

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A sub-10mW real-Time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Benatti, S., Rovere, G., Bossler, J., Montagna, F., Farella, E., Glaser, F., et al. (2017). A sub-10mW real-Time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC. Institute of Electrical and Electronics Engineers Inc. [10.1109/IWASI.2017.7974234].

Availability:

This version is available at: <https://hdl.handle.net/11585/609280> since: 2018-09-05

Published:

DOI: <http://doi.org/10.1109/IWASI.2017.7974234>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the post peer-review accepted manuscript of:

S. Benatti, G. Rovere, J. Bossert, F. Montagna, E. Farella, F. Glaser, P. Schonle, T. Burgert, S. Fateh, Q. Huang and L. Benini, "A sub-10mW real-time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC," *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, Vieste, 2017, pp. 139-144. doi: 10.1109/IWASI.2017.7974234

The published version is available online at: <https://doi.org/10.1109/IWASI.2017.7974234>

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

A sub-10mW Real-Time Implementation for EMG Hand Gesture Recognition based on a Multi-Core Biomedical SoC

Simone Benatti*, Giovanni Rovere†, Jonathan Bösser†, Fabio Montagna*, Elisabetta Farella‡, Florian Glaser†, Philipp Schönle†, Thomas Burger†, Schekeb Fateh†, Qiuting Huang† and Luca Benini* †

*DEI, University of Bologna, Italy. Email: {simone.benatti, fabio.montagna, luca.benini}@unibo.it

†Integrated System Laboratory, ETH Zurich, 8092 Zurich, Switzerland.

Email: {rovereg, jboesser, glaserf, schoenle, burger, fateh, huang, lbenini}@iis.ee.ethz.ch

‡ICT Center - FBK, Trento, Italy Email: efarella@fbk.eu

Abstract—Real-time biosignal classification in power-constrained embedded applications is a key step in designing portable e-health devices requiring hardware integration along with concurrent signal processing. This paper presents an application based on a novel biomedical System-On-Chip (SoC) for signal acquisition and processing combining a homogeneous multi-core cluster with a versatile bio-potential front-end. The presented implementation acquires raw EMG signals from 3 passive gel-electrodes and classifies 3 hand gestures using a Support Vector Machine (SVM) pattern recognition algorithm. Performance matches state-of-the-art high-end systems both in terms of recognition accuracy ($> 85\%$) and of real-time execution (gesture recognition time $\ll 300\text{ms}$). The power consumption of the employed biomedical SoC is below 10mW , outperforming implementations on commercial MCUs by a factor of 10, ensuring a battery life of up to 160 hours with a common Li-ion 1600mAh battery.

I. INTRODUCTION

The fast growth of miniaturized and efficient electronics enables the development of unobtrusive and portable personal health care systems. Electrode-based biosignals like ECG, EEG, EMG (ExG) resulting from the underlying physiological body activity allow to infer biomedical parameters linked to the subject health status. ExG application ranges from consumer electronics for fitness and wearable applications to medical-grade devices enabling patient monitoring and rehabilitation. For instance, ECG monitoring is employed in smart-watches for fitness heartbeat monitoring [1], pacemakers [2] and holter devices [3]. EEG based systems are used in medical applications for the treatment of neurological disorders, like Parkinsons [4], epilepsy [5], spinal cord injuries [6], [7], [8] as well as attention loss, drowsiness [9] and autism detection [10]. Implantable intramuscular EMG recording devices have been explored in research [11] but a significant part of EMG applications are devoted to wearable surface EMG devices thanks to their unobtrusiveness, e.g., controllers of the upper limb prostheses [12] and hand gesture recognition systems intended for consumer Human-Machine interaction [13]. The surface EMG signals are the superposition of the electrical activity of underneath muscles when contractions occur [14]. Typical amplitude ranges from $\pm 10\mu\text{V}$ to $\pm 10\text{mV}$ with a

maximum bandwidth of 2kHz depending on the size of the contracting muscle, on the distance between fibers and the electrodes and on the properties of the latter.

