EffiCSense: an Architectural Pathfinding Framework for Energy-Constrained Sensor Applications

Jonah Van Assche, Ruben Helsen, Georges Gielen
ESAT-MICAS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven (Heverlee), Belgium
Email: jonah.vanassche@kuleuven.be

***Abstract*—This paper introduces EffiCSense, an architectural pathfinding framework for mixed-signal sensor front-ends for both regular and compressive sensing systems. Since sensing systems are often energy constrained, finding a suitable architecture can be a long iterative process between high-level modeling and circuit design. We present a Simulink-based framework that allows for architectural pathfinding with high-level functional models while also including power consumption models of the different circuit blocks. This allows to directly model the impact of design specifications on power consumption and speeds up the overall design process significantly. Both architectures with and without compressive sensing can be handled. The framework is demonstrated for the processing of EEG signals for epilepsy detection, comparing solutions with and without analog compressive sensing. Simulations show that using the compression, an optimal design can be found that is estimated to be 3.6 times more power-efficient compared to a system without compression, consuming 2.44μW for a detection accuracy of 99.3%.**

***Index Terms*—Compressive sensing, Sensor Front-End, Simulink, System Modeling**

I. Introduction

In recent years, there has been a surge in portable and implantable, always-on devices. These devices allow for a constant monitoring of patients outside of healthcare facilities and provide numerous benefits such as early detection of diseases, direct intervention in case of emergency, etc. [1].

In Fig. 1 a) a typical biomedical monitoring system is depicted. For battery-operated sensor platforms, data transmission or storage is a dominant contributor to the power consumption [2], [3], reducing battery life. Since these applications typically have to work for long periods of time, some form of data compression is required. However, using complex compression schemes on-chip is not always an option, since the digital circuits required often also consume a significant amount of power.

There is thus a trade-off in the power consumption of the transmitter, the processing circuits and the applied compression [4]. When designing such a platform for an application, it is not always clear where the bottleneck is in terms of power consumption and what the optimal system architecture is for a given application. Moreover, when introducing a new circuit or compression technique, it is not always clear what the impact will be at the system level. Hence, a long iterative process between high-level modeling and simulation on the one hand and circuit design on the other hand is needed to determine the optimal chip architecture and block specifications. This paper therefore describes the EffiCSense pathfinding framework based on the MATLAB Simulink environment [5] that addresses this problem. By combining both high-level functional and power models in a plug-and-play fashion, the effect of architectural design choices on the overall signal acquisition performance and power consumption can be estimated simultaneously, allowing to find a power-optimal configuration for a specific application.

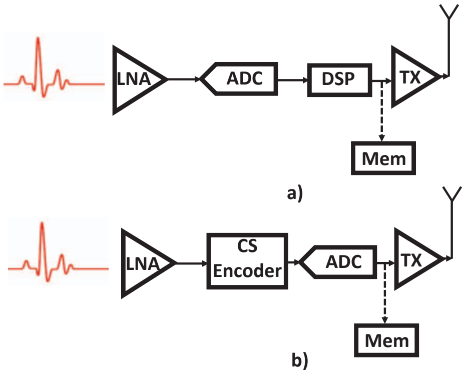


Fig. 1: a) Architecture of a classical biomedical system and b) a compressive sensing based system.

A recent technique for data compression is compressive sensing (CS). It simultaneously compresses and samples the signal, lowering both the data rate as well as the power consumption of the ADC. Due to its many degrees of freedom (choice of sensing matrix, architecture, reconstruction, etc.) there are many different designs and proposals in literature [2], [6]–[9], but there is no clear view on which technique is more or less efficient for a target application. The modeling in the EffiCSense framework is demonstrated in this paper for a new CS technique that uses only capacitors for the CS encoder (see Fig. 1 b), implementing a passive charge-sharing switch-capacitor circuit, that is more power efficient compared to active implementations using OTAs [10]. Because of its analog nature, the proposed technique is however susceptible to typical analog imperfections like mismatch and noise. To demonstrate the framework, we use EffiCSense to estimate the effect of design choices on the overall power consumption and simulate the effect of the analog imperfections on the achieved signal quality.

