Scalable EEG Seizure Detection on an Ultra Low Power Multi-Core Architecture

S.Benatti*, F.Montagna*, D.Rossi*, L.Benini*‡
*DEI, University of Bologna, Italy. Email: {simone.benatti,fabio.montagna, davide.rossi,luca.benini}@unibo.it
‡Integrated System Laboratory, ETHZ, Zurich, Switzerland. Email: lbenini@iis.ee.ethz.ch

***Abstract*—Energy efficient processing architectures represent key elements for wearable and implantable medical devices. Signal processing of neural data is a challenge in new designs of Brain Machine Interfaces (BMI). A highly efficient multi-core platform, designed for ultra low power processing allows the execution of complex algorithms complying with real time requirements. This paper describes the implementation and optimization of a seizure detection algorithm on a multi-core digital integrated circuit designed for energy efficient applications. The proposed architecture is able to implement ultra low power parallel processing seizure detection on 23 electrodes within a power budget of 1 mW, outperforming implementations on commercial MCUs by up to 100 times in terms of performance and up to 80 times in terms of energy efficiency still providing high versatility and scalability, opening the way to the development of efficient implantable and wearable smart systems.**

I. Introduction

Recent advancements in Brain Machine Interfaces (BMI) are paving the way to systems for treating various neural diseases. Among these studies, treating epilepsy has a great impact on public health, since this neural disorder affects approximately 1% of the world population and can result in severe and disabling pathologies. In epilepsy, the normal pattern of neuronal activity becomes disturbed, causing depression, convulsions or loss of consciousness. The clinical measurement of the brain electrical activity through the analysis of the EEG traces, and the expertise of the neurologist can diagnose the epileptic seizure recognizing certain changes in patterns of amplitude and frequency of the neural signal. The therapeutic approach is mainly pharmacological or surgical. Unfortunately, for about 30% of epileptic subjects, seizures cannot be controlled with drugs delivery nor surgical techniques; but react to neuromodulation [1], a technique based on direct electrical stimulation of the brain tissue. In this scenario, the development of automatic closed loop neuromodulation systems can reduce the time of reaction many orders of magnitude more than human intervention. Furthermore, a closed-loop system provides stimulation only when triggered by seizure detection, hence it is less traumatic wrt first generations of neuromodulators, which just deliver continuous, constant stimulation [2].

The design of these systems requires a holistic approach in the development of sensors, digital architectures and algorithms to process the brain signals. Neuromodulation systems are based on algorithms that analyze the EEG signal to detect changes that may represent seizure activity [3]. While the trend in research goes toward design of dense multichannel systems, with large and dense arrays of sensors, to allow a fine grain coverage of the brain surface and target wider areas, commercial SoA devices like Medtronic Activa PC+S and Neuropace RNS [4] are only able to manage up to 4 electrodes with a latency of 500ms due to their limited computing power. A smaller detection latency is also desirable to react as fast as possible to a seizure or perform early data acquisition. Several studies show that machine learning techniques can achieve high accuracy in seizure detection [5]. Many inspiring solutions have been proposed both at chip [6] and system level [7]. Furthermore, extensive research is being done on signal processing as well, helping to detect when the seizure begins [8]. Most of these approaches include dimensionality reduction, feature extraction and pattern recognition algorithms for classification [9].

Several research ASICs, designed for very specific tasks, reach remarkable performance in terms of power consumption but they totally lack flexibility. The work of [10], based on a single channel, can detect a limited class of seizures with 50uW. In [11], a subdermally implanted system is presented, which acquires up to 8 EEG channels and performs seizure count with less than 3uW per channel. However, it is not usable in a closed loop system due to the high false positive rate of the extremely simple algorithm adopted. Since the computational requirements for these algorithms are challenging and the complexity scales up with the number of sensors, the design of a scalable digital architecture must target energy efficiency for a wide range of workloads.