EMG signals can be processed to recognize the user's intended gestures by analyzing muscular activation patterns. Due to the intrinsic variability of biosignals, machine learning approach is mandatory to reach high recognition accuracy. However this requires substantial computational capabilities that only became suitable for wearable platforms in the last years. In fact, early studies on EMG gesture recognition were performed offline on benchtop platforms [15], while the remarkable advances in digital low-power design and in efficient computational architectures [16], [17] allowed to implement real-time systems that can execute algorithms like advanced filtering [18], dimensionality reduction [17] and pattern recognition [19]. Nevertheless, the design of an efficient real-time system for low-power acquisition and EMG processing is still an open challenge that requires a multilevel approach, ranging from the design of the Analog Front-End (AFE) and of an efficient digital processor to the system level architecture and the algorithms. The development of various single purpose and general purpose biomedical ICs have been reported in literature, however none of them cover the requirements of the presented application in terms of number of channels or processing power.

In this paper, we propose the implementation of real-

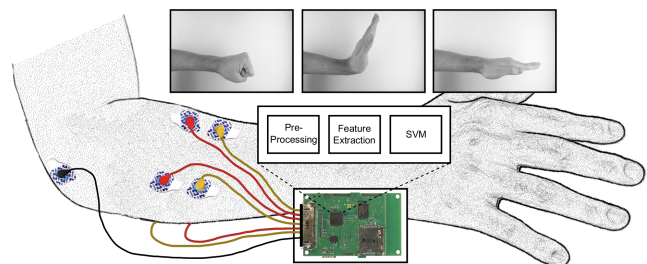


Fig. 1. Architecture of the implemented hand gesture recognition based on EMG signal acquisition, processing and classification.

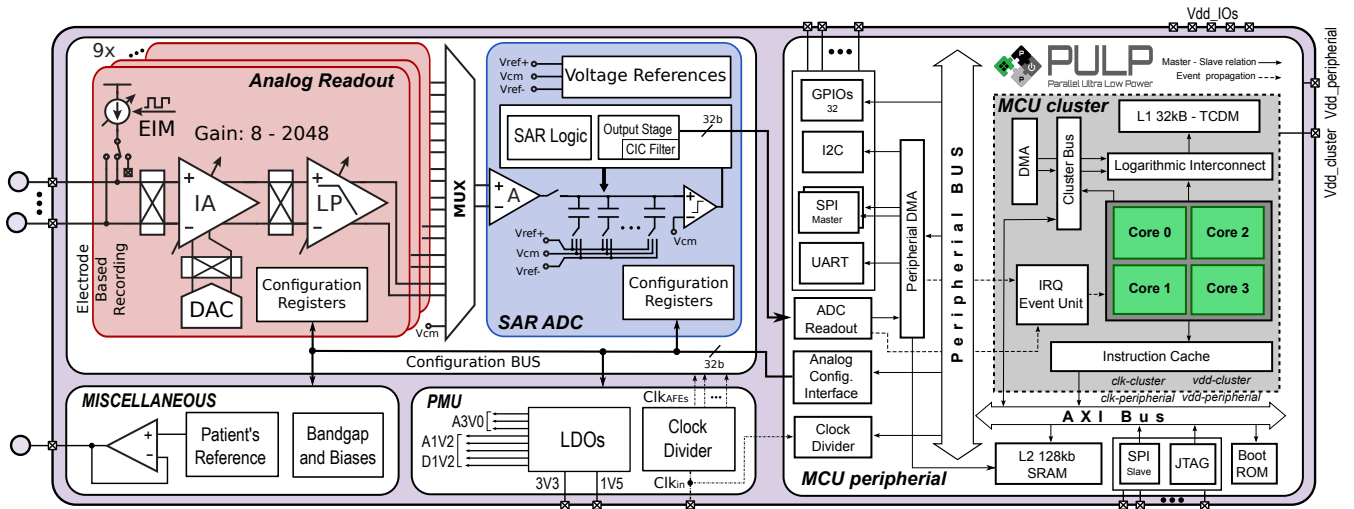


Fig. 2. Block diagram of the biomedical SoC with emphasis on the EMG signal acquisition (Analog Readout and SAR ADC) and processing (MCU).