This paper is organized as follows. In Section II, the Simulink-based pathfinding framework to simulate and optimise sensing architectures is described. In Section III, the modeling of the new compressive sensing technique is presented. In Section

978-3-9819263-6-1/DATE22/©2022 EDAA

TABLE I: Mixed-Signal and Power Modeling Frameworks

	High-Level Behavioral Modeling [11]	FOM-based [2], [12]	EffiCSense
Target Application	$\Delta\Sigma$ ADCs	CS applications	Sensor Front-Ends
Mixed-Signal Modeling	Yes	No	Yes
Power Modeling	No	Yes	Yes
Method	/	FOM/Ideal Model	FOM/Analytical Model
Application Specific	No	Yes	No

IV, experimental results using the EffiCSense framework for an epilepsy detection application show the advantages of the developed framework. Conclusions are given in Section V.

II. Architectural Pathfinding Framework

High-level modeling of circuits is a well established design step for mixed-signal systems [11], [13], providing a quick way of verifying design solutions, that would take much longer by running detailed circuit simulations. However, with designs that are becoming increasingly complex (with many possible design parameters) and energy constrained, it is desirable to directly estimate the power consumption of the entire system, since iterations between the system level and the circuit level would take up too much time (and can possibly lead to a sub-optimal design). There exist several techniques to estimate the power consumption of a design, as listed in Table I.

For mixed-signal systems, in particular compressive sensing systems, there exists literature that analyses the effect of compressive sensing techniques on the power consumption at system level [2], [12]. They use a combination of Figures-of-Merit (FOMs) and idealised circuit models to estimate the power consumption of different building blocks. LNAs, ADCs, LDOs, VCOs, etc. all have FOMs that express how well the designed circuit performance relates between speed, accuracy and power. Using FOMs, energy efficiency trends of real circuit implementations can be found. However, not all relevant parameters are captured by a FOM. For example, the Walden FOM for ADCs only captures the total SNDR of the ADC, but doesn't reveal any other information about the ADC that is relevant for system-level modeling. Also, idealised models of building blocks are not useful for mixed-signal design, as it are the non-idealities of a building block that are important for a designer. A more useful method for estimating power consumption of design parameters compared to using FOMs is to use power bound analysis [14]. An abundance of literature is available on the analysis of different building blocks/circuits [3], [14], [15], in which the limits of these circuits are explored. While these models often make assumptions, they can offer a first-order estimate on the power consumption when used in the right way. For system-level exploration, these estimates can provide an insight in how the power consumption and performance of different building blocks relate to each other and to the system as a whole, and on which blocks most design emphasis should be. This is why EffiCSense uses as much as possible this type of models. Here is also a key contribution of EffiCSense: it provides a library of models for sensing systems that simultaneously include functional parameters and estimates for power consumption that are a function of those parameters. This means that it can be used for real mixed-signal designs. As it is an open framework, other collaborators will eventually be able to contribute/extend the library. An overview of how EffiCSense is built up is given in Fig. 2, we now describe each step in more detail.

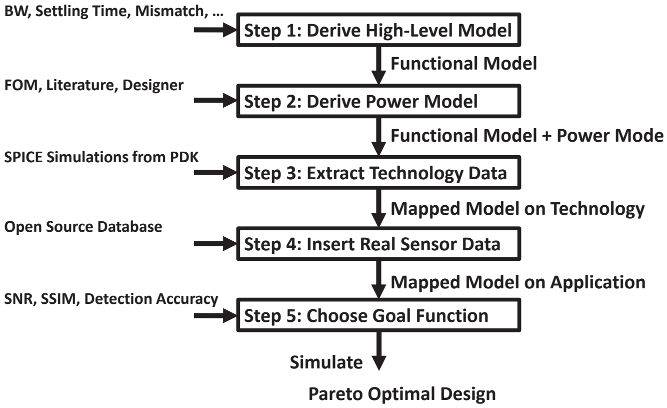

BW, Settling Time, Mismatch, ...
Step 1: Derive High-Level Model
Functional Model
FOM, Literature, Designer
Step 2: Derive Power Model
Functional Model + Power Mode
SPICE Simulations from PDK
Step 3: Extract Technology Data
Mapped Model on Technology
Open Source Database
Step 4: Insert Real Sensor Data
Mapped Model on Application
SNR, SSIM, Detection Accuracy
Step 5: Choose Goal Function
Simulate
Pareto Optimal Design

Fig. 2: Flowchart of the EffiCSense pathfinding framework.