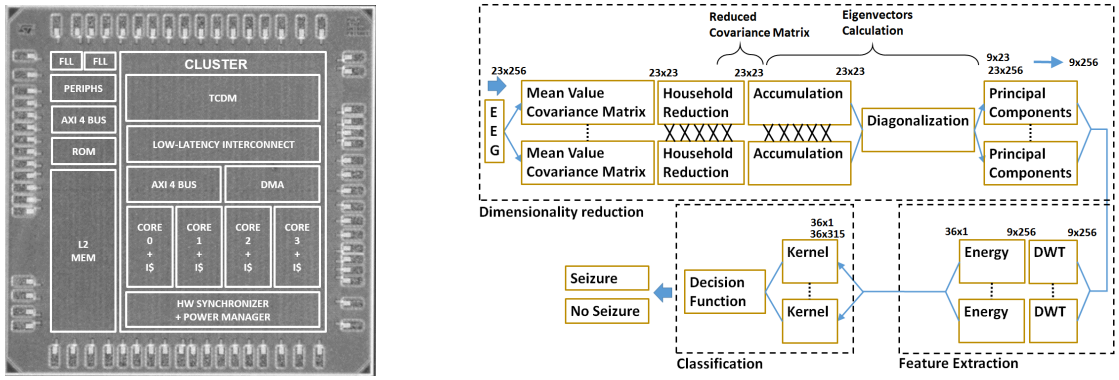
The open challenge we address in this work is the design of an efficient programmable framework for seizure detection, based on the combination of parallel processing and near threshold computing on Parallel Ultra Low Power (PULP) platform [12], a scalable and energy efficient multi-core architecture for sub-mW, deeply embedded applications. Taking data from an online EEG dataset [5], we show that PULP is able to compute a seizure detection on 23 electrodes in less than 5ms, improving state of the art of commercial systems by more than 5x if we compare the number of electrodes, and by 100x in terms of detection latency. Still, we compare the results with implementations on 32-bit ARM Cortex M4-based MCUs showing that, by virtue of our powerful and energy-efficient architecture we reduce the energy required to compute the algorithm by 15x to 80x, depending on the detection latency requirements. Furthermore, the proposed computational framework based on a fully software programmable multi-core architecture is highly scalable, versatile and can be used for a wide range of architectures and applications.

II. Materials and Methods

A. PULP platform

PULP is a programmable multi-core computing platform that exploits parallel, near-threshold operation and low-power 28nm FD-SOI technology to match computational requirements of near-sensors processing applications constrained by power budgets ranging from sub mW to few mW. The first implementation of the PULP architecture is described in [12], while the second is described in [13]. Figure 1(a) shows its die micrograph used for characterization of the power models used in this work. The third generation PULP architecture exploited in this work (PULPv3) is summarized in the following.

978-1-5090-2959-4/16/$31.00 2016 IEE



(a) (b)

Fig. 1. Layout of the PULP chip (a) and Seizure Detection Algorithm with computational kernels (b)

SoC architecture. The compute engine is a cluster with a parametric number of OpenRISC cores sharing 4kB of instruction cache. The core features optimizations and ISA extensions for energy efficient digital signal processing; in the context of this work we consider a configuration of the core augmented with floating-point units [14]. The L1 memory is composed of a 64kB, single cycle latency, multi-banked, Tightly Coupled Data Memory (TCDM) working as software-managed scratchpad memory. The TCDM features a *2 * number of cores* word-level interleaved banks connected to the processors through a non-blocking interconnect to reduce banking conflict probability. Off-cluster 256kB L2 memory access is managed by a tightly coupled DMA optimized for low power, connected to an AXI-4 interconnect enabling efficient data transfers. To provide high energy efficiency for a wide range of workloads, the cluster and the rest of the SoC are in different power and clock domains controlled by two Frequency-Locked Loops and external voltage regulators.

Programming model. A lightweight implementation of OpenMP 3.0 [15] has been tailored to PULPs explicitly managed, scratchpad-based memory hierarchy on top of a GCC 4.9 and LLVM 3.7 toolchains. To achieve high energy efficiency PULP includes special hardware for accelerating key software patterns such as barriers. Moreover, to reduce the power wasted by unused cores when worker threads are idling (e.g., in sequential regions of the program), PULP supports a clock-gating based thread docking scheme to reduce power of idle cores [12]. Control of power management knobs is fully integrated in the OpenMP runtime, hence completely transparent from the programmer viewpoint.

B. Seizure Detection On PULP

A block diagram of the seizure detection algorithm is shown in fig. 1(b), providing details of the whole processing chain and of its parallelization scheme.