time acquisition, processing and classification of a 3-channel EMG signal based on a biomedical SoC [20] platform. It supports up to 9 passive electrode-based medical-grade ExG channels, each configurable to trade-off unnecessary precision and bandwidth for power. Furthermore it integrates a homogeneous multi-core digital processor where energy efficiency is increased by voltage and frequency scaling while computation power can be recovered by parallel computation. This solution provides all the hardware required for the hand gesture recognition on a single chip, while maintaining the versatility of a general purpose programmable multi-core platform. Our SoC-based platform executes the EMG classification in $560 \mu s$, significantly lower than the real-time requirement of $300 ms$ for upper limb prosthetic controllers [21], reaching also a performance gain of more than 10x in terms of power consumption w.r.t. current commercial ST Microelectronic Cortex M4 based solutions [21].

II. HARDWARE IMPLEMENTATION

A. The Biomedical SoC Architecture

The biomedical SoC used for this experiment is based on the design developed at ETH Zurich [20]. The chip includes a quad-core microcontroller (MCU) equipped with 9 electrode-based analog channels for ExG signal amplification with Electrode Impedance Measurements (EIM) for active lead-off detection. The parallel MCU thus provides an efficient mean to real-time process sparse biomedical signals on-chip in a concurrent fashion. The time-multiplexed ADC allows for maximum hardware reuse of constantly-biased analog circuits while using the Successive Approximation Register (SAR) architecture allows for faster analog channel switching and guarantees precise synchronization among them. On top of that $1.5 V$ and $3.3 V$ capacitor-free Low-DropOut regulators (LDOs) [22] decouple sensitive analog supplies and allow to power-gate unused (not shown in Fig. 2) analog circuits. A flexible clock-division and -distribution scheme allows to

independently tune clock signals to lower gate switching dynamic power.

1) *MCU*: The employed MCU is an implementation derived from the PULP (Parallel-Ultra-Low-Power) Platform [23], [24]. It comes with 4 general-purpose *openRISC* cores sharing the level 1 (L1) memory - tightly coupled data memory (TCDM). L1 data can be accessed by all the cores within a single cycle through a logarithmic interconnect, not requiring explicit copies of data for data exchange between cores. Unused cores are set to clock-gated idle mode to save dynamic power while if all the cores are in idle mode the whole cluster region is clock-gated as well. Leakage currents ($25 \mu W$) are eliminated by power-gating the cluster region if in idle for more than $40 ms$. Besides standard communication interfaces (JTAG, UART, SPI, I2C, GPIO) the MCU peripheral region is equipped with an ADC readout block as point-of-entry for incoming digitized signals. The raw data stream is usually buffered directly in the L2 memory, routed via a Peripheral Direct Memory Access (PDMA) without requiring core intervention. Once buffers are filled cluster and/or cores are waken up through interrupt requests handled by the event-unit. To optimize power savings the MCU cluster and the MCU peripherals belong to different domains operating at different voltages and clock frequencies. Reliable signal domain crossing is guaranteed by the insertion of levelshifters and FIFO synchronizers. At startup a boot-ROM allows to load program code from external flash memory to the L2 via SPI while applications can have memory-mapped access to all the MCU peripherals and the analog configuration registers.

2) *Analog Readout*: The low-power ($150 \mu W$) Analog Readout (AR) consist of a low-noise ($IR = 1.09 \mu V$, $BW = 150 Hz$) Instrumentation Amplifier (IA) followed by a Low Pass Filter (LPF) to reduce off-band components and to prevent aliasing effect due to the subsequent time-sampling. Both the AR gain and cut-off frequencies are selectable over a wide range to cover many applications while chopping modulation

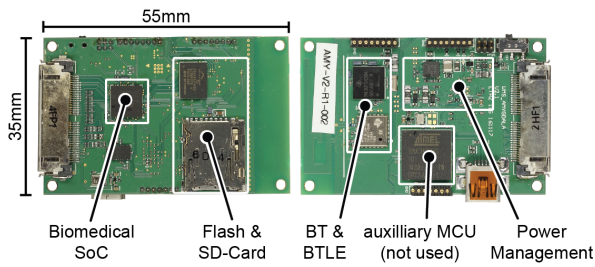


Fig. 3. System platform used for the implemented hand gesture recognition application.