Step 1: Derive High-Level Model

The first step in the design is to identify the relevant building blocks of the mixed-signal system to be used for the target applications. As shown in Fig. 1 a), a general biomedical system contains an LNA to amplify the weak biosignals that are recorded, an ADC to digitize the analog signal (the most commonly used architecture is the SAR ADC, which is also used in this work), a DSP block for signal conditioning and a transmitter/memory to transmit/store the data. Starting from this basic system architecture, a suitable circuit topology can be chosen for each of the blocks (e.g. what type of LNA, what type of ADC, etc.). Then, each block is functionally modeled (or a model is taken from the library). For example, an LNA can be modeled by first adding white noise to the signal, then multiplying this sum with the gain and limiting the bandwidth by low-pass filtering this signal, then adding a non-linearity and possibly clipping of the output signal. This illustrative Simulink model is shown in Fig. 3.

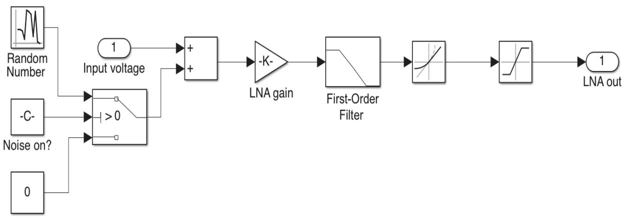


Fig. 3: A simple LNA model showing gain, noise, non-linearity and clipping.

TABLE II: Power Models of the Different Building Blocks

Circuit	Power Model	Reference
LNA	$V_{dd} \cdot max \begin{cases} \text{GBW}_{\text{LNA}} \cdot 2\pi \cdot C_{load} \cdot \frac{g_m}{I_d} \\ V_{ref} \cdot f_{clk} \cdot C_{load} \\ \left(\frac{NEF}{\text{LNA}_{\text{noise floor}}}\right)^2 \cdot 2\pi \cdot 4kT \cdot BW_{LNA} \cdot V_T \end{cases}$	[16]
Sample & Hold	$V_{ref} \cdot f_{clk} \cdot 12kT \frac{2^{2N}}{V_{FS}^2}$	[14]
Comparator	$2N \cdot \ln(2) \cdot (f_{clk} - f_{sample}) \cdot C_{load} \cdot V_{FS} \cdot V_{eff}$	[14]
SAR logic (N bit)	$\alpha \cdot (2N+1) \cdot C_{logic} \cdot V_{dd}^2 \cdot (f_{clk} - f_{sample}) \qquad (\alpha = .4)$	[17]
DAC	$\frac{2^N \cdot f_{clk} \cdot C_u}{N+1} \left\{ \left(\frac{5}{6} - \left(\frac{1}{2}\right)^N - \frac{1}{3} \cdot \left(\frac{1}{2}\right)^{2N} \right) \cdot V_{ref}^2 - \frac{1}{2} V_{in}^2 - \left(\frac{1}{2}\right)^N \cdot V_{in} \cdot V_{ref} \right\}$	[15]
Transmitter	$\frac{f_{clk}}{N+1} \cdot N \cdot E_{bit}$	[4], [12]
CS Encoder Logic	$\alpha \cdot (\lceil \log_2(N_\Phi) \rceil + 1) \cdot N_\Phi \cdot 8C_{\text{logic}} \cdot V_{\text{dd}}^2 \cdot (f_{\text{clk}}) \qquad (\alpha = 1)$	[17]

