energy benefits without any performance loss. A second drawback of low-voltage NVMs is that their write latencies are much higher than that of volatile alternatives. This aspect may pose a challenge for high-performance applications, but is less crucial in an ultra-low power scenario, where deeply scaled voltage supply levels are adopted, resulting in low operating frequencies. For example, a system cycle time of 50ns (20MHz), encapsulates both the read and write latencies of the considered NVM memories (read 1-2ns, write 10-20ns) [141], so that read and write operations consume a single cycle. Moreover, the low operating frequencies can be exploited to further lower the read and write energy of the considered memory technology, without any device-level modifications. This is indeed key to increase the energy-efficiency of low-power computing systems. The required write current (or voltage) can be relaxed by increasing the write pulse width. This relaxation is accompanied by a reduction of voltage (or current), due to the I-V characteristics of the access transistor. In this work this methodology is followed to reduce the write energy of STTRAM and RRAM, while doing the circuit level characterization, based on the following relationships.

STTRAM writing and reading current can be tuned to benefit from low operating frequencies, based on the relationship between writing current (I_c) and pulse width (t: time needed to change the magnetic material state) [155]:

$$I_c = I_{co}\left(1 - \frac{1}{\Delta}\ln\left(\frac{t}{\tau_0}\right)\right) \tag{4.1}$$

where I_{co} is the threshold write current (STTRAM-material dependent), Δ is the thermal stability factor (STTRAM-material dependent and affects the retention time of the memory) and is the nominal switching time (1ns). This relationship enables the tuning of write current, hence reducing the write energy of STTRAM.

Similarly, for RRAM, the applied write voltage can also be reduced to relax the pulse width [156]:

$$\tau_{set} = \frac{\Delta\Phi}{A}\exp\left(\frac{E_A - aqV}{kT_0 + \frac{V^2}{8\rho k_{th}}}\right) \tag{4.2}$$

where $\Delta\Phi$ is the change in conductive filament required for a sufficient change of resistance, A is the filament diameter, E_A is the activation energy required to set, ρ is the electrical resistivity of the conductive filament, k_{th} is the thermal conductivity of the conductive filament, and T_0 is the ambient temperature. However, τ_{set} increases faster than quadratically with decreasing V. Thus, the write energy increases as the voltage is increased since $E_{write} \propto V^2\tau_{set}$.

4.3.1.2 3D Integration

Vertical (3D) integration of circuits, whereby circuits are stacked vertically over one another, offers benefits ranging from processing and circuit-level optimization to architecture and system-level optimization. In this work I capitalize on the following key benefits of 3D integration:

1. Massive connectivity between various components, that are otherwise infeasible or very hard in planar 2D integration (with reasonable routing overhead).
2. Heterogeneous technology integration of the various tiers, to maximize the efficiency of the overall systems (area, performance, energy, cost, etc.).

Traditionally, 3D integration relies on through-silicon via (TSV) technology. However, these TSVs can occupy significantly large area footprint [157]. Moreover, they require large keep-out-zones where no transistors may be placed, further inhibiting the ability for dense integration. To achieve fine-grained integration, one can rely on monolithic 3D integration, whereby each vertically-stacked tier of circuits is fabricated directly over previous fabricated tiers [158] [143]. This technology enables using traditional inter-layer vias (ILVs) to connect each tier of circuits. The significantly smaller via size and absence of keep-out-zones, compared to TSVs, allows the design of a massive vertical connectivity, without prohibitively large area overheads.

Such solution is adopted in the proposed platform to link non-volatile and volatile memory elements. The low overhead of data/instructions transfer granted by ultra-wide ILV-based interconnect, coupled with the high locality characterizing the applications, result in an energy-efficient and low-overhead nano-engineered architecture.

Contrary to TSV-based 3D, monolithic 3D integration requires stacked subsequent tiers of circuits to be fabricated with low temperature (<400°C) to preserve the performance of the ones already finished. The considered NVM can be manufactured at the required low temperature [159]. However, silicon requires high temperature in the fabrication process (temperature exceeding 1000oC is needed for various steps such as dopant activation [160]), which renders it unusable in any tier, except the first one. While there are efforts to overcome such limitation [158] [161], carbon nanotube field-effect transistors (CNFETs) naturally overcome this temperature barrier since all processing steps on the main wafer are below 200°C. Systems that monolithically integrate non-volatile memories and CNFETs on silicon CMOS technology have already been experimentally demonstrated [144].

I use low-voltage NVM and monolithic 3D integration to reduce energy consumption and overall area footprint in combination with a state-of-the-art low-power multi-core architecture for bio-signal processing. CNFETs are monolithically integrated on top of traditional silicon CMOS logic to construct the non-volatile memory system cells. Although CNFETs have been projected to provide an order of magnitude energy-efficiency improvement over traditional silicon CMOS [162], I only use CNFET in the NVM access circuitry (e.g., row decoders, selection transistors) found in upper tiers, thus, neglecting the potential energy-efficiency impact of CNFET in those circuits.

4.3.2 Proposed Nano-Engineered Architecture

To fully exploit the synergistic combination of ultra-low-power architectures for biomedical monitors and new technological devices, I propose a novel system that enables high energy efficiency with a low area footprint, thanks to the new opportunities provided by the low-energy NVM and ultra-dense 3D integration technologies.

4.3.2.1 System Architecture

The target platform, presented in Figure 4.11, consists of a multi-tier chip with the bottom tier hosting the processing logic and the upper tier hosting the non-volatile storage. High-density vertical connections link the tiers among them.

The architecture of the bottom tier is similar to the one introduced in Chapter 3. It features eight low-power RISC cores interfaced to 16 data memory and 8 instruction memory banks. The data interface is 16-bit wide, while each instruction word is 24-bits wide. The links between cores and memory banks are implemented by a mesh-of-trees logarithmic interconnect [114], that provides single-cycle access to the banks and perform arbitration in case of conflict among

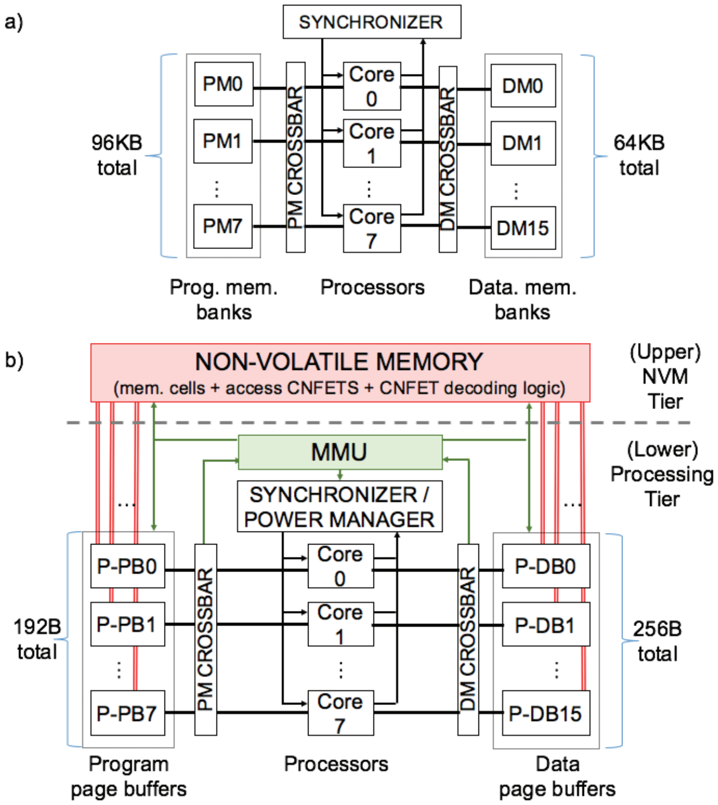


Figure 4.11 – Block diagrams of the a) multi-core architecture introduced in Chapter 3, featuring volatile SRAM and b) the target NVM-based platform

several memory requests. In the baseline architecture, the entire program and data content reside in the above-mentioned banks, resulting in a flat (SRAM-based) memory structure. Conversely, in the proposed nano-engineered system those volatile memories are realized as page buffers of only few words each, collectively acting as a cache for the non-volatile storage residing in the upper tier. The small size of the buffers, in conjunction with the high-density inter-tier connectivity enabled by monolithic integration, enables data transfers in-between buffers and NVMs with low overheads (as detailed later). It is also crucial for the efficient implementation of the *deep-sleep* mode, in which both the top (non-volatile) and the bottom (volatile) tiers are power gated during idle periods.

The architecture proposed in this section can also leverage the benefits of the multi-core approach introduced in [84]. By allocating workloads on several resources, a low system clock frequency of only few MHzs can be adopted, enabling aggressive voltage/frequency scaling without violating real time constraints. Crucially, a low frequency operating point also allows single-cycle transfers to and from the non-volatile sub-system, even when the longer latencies of RRAMs and STTRAMs with respect to SRAMs are considered.

Furthermore, the system supports multiple-instruction multiple-data (MIMD) and single-instruction multiple-data (SIMD) execution modes. MIMD is beneficial for bio-signal applications consisting of a sequence of algorithms applied over a stream of data [10], while SIMD increases the efficiency when the same DSP is applied on multiple sources, by dramatically reducing the number of required instruction fetches [84].

A hardware synchronizer unit derived from the one described in Section 3.3.2.2 orchestrates the run-time behavior of the system. First, this unit manages SIMD and MIMD execution, keeping track of data-dependent branches and producer-consumer relationships among cores, respectively. Second, it interfaces with the Memory Management Unit, stalling cores who experience a miss in the data or instruction buffers. Third, it allows the system to transition to (and recover from) deep-sleep sensing, signaling the processors to store and read back their state in the non-volatile storage. These last two features are here introduced for the first time and become essential for the correct operation of the proposed NVM-based system.

4.3.2.2 Memory Management Approach

The full-custom page buffers have been implemented as arrays of latches that incorporate a direct input line connected to each bit cell allowing a single-cycle page storage or readout. These massively parallel transactions require an ultra-wide connection with the NVM. While this arrangement can be realized with 2D, 3D TSV or monolithic 3D integration, the area overhead favors monolithic integration, as shown later in the experimental evaluation.

Targeting low-voltage RRAM or STTRAM NVMs significantly reduces the access energy with respect to standard solutions, such as FLASH-based NVMs. Therefore, the size of volatile buffers can be much smaller than the application data and instruction size, avoiding full shadowing and relying instead on *on-demand* page transfers to volatile banks. A proper selection of the size of the page buffer is therefore essential to maximize efficiency, since the memory power consumption will depend on the page transfer rate (which decreases when buffers are large) and the leakage and energy per access of these caches (which, on the contrary, favors the usage of small buffers).