of the analog signal is required to avoid in-band flicker-noise ($1/f$). The EIM for lead-off detection can be activated by injecting high frequency currents into the electrodes and thus measuring the electrode voltage drop, inversely proportional to the quality of the electrode-skin interface. Field measurements with actual human-connected electrodes often experience a large ($\pm 300\text{ mV}$) wandering baseline at the AR input, causing the IA to saturate. AC-coupling is not a viable solution due to large required capacitors. Our approach exploits a current-steering DAC to compensate for the large offsets with a digitally controlled servo-loop activated once IA analog saturation is detected.

3) *SAR-ADC*: In this fully differential SAR-ADC design [25], the sampling/settling time and the bias current of the analog buffers are programmed for each of the successive-approximation steps. A majority voting mechanism after the SAR-ADC dynamic comparator is employed to reduce thermal noise floor while Cascaded Integrated Comb (CIC) filters further lower it while performing downsampling. A calibration algorithm compensates for the capacitor-array mismatches, boosting the overall ADC resolution by 1.8 Effective Number of Bits (ENOB) with minor power penalty. SAR-ADC high configurability allows for power savings by trading-off power for speed and precision to just cover the application requirements. The ADC sampling frequency (f_s up to 286 kHz) and precision (ENOB up to 13.5 bits) can be thus finely controlled on-the-fly by the MCU over an asynchronous low-latency (3 clock cycles) 32-bit configuration bus.

B. System Platform

Fig. 3 shows an image of the used system platform. The board is based on a 8-layer PCB and includes the biomedical SoC with the integrated 9 channel AFE, a Bluetooth (BT/BTLE) link, consisting of a MSP430 running the BT protocol stack and a RF-transceiver for the communication with a host device. Furthermore there are a non-volatile flash memory containing the application of the biomedical SoC and a SD-Card to enable dumping data for debugging reasons and/or further processing. The board can be operated either battery powered with a supply between 3.7 and 4.2 V or by an external 5 V power supply. The on-board power management provide stable output voltagess for the biomedical SoC (0.9 V, 1.5 V and 3.3 V). The board configuration specific for this application allows to acquire 3 differential EMG channels

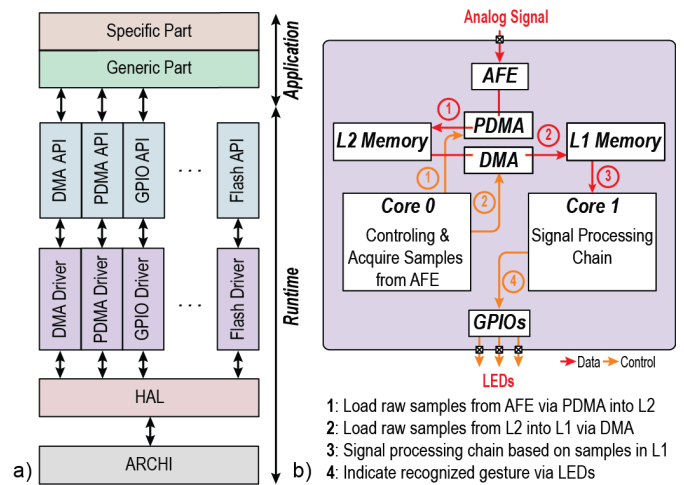


Fig. 4. (a) Firmware framework, including a runtime consisting of code providing simplified access to the underlying hardware, and the application with a generic and a specific part and (b) programming model indicating the core assignments for the tasks of the application.

including off-chip RC band-pass filters with lower and upper cutoff frequencies equal to 0.3 Hz and 12 MHz respectively, to minimize the high frequency electrical noise and the baseline wander of the EMG signal. An additional electrode is used to drive the otherwise floating patient common mode voltage at 0.9 V .