TABLE III: Extracted Technology Parameters & Design parameters

Technology Parameter/Constant	Symbol	Value
Gate Capacitance Logic	C_{logic}	1 fF
gm/I$_\text{D}$	/	20 / V
MIM Capacitor Density	/	.001025 F/μm^2
Minimum MIM Capacitor	$\text{C}_{\text{u,min}}$	1 fF
Capacitor mismatch	C_{pk}	$3.48 \cdot 10^{-9}$ %/μm^2
Leakage Current	I_{leak}	1 pA
Transmission energy per bit	E_{bit}	1 nJ
Thermal Voltage	V_T	25.27 mV
Design Parameter	Symbol	Value
Input bandwidth	BW_{in}	256 Hz
Φ dimensions	M, N_Φ	75-150-192, 384
C_{hold}/C_{sample}	/	1 - 20
ADC Resolution	N	6 - 8 bit
Supply Voltage	V_{dd}	2 V
Sample frequency	f_{sample}	$2.1 \cdot BW_{Input}$
Clock frequency	f_{clk}	$(N+1) \cdot f_{sample}$
Full Scale & Reference Voltage	V_{FS} & V_{ref}	2 V
LNA Bandwidth	BW_{LNA}	$3 \cdot BW_{Input}$

Step 2: Derive Power Models

Once the functional model and relevant non-idealities for a block are developed, a power model can be derived as a function of the relevant circuit parameters. As mentioned, there already exist validated power models for many circuits/building blocks in literature. As discussed, for the power models, we mostly use power bound theoretical models. Table II gives an overview of the different power models for each building block in the library, the parameters can be found in Table III.

Step 3: Extract Relevant Technology Parameters

The choice of technology has a large impact on the system-level performance. However, technology data is not freely available. Therefore, some parameters should be extracted through circuit simulations in the specific technology. For the framework demonstrations, we extracted parameters from a gpdk045 predictive technology, using the Cadence Virtuoso simulation environment. The extracted relevant technology parameters that are used in the power models are listed in Table III.

Step 4: Insert Real Sensor Data

To simulate the application, actual sensor data must be used to asses the system-level performance. Many databases exist that offer a wide range of recorded biomedical signals, with and without artefacts [18], [19]. For the epilepsy detection application used in this paper, we used EEG signals from [18]. Once a proper database has been chosen, the signals can be upsampled to mimic a continuous-time signal. The EEG signals sampled at 173.61 Hz were upsampled to 512 Hz, as it was determined that this gave a sufficient quality signal.

Step 5: Choose Goal Function

After completing Step 1 to Step 4, the system can now be simulated and both the performance and power consumption can be analyzed. The last step is to choose a proper goal function. A mixed-signal sensing system ultimately aims to capture a signal with a certain signal quality. SNR or SNDR are obvious parameter to describe this, but other metrics can be thought of as well, such as the accuracy in detecting certain signal characteristics (like the seizure classification accuracy in EEG signals for example). Depending on the application, a proper goal function should be chosen. After doing so, simulations can be executed. For a mixed-signal designer, it is often important to see the trade-offs between different parameters. Our EffiCSense framework allows to sweep any modeled range of parameters to estimate their effect on the system performance and power consumption. As example, a sweep of the input-referred noise of the LNA versus the SNDR for the system in Fig. 1 a) (using a sine as input) is shown in Fig. 4, to illustrate the capabilities of the framework. The figure shows how the input-referred noise of the LNA influences the power consumption and SNDR. It also shows how the power consumption is distributed across the different building blocks.

III. Modeling Use Case: Analog Compressive Sensing

To highlight the capabilities of the EffiCSense framework, it is shown here how a novel passive analog CS front-end is modeled and entered into the framework. The CS front-end is shown in Fig. 5. In the proposed architecture, an array of switched capacitors is added to the baseline architecture of Fig. 1 a). This array functions as the compressive sensing encoder (see