The decision of issuing a page transfer is taken by a light-weight Memory Management Unit (MMU), which, in conjunction with the synchronizer, coordinates the page transfers between the NVM and the buffers, stalling the cores' execution while transfers are in progress. This unit works as a simple content-addressable-unit (CAM), which makes the translation between the address of a word request and the location of the corresponding page, checking if it is available in any of the buffers. Otherwise, the unit selects the page to be replaced.

4.3.2.3 Deep-sleep Sensing

The run-time management of bio-sensing analysis systems based on non-volatile memories present a marked different with respect to volatile-based alternatives. In the latter case, goal of the power manager is usually to minimize the idle time by setting a clock frequency that barely meets real time constraints [84] [20]. Conversely, for platforms embedding NVMs higher operating frequencies are desirable during computing bursts, in order to reduce active time.

In fact, in idle mode, the full digital architecture can be power gated, while new samples are acquired. I term this state “deep-sleep sensing”. This strategy is possible thanks to the availability of persistent memory provided by the NVM. Before entering deep-sleep sensing, a copy of the application state, namely the the data in the page buffers and the processor registers, is transferred to non-volatile storage. At this point, both the memories and the processing elements can be safely power-gated. At power up, each processor reloads the content of its own registers and execution can seamlessly resume from where it was interrupted.

The transition from and to the power gated state is managed by monitoring the activity of each individual core and by the built-in ADC module. The system is power gated when all processors are idle and resumed when new samples are available.

4.3.3 Experimental Framework

The following sections show the performed evaluation to highlight the benefits of the proposed nano-engineered ultra-low power architecture for bio-signal processing system.

4.3.3.1 Circuit-level Characterization

A full physical design (through place and routing) of various components (i.e., processing core, SRAM-based memory, latch-based page buffers, and the integration of NVM memory with such system to account for the ILV energy and delay overhead) was performed using a 28nm process design kit (PDK) (1.0V VDD) to extract area, power and performance characteristics. Inter-component wiring was considered while extracting the previously mentioned characteristics, in all the studied 2D and 3D (TSV-based and monolithic) architectures. The operating frequency was set to 20MHz for the studied NVM-based configurations (c.f. Section 4.3.3.3). It is important to note that, while a platform featuring multiple voltage islands (e.g. higher voltage level for the NVMs) would lead to further energy savings on the logic side (i.e. processors, volatile memory, etc.), the overhead of the resulting platform due to its higher complexity would not compensate for the potential energy savings. In particular, this feature would require a more complex access circuitry for the non-volatile cells and therefore, both area- and energy-wise, this component would become less efficient. As a consequence, in this work, I consider a single-voltage domain targeting the nominal voltage given by the employed PDK.

For the considered NVM, the required read and write current values and pulse widths based on the equations 4.1 and 4.2 were identified to fully benefit from the long cycle time (50ns), while device-level parameters were obtained from the literature [142]. In particular, the write pulse width of the NVM cells was set to 25ns to account for the additional overhead spent in memory access circuitry. Then, detailed SPICE simulations were performed to deduce the applied voltage on the associated transistors with the NVM cells and the transistor width

	Dyn. Energy (pJ/bit)		Leakage (μW)
Processing core	10.9 (pJ/operation)		41.37
8x12 KB PM SRAM Bank	0.2 (rd)		3.53
16x4 KB DM SRAM Bank	0.23 (rd)	0.27 (wr)	1.90
24 B PM Page buffer	0.01 (rd)		6.81
16 B DM Page buffer	0.01 (rd)	0.02 (wr)	4.65
96 KB STTRAM PM	0.13 (rd)		1.05
64 KB STTRAM DM	0.13 (rd)	1.13 (wr)	0.66
96 KB RRAM PM	3.2 (rd)		3.46
64 KB RRAM DM	3.3 (rd)	6.7 (wr)	2.31

Table 4.4 – Parameters of various key components of the studied nano-engineered architectures

required to provide the current needed to program the considered NVM. Finally, the values deduced form SPICE were linked with NVSim [146] to estimate the corresponding parameters of the overall memory array (including memory interface circuitry). Table 4.4 summarizes the power consumption of the main blocks of the target system.

4.3.3.2 Bio-signal Processing Benchmarks

In order to evaluate the considered architectures, in this study I have employed the bio-signal processing benchmarks descried in Section 4.2.4.2. Hereafter I only provide a brief description of each of the applications:

- **8L-CS:** Eight-lead Compressed Sensing based on the algorithm described in Section 2.2.4 of Chapterch:sw. It is mapped on 8 processing cores.
- **3L-MF:** Three-lead Morphological filtering based on the algorithm described in Section 2.2.1.2 of Chapter 2. It is mapped in 3 processing cores.
- **3L-MMD:** Three-lead Morphological Delineation performs morphological filtering, RMS combination and delineation of the resulting signal based on the algorithm described in Section 2.2.2.2 of Chapterch:sw. It is mapped on 5 processing cores.
- **RPCLASS:** Selective advanced ECG processing triggered by an abnormality detector implemented as a neuro-fuzzy classifier described in Section 2.3 of Chapter 2. It is mapped on 6 processing cores.

4.3.3.3 Explored Nano-engineered Architectures

An exhaustive design-space-exploration (DSE) was performed according to two main characteristics of the studied architecture: memory volatility and 3D integration technology. As a result, the following four platforms were considered:

- **2D_Baseline:** As in [124], this platform only employs SRAM memories and a planar arrangement. Therefore, it does not include neither NVMs nor 3-dimensional integration of any kind. The system implements the synchronization mechanism described in Section 3.4.2.2 of Chapter 3 and is able to power gate unused program memory banks at boot time.

- **2D_NVM:** This platform adopts the architectural concept described in this work, while integrating the proposed memory subsystem, even including the NVM, in the same die as the processing logic and latch-based page buffers

- **3D_TSV:** In this platform NVM is placed on another tier than the cores and the page buffers, while through-silicon vias (TSVs) are used to connect those tiers. The TSV dimensions for this architecture are optimistically deduced from [163] assuming a 5 μm pitch (i.e., TSV keep-out zone).

- **3D_TARGET:** This platform is the target system where monolithic 3D integration is used to connect the processing logic tier with the NVM tier. The impact of monolithic 3D integration (i.e., ILVs) are taken into account for all analysis performed later.

The comparative evaluation does not consider volatile three-dimensional systems, as such strategy would not lead to an energy-efficient implementation. In fact, the in this case at least the upper layer would always be powered (even during idle periods) to save the system state, resulting non-zero power due to leakage. Dissipation can be reduced, but not eliminated, by putting the memory in a retentive state, that is, powering it at a really low level, in which it cannot be accessed but still holds the stored values. This approach, previously introduced in Section 4.2 is therefore in-between the 2D_Baseline and 3D considered architectures. Area-wise, it would also not be particularly appealing, as 85% of the area is devoted to memories in 2D_Baseline.

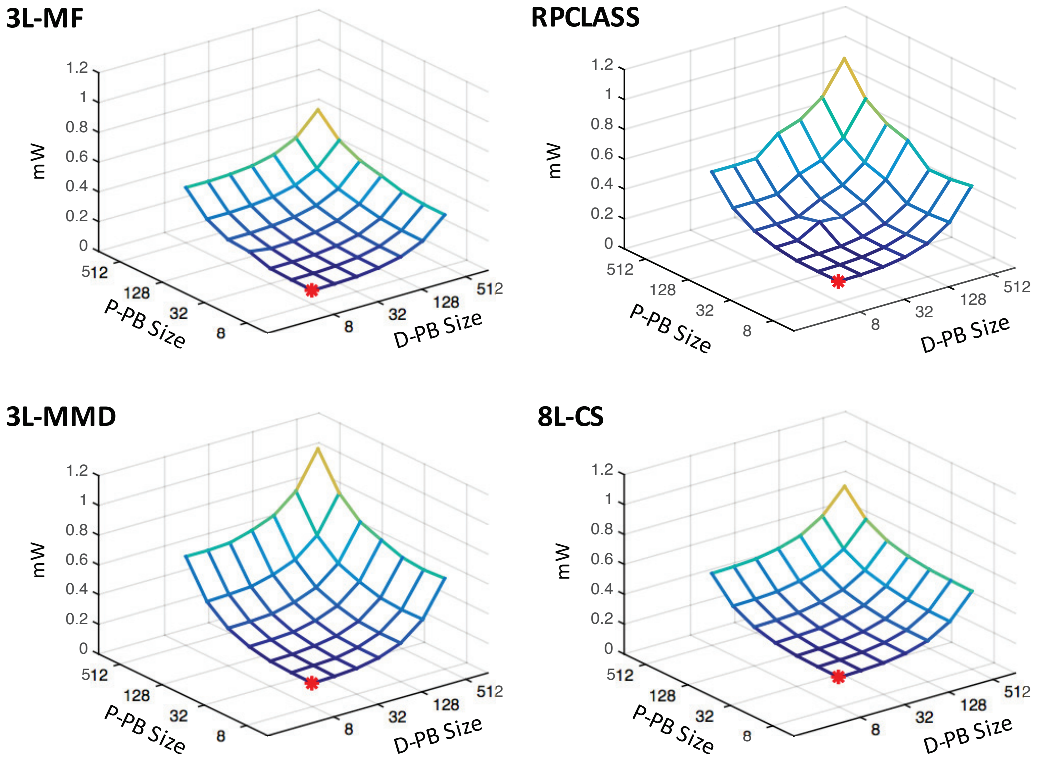
I also examine FLASH as a NVM option, for both 2D and 3D cases. The long write latency (i.e., 120μs [25] [164]) of FLASH memory renders the entire system non real-time. The time needed to write to the NVM in this case exceed the idle periods between subsequent samples. As a result, deep sensing policy could not be applied.

4.3.4 Experimental Evaluation

4.3.4.1 Page Buffer Sizing

The memory hierarchy of the 3D_TARGET platform comprises NVM blocks (in the upper tier) and smaller latch-based volatile page buffers (in the lowest one, along with processing elements). These are connected on one side to the crossbars and, ultimately, to the multiple cores, on the other side to the NVM via an ultra-dense vertical interconnect.

Program and data page buffers can be sized differently, since they are interfaced to the cores



(a) (b)

Figure 4.12 – Average power consumption of 3D_TARGET employing different instruction page buffer (I-PB) and data page buffer (D-PB) sizes (in number of words). Local minima are marked with a red dot

through independent networks. Figure 4.12 shows an exploration of the proposed architecture employing 7 different page buffer sizes ranging from 8 to 512 words and executing all the 4 studied benchmarks. 49 different configurations have been tested for each of the applications. As it can be observed, variants employing page buffers with reduced amounts of words perform better. This situation highlights, the inherent code locality present in embedded bio-signal processing algorithms, which favors more compact implementation, having to lower leakage and energy-per access, even at the cost of an increased amount of page transfers. Among the different options, the best alternative, used hereafter to perform the rest of the experiments, is the one utilizing 8-word banks for both instruction and data page buffers.