III. BIOMEDICAL SOC FIRMWARE

The presented platform is designed to cover a variety of use-cases with different requirements and applications altogether. Nevertheless, a considerable part of the firmware will be very similar throughout all applications – e.g. interfacing internal SoC peripherals – and can therefore be shared. This leads to a firmware framework as shown in Fig. 4 a), composed of hardware related code called runtime, a generic application part shared between applications, and a specific part comprising of the application specific algorithms. The runtime provides facilitated access to the various hardware blocks of the biomedical SoC – e.g. employing the DMA or changing the gain of an AFE channel, including firmware layers directly interfacing the biomedical SoC hardware – the Architectural layer (ARCHI) and the Hardware Abstraction Layer (HAL) – as well as the drivers and their respective Application Programming Interfaces (APIs), which define simplified interfaces to hardware blocks such as the GPIOs or the various serial interfaces. By using the runtime, the user can focus on the application specific parts of the firmware and access optimized drivers through low-complexity APIs tailored to the user’s needs. The generic part of the application is in charge of setting up the SoC according to the application’s needs and configuring and operating the interfaces to the SoC peripherals by using the aforementioned APIs. This part is mainly deployed on one of the cluster’s cores and there may be no significant difference from one application to another. The specific part represents the core functionality

of the application with its bespoke algorithms and may be distributed on multiple cores. This enables parallel as well as concurrent computation and allows for highly optimized algorithms regarding core frequency and therefore power consumption.

The programming model, i.e. how the parts of the firmware are deployed among the cores, is implemented as follows: the generic and the specific part of the application are strictly separated by employing them on different cores. The presented hand gesture recognition application makes use of 2 out of the 4 cores of the cluster, shown in Fig. 4 b). The cores not used by the application are clock-gated and available for further computation. Otherwise each core starts up initializing the resources needed for the assigned tasks and stops, i.e., is clock-gated, at a synchronization barrier. Once every core reached its respective barrier, the clock is enabled and the application execution resumes. The first core of the cluster – Core 0 – is assigned the specific part of the firmware that configures the SoC’s processing unit as well as the AFE accordingly. Afterwards, the core configures the PDMA to transfer the acquired samples from the AFE to the L2 memory and the DMA to copy them into the L1 memory where they are distributed to separate buffers for each channel. Concurrently, the second core employed by the application is processing the acquired and buffered samples according to the algorithms described in Section IV.

IV. HAND GESTURE RECOGNITION APPLICATION

The block diagram depicted in Fig. 5 shows the computational steps of the biomedical signal processing chain, consisting of 4 main kernels, i.e. Notch Filter, Offset Compensation, Envelope Extraction and SVM.

1) *Notch filter*: The processing chain starts with a IIR Notch filter that aims to eliminate the interference caused by the AC frequency of the powerlines, which represents one of the most significant noise sources in biosignal processing. In this application, a third order Notch filter is implemented, with a Q-factor equal to 50. The filter coefficients are calculated off-line and allocated in the L1 memory as constants.

2) *Offset Compensation*: The input signals can be affected by a voltage offset caused by motion artifacts of the subject. This offset must be periodically evaluated and removed: hence, on each EMG channel a moving average is applied, by implementing a circular buffer to keep track of the last n samples. After an evaluation of the trade-off between performance and memory requirements, the length of the averaging window is fixed to $n = 60$. The output of the *Offset Compensation* kernel is the absolute value of the sample minus the mean value of the sliding window.

3) *Envelope Extraction*: Once the signal is offset-compensated we extract its temporal envelope by calculating the Root Mean Square (RMS). The RMS is computed on the last $n = 60$ values with a 1-sample sliding window. Thus, the output of this kernel is the feature vector, which is the input of the SVM algorithm.

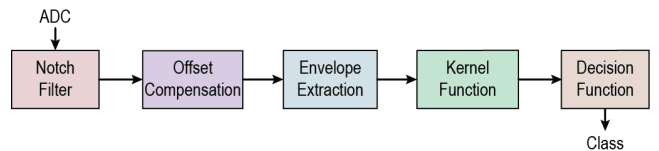


Fig. 5. Signal processing chain of computational kernels for EMG hand gesture recognition application.