Dimensionality Reduction. Using an orthogonal transformation, we convert the possibly correlated data acquired from p sensors into a set of linearly uncorrelated components (l). This transformation, widely used in neural processing [8], is named Principal Component Analysis (PCA), and represents an input dataset into a new coordinates system through the linear transformation:

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{P} \tag{1}$$

where $\mathbf{X}_{n \times p}$ is the input data matrix, $\mathbf{P}_{p \times l}$ is the transformation matrix and $\mathbf{Y}_{n \times l}$ is matrix of the reduced data. In this implementation, we reduce the dimensionality of the data from 23 to 9 components maintaining more than the 90% of the variance of the original input data.

As shown in Fig. 1(b), the PCA is composed of five main sub-kernels, featuring very different parallelism properties. *Mean Value and Covariance Matrix* can be efficiently parallelized on multiple cores, since OpenMP threads can process independently the 23 EEG input channels, producing one raw of the 23x23 covariance matrix each. *Householder Reduction and Accumulation* characterize a complex parallelization scheme where block level parallelism cannot be exploited. In these kernels the generation of each column of the reduced covariance matrix requires the output of the previous column to be processed. The same process is then replicated by rows, hence requiring additional synchronization barriers. Moreover, several OpenMP reductions (e.g. tree additions) are required to calculate scaling factors to compute intermediate matrixes in a bidiagonal form. *Diagonalization* is the most challenging kernel since the diagonal matrix is calculated in an iterative manner which is more efficient from the computational point of view. Moreover, the i column of the output matrix of this kernel is calculated from the $i-1$ column of the input matrix, and very small parallelism is available with the 23x23 processed matrix. The output of this kernel are eigenvectors and eigenvalues sorted with descending order in a diagonal matrix. *Principal Component* can be efficiently parallelized at data level, since it requires the multiplication of the 23x256 input EEG components matrix with the 9x23 eigenvector diagonal matrix connected to most significant eigenvalues, resulting in a 9x256 principal components matrix.

Feature Extraction. The *Discrete Wavelet Transform* (*DWT*) performs an efficient time-frequency analysis, providing information on the frequency content of a signal in the time domain. The signal is decomposed through a bank of low pass (LPF) and high pass (HPF) filters. The output of a level n decomposition results in a series of coefficients, named *Detail Coefficients* ($D_{(n)}$) for the HPF and *Approximation Coefficients* ($A_{(n)}$) for the LPF. We apply a 4 levels DWT on a 256 samples sliding window to obtain the D_{1-4}. Once the detail coefficients are extracted, we calculate their energy to retrieve the information related to the frequency content of each sub-band with the following equation:

$$\mathbf{E}_{D_{(n)}} = \sum_{i=0}^{k} \left| D_{(n)}[i] \right|^2 \tag{2}$$

where k is the length of the coefficient vector of level n. These kernels can be efficiently calculated since the structures of digital filters and summations can be divided in parallel threads and parallelized at block level. However, in this case maximum available parallelism is 9, since one OpenMP thread is created for each principal component.

Pattern Recognition. Among the algorithms used in EEG

TABLE I. EXECUTION OF SEIZURE DETECTION ON DEEPLY EMBEDDED COMPUTING PLATFORMS

	ARM Cortex M4		PULP 1 core	PULP 2 cores			PULP 4 cores			PULP 8 cores		
Kernel	kCycles	load %	kCycles	kCycles	Speedup[a]	Sleep[b]	kCycles	Speedup[a]	Sleep[b]	kCycles	Speedup[a]	Sleep[b]
PCA	2600	82	2072	1235	1,68	19,7	764	2,71	30,4	487	4,25	38,9
Mean+Cov.	1300	41	872	444	1,96	1,1	223	3,91	1,5	113	7,72	2,8
Householder Red.	272	9	193	131	1,47	16,2	87	2,22	24,7	62	3,11	31,9
Accumulate	133	4	96	57	1,68	10,3	34	2,82	16,9	22	4,36	26,1
Diagonalize	190	6	316	308	1,03	65,6	269	1,17	72,8	209	1,51	76,4
Compute PC	709	22	571	293	1,95	0,3	147	3,88	0,6	75	7,61	1,6
DWT+ENERGY	192	6	173	97	1,78	20,6	59	2,93	34,1	40	4,33	50,5
SVM	369	12	426	222	1,92	3,4	114	3,74	5,8	61	6,98	12,3
TOT	3165	100	2647	1552	1,71	18,2	933	2,84	28,5	582	4,55	38,6