4.3.4.2 NVM Optimization

I explored various properties of the considered STTRAM and RRAM technologies and their corresponding impact on the overall nano-engineered performance. In particular, I studied the impact on: 1) energy consumption; 2) footprint; and 3) endurance.

By optimizing both NVM technologies, through increasing the write pulse width and corre-

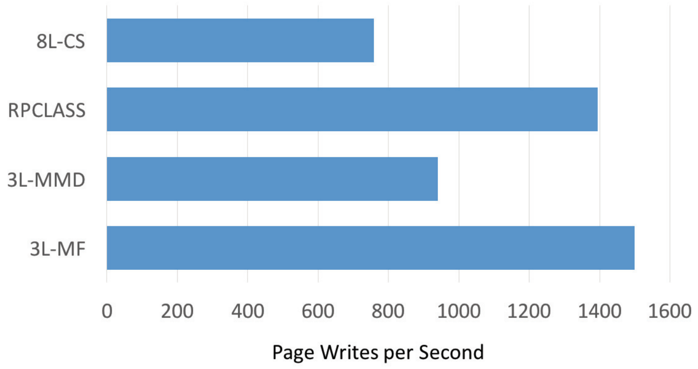


Figure 4.13 – Average page write frequency for the four studied benchmarks

spondingly reducing the write current and or voltage, STTRAM provides lower energy than RRAM by 5x, as shown in Table 4.4. At the system level, the results indicate STTRAM increases the energy-efficiency by 50% compared to RRAM-based configuration.

While STTRAM enhances the energy efficiency of the target system, RRAM is denser enabling higher memory capacity per unit footprint. As Table 4.3 shows, RRAM cell is 33% smaller than STTRAM. Thus, for a fixed-capacity memory with high area efficiency (most of the footprint is occupied by only the cells, while less area is occupied by the interface circuitry), it is expected that RRAM-based memory would have smaller footprint. In the case of 3D_TARGET however, the NVM has 30% area efficiency, which resulted in similar footprint in both memories.

I studied endurance of both NVM technologies, to deduce the lifetime of the platform until a cell cannot be used. STTRAM provides a better write endurance ($\sim 10^{15}$ writes) than RRAM (up to 10^{12}), which makes STTRAM favorable. However, by studying the specific write patterns of the targeted applications to the NVM via the studied architectures, I observed that a high endurance may not be needed in the target scenario. Figure 4.13 shows average write rates of the most used pages for the studied benchmarks. With a maximum of 1500 writes/second, a write endurance of 10^{12} is sufficient for a lifetime of >20 years. This low-endurance requirement is a significant advantage for RRAM. When this combined with the manufacturing advantages of RRAM (RRAM is favorable as it uses materials that are common in current semiconductor manufacturing [142]), RRAM may be seen as a more suitable NVM. From a research perspective, however, in this work I opt to use STTRAM in the following sections due to its energy superiority over RRAM.

4.3.4.3 Run-time Performance

The benefits of the proposed memory management enabled by the NVM integration can be observed in detail in Table 4.5. As a first observation, the amount of processing time is below 10% for all the evaluated benchmarks allowing for long periods of deep-sleep sensing in which the platform is inactive. In addition, for the chosen configuration, the amount of active cycles

	8L-CS	3L-MF	3L-MMD	RPCLASS
Processing time (%)	5.49	4.67	8.16	7.04
Data exchange cycles (%)	2.31	5.39	4.30	5.79
NVM → Buffer (Avg. MB/s	44.60	51.41	79.34	137.36
Buffer → NVM (Avg. MB/s	18.25	30.96	28.60	32.57

Table 4.5 – Runtime metrics of the analyzed benchmarks using 8-word program and data page buffers

devoted to page transfer among the NVM and the page buffers is in the worst case 5.79%, which minimally impacts the overall power budget.

Moreover, Table 4.5 shows that the utilization of the MMU does not dominate the execution time, so that even a poor choice for the replacement policy cannot determine run-time bottlenecks. The low amount of processing time will always allow to meet real-time constraints with ample margins, justifying the choice of focusing on average, instead of worst case, execution times. Finally, the ultra-wide interconnect between the processing tier and the NVM tier can support high transfer rates between the NVM and the page buffers with negligible performance degradation.

4.3.4.4 Area Footprint

Out of the studied architectures, 2D_NVM presents the largest area footprint (0.4568 mm^2). In Figure4.14, I compare the area breakdown of the other three evaluated architectures against the one of this system. 2D_Baseline is smaller than 2D_NVM, due to the inclusion of the 2-level memory structure. The ensuing overhead is chiefly determined by the bit-level interconnect between page buffers and non-volatile memories, when realized in a planar technology.

Such complex routing can be much more effectively implemented in a three-dimensional circuit. However, in the case of 3D_TSV, the space dedicated to the through-silicon-vias is considerable, accounting for more than 60% of the whole NVM tier, reaching an overall reduction with respect to 2D_NVM of 49.1%. It is worth mentioning that the TSVs area overhead directly impacts the NVM as the TSVs connect the metal layers of the processing cores to the metal layer of the NVM by passing through the NVM active layer (i.e., the layer with the transistors used to access the NVM cells). Nevertheless, the choice of the monolithic integration strategy used in the 3D_TARGET architecture provides the most compact design resulting in a 4.98X and 2.53X area saving when compared to 2D_NVM and 3D_TSV respectively.

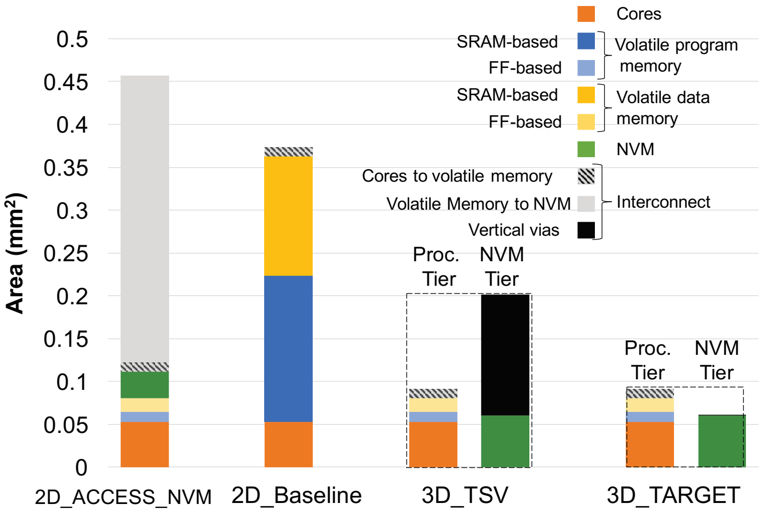


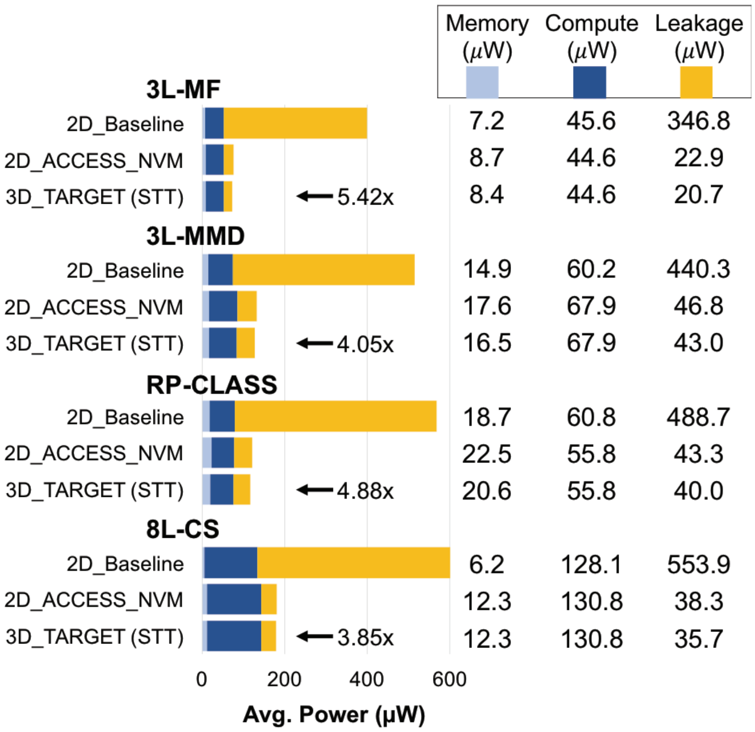
Figure 4.14 – Area breakdown for the studied nano-engineered architectures

4.3.4.5 Power Consumption

Figure 4.15 shows the power consumption comparison for each of the studied benchmarks and the corresponding breakdown of the three main contributors, namely memory, compute, and leakage (similar to the breakdown in Figure 4.15). Only results of 3D_TARGET are reported since according to the performed experiments, 3D_TSV consumes at least 1% more energy in all the cases. Figure 4.16 also shows the detailed power breakdown of all the components in 3D_TARGET. The energy consumption of the target nano-engineered bio-signal processing architectures is hugely decreased with respect to the baseline implementation, thanks to deep-sleep sensing enabled by the non-volatility. In particular, compared to 2D_Baseline, the other alternatives improve their energy efficiency up to 81% (in the case of 3D_TARGET). The efficiency of the NVM-based memory subsystem leads to 4x energy gains with respect to the volatile SRAM of 2D_Baseline, resulting in large savings in both 2D and 3D architectures. These savings are mainly derived from the adopted power gating strategy, which turns off the entirety of the platform when it is not actively processing.

4.4 Summary and Concluding Remarks

In this chapter, I have explored two different approaches to improve the energy efficiency of low-power multi-core platforms by redesigning their memory subsystem. First, I explore the utilization of hybrid memory banks provided with a small reliable memory region implemented with standard cell memories. This small memory portion can operate at ultra-low voltage levels without suffering of data degradation. Thanks to this memory arrangement, the architecture can adopt a more efficient power management strategy. During *sensing* periods, data is buffered into the reliable memory section and when enough data is available, the platform voltage is raised and the data is processed accordingly. Experimental results show



	Memory (μW)	Compute (μW)	Leakage (μW)
3L-MF			
2D_Baseline	7.2	45.6	346.8
2D_ACCESS_NVM	8.7	44.6	22.9
3D_TARGET (STT)	8.4	44.6	20.7
3L-MMD			
2D_Baseline	14.9	60.2	440.3
2D_ACCESS_NVM	17.6	67.9	46.8
3D_TARGET (STT)	16.5	67.9	43.0
RP-CLASS			
2D_Baseline	18.7	60.8	488.7
2D_ACCESS_NVM	22.5	55.8	43.3
3D_TARGET (STT)	20.6	55.8	40.0
8L-CS			
2D_Baseline	6.2	128.1	553.9
2D_ACCESS_NVM	12.3	130.8	38.3
3D_TARGET (STT)	12.3	130.8	35.7

Figure 4.15 – Power consumption of the studied architectures. The breakdown on the right half of the figure is in μW

that up to 50% power consumption can be saved adopting this strategy while only increasing the overall system area by 0.1%.