4) *Support Vector Machine*: Among pattern recognition algorithms, SVM has the main advantages of being theoretically robust and efficiently implementable [26]. SVM is a supervised learning algorithm belonging to the framework of statistical learning and its goal is to find the optimal separation hyperplane between 2 classes of vectors. This decision boundary, defined during the training stage, is obtained by a subset of vectors of the input space, named Support Vectors (SVs). The length of the model is variable and depends on multiple factors such as the nature of the data, the quality of the training set and the pre-processing capabilities. Data used to test the performance of the application led to a model composed by 31 SVs. In this application, all the SVs can be contained in the fast-access L1 memory of the chip but for larger models it is possible to load the vectors from larger L2 memory using the DMA and double buffering.

V. APPLICATION RESULTS

The performance of the proposed system was compared against the previously presented solution [12] in terms of real-time algorithm execution and of power consumption. One right-handed healthy subject with no history of neurological or psychiatric disorders participated in the experiment. The 3 fully differential EMG channels are placed on the forearm in correspondence of the *flexor radialis carpi*, *flexor ulnaris carpi* and *extensor communis digitorum*. Each EMG channel is sampled at 1 kS/s , which has been shown to be suitable for gesture recognition applications [27]. During the training phase, the subject executed the gestures holding the contractions for 3 seconds and separating each gesture with a 3 second muscular relaxation. Each gesture is repeated 5 times and the collected data is streamed to a PC platform where they are acquired and segmented to calculate the Support Vectors (SVs) of the SVM model. The set of the gestures includes the power grasp (closed punch), the open hand and the rest position. They correspond to the 3 classes recognized by the SVM algorithm. On the PC side, the classification algorithm is the libSVM [28], a multiplatform implementation of the SVM, while on the embedded platform we adapted the code to enable the classification with fixed-point arithmetic. By virtue of the SVM multi-platform implementation, the accuracy of the gesture recognition is evaluated offline, reaching 88% of correct gesture classifications and showing results comparable with our previous work [27]. We also made a comparison between the fixed- and floating-point implementations that shows same performance in terms of accuracy. The real-time execution of the application on a time-line is shown in Fig. 6, whereby overall core activity a) and respective activity of

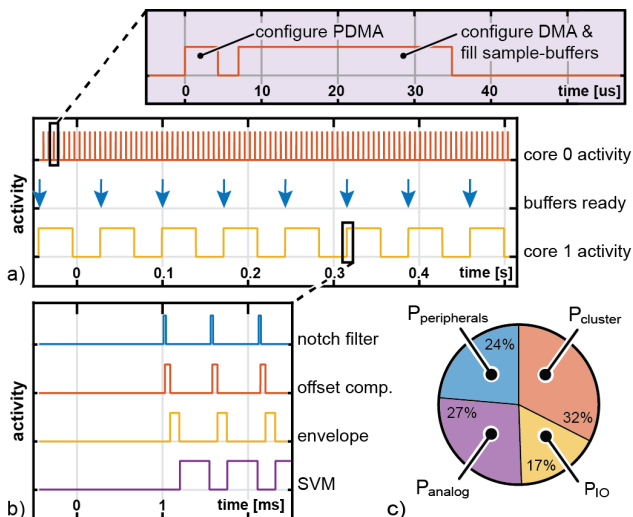


Fig. 6. Execution of the application tasks depicted on a timeline showing continuous acquiring and processing of samples, depicting (a) overall core activity, (b) time distribution among the 4 kernels and (c) power contributions of the analog and digital parts: P_{analog} refers to the analog circuitry, $P_{peripherals}$ and P_{IO} are related to the chip peripherals and to the digital Input-output respectively and $P_{cluster}$ represents the percentage of the total power consumption used by multi-core.

the 4 kernels b) is depicted, given $f_{MCU} = 48 MHz$. With $f_s = 1 kHz$ the time-budget to process a buffer containing 72 samples of each channel is $72 ms$. It can be seen that this timing requirement is easily met as the processing time of one buffer does not exceed $41 ms$. The execution times of the four kernels per one sample of each channel results in: notch filter $< 31 \mu s$, offset comp. $< 64 \mu s$, envelope $< 115 \mu s$ and SVM $< 350 \mu s$. The number of cycles of the kernels are shown in Table I.