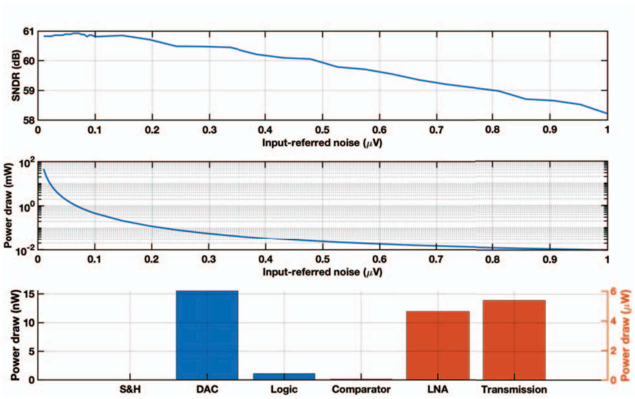


Fig. 4: Sweeping the input-referred noise of the standard acquisition system model of Fig. 1 a) and impact on the system SNDR and power consumption. The bottom plot shows the distribution of the power across the different blocks.

Fig. 1 b)) and performs passive matrix multiplications, which removes the need for active integrators and their power-hungry OTAs [2].

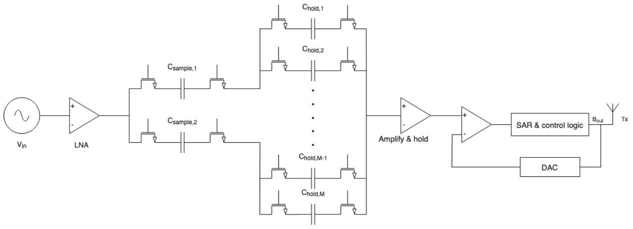


Fig. 5: Proposed passive, charge-sharing CS architecture.

Charge-sharing multiplication

The proposed CS architecture is based on passive charge sharing using capacitors [10]. Fig. 6 shows the example of a stray-insensitive charge-sharing architecture. In clock phase ϕ_1, C_1 is charged to v_{in} at that time, V_1, accumulating a charge of $C_1 \cdot V_1$. Then, during phase ϕ_2, this charge is shared across C_1 and C_2. This results in a voltage of $\frac{C_1}{C_1+C_2}V_1$ over both capacitors. In a second ϕ_1, C_1 is charged to a new voltage V_2, while the charge on C_2 is preserved. In ϕ_2, the charge on C_1 is again shared over both capacitors. By repeating this procedure, a weighted sum of the sampled input voltages can be made:

$$V_{\text{sum}} = \sum_{j=1}^{N} V_j \cdot \frac{C_1}{C_1 + C_2} \cdot \left(\frac{C_2}{C_1 + C_2}\right)^{N-j}. \qquad (1)$$

The charge-sharing technique thus makes it possible to perform additions and multiplications and can thus act as an analog counterpart to digital multiply and accumulate (MAC) operations.

Operation principle

The core operation of compressive sensing is a matrix-vector multiplication $\vec{y} = \Phi\vec{x}$ with $\vec{x}$ an $N \times 1$ input vector and $\vec{y}$ an $M \times 1$ output vector (M<<N). Each consecutive sample must be multiplied with a single column of the sensing matrix Φ. Typical sensing matrices have a high computational workload for each input sample, requiring many MAC operations. Additionally, large amounts of on-chip storage is required to store the matrix. By using random, sparse and binary matrices, it is possible to lower the computational complexity and minimize the memory requirements [9]. Our work uses such s-Sparse Random Binary Matrices (s-SRBM), where s indicates the amount of non-zero elements in a matrix column and thus the amount of partial sums a sample should be added to. Because each column of an s-SRBM only contains s ones, only s additions need to be performed. The architecture, shown in Fig. 5, implements this operation with a capacitor bank and a set of switches. Samples are taken and stored on the capacitors C_{sample}, while intermediate results are stored on the capacitors C_{hold}. The sizes of these capacitors are determined by the size of Φ, the matching requirements and the noise specifications. The network of stray-insensitive switches then determines on which C_{hold} a sample is redistributed, thus determining to which intermediate products it is added. The architecture shown in Fig. 5 has two C_{sample} capacitors, it is designed for 2-SRBM matrices, taking and redistributing two samples each cycle. In this way, the compressive sensing operation is performed fully passively!