Second, I have proposed a completely re-designed two-level memory subsystem that incorporates a non-volatile memory. The new subsystem is provided with a volatile level consisting of a set of small page buffers that collectively act as a cache for the NVM. These much smaller buffers replace the typical SRAM banks and can be accessed respecting a latency of only one clock cycle. Page transfers between the NVM and the volatile buffers can be also performed in a single clock cycle due to the existing bit-level ultra-wide interconnect enabled by the chosen fabrication technology. The architecture is realized as a two-tier monolithically 3D-integrated chip, in which processing logic and volatile memories reside in the bottom tier while the NVM is hosted in the upper one. The data transfers between instruction/data page buffers and the NVM are managed by a custom-design lightweight Memory Management Unit (MMU), which collaborates with the hardware synchronization unit to orchestrate the runtime execution. An advanced power management strategy allowing for fine-grained power-gating can be adopted thanks to this memory arrangement. Experimental results show that, the obtained chip is 5x more compact than the 2D one based on SRAMs or hybrid memories. Moreover, the proposed architecture can reach up to 5.42x reduction thanks to the decreasing in leakage power.

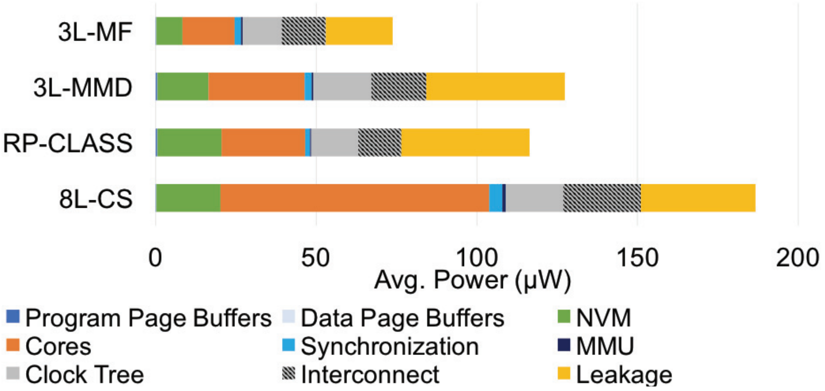


Figure 4.16 – Power consumption breakdown of all considered applications when executed on 3D_TARGET

5 Conclusions and Future Work

To conclude this thesis, hereafter I list the main contributions of my research work and highlight the impact that it has had in the international community. Then, at the end of this chapter, I provide several research directions for future work based on the results obtained in this thesis.

5.1 Summary and Contributions

In this thesis I have proposed a set of hardware/software co-design techniques to improve the energy efficiency of biomedical monitors. In order to do so, I have explored different alternatives to reduce the computational power and transmission volume required by state-of-the-art bio-signal processing applications executing on these devices. At the same time, I have proposed optimized ultra-low power multi-core architectures that exploit some of the application domain-specific characteristics to achieve the desired energy efficiency.

The following list provides a more detailed summary of the contributions introduced in the different chapters, and discusses the results of this thesis:

- **Optimized Embedded Digital Signal Processing for Health Monitoring:** In Chapter 2, I have discussed two complementary strategies to optimize state-of-the-art embedded bio-signal processing applications by reducing their computational complexity. On the one hand, at the sensor node level, I have introduced the utilization of a heartbeat classifier to perform selective advanced DSP in ECG single-node biomedical monitors. The classifier only activates the costly DSP analysis routines in case of detecting an abnormality in the heartbeat morphologies. The proposed implementation consists of a lightweight, yet accurate, heartbeat classifier based on a neuro-fuzzy structure coupled with a novel technique to perform feature extraction. This technique employs *Random Projections* to reduce the high dimensionality of the heartbeats representation down to a small set of features that are fed to the optimized classifier. Experimental results show that the accuracy of the proposed classifier when identifying abnormalities can reach up to 98.9% keeping a low rate of mis-classifications. With respect to a typical system that is continuously performing DSP analysis, my proposed approach can reduce the duty cycle and transmission volume of an ECG biomedical monitor by up to 60% and 64%, respectively.

On the other hand, at the sensor network level, I have proposed an energy- and transmission-aware WBSN devoted to the identification of physical activities. The system is composed of several nodes deployed throughout the body of the subject and interfaced with a smartphone within the same network. In particular, two classification schemes, which trade accuracy for transmission volume, are proposed. On the one side, the highly accurate smartphone-centric alternative is based on an NFC that exploits the high computational resources available on the mobile phone. On the other side, the transmission-aware scheme performs on-node classification employing a less complex decision tree and minimizing the data transmission. According to the obtained results, the high-precision classification reaches 97.2% accuracy, outperforming the state-of-the-art alternatives. Moreover, the on-node option reduces the transmission volume by up to 86% with a small classification degradation, leading to a 88% accuracy.

Publications: The work presented in this chapter has been welcomed by the embedded system community. First, the heartbeat classifier was outlined in a publication at the *Design Automation and Test in Europe (DATE) Conference* [57] and then extended and accepted for publication on *MDPI Sensors Journal* [12]. The WBSN for activity monitoring was presented at the *IEEE EUC Conference* [165].

Patent: A patent application has been granted by the U.S. Patent and Trademark Office (USPTO) containing the work about the embedded heartbeat classifier (Patent *"Method for detecting abnormalities in an electrocardiogram"*, no. US 9468386).

- **Synchronization-Based Ultra-Low Power Multi-Core Architectures:** In Chapter 3, I propose an ultra-low power multi-core architecture featuring a synchronization technique that allows to efficiently execute bio-signal processing algorithms exploiting the intrinsic parallelism of these applications. The platform is composed of 8 TamaRISC processing cores interfaced with multi-banked instruction and data memories through combinational crossbar interconnects. The architecture aims at minimizing power consumption by relaxing the system clock constraint and applying voltage scaling. To that end, the application is parallelized dividing the workload among the different cores.

  In a first version, I propose a hardware/software technique devoted to maximize lock-step execution in multi-channel parallel applications following a single-instruction multiple-data (SIMD) paradigm. It consists of a dedicated hardware synchronizer and an instruction set extension, which jointly allow to recover synchronization after the execution of data-dependent segments of code. The obtained results show that, when compared with a state-of-the-art multi-core equivalent, the proposed synchronization-based architecture provides up to 38% energy savings while only increasing the area footprint by 2%.

In a second version, I generalize the technique to any bio-signal processing application presenting an arbitrarily high degree of parallelism. I propose mechanisms to concurrently manage lock-step execution of code and producer-consumer relationships among cores. Apart from an SIMD-like strategy, the technique employs the software pipelining paradigm to split the application workload among cores. The methodology describes the necessary hardware and software support as well as the steps to adapt an existing application in order to adopt the proposed technique. According to the experimental results, the proposed architecture can obtain up to 40% energy savings while running real-world ECG processing benchmarks.

Publications: This work has been highly appreciated by the community and has lead to two consecutive publications at the *Design Automation and Test in Europe (DATE) Conference*. More precisely, the first version of the technique was published in 2013 [122] and the second one in 2014 [124].

- **Energy-Efficient Memory Subsystems for ULP Multi-Core Architectures:** In chapter 4, I explore two different approaches to improve the energy efficiency of low-power multi-core platforms by redesigning their memory subsystem. First, I explore the utilization of hybrid memory banks provided with a small reliable memory region implemented with standard cell memories. This small memory portion can operate at ultra-low voltage levels without suffering of data degradation. Thanks to this memory arrangement, the architecture can adopt a more efficient power management strategy. When no data is available to be processed, the full processing architecture is stalled (clock-gated) and the system supply voltage is aggressively reduced. During these *sensing* periods data is buffered into the reliable memory section. When enough data is available, the platform voltage is raised and the data is processed accordingly. Experimental results show that up to 50% power consumption can be saved adopting this strategy while only increasing the overall system area by 0.1%.

  Second, I propose a completely re-designed two-level memory subsystem that incorporates a non-volatile memory as the new main storage feature. The new subsystem is provided with a volatile level consisting of a set of small page buffers that collectively act as a cache for the NVM. These much smaller buffers replace the typical SRAM ones and can be accessed respecting a latency of only one clock cycle. Page transfers between the NVM and the volatile buffers can be also performed in a single clock cycle due to the existing bit-level ultra-wide interconnect among them. Such a high bandwidth is possible thanks to the chosen fabrication technology. The architecture is realized as a two-tier monolithically 3D-integrated chip, in which processing logic and volatile memories reside in the bottom tier while the NVM is hosted in the upper one. The data transfers between instruction/data page buffers and the NVM are managed by a custom-design lightweight Memory Management Unit (MMU), which collaborates with the hardware synchronization unit to orchestrate the runtime execution. Thanks to this memory organization, a more advanced power management strategy allowing for fine-grained power-gating is adopted. Experimental results show that, the obtained chip is 5x more compact than the typical 2D one based on SRAMs or hybrid memories. Energy-wise, the proposed architecture can reach up to 5.42x reduction thanks to decreasing leakage power consumption by more 80%.

Publications: These works have recently been submitted and very well received by the embedded system and computer architecture community. The work of the platform featuring the hybrid memory subsystem has been accepted for publication in the journal *IEEE Transactions on Computers* [166]. In addition, the 3D-integrated NVM-based architecture was presented at the *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)* [167].

5.2 Future Work

Based on my research findings exposed in this thesis, in this section I provide some research directions that can be taken in the field. In particular, I highlight some of the short- and long-term lines that can derive from my work.