Considering $300 ms$ [12] as the threshold for the real-time gesture recognition of a hand controller, it is noticeable that the proposed system is well within this requirement. Power consumption is another key point in designing an efficient wearable system. Fig. 6 (c) shows the power breakdown of the analog and digital circuitry of the SoC. The system runs with an external clock of $48 MHz$ provided by the control MCU of the board, the power consumption of the biomedical SoC digital part results from $P_{peripherals} + P_{IO} + P_{cluster}$. The peripherals and the cluster are supplied with $0.9 V$, consuming $2.47 mA$ and $3.41 mA$ respectively, with a total power consumption of $5.29 mW$. The IOs are supplied by $3.3 V$ and they drain a current of $486 \mu A$, leading to a power consumption of $1.6 mW$. The analog circuitry P_{analog} is supplied by 1.5 and $3.3 V$ with a total power consumption of $2.54 mW$. The measured power of the biomedical SoC results in $9.43 mW$. The proposed system gains more than 10x in terms of energy efficiency w.r.t. the previous work [21], where the current consumption of the Cortex M4 is $40 mA$ ($238 \mu A/MHz$) at $2.5 V$ for the processing part with the microcontroller clocked at $168 MHz$. In addition, the power consumption of the external AFE is $15 mA$ at $1.2 V$, resulting in an overall power consumption of $118 mW$.

TABLE I
EXECUTION OF GESTURE RECOGNITION APPLICATION ON DEEPLY EMBEDDED COMPUTING PLATFORMS

Kernel (sub-kernels)	Cycles	L2 Mem. [byte]	L1 Mem. [byte]
Pre-Processing			
Notch filter	1300	–	60
Offset Compensation	2900	–	744
Envelope Extraction	5500	558 ^b	756
Total Pre-Processing	9800 ^a	558	1590
SVM	16400	558 ^b	12
Total	26200	558	1602

^a considering execution of pre-processing kernels on 3 acquisition ch. ^b shared by multiple kernels

VI. CONCLUSION

A real-time implementation for the acquisition, processing and classification of EMG signals implemented on a wearable system is presented. A novel biomedical SoC combining a versatile 14-bit ADC and a multi-core digital processor based on the OpenRISC architecture is introduced as main component of this system along with the signal pre-processing and a fixed-point implementation of the libSVM algorithm for hand gesture recognition based on 3 EMG channels sampled at $1 kHz$. The proposed implementation is proven to be appropriate for a biomedical application and to comply with both the power and timing constraints implied by such a system. The duration for processing one sample by the entire processing chain $< 560 \mu s$ fully satisfies the real-time constraint and the overall power consumption of $9.43 mW$ including cluster, peripherals and IOs outperforms the real-time implementation on a current commercial SOA cortex M4 based platform by a factor of 10, leading to a significant advantage regarding battery lifetime. Given the system versatility and proven energy efficiency further biomedical applications can be quickly developed making use of the same platform and firmware runtime.

ACKNOWLEDGMENT

The authors would like to thank Xu Han, Noé Brun and Tom Kleier for their assistance during the ASIC and board design. This project has been partially funded by ACP AG, Nano-Tera.ch RTD Project *WearMeSoC* and 157048 grant *Transient Computing Systems* which are financed by the Swiss Confederation and scientifically evaluated by SNSF. This work was also funded by the ERC MultiTherman project (ERC-AdG-291125), and by the OPRECOMP project (grant 732631).

REFERENCES

- [1] SmartCardia, <http://www.smartcardia.com/inyu/>, 2013.
- [2] P. S. G. V. Parsonnet and N. P. Smyth, "Implantable cardiac pacemakers: Status report and resource guideline," *The American journal of cardiology*, vol. 34, no. 4, pp. 487–500, 1974.
- [3] Medtronic REVEAL, <http://www.medtronicdiagnostics.com/>, 2011.
- [4] Medtronic DBS, <http://www.medtronicdbs.com/>, 2011.
- [5] Neuropace, <http://www.neuropace.com/>, 2011.
- [6] M. Capogrosso et al., "A brain–spine interface alleviating gait deficits after spinal cord injury in primates," *Nature*, vol. 539, no. 7628, pp. 284–288, 2016.