[a] Speed-Up with respect to single-core PULP paltform.
[b] Mean value of sleep cycles of slave cores (%)

seizure detection, Support Vector Machine (SVM) is a supervised classifier widely used for its solid theoretical background that guarantees global minimum convergence with high computational efficiency [9]. The separation hyperplane between two classes of vectors is represented by a set of data vectors, named Support Vectors (SVs) which belong to the border between the classes [16]. The mean dimension of SVs matrix calculated for this setup is 36x315. In our application the input of the SVM classifier is a 36-dimensional vector calculated in Eq. (2) from the D_i coefficients. Having two possible classes, denoted as Cl_1 and Cl_2, the formula of the decision function to classify a new input instance is:

$$f(\mathbf{x}) = \sum_{i=1}^{N_{SV}} y_i \alpha_i K\langle \mathbf{x}, \mathbf{s}_i \rangle - \rho \quad \begin{cases} f(\mathbf{x}) > 0, \mathbf{x} \in Cl_1 \\ f(\mathbf{x}) < 0, \mathbf{x} \in Cl_2 \end{cases} \tag{3}$$

where $\mathbf{x}$ is the input features vector, $\mathbf{s}_i$ are the support vectors, α_i y_i are precalculated coefficients, ρ is a bias term and $K\langle \cdot, \cdot \rangle$ denotes the Radial Basis Function (RBF) kernel function, expressed by:

$$K\langle \mathbf{x}, \mathbf{s}_i \rangle = \exp\left(-\frac{||\mathbf{x} - \mathbf{s}_i||}{2\sigma^2} \right) \tag{4}$$

where σ is the variance. The parallelization of the RBF kernel function is highly efficient since each core can take over the computation of one of the 315 $K\langle\rangle$ values. The Decision Function is computed sequentially due to its small computational requirements, comparable to the cost of the OpenMP threads creation and termination, hence not suitable for parallelization.

III. EXPERIMENTAL RESULTS

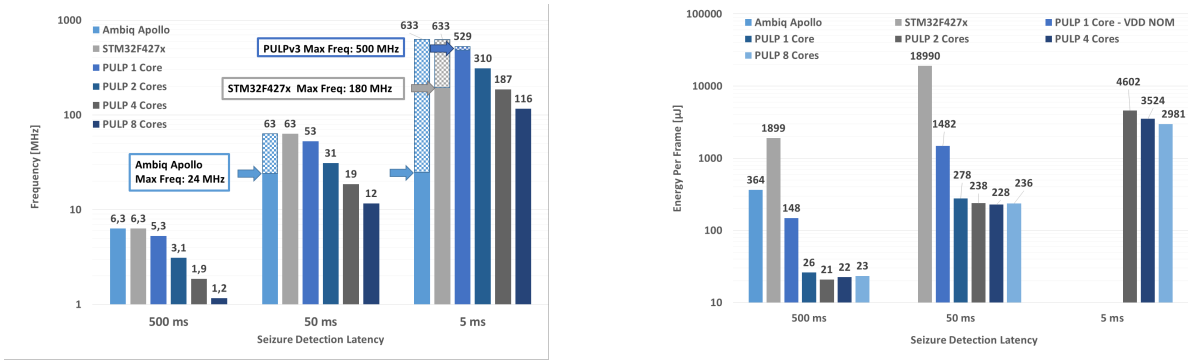
The EEG data are taken from the CHB-MIT data set, which collects samples collected from 23 pediatric subjects affected by intractable seizures. The electrodes are placed following the *International 10-20 System* and the EEG signals are acquired with 256Hz sampling frequency and 16-bit resolution. We took the EEG data from 8 patients randomly chosen among the traces that present seizures. To test the system we initially tuned up the processing chain and executed it on Matlab to verify the accuracy of the seizure recognition. The algorithm reaches 98.9% accuracy in the seizure detection with a *sensitivity* of 0.85. After this validation, the seizure detection algorithm was evaluated on PULP and two commercial off the shelf MCUs integrating an ARM Cortex M4 processor. We consider a high-end MCU (STM32F427) [17] and a low-end MCU (Ambiq Apollo) [18], that represent the two extremes in the market for high performance and low power, for Cortex M4 class architectures, both implemented in 90nm CMOS technology.