A set of short-term developments that continues the proposed approach to achieve higher energy-efficiency in biomedical monitors can be summarized as follows:

- **Development of Energy-Efficient Domain-Specific Processing Architectures:** While in this thesis I have not covered this aspect, there is an important opportunity to obtain large energy gains by offloading the execution of complex algorithms to more efficient dedicated hardware units. This option, which in principle may sound counterintuitive due to its limited flexibility, does not have to be studied in a per-application basis but at the target domain level (i.e. bio-signal processing applications) where some basic computing intensive routines may be shared by the different applications. In addition, and as an extension to the work presented in Chapter 4, the energy efficiency of this subset of architectures can be greatly improved by carefully tailoring the memory subsystems, exploiting some domain-specific application characteristics [168], or employing standard memories in combination with error correction techniques to work at near-threshold regimes [109].
- **Exploration of Heterogeneous Architectures:** In this thesis, I have focused in the utilization of general purpose low-power homogeneous multi-core architectures. However, an interesting area of research is the integration of processing units of different nature (e.g. general purpose cores, parallel digital signal processors, hardware accelerators, etc.). In this context, some preliminary works have already shown to provide promising results. At the Embedded Systems Laboratory of EPFL (Switzerland), it has been proved that, by incorporating a Coarse-Grained Reconfigurable Array to a low-power multi-core system, considerable energy savings can be attained when executing computing-intensive *kernels* [169].
- **Advanced Low-overhead Memory Protection and Correction:** In this thesis I have pro-

posed two optimized memory subsystems assuming that no error can be tolerated at the application level. However, bio-signal processing algorithms present some resilience to errors that can be exploited in order to reduce the power consumption of the system. For instance, big data buffers stored in data memory may tolerate different kind of static or dynamic bit-flips with no major impact in the quality of results. This aspect can be leveraged in order to apply a more aggressive voltage scaling. At the same time, some already existing error detection and correction techniques could be applied in order to improve the robustness of the application, leading to a tradeoff between area and energy consumption.

- **Approximate Computing for Bio-Signal Processing:** Following the same rationale, approximate computing relies on the ability of applications and systems to tolerate some inaccuracies in the computed results. By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow considerable improvements in energy efficiency. Given the algorithmic resilience of bio-signal processing applications, this technique can be applied to further reduce the power consumption of the architectures proposed in this thesis.
- **Advanced Hybrid Classification Framework:** Given the classification framework presented in the Chapter 2 of this thesis, several techniques can be employed to further improve its accuracy and energy efficiency. On the one hand, the classification framework based on an optimized embedded neuro-fuzzy classifier can be improved by employing advanced arithmetic manipulations in the logarithmic domain while computing the fuzzification values, which would reduce the complexity of the calculations while keeping the precision obtained by the proposed piecewise approximation. In addition, the compression of the RP matrix and the projection itself can be done by means of incorporating a hardware accelerator coupled to the data memory unit of the target platform. On the other hand, the proposed WBSN for activity monitoring can be updated to efficiently combine the two presented configurations by employing a hybrid classification scheme. This new system could perform a computationally intensive classification on the smartphone periodically or when a sudden change is detected on the node-based classifier and use the outcome to correct and re-configure the implemented on-node decision tree.

In addition and stemming from the proposed work, some long-term research lines, which would reduce the power consumption and increase the reliability and performance of next generation biomedical monitors, are the following:

- **Logic Error Detection and Mitigation:** In the nano-scale era many reliability issues will affect the performance and correct functioning of logic circuits, as long as we keep on advancing in the trend of reducing transistor dimensions. Ultra-low power architectures, such as the ones presented in this thesis, will also need to cope with this problems, specially when voltage scaling is applied. Transistors performance degradation can lead to combinational logic failures in the form of timing violations that result in circuit functional errors. In this context, Negative Bias Temperature Instability (NBTI) is one of the major threats, which affects the PMOS transistors under a negative gate stress temporary degrading their performance. The online preliminary detection of this spurious errors and its mitigation represents a challenging area of research that not only applies to the specific field of low-power architecture design but in general to the digital circuit design domain.

- **3D-Integration Exploiting Emerging Nano-Technologies:** Building upon the 3D-integrated architecture proposed in the Chapter 4 of this thesis, new emerging nano-technologies such as carbon nanotubes field-effect transistors (CNFETs) could be employed to further improve the energy efficiency and performance of ultra-low-power architectures devoted to bio-signal processing. In addition, even though the presented platform has been characterized using experimental evidence obtained in academic facilities [144], the employed fabrication processes have not been validated yet in an industrial scenario, which may impose further design constraints not considered in this work.

Bibliography

[1] "Non-communicable diseases," tech. rep., World Health Organization. http://www.who.int/mediacentre/factsheets/fs355/en/; Last accessed on Sep. 10th, 2016.

[2] H. B. Hubert, M. Feinleib, P. M. McNamara, and W. P. Castelli, "Obesity as an independent risk factor for cardiovascular disease: a 26-year follow-up of participants in the framingham heart study.," *Circulation*, vol. 67, no. 5, pp. 968–977, 1983.

[3] U. Nations, "World population ageing." http://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2015_Report.pdf; Last accessed on Sept. 10th, 2016.

[4] "Cardiovascular diseases," tech. rep., World Health Organization. http://www.who.int/cardiovascular_diseases/en/; Last accessed on Jun. 17th, 2016.

[5] "Global status report on non-communicable diseases, 2014," tech. rep., World Health Organization. http://apps.who.int/iris/bitstream/10665/148114/1/9789241564854_eng.pdf?ua=1; Last accessed on Sept. 10th, 2016.

[6] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," *IEEE Wireless Communications Magazine*, vol. 17, no. 1, pp. 80–88, 2010.

[7] E. Jovanov, A. Milenkovic, C. Otto, and P. C. De Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and rehabilitation*, vol. 2, no. 1, p. 1, 2005.

[8] I. Beretta, F. Rincon, N. Khaled, P. R. Grassi, V. Rana, and D. Atienza, "Design exploration of energy-performance trade-offs for wireless sensor networks," in *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, (New York, NY, USA), pp. 1043–1048, ACM, 2012.

[9] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *Architecture of Computing Systems (ARCS), 2010 23rd International Conference on*, pp. 1–10, Feb 2010.

[10] F. Rincon, J. Recas, N. Khaled, and D. Atienza, "Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, pp. 854–863, Nov 2011.

[11] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, pp. 2456–2466, Sept 2011.

[12] R. Braojos, I. Beretta, G. Ansaloni, and D. Atienza, "Early classification of pathological heartbeats on wireless body sensor nodes," *Sensors*, vol. 14, no. 12, p. 22532, 2014.

[13] A. Y. Dogan, J. Constantin, D. Atienza, A. Burg, and L. Benini, "Low-power processor architecture exploration for online biomedical signal analysis," *IET Circuits, Devices Systems*, vol. 6, pp. 279–286, Sept 2012.

[14] 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*, DAC '10, (New York, NY, USA), pp. 543–548, ACM, 2010.

[15] Y. Hao and R. Foster, "Wireless body sensor networks for health-monitoring applications," *Physiological Measurement*, vol. 29, no. 11, p. R27, 2008.

[16] T. Berset, I. Romero, A. Young, and J. Penders, "Robust heart rhythm calculation and respiration rate estimation in ambulatory ECG monitoring," in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pp. 400–403, Jan 2012.

[17] F. Massé, M. V. Bussel, A. Serteyn, J. Arends, and J. Penders, "Miniaturized wireless ECG monitor for real-time detection of epileptic seizures," *ACM Trans. Embed. Comput. Syst.*, vol. 12, pp. 102:1–102:21, July 2013.

[18] J. Kwong and A. P. Chandrakasan, "An energy-efficient biomedical signal processing platform," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 1742–1753, July 2011.

[19] S. R. Sridhara, M. DiRenzo, S. Lingam, S. J. Lee, R. Blazquez, J. Maxey, S. Ghanem, Y. H. Lee, R. Abdallah, P. Singh, and M. Goel, "Microwatt embedded processor platform for medical system-on-chip applications," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 721–730, April 2011.

[20] 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 *2008 IEEE Symposium on VLSI Circuits*, pp. 188–189, June 2008.

[21] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw, "Exploring variability and performance in a sub-200-mv processor," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 881–891, April 2008.

[22] R. G. Dreslinski, M. Wieckowski, D. Blaauw, and D. Sylvester, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, 2010.

[23] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin, "A 2.60pj/inst subthreshold sensor processor for optimal energy efficiency," in *2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.*, pp. 154–155, 2006.

[24] D. Bortolotti, H. Mamaghanian, A. Bartolini, M. Ashouei, J. Stuijt, D. Atienza, P. Vandergheynst, 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*, ISLPED '14, (New York, NY, USA), pp. 45–50, ACM, 2014.

[25] H. Mitani, K. Matsubara, H. Yoshida, T. Hashimoto, H. Yamakoshi, S. Abe, T. Kono, Y. Taito, T. Ito, T. Krafuji, K. Noguchi, H. Hidaka, and T. Yamauchi, "7.6 a 90nm embedded 1t-monos flash macro for automotive applications with 0.07mj/8kb rewrite energy and endurance over 100m cycles under tj of 175??c," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 140–141, Jan 2016.

[26] Y. Taito, M. Nakano, H. Okimoto, D. Okada, T. Ito, T. Kono, K. Noguchi, H. Hidaka, and T. Yamauchi, "7.3 A 28nm embedded SG-MONOS flash macro for automotive achieving 200MHz read operation and 2.0 MB/S write throughput at T i, of 170°C," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pp. 1–3, IEEE, 2015.

[27] J. Lewandowski, H. E. Arochena, R. N. G. Naguib, K.-M. Chao, and A. Garcia-Perez, "Logic-centered architecture for ubiquitous health monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1525–1532, Sept 2014.

[28] C. Park, P. H. Chou, Y. Bai, R. Matthews, and A. Hibbs, "An ultra-wearable, wireless, low power ECG monitoring system," in *Biomedical Circuits and Systems Conference, 2006. BioCAS 2006. IEEE*, pp. 241–244, Nov 2006.

[29] J. Proulx, R. Clifford, S. Sorensen, D.-J. Lee, and J. Archibald, "Development and evaluation of a Bluetooth© EKG monitoring sensor," in *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pp. 507–511, 2006.

[30] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer a wireless sensor platform for noninvasive biomedical research," *IEEE Sensors Journal*, vol. 10, pp. 1527–1534, Sept 2010.

[31] D. Malan, F. J. Thaddeus, M. Welsh, and S. Moulton, "CodeBlue : An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care," in *Proceeding on the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, pp. 12–14, ACM, 2004.

[32] J. Lester, T. Choudhury, and G. Borriello, *Pervasive Computing: 4th International Conference, PERVASIVE 2006, Dublin, Ireland, May 7-10, 2006. Proceedings*, ch. A Practical Approach to Recognizing Physical Activities, pp. 1–16. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[33] F. Casamassima, E. Farella, and L. Benini, "Context aware power management for motion-sensing body area network nodes," in *Proceedings of the Conference on Design, Automation & Test in Europe*, DATE '14, (3001 Leuven, Belgium, Belgium), pp. 170:1–170:6, European Design and Automation Association, 2014.

[34] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 156–167, Jan 2006.