- [7] A. Moin et al., "Powering and communication for omni: A distributed and modular closed-loop neuromodulation device," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.
- [8] F. T. Sun et al., "Closed-loop neurostimulation: the clinical experience," *Neurotherapeutics*, vol. 11, no. 3, pp. 553–563, 2014.
- [9] V. Kartsch, S. Benatti, D. Rossi, and L. Benini, "A wearable EEG-based drowsiness detection system with blink duration and alpha waves analysis," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2017.
- [10] A. B. Brandwein et al., "Neurophysiological indices of atypical auditory processing and multisensory integration are associated with symptom severity in autism," *Journal of autism and developmental disorders*, vol. 45, no. 1, pp. 230–244, 2015.
- [11] D. Seo et al., "Wireless recording in the peripheral nervous system with ultrasonic neural dust," *Neuron*, vol. 91, no. 3, pp. 529–539, 2016.
- [12] S. Benatti, B. Milosevic, E. Farella, E. Gruppioni, and L. Benini, "A prosthetic hand body area controller based on efficient pattern recognition control strategies," *Sensors*, vol. 17, no. 4, p. 869, 2017.
- [13] Thalmic lab MYO, <https://www.thalmic.com/>, 2011.
- [14] J. V. Basmajian and C. De Luca, "Muscles alive," *Muscles alive: their functions revealed by electromyography*, vol. 278, p. 126, 1985.
- [15] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, "Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay," *IEEE TNSRE*, 2011.
- [16] J. Constantin et al., "TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing," in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, Oct 2012, pp. 159–164.
- [17] S. Benatti, F. Montagna, D. Rossi, and L. Benini, "Scalable EEG seizure detection on an ultra low power multi-core architecture," in *Biomedical Circuits and Systems Conference (BioCAS), 2016 IEEE*. IEEE, 2016, pp. 86–89.
- [18] M. Tomasini, S. Benatti, B. Milosevic, E. Farella, and L. Benini, "Power line interference removal for high-quality continuous biosignal monitoring with low-power wearable devices," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3887–3895, May 2016.
- [19] 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.
- [20] P. Schönle et al., "A multi-sensor and parallel processing SoC for wearable and implantable telemetry systems," *Proc. IEEE European Solid-State Circuits Conf. (ESSCIRC)*, 2017.
- [21] S. Benatti, F. Casamassima, B. Milosevic, E. Farella, P. Schnle, S. Fateh, T. Burger, Q. Huang, and L. Benini, "A versatile embedded platform for emg acquisition and gesture recognition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 5, pp. 620–630, Oct 2015.
- [22] X. Han, T. Burger, and Q. Huang, "An output-capacitor-free adaptively biased LDO regulator with robust frequency compensation in 0.13 μ m CMOS for SoC application," *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 2699–2702, May. 2016.
- [23] F. Conti et al., "An IoT endpoint System-on-Chip for secure and energy-efficient near-sensor analytics," *IEEE Trans. Circuits Syst. – I: Reg. Papers*, to be published.
- [24] M. Gautschi et al., "Tailoring instruction-set extensions for an ultra-low power tightly-coupled cluster of openrisc cores," in *2015 IFIP/IEEE VLSI-SoC*, Oct 2015, pp. 25–30.
- [25] S. Fateh et al., "A reconfigurable 5-to-14 bit SAR ADC for battery-powered medical instrumentation," *IEEE Trans. Circuits Syst. – I: Reg. Papers*, vol. 62, no. 11, pp. 2685–2694, Nov. 2015.
- [26] Y. Liu et al., "Automatic seizure detection using wavelet transform and svm in long-term intracranial EEG," *TNSRE, IEEE*, vol. 20, no. 6, 2012.
- [27] S. Benatti et al., "EMG-based hand gesture recognition with flexible analog front end," in *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*. IEEE, 2014, pp. 57–60.
- [28] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.