DOI: 10.1109/TBCAS.2018.2830659

Exploiting All Programmable SoCs in Neural Signal Analysis: a Closed-Loop Control for large-scale CMOS multielectrode arrays

Giovanni Pietro Seu, Gian Nicola Angotzi, Fabio Boi, Luigi Raffo, Luca Berdondini[1] and Paolo Meloni[1]

Microelectrode array (MEA) systems with up to several thousands of recording electrodes and electrical or optical stimulation capabilities are commercially available or described in literature. By exploiting their sub-millisecond and micrometric temporal and spatial resolutions to record bioelectrical signals, such emerging MEA systems are increasingly used in neuroscience to study the complex dynamics of neuronal networks and brain circuits. However, they typically lack the capability of implementing real-time feedback between the detection of neuronal spiking events and stimulation, thus restricting large-scale neural interfacing to open-loop conditions.

In order to exploit the potential of such large-scale recording systems and stimulation, we designed and validated a fully reconfigurable FPGA-based processing system for closed-loop multi-channel control. By adopting a Xilinx Zynq®-All-Programmable System on Chip that integrates reconfigurable logic and a dual-core ARM-based processor on the same device, the proposed platform permits low-latency pre-processing (filtering and detection) of spikes acquired simultaneously from several thousands of electrode sites. To demonstrate the proposed platform, we tested its performances through ex vivo experiments on the mice retina using a state-of-the-art planar high-density microelectrode array that samples 4096 electrodes at 18kHz and record light-evoked spikes from several thousands of retinal ganglion cells simultaneously. Results demonstrate that the platform is able to provide a total latency from whole-array data acquisition to stimulus generation below 2 ms. This opens the opportunity to design closed-loop experiments on neural systems and biomedical applications using emerging generations of planar or implantable large-scale MEA systems.

I. Introduction

Microelectrode arrays (MEAs) have been used in neuroscience since the very early 80^s to study the mechanisms underlying the dynamics of neural networks and brain circuits [1]. Thanks to major technological advancements occurred in this field in the last decade, CMOS-based MEAs capable of simultaneously record from thousands of densely integrated sensing electrodes are nowadays increasingly used to monitor the electrical activity of large neuronal populations with fine spatio-temporal resolution either *in vitro* or, more recently, *in vivo* [2]. Furthermore, these devices can integrate in the same chip both recording and stimulating electrodes or can be combined with external electrical or optical stimulation, thus enabling to perturb neural systems for studying their response dynamics by resolving the spiking activity of large number of neurons and mapping bioelectrical signals with an unprecedented spatial and temporal detail as in [3]–[5]. When delivered in open-loop conditions, however, such perturbations pose several limitations in interfacing neural systems. For instance, by being complex dynamical systems, stimuli elicit very different responses in each trial as resulting from the interaction of the stimulus with the ongoing neuronal activity and history-dependent network state. To achieve stable responses, or to modulate them predictably, stimulation settings would need to adapt to the dynamics of the neural network. This poses the need for efficient real-time processing platforms enabling the creation of closed-loop systems to control stimulation on the basis of the detected neural activity from large arrays of recording electrodes. Such closed-loop systems allow to perform real-time model estimation and dynamic characterization of neural networks properties [6