[35] P. Laguna, R. Jané, and P. Caminal, "Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database," *Computers and Biomedical Research*, vol. 27, no. 1, pp. 45 – 60, 1994.

[36] Y. Sun, K. L. Chan, and S. M. Krishnan, "Characteristic wave detection in ECG signal using morphological transform," *BMC Cardiovascular Disorders*, vol. 5, no. 1, pp. 1–7, 2005.

[37] Y. Sun, K. L. Chan, and S. M. Krishnan, "ECG signal conditioning by morphological filtering," *Computers in Biology and Medicine*, vol. 32, no. 6, pp. 465 – 479, 2002.

[38] J. Constantin, A. Dogan, O. Andersson, P. Meinerzhagen, J. N. Rodrigues, D. Atienza, and A. Burg, "TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing," in *VLSI and System-on-Chip, 2012 (VLSI-SoC), IEEE/IFIP 20th International Conference on*, pp. 159–164, Oct 2012.

[39] C. Meyer and H. Keiser, "Electrocardiogram baseline noise estimation and removal using cubic splines and state-space computation techniques," *Computers and Biomedical Research*, vol. 10, no. 5, pp. 459 – 470, 1977.

[40] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications.* Academic Press, 2005.

[41] Z. Dokur, T. Ölmez, E. Yazgan, and O. K. Ersoy, "Detection of ECG waveforms by neural networks," *Medical engineering & physics*, vol. 19, no. 8, pp. 738–741, 1997.

[42] S. Graja and J. M. Boucher, "Hidden markov tree model applied to ECG delineation," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, pp. 2163–2168, Dec 2005.

[43] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on Biomedical Engineering*, vol. 42, pp. 21–28, Jan 1995.

[44] J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, and P. Laguna, “A wavelet-based ECG delineator: evaluation on standard databases,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 570–581, April 2004.

[45] N. Boichat, N. Khaled, F. Rincon, and D. Atienza, “Wavelet-based ECG delineation on a wearable embedded sensor platform,” in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pp. 256–261, June 2009.

[46] A. Cohen and J. Kovacevic, “Wavelets: The mathematical background,” in *Proc. IEEE*, pp. 514–522, 1996.

[47] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[48] X. Jiang, L. Zhang, Q. Zhao, and S. Albayrak, “ECG arrhythmias recognition system based on independent component analysis feature extraction,” in *TENCON 2006. 2006 IEEE Region 10 Conference*, pp. 1–4, Nov 2006.

[49] R. Ceylan and Y. Özbay, “Comparison of FCM, PCA and WT techniques for classification ECG arrhythmias using artificial neural network,” *Expert Systems with Applications*, vol. 33, no. 2, pp. 286 – 295, 2007.

[50] M. Paoletti and C. Marchesi, “Discovering dangerous patterns in long-term ambulatory ECG recordings using a fast QRS detection algorithm and explorative data analysis,” *Computer Methods and Programs in Biomedicine*, vol. 82, no. 1, pp. 20 – 30, 2006.

[51] S.-N. Yu and K.-T. Chou, “Integration of independent component analysis and neural networks for ECG beat classification,” *Expert Systems with Applications*, vol. 34, no. 4, pp. 2841 – 2846, 2008.

[52] C. Ye, M. T. Coimbra, and B. V. K. V. Kumar, “Arrhythmia detection and classification using morphological and dynamic features of ECG signals,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 1918–1921, Aug 2010.

[53] I. Iliev, V. Krasteva, and S. Tabakov, “Real-time detection of pathological cardiac events in the electrocardiogram,” *Physiological Measurement*, vol. 28, no. 3, p. 259, 2007.

[54] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE Transactions on Information Theory*, vol. 52, pp. 5406–5425, Dec 2006.

[55] I. Bogdanova, F. Rincón, and D. Atienza, “A multi-lead ECG classification based on random projection features,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 625–628, March 2012.

[56] D. Achlioptas, “Database-friendly random projections,” in *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS ’01, (New York, NY, USA), pp. 274–281, ACM, 2001.

[57] 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, March 2013.

[58] V.-E. Neagoe, I.-F. Iatan, and S. Grunwald, “A neuro-fuzzy approach to classification of ECG signals for ischemic heart disease diagnosis,” in *AMIA Annual Symposium*, pp. 494–498, 2003.

[59] I. Jolliffe, “Principal component analysis,” *Springer Series in Statistics (*, 2002.

[60] R. Ceylan, Yüksel Özbay, and B. Karlik, “A novel approach for classification of ECG arrhythmias: Type-2 fuzzy clustering neural network,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6721 – 6726, 2009.

[61] C. T. Sun and J. S. Jang, “A neuro-fuzzy classifier and its applications,” in *Fuzzy Systems, 1993., Second IEEE International Conference on*, pp. 94–98 vol.1, 1993.

[62] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, vol. 6, no. 4, pp. 525 – 533, 1993.

[63] B. Cetişli and A. Barkana, “Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training,” *Soft Computing*, vol. 14, no. 4, pp. 365–378, 2009.

[64] P. de Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ECG morphology and heartbeat interval features,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1196–1206, July 2004.

[65] R. Braojos, G. Ansaloni, D. Atienza, and F. J. Rincon, “Embedded real-time ECG delineation methods: A comparative evaluation,” in *Bioinformatics Bioengineering (BIBE), 2012 IEEE 12th International Conference on*, pp. 99–104, Nov 2012.

[66] V. Rokhlin, A. Szlam, and M. Tygert, “A randomized algorithm for principal component analysis,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1100–1124, 2010.

[67] J. Yang and V. Honavar, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, ch. Feature Subset Selection Using a Genetic Algorithm, pp. 117–136. Boston, MA: Springer US, 1998.

[68] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13).

[69] M. Milis, K. Michaelides, A. Kounoudes, G. Ansaloni, D. Atienza, F. Giroud, P. F. Ruedi, and F. Masson, "Icyheart: Highly integrated ultra-low-power soc solution for unobtrusive and energy efficient wireless cardiac monitoring: Research project for the benefit of specific groups (fp7, capacities)," in *Bioinformatics Bioengineering (BIBE), 2012 IEEE 12th International Conference on*, pp. 105–109, Nov 2012.

[70] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: A wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, (New York, NY, USA), pp. 183–196, ACM, 2009.

[71] S. J. Preece*, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 871–879, March 2009.

[72] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213 – 2220, 2008.

[73] C. Lombriser, N. B. Bharatula, D. Roggen, and G. Tröster, "On-body activity recognition in a dynamic sensor network," in *Proceedings of the ICST 2Nd International Conference on Body Area Networks*, BodyNets '07, (ICST, Brussels, Belgium, Belgium), pp. 17:1–17:6, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[74] F. Rincón, N. Boichat, V. Barbero, N. Khaled, and D. Atienza, "Multi-lead wavelet-based ECG delineation on a wearable embedded sensor platform," in *Computers in Cardiology, 2009*, pp. 289–292, Sept 2009.

[75] L. Bao and S. S. Intille, *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings*, ch. Activity Recognition from User-Annotated Acceleration Data, pp. 1–17. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[76] G. Bieber and C. Peter, "Using physical activity for user behavior analysis," in *Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '08, (New York, NY, USA), pp. 94:1–94:6, ACM, 2008.

[77] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 289–296, 2001.

[78] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, pp. 1541–1546, AAAI Press, 2005.

[79] NXP, *JN5148-001 IEEE802.15.4 Wireless Microcontroller (JN-DS-JN5148-001 1v9) [Online]*, 2013.

[80] IEEE, "IEEE Std 802.15.4-2006 (revision of IEEE Std 802.15.4-2003)," 2006.

[81] STMicroelectronics, *LSM330DLC iNEMO inertial module: 3D accelerometer and 3D gyroscope [Online]*, 2012.

[82] STMicroelectronics, *LIS3LV02DQ MEMS Inertial Sensor [Online]*, 2005.

[83] T. R. Burchfield, S. Venkatesan, and D. Weiner, "Maximizing throughput in ZigBee wireless networks through analysis, simulations and implementations," in *Proceedings of the 1st International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks*, 2007.

[84] 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.

[85] A. Y. Dogan, J. Constantin, D. Atienza, A. Burg, and L. Benini, "Low-power processor architecture exploration for online biomedical signal analysis," *IET Circuits, Devices Systems*, vol. 6, pp. 279–286, Sept 2012.

[86] M. Ashouei, J. Hulzink, M. Konijnenburg, J. Zhou, F. Duarte, A. Breeschoten, J. Huisken, J. Stuyt, H. de Groot, F. Barat, J. David, and J. V. Ginderdeuren, "A voltage-scalable biomedical signal processor running ECG using 13pJ/cycle at 1MHz and 0.4V," in *2011 IEEE International Solid-State Circuits Conference*, pp. 332–334, Feb 2011.

[87] "Cortex-m0 processor." http://www.arm.com/products/processors/cortex-m/cortex-m0.php. Last accessed on Jun. 17th, 2016.

[88] W. Massagram, N. Hafner, M. Chen, L. Macchiarulo, V. M. Lubecke, and O. Boric-Lubecke, "Digital heart-rate variability parameter monitoring and assessment ASIC," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, pp. 19–26, Feb 2010.

[89] X. Liu, Y. Zheng, M. W. Phyu, F. N. Endru, V. Navaneethan, and B. Zhao, "An ultra-low power ECG acquisition and monitoring ASIC system for WBAN applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, pp. 60–70, March 2012.

[90] S.-C. Huang, H.-M. Wang, and W.-Y. Chen, "A ±6ms-accuracy, 0.68mm2, and 2.21 μw qrs detection ASIC," *VLSI Des.*, vol. 2012, pp. 20:20–20:20, Jan. 2012.

[91] M. W. Phyu, Y. Zheng, B. Zhao, L. Xin, and Y. S. Wang, "A real-time ECG QRS detection ASIC based on wavelet multiscale analysis," in *Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian*, pp. 293–296, Nov 2009.

[92] S. Y. Hsu, Y. L. Chen, P. Y. Chang, J. Y. Yu, T. F. Yang, R. J. Chen, and C. Y. Lee, "A micropower biomedical signal processor for mobile healthcare applications," in *Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian*, pp. 301–304, Nov 2011.

[93] C. Y. Chiang, H. H. Chen, T. C. Chen, C. S. Liu, Y. J. Huang, S. S. Lu, C. W. Lin, and L. G. Chen, "Analysis and design of on-sensor ECG processors for realtime detection of vf, vt, and pvc," in *2010 IEEE Workshop On Signal Processing Systems*, pp. 42–45, Oct 2010.

[94] J. D. Boeck, "Game-changing opportunities for wireless personal healthcare and lifestyle," in *2011 IEEE International Solid-State Circuits Conference*, pp. 15–21, Feb 2011.