We tested our seizure detection implementation, imposing 3 latency constraints: 500ms, 50ms and 5ms. While the 5ms requirement is not necessary for clinical applications, it is useful to show performance of the system with different workloads. The evaluation was conducted executing the processing steps of the seizure detection application on a demo board, to analyze the performance of the two Cortex M4-based MCUs and on the instruction accurate simulator of the PULP platform with 1,2,4,8 cores extended with FPUs. The operating frequency and power consumption of the PULP platform at different voltage levels have been extracted from post-layout timing and power analysis of an instance of PULPv3 SoC, accurately calibrated with models and measurements performed on the first silicon prototype of PULP [12], and finally adapted to the configurations adopted. A 4-cores PULP cluster with FPUs achieves 500MHz at 1V and 112μW/MHz and 50MHz at 0.5V and 24μW/MHz. For fair comparison we only consider the power of the processors for the MCUs and the power of the cluster for PULP, excluding the power consumption of peripheral subsystem which would be similar, and negligible with respect to the digital processing power.

A. Evaluation of Performance and Energy Metrics

Table I summarizes the execution time (clock cycles) of the seizure detection application on the reference platforms. *PCA* requires 82% of the overall computation time on the Cortex M4, while *DWT*, *Energy* and *SVM* contribute to the remaining computational load (18%). When executing the algorithm exploiting parallel processing over multiple cores of the PULP platform, the execution time reduces by up to 4.55x with 8 cores. It can be noted that for the kernels with high parallelism, like *Mean value + Covariance*, *Compute PC* and *SVM*, that account for 75% of the overall computational load during sequential execution, the speed-up is nearly ideal. *Householder Reduction* and *Accumulate* require parallel computations on small chunks of data and several synchronization points, which increase the overhead of the OpenMP runtime. *Diagonalize* is an iterative kernel affected by pathological Amdahl bottleneck caused by the dependencies between matrix elements calculated during the iterations, which force most of this kernel to be executed sequentially. For this reason, we see in table I the high percentage of sleep cycles of the slave cores of this kernel. Finally, even though *DWT + Energy* kernel is highly parallelizable, it is affected by workload unbalance, since it requires the elaboration of 9 PC components over 8 cores, limiting the overall speedup of this kernel to 4.33x and leading to a highly increasing ratio of sleep cycles of the slave cores when increasing parallelism.

Fig. 2(a) shows the operating frequency required to run the seizure detection on a data frame (23 channels x 256 samples) within a given latency (500ms, 50ms and 5ms). The graph highlights that with the limited capabilities of Ambiq Apollo, the constraints can be satisfied only with 500ms frame period, while STM32F427 can also satisfy the constraints



(a) (b)

Fig. 2. Frequency (a) and Energy (b) required to execute seizure detection algorithm on Cortex M4 and PULP platform for different frame periods.

for 50 ms latency. By virtue of its scalable architecture, the PULP platform can satisfy the requirements with all constraints, being able to consume an average power of 47μW, 473μW, and 5.9mW for 500ms, 50ms, and 5ms latency. It can be noted that increasing the number of cores the operating frequency required to achieve the real-time constraints on PULP decreases, allowing to reduce the supply voltage as well. This leads to significant improvement in power density, thanks to the quadratic dependency of dynamic power with supply voltage. This highlights a trade-off between the parallelization efficiency, which decreases with the number of cores, and voltage scaling.

This scenario is highlighted in Fig. 2(b), which also shows the comparison with off-the shelf MCUs, that operates at nominal voltage supply of 1.8V and 2.5V. The difference in energy between the MCUs and the single-core PULP platform at nominal supply voltage is mainly given by technology gap, different implementation strategy and architectural complexity, and leads to 12x to 2.5x lower energy. More interesting is the exploitation of parallel near threshold computing on the PULP platform, which leads to a further improvement of 6x with respect to sequential processing, and to an improvement of 72x and 15x in terms of energy consumption with respect to commercial MCUs. Energy efficiency is further increased thanks to the power management techniques applied to idle cores during sequential execution or barriers, leading to an average energy reduction of 5%, 12%, 21%, when executing on 2, 4, and 8 cores, respectively. From an application perspective, these results show that the optimization of the parallel processing tailored for a highly efficient HW/SW platform allows to scale up the complexity of the system (eg. the number of channels), without losing the real-time requirements. This combined approach can dramatically extend the battery life of a closed loop neural stimulation system targeting also a power budget compatible with implantable energy harvesters [19].