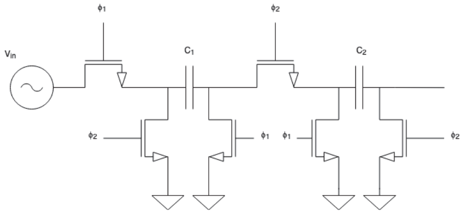


Fig. 6: Stray-insensitive switches used in the architecture of Fig. 5.

Modeling the new CS front-end

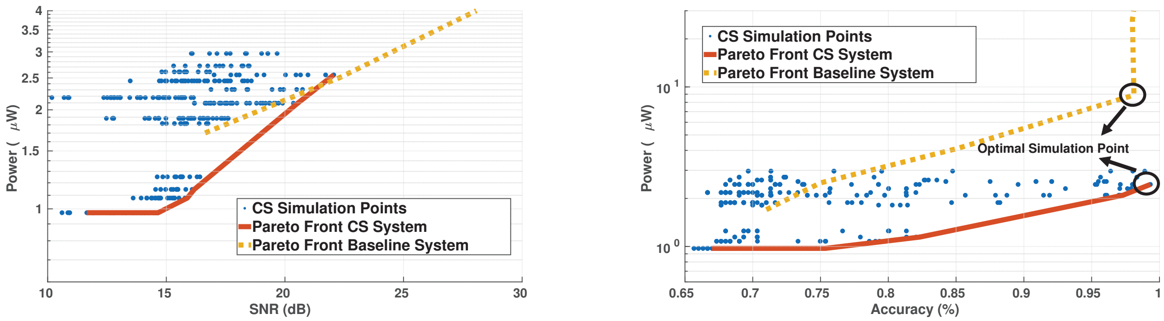
Starting from the classical system, the CS encoder now needs to be included in the system-level architecture (see Fig. 1b). This is done like any of the other blocks: derive a functional model based on the desired front-end, include non-linear behavior, noise and derive a power model. The results of this new CS system can then be compared to the baseline system very easily, allowing the designer to more quickly explore different kinds of front-ends (e.g. digital vs analog or active vs passive compressive sensing). The CS logic consists of a shift register (consisting of D flip-flops) that stores the values of the sensing matrix and of the switches in the charge-sharing network. Assuming minimal sized switches and inverters are used in the charge-sharing network and flip-flops, an expression for the CS Encoder power can be estimated by multiplying the amount of gates/switches needed (this depends on the size of the sensing matrix) with the minimal logic capacitance. From this total logic capacitance, the dynamic power consumption can be easily estimated. The power expression for this CS encoder logic can be found in Table II. Besides this new expression, the load of the LNA should also be taken equal to the C_{hold} value of the CS encoder, since this becomes the load capacitance of the LNA (as can be seen in Fig. 5).

IV. Experimental Results

The developed EffiCSense framework with the extended library is now demonstrated for the application of epilepsy detection. The impact of using analog CS is compared to the case without CS, to evaluate the possible advantage of the analog CS technique. To evaluate the accuracy of the front-end, the neural network developed in [20] is used to detect epileptic seizures in EEG signals. The minimum required detection accuracy that is set for this application is 98 %. Since power consumption is a primary concern for wearable sensor systems, power was chosen as the optimization goal.

Fig. 7 shows the results from a sweep over a search space (the different design parameters that were swept can be found in Table III). Each point represents a set point that was evaluated over all 500 23.6-second signals from the dataset and averaged. The left plot shows the Pareto fronts for both the baseline system and the compressive sensing system, as a function of the achieved SNR. A trend can clearly be distinguished: for lower SNR, the CS technique gives an advantage, while the classical system outperforms the CS system for higher SNR. Fig. 7 b) plots the same search space as Fig. 7 a), but with detection accuracy used to evaluate the output of the sensor front-end. The trend line of the CS system now clearly shows that the CS system outperforms the baseline system over the whole detection range. This shows that the choice of the accuracy metric can really have an impact on the system parameters and the optimal solution. The optimal design solution for the baseline system reaches an accuracy of 98.1% at a power consumption of 8.8 μW, while the CS system reaches an accuracy of 99.3% while at a consumption of 2.4 μW, which is a power saving of 3.6 times. For these two optimal solutions, the distribution of the power over the different building blocks is given in Fig. 8. It becomes clear that the CS system saves power by allowing a lower transmission power and LNA power compared to the baseline system. The first insight is to be expected, since the compressive sensing technique takes fewer samples than the baseline architecture, but the second insight of the lower LNA power is less straight-forward. This showcases why EffiCSense is a useful design tool: it allows a designer to explore trends, even one that seems less straightforward at first sight. The reason why the LNA power is lower is due to the higher tolerated noise floor of the compressive sensing technique. This does not necessarily lead to a lower accuracy, since for the compressive sensing technique, it is the output of the CS encoder (which is the sum of multiple samples) that is digitized and not the signal at the sensor input. There is a sort of averaging effect in place, which allows for a higher noise floor to be tolerated compared to the baseline system. The CS technique does increase the power consumption of the digital logic (mainly due to the extra logic for the CS encoder), but this is only a marginal increase compared to the power savings in the LNA and the transmission, and therefore can be accepted.

Besides power consumption, chip area can also be a concern for designers. In mixed-signal designs, most area is occupied by capacitors, and an estimate on the amount of area used by a design can be derived from the total amount of capacitors that is used. In Fig. 9, the total amount of capacitors (expressed in multiples of the minimum technology capacitor $C_{u,min}$), is plotted as a function of the accuracy. The CS technique clearly increases the area of the system significantly compared to the baseline system. It is also possible to do an area constrained search: the resulting Pareto fronts for different constraints are depicted in Fig. 10. In comparison with the unconstrained search, it is clear that the area constraints limit the maximum accuracy that can be achieved, keeping the capacitor area small drastically limits the maximum achievable accuracy. Thus, the CS technique can be more favorable compared to the baseline system, if an increase in chip area is tolerated. In a realistic scenario, if a chip is bondpath limited (that is, the chip area is determined by the amount of bondpaths required), this increase in active area can be feasible.



(a) (b)

Fig. 7: a) SNR vs power consumption and b) Detection accuracy vs power consumption, where the best configuration of the search space that fulfills the constraints is indicated. Note that the use of different accuracy metrics can lead to different optimal solutions!

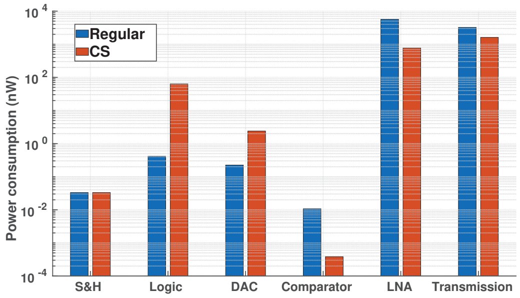


Fig. 8: Comparison between the optimal simulation point of the baseline system (in blue) and the compressive sensing system (in orange). A significant reduction in transmission power and LNA power consumption is noted.

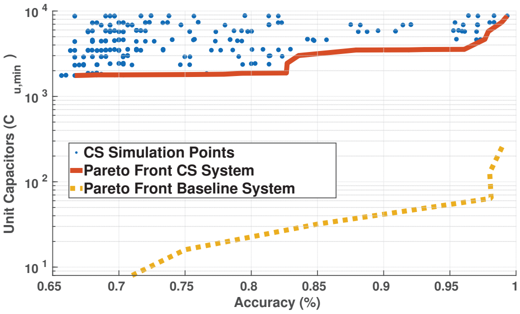


Fig. 9: Accuracy vs size of capacitors used in the design (expressed as a function of the minimum technology capacitor $C_{u,min}$).

V. Conclusion

This paper has introduced EffiCSense, a framework aimed at architectural pathfinding exploration of energy-constrained sensing applications, with and without compressive sensing. The high-level framework provides both performance and power estimates based on combined functional and power models. The steps in deriving a model for a new architecture have been explained and applied to a novel passive compressive sensing front-end. This model was then used to optimize the system architecture for epilepsy detection on an EEG dataset. The compressive sensing technique is estimated to achieve a 3.6x power reduction compared to a traditional system, consuming only 2.44 μW for a 99.3 % detection accuracy. The use case also demonstrated multiple system-level trends, and a clear trade-off between power efficiency and area.

Acknowledgment

This research has been funded in part by the "Time is money" (3E180440) internal grant of KU Leuven.

References

[1] Andreu-Perez *et al.*, "From wearable sensors to smart implants-–toward pervasive and personalized healthcare," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 12, pp. 2750–2762, 2015.

[2] F. Chen *et al.*, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

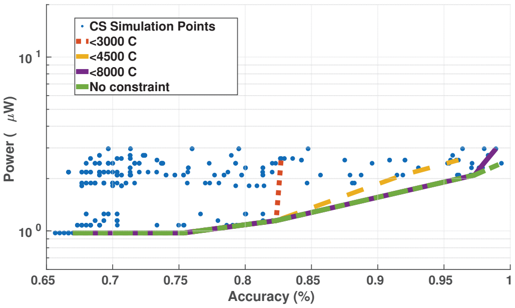


Fig. 10: Detection accuracy vs power consumption, with a constraint on the maximum area (total capacitance value) that can be used.

[3] M. Saberi *et al.*, "Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 8, pp. 1736–1748, 2011.

[4] D. Bortolotti *et al.*, "An ultra-low power dual-mode ecg monitor for healthcare and wellness," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015, pp. 1611–1616.

[5] "Simulink - Simulation and Model-Based Design - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/products/simulink.html

[6] W. Guo *et al.*, "A Fully Passive Compressive Sensing SAR ADC for Low-Power Wireless Sensors," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 8, pp. 2154–2167, 2017.

[7] A. Anvesha *et al.*, "A 65nm compressive-sensing time-based ADC with embedded classification and INL-aware training for arrhythmia detection," *2017 IEEE, BioCAS 2017 - Proceedings*, vol. 2018-January, pp. 1–4, 2018.

[8] Z. Zhang *et al.*, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221–224, 2013.

[9] W. Zhao *et al.*, "On-Chip Neural Data Compression Based on Compressed Sensing with Sparse Sensing Matrices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 242–254, 2018.

[10] E. H. Lee and S. S. Wong, "Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 261–271, 2017.

[11] P. Malcovati *et al.*, "Behavioral modeling of switched-capacitor sigma-delta modulators," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 3, pp. 352–364, 2003.

[12] D. E. Bellasi and L. Benini, "Energy-efficiency analysis of analog and digital compressive sensing in wireless sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 11, pp. 2718–2729, 2015.

[13] S.-Y. Lee and C.-J. Cheng, "Systematic design and modeling of a ota-c filter for portable ecg detection," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 1, pp. 53–64, 2009.

[14] T. Sundström *et al.*, "Power dissipation bounds for high-speed nyquist analog-to-digital converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 509–518, 2009.

[15] J. Van Assche and G. Gielen, "Power Efficiency Comparison of Event-Driven and Fixed-Rate Signal Conversion and Compression for Biomedical Applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 746–756, 2020.

[16] M. S. Steyaert *et al.*, "A Micropower Low-Noise Monolithic Instrumentation Amplifier For Medical Purposes," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 6, pp. 1163–1168, 1987.

[17] T. Bos *et al.*, "Architecture optimization for energy-efficient resolution-scalable 8-12-bit SAR ADCs," *Analog Integrated Circuits and Signal Processing*, vol. 97, no. 3, pp. 437–448, 2018.

[18] "Bonn university eeg dataset," https://physionet.org/content/chbmit/1.0.0/.

[19] "Chb-mit dataset," https://physionet.org/content/chbmit/1.0.0/.

[20] I. Ullah *et al.*, "An automated system for epilepsy detection using EEG brain signals based on deep learning approach," *arXiv*, no. Ml, 2018.