[95] F. Aihua, B. Chunhua, N. Xinbao, H. Aijun, and Z. Jianjun, "Portable electrocardiogram monitor based on ARM," in *2008 International Conference on Information Technology and Applications in Biomedicine*, 2008.

[96] J. Justesen and S. C. Madsen, "Wearable wireless ECG monitoring hardware prototype for use in patients own home," in *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–3, IEEE, 2009.

[97] C. Ghule, D. Wakde, G. Virdi, and N. R. Khodke, "Design of portable ARM processor based ECG module for 12 lead ECG data acquisition and analysis," in *2009 International Conference on Biomedical and Pharmaceutical Engineering*, pp. 1–8, IEEE, 2009.

[98] "Texas instruments MSP430 family [online]." http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/overview.page?DCMP=MCU_%2520other&HQS=msp430. Last accessed on Sep. 4th, 2016.

[99] M. Johnson, M. Healy, P. van de Ven, M. J. Hayes, J. Nelson, T. Newe, and E. Lewis, "A comparative review of wireless sensor network mote technologies," in *Sensors, 2009 IEEE*, pp. 1439–1442, Oct 2009.

[100] "Shimmer [online]." http://www.shimmersensing.com/. Last accessed on Sep. 4th, 2016.

[101] S. C. Jocke, J. F. Bolus, S. N. Wooters, A. Jurik, A. Weaver, T. Blalock, and B. Calhoun, "A 2.6-μw sub-threshold mixed-signal ECG SoC," in *2009 Symposium on VLSI Circuits*, pp. 60–61, IEEE, 2009.

[102] Y. Zhang, Y. Shakhsheer, A. T. Barth, H. C. Powell Jr, S. A. Ridenour, M. A. Hanson, J. Lach, and B. H. Calhoun, "Energy efficient design for body sensor nodes," *Journal of Low Power Electronics and Applications*, vol. 1, no. 1, pp. 109–130, 2011.

[103] R. Fasthuber, F. Catthoor, P. Raghavan, and F. Naessens, *Energy-efficient communication processors*. Springer, 2013.

[104] J. H.-F. Constantin, "Processor development in LISA for biomedical applications," Master's thesis, Eidgenössische Technische Hochschule Zürich (EPFL), 2010.

[105] “Microchip 16-bit PIC24 MCUs family [online].” http://www.microchip.com/design-centers/16-bit. Last accessed on Sep. 4th, 2016.

[106] C. Piguet, J.-M. Masgonty, C. Arm, S. Durand, T. Schneider, F. Rampogna, C. Scarnera, C. Iseli, J.-P. Bardyn, R. Pache, *et al.*, “Low-power design of 8-b embedded coolrisc microcontroller cores,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, pp. 1067–1078, 1997.

[107] A. Dogan, *Energy-Aware Processing Platform Exploration for Embedded Biosignal Analysis.* PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2013.

[108] T. Gemmeke, M. M. Sabry, J. Stuijt, P. Raghavan, F. Catthoor, and D. Atienza, “Resolving the memory bottleneck for single supply near-threshold computing,” in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, March 2014.

[109] C. Silvano *et al.*, *Near Threshold Computing.* Springer International Publishing, 2016.

[110] R. G. Dreslinkski, B. Zhai, T. Mudge, D. Blaauw, and D. Sylvester, “An energy efficient parallel architecture using near threshold operation,” in *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*, PACT ’07, (Washington, DC, USA), pp. 175–188, IEEE Computer Society, 2007.

[111] Y. Pu, P. de Gyvez, H. Corporaal, Y. Ha, *et al.*, “An ultra-low-energy multi-standard jpeg co-processor in 65 nm CMOS with sub/near threshold supply voltage,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 3, pp. 668–680, 2010.

[112] 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,” *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 2, pp. 257–267, 2014.

[113] B. Biisze, F. Bouwens, M. Konijnenburg, M. De Nil, M. Ashouei, J. Hulzink, J. Zhou, J. Stuyt, J. Huisken, H. De Groot, *et al.*, “Ultra low power programmable biomedical soc for on-body ECG and EEG processing,” in *Solid State Circuits Conference (A-SSCC), 2010 IEEE Asian*, pp. 1–4, IEEE, 2010.

[114] A. Rahimi, I. Loi, M. R. Kakoee, and L. Benini, “A fully-synthesizable single-cycle interconnection network for shared-l1 processor clusters,” in *2011 Design, Automation Test in Europe*, pp. 1–6, March 2011.

[115] D. E. Culler, J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach.* Gulf Professional Publishing, 1999.

[116] 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*, GLSVLSI ’09, (New York, NY, USA), pp. 141–144, ACM, 2009.

[117] C. Stoif, M. Schoeberl, B. Liccardi, and J. Haase, "Hardware synchronization for embedded multi-core processors," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2557–2560, May 2011.

[118] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph, and M. Snir, "The nyu ultracomputer - designing an mimd shared memory parallel computer," *IEEE Transactions on Computers*, vol. C-32, pp. 175–189, Feb 1983.

[119] T. J. Rolfe, "On a fast integer square root algorithm," *SIGNUM Newsl.*, vol. 22, pp. 6–11, Oct. 1987.

[120] J. Yoo, L. Yan, D. El-Damak, M. A. B. Altaf, A. H. Shoeb, and A. P. Chandrakasan, "An 8-channel scalable eeg acquisition soc with patient-specific seizure classification and recording processor," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 214–228, Jan 2013.

[121] J. Willems, P. Arnaud, J. van Bemmel, R. Degani, P. Macfarlane, and C. Zywietz, "Common standards for quantitative electrocardiography: goals and main results. CSE working party," *Methods of information in medicine*, vol. 29, pp. 263–271, September 1990.

[122] A. Y. Dogan, R. Braojos, J. Constantin, G. Ansaloni, A. Burg, and D. Atienza, "Synchronizing code execution on ultra-low-power embedded multi-channel signal analysis platforms," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pp. 396–399, March 2013.

[123] 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 *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, pp. 123–134, Feb 2008.

[124] R. Braojos, A. Y. 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, March 2014.

[125] 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*, ISLPED '98, (New York, NY, USA), pp. 76–81, ACM, 1998.

[126] B. H. Calhoun and A. Chandrakasan, "Analyzing static noise margin for sub-threshold SRAM in 65nm CMOS," in *Proceedings of the 31st European Solid-State Circuits Conference, 2005. ESSCIRC 2005.*, pp. 363–366, Sept 2005.

[127] 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*, ASP-DAC '07, (Washington, DC, USA), pp. 798–803, IEEE Computer Society, 2007.

[128] W. Nebel and J. Mermet, *Low power design in deep submicron electronics*, vol. 337. Springer Science & Business Media, 2013.

[129] 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*, MobiCom ’01, (New York, NY, USA), pp. 251–259, ACM, 2001.

[130] 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*, FPGA ’04, (New York, NY, USA), pp. 183–189, ACM, 2004.

[131] 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,” *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 1625–1637, July 2013.

[132] Y. Pu, J. P. de Gyvez, 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 *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, pp. 146–147,147a, Feb 2009.

[133] J. Hulzink, M. Konijnenburg, M. Ashouei, A. Breeschoten, T. Berset, J. Huisken, J. Stuyt, H. de Groot, F. Barat, J. David, and J. V. Ginderdeuren, “An ultra low energy biomedical signal processing system operating at near-threshold,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, pp. 546–554, Dec 2011.

[134] N. Verma and A. P. Chandrakasan, “A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 141–149, Jan 2008.

[135] R. G. Dreslinski, G. K. Chen, T. Mudge, D. Blaauw, D. Sylvester, and K. Flautner, “Reconfigurable energy efficient near threshold cache architectures,” in *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pp. 459–470, Nov 2008.

[136] I. J. Chang, D. Mohapatra, and K. Roy, “A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 101–112, Feb 2011.

[137] O. Andersson, B. Mohammadi, P. Meinerzhagen, A. Burg, and J. N. Rodrigues, “Dual-VT 4kb sub-VT memories with <1 pW/bit leakage in 65 nm CMOS,” in *ESSCIRC (ESSCIRC), 2013 Proceedings of the*, pp. 197–200, Sept 2013.

[138] W. Kim, D. Brooks, and G. Y. Wei, “A fully-integrated 3-level dc-dc converter for nanosecond-scale dvfs,” *IEEE Journal of Solid-State Circuits*, vol. 47, pp. 206–219, Jan 2012.

[139] 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.

[140] “2.4-ghz Bluetooth© low energy system-on-chip.” http://www.ti.com/lit/ds/symlink/cc2540.pdf. Last accessed on Jun. 17th, 2016.

[141] A. D. Kent and D. C. Worledge, “A new spin on magnetic memories,” *Nature nanotechnology*, vol. 10, no. 3, pp. 187–191, 2015.

[142] H.-S. P. Wong, C. Ahn, J. Cao, H.-Y. Chen, S. W. Fong, Z. Jiang, C. Neumann, S. Qin, J. Sohn, Y. Wu, S. Yu, and X. Zheng, “Stanford memory trends,” tech. rep. Last accessed on June 17th, 2016.

[143] H. Wei, M. Shulaker, H. S. P. Wong, and S. Mitra, “Monolithic three-dimensional integration of carbon nanotube fet complementary logic circuits,” in *2013 IEEE International Electron Devices Meeting*, pp. 19.7.1–19.7.4, Dec 2013.

[144] M. M. Shulaker, T. F. Wu, A. Pal, L. Zhao, Y. Nishi, K. Saraswat, H. S. P. Wong, and S. Mitra, “Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs,” in *2014 IEEE International Electron Devices Meeting*, pp. 27.4.1–27.4.4, Dec 2014.

[145] M. M. Shulaker, T. F. Wu, M. M. Sabry, H. Wei, H. S. P. Wong, and S. Mitra, “Monolithic 3D integration: A path from concept to reality,” in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1197–1202, March 2015.

[146] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, “Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement,” in *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, pp. 554–559, June 2008.

[147] A. Nigam, C. W. Smullen, IV, V. Mohan, E. Chen, S. Gurumurthi, and M. R. Stan, “Delivering on the promise of universal memory for spin-transfer torque RAM (STT-RAM),” in *Proceedings of the 17th IEEE/ACM International Symposium on Low-power Electronics and Design*, ISLPED ’11, (Piscataway, NJ, USA), pp. 121–126, IEEE Press, 2011.

[148] F. Sampaio, M. Shafique, B. Zatt, S. Bampi, and J. Henkel, “Energy-efficient architecture for advanced video memory,” in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD ’14, (Piscataway, NJ, USA), pp. 132–139, IEEE Press, 2014.

[149] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, “A novel architecture of the 3D stacked MRAM L2 cache for CMPs,” in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, pp. 239–249, Feb 2009.