IV. Conclusion and Future Work

The proposed work shows the strong impact of the PULP architecture in the design of a real time embedded system for neural processing. The combination of the near threshold operation with the parallel multi-core architecture of PULP outperforms commercial solution by 10-100 times in terms of performance and up to 80 times in terms of energy efficiency. Moreover, as opposed to ASIC solutions, the proposed platform maintains the flexibility typical of programmable processors suitable to implement a versatile and scalable neural processing framework. Future works target the tuning of the PULP architecture with dedicated HW optimization in the ISA design for low power signal processing and also more aggressive algorithmic strategies to improve the parallel speedup in neural computing algorithms and the energy efficiency of next generation neural computing systems.

Acknowledgment

This work has been partially supported by the FP7 ERC Advance project MULTITHERMAN (g.a. 291125) and by the SNF project MicroLearn: Micropower Deep Learning

References

[1] F. A. Al-Otaibi *et al.*, "Neuromodulation in epilepsy," *Neurosurgery*, vol. 69, no. 4, pp. 957–979, 2011.

[2] F. T. t. Sun, "Closed-loop neurostimulation: the clinical experience," *Neurotherapeutics*, vol. 11, no. 3, pp. 553–563, 2014.

[3] H. T. Ocbagabir *et al.*, "Efficient eeg analysis for seizure monitoring in epileptic patients," in *LISAT, IEEE*, 2013.

[4] Neuropace, http://www.neuropace.com/, 2011.

[5] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[6] M. A. Bin Altaf *et al.*, "A 16-channel patient-specific seizure onset and termination detection soc with impedance-adaptive transcranial electrical stimulator," *IEEE Journal of Solid-State Circuits*, Nov 2015.

[7] P. Afshar *et al.*, "A translational platform for prototyping closed-loop neuromodulation systems," *Closing the Loop Around Neural Systems*, p. 367, 2014.

[8] T.N. Alotaiby *et al.*, "Eeg seizure detection and prediction algorithms: a survey," *EURASIP Journal on Advances in Signal Processing*, 2014.

[9] Y. Liu *et al.*, "Automatic seizure detection using wavelet transform and svm in long-term intracranial eeg," *TNSRE, IEEE*, vol. 20, no. 6, 2012.

[10] M. T. Salam, M. Sawan, and D. K. Nguyen, "A novel low-power-implantable epileptic seizure-onset detector," *IEEE TBCAS*, 2011.

[11] B. Do Valle *et al.*, "Low-power, 8-channel eeg recorder and seizure detector asic for a subdermal implantable system," *IEEE TBCAS*, 2016.

[12] D. Rossi *et al.*, "A 60 GOPS/W, -1.8V to 0.9V body bias ULP cluster in 28nm UTBB FD-SOI technology," *Solid-State Electronics*, vol. 117, pp. 170–184, Mar. 2016.

[13] D. Rossi *et al.*, "193 MOPS/mW 162 MOPS, 0.32V to 1.15V Voltage Range Multi-Core Accelerator for Energy-Efficient Parallel and Sequential Digital Processing," in *Cool Chips*, 2016.

[14] M. Gautschi *et al.*, "4.6 a 65nm cmos 6.4-to-29.2 pj/flop 0.8v shared logarithmic floating point unit for acceleration of nonlinear function kernels in a tightly coupled processor cluster," in *ISSCC 2016*.

[15] OpenMP Programming Model, http://openmp.org, 2016.

[16] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992.

[17] stm32f427vg, http://www.st.com/resource/en/datasheet/stm32f427vg.pdf, 2016.

[18] Ambiq Apollo, http://ambiqmicro.com/system/files/Apollo_MCU_Data_SheetDS0010V0p45.pdf, 2016.

[19] B. Rapoport *et al.*, "A glucose fuel cell for implantable brain–machine interfaces," *PloS one*, 2012.