[150] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, “Relaxing non-volatility for fast and energy-efficient STT-RAM caches,” in *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 50–61, Feb 2011.

[151] Z. Sun, X. Bi, H. H. Li, W.-F. Wong, Z.-L. Ong, X. Zhu, and W. Wu, “Multi retention level STT-RAM cache designs with a dynamic refresh scheme,” in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-44, (New York, NY, USA), pp. 329–338, ACM, 2011.

[152] W. Xu, H. Sun, X. Wang, Y. Chen, and T. Zhang, “Design of last-level on-chip cache using spin-torque transfer RAM (STT RAM),” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, pp. 483–493, March 2011.

[153] B. Ransford, J. Sorber, and K. Fu, “Mementos: System support for long-running computation on rfid-scale devices,” in *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XVI, (New York, NY, USA), pp. 159–170, ACM, 2011.

[154] H. Jayakumar, A. Raha, and V. Raghunathan, “QUICKRECALL: A low overhead hw/sw approach for enabling computations across power cycles in transiently powered computers,” in *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, pp. 330–335, Jan 2014.

[155] K. C. Chun, H. Zhao, J. D. Harms, T. H. Kim, J. P. Wang, and C. H. Kim, “A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMs for high-density cache memory,” *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 598–610, Feb 2013.

[156] D. Ielmini, “Modeling the universal set/reset characteristics of bipolar RRAM by field- and temperature-driven filament growth,” *IEEE Transactions on Electron Devices*, vol. 58, pp. 4309–4317, Dec 2011.

[157] Z. Xu and J. Q. Lu, “Through-silicon-via fabrication technologies, passives extraction, and electrical modeling for 3-D integration/packaging,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 26, pp. 23–34, Feb 2013.

[158] P. Batude, M. Vinet, B. Previtali, C. Tabone, C. Xu, J. Mazurier, O. Weber, F. Andrieu, L. Tosti, L. Brevard, B. Sklenard, P. Coudrain, S. Bobba, H. B. Jamaa, P. E. Gaillardon, A. Pouydebasque, O. Thomas, C. L. Royer, J. M. Hartmann, L. Sanchez, L. Baud, V. Carron, L. Clavelier, G. D. Micheli, S. Deleonibus, O. Faynot, and T. Poiroux, “Advances, challenges and opportunities in 3D CMOS sequential integration,” in *Electron Devices Meeting (IEDM), 2011 IEEE International*, pp. 7.3.1–7.3.4, Dec 2011.

[159] S. Wong, A. El, P. Griffin, Y. Nishi, F. Pease, and J. Plummer, “Monolithic 3D integrated circuits,” in *2007 International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, pp. 1–4, April 2007.

[160] A. L. P. Rotondaro, M. R. Visokay, J. J. Chambers, A. Shanware, R. Khamankar, H. Bu, R. T. Laaksonen, L. Tsung, M. Douglas, R. Kuan, M. J. Bevan, T. Grider, J. McPherson, and L. Colombo, "Advanced CMOS transistors with a novel HfSiON gate dielectric," in *VLSI Technology, 2002. Digest of Technical Papers. 2002 Symposium on*, pp. 148–149, June 2002.

[161] F. Carta, S. M. Gates, A. B. Limanov, J. S. Im, D. C. Edelstein, and I. Kymissis, "Sequential lateral solidification of silicon thin films on cu beol-integrated wafers for monolithic 3-D integration," *IEEE Transactions on Electron Devices*, vol. 62, pp. 3887–3891, Nov 2015.

[162] D. Frank and L. Chang, "Technology optimization for high energy-efficiency computation," 2012. IEDM Short Course.

[163] M. Jung, J. Mitra, D. Z. Pan, and S. K. Lim, "TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC," *Commun. ACM*, vol. 57, pp. 107–115, Jan. 2014.

[164] M. Nakashima, "High performance and highly reliable ssd," 2015.

[165] R. Braojos, I. Beretta, J. Constantin, A. Burg, and D. Atienza, "A wireless body sensor network for activity monitoring with low transmission overhead," in *Embedded and Ubiquitous Computing (EUC), 2014 12th IEEE International Conference on*, pp. 265–272, Aug 2014.

[166] R. Braojos, D. Bortolotti, A. Bartolini, G. Ansaloni, L. Benini, and D. Atienza, "A synchronization-based hybrid-memory multi-core architecture for energy-efficient biomedical signal processing," *IEEE Transactions on Computers*, vol. PP, no. 99, pp. 1–1, 2016.

[167] R. Braojos, D. Atienza, M. M. S. Aly, T. F. Wu, H.-S. P. Wong, S. Mitra, and G. Ansaloni, "Nano-engineered architectures for ultra-low power wireless body sensor nodes," in *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, CODES '16, (New York, NY, USA), pp. 23:1–23:10, ACM, 2016.

[168] F. Catthoor, P. Raghavan, A. Lambrechts, M. Jayapala, A. Kritikakou, and J. Absar, *Ultra-low energy domain-specific instruction-set processors.* Springer Science & Business Media, 2010.

[169] L. G. Duch, S. S. Basu, R. Braojos Lopez, G. Ansaloni, L. Pozzi, and D. Atienza Alonso, "A multi-core reconfigurable architecture for ultra-low power bio-signal analysis," in *Biomedical Circuits and Systems (BioCAS)*, no. EPFL-CONF-220721, 2016.

Rubén **Braojos López**

Avenue du Chablais, 42 ● Lausanne, 1007 ● Switzerland
E-mail: rubenbraojoslopez@gmail.com ● Mobile: (+41) 76 712 63 18 - (+34) 647 941 299
Date and place of birth: 4th May, 1987. Toledo (Spain)

INTERESTS

Computer Architecture, Low-power Multi-Core Platforms, Embedded Systems, Wireless Body Sensor Networks, Bio-signal Processing, Embedded Software Optimization

EDUCATION

09/2011-	PhD candidate at the Embedded Systems Laboratory (ESL) École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
08/2009-07/2010	Erasmus Exchange Scholarship. École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. (Last year of the degree in Computer Engineering)
09/2005-07/2010	Bachelor of Science and Master of Science in Computer Engineering. Universidad Complutense de Madrid, Spain.

WORK EXPERIENCE

06/2015-07/2015	Research Intern IMEC Belgium
05/2011-09/2011	Software Engineer Credit Suisse AG, Lausanne, Switzerland
07/2010-05/2011	Embedded Software Engineer Embedded Systems Laboratory (ESL), Switzerland

TEACHING:

2010-2016	Teaching assistant for the Bachelor-level course “Microprogrammed Embedded Systems” (EE-310) École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

TECHNICAL SKILLS

OPERATING SYSTEMS	UNIX based OS (Linux, MacOS, iOS), Windows Family OS, Real Time OS (uCOS-II, TinyOS)
HARDWARE DESIGN	VHDL, Verilog, SystemC
PROGRAMMING LANGUAGES	C/C++, Java, Pascal, PHP, ARM Assembly Language, Matlab, Shell scripting, Perl, databases (MySQL)

LANGUAGES

Spanish: Native Speaker
English: Proficient (C2)
French: Post-intermediate (B2/C1)

SELECTED PUBLICATIONS

- R. Braojos, T. F. Wu, G. Ansaloni, M. Sabry, S. Mitra, H.-S. P. Wong and D. Atienza. ***“Nano-Engineered Architectures for Ultra-Low Power Wireless Body Sensor Nodes.”*** International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2016
- R. Braojos, D. Bortolotti, A. Bartolini, G. Ansaloni, L. Benini, and D. Atienza. “**A Synchronization-Based Hybrid-Memory Multi-Core Architecture for Energy-Efficient Biomedical Signal Processing”**, accepted at IEEE Transactions on Computers
- R. Braojos, I. Beretta, G. Ansaloni and D. Atienza. “**Early Classification of Pathological Heartbeats on Wireless Body Sensor Nodes”**, in Sensors, vol. 14, num. 12, p. 22532-22551, 2014.
- R. Braojos, H. Mamaghanian, A. Dias Junior, G. Ansaloni, D. Atienza, F. J. Rincon and S. Murali. ***“Ultra-Low Power Design of Wearable Cardiac Monitoring Systems.”*** Design Automation Conference (DAC), 2014
- R. Braojos, I. Beretta, J. H.-F. Constantin, A. P. Burg and D. Atienza. ***“A Wireless Body Sensor Network For Activity Monitoring With Low Transmission Overhead.”*** IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2014.
- J. Milosevic, A. Dittrich, A. Ferrante, M. Malek, D. C. Rojas, R. Braojos, G. Ansaloni and D. Atienza. **“Risk Assessment of Atrial Fibrillation: a Failure Prediction Approach.”**, Computers in Cardiology Conference, 2014.
- R. Braojos, I. Beretta, G. Ansaloni and D. Atienza. **“Hardware/Software Approach for Code Synchronization in Low-Power Multi-Core Sensor Nodes”**. Design Automation and Test in Europe (DATE) Conference, 2014.
- R. Braojos, G. Ansaloni and D. Atienza. **“A Methodology for Embedded Classification of Heartbeats Using Random Projections”**. Design Automation and Test in Europe (DATE) Conference, 2013
- A. Y. Dogan, R. Braojos, J. H.-F. Constantin, G. Ansaloni, A. P. Burg and D. Atienza. **“Synchronizing Code Execution on Ultra-Low-Power Embedded Multi-Channel Signal Analysis Platforms”**. Design Automation and Test in Europe (DATE) Conference, 2013
- R. Braojos, G. Ansaloni, D. Atienza, R. Vallejos and F. Javier. **“Embedded Real-Time ECG Delineation Methods: a Comparative Evaluation”**. IEEE 12th International Conference on BioInformatics and BioEngineering (BIBE), 2012.

PATENTS

- Braojos Lopez et al., **"Method for Detecting Abnormalities in an Electrocardiogram"**.
  U.S. Patent no. US9468386 B2, (October 18th , 2016)

OTHER ACHIEVEMENTS

- SIGDA/ACM Travel Grant, PhD Forum at DAC Conference, U.S.A. 2015
- Best Poster Award Winner, Nano-Tera.ch Annual Meeting, Switzerland 2014
- Hi-PEAC Paper Award Winner, DAC Conference, U.S.A. 2014
- Best Paper Candidate. DATE Conference, France, 2013
- Fellowship holder during the academic year 2008/2009 in the DSIC (Department of Computer Systems and Computing) from the Faculty of Computer Science of the Complutense University of Madrid.
- Fellowship “Aprovechamiento Academico Excelente” (*“Excellent academic performance”*) from the regional government of Madrid during the academic years 2005/2006, 2006/2007 and 2007/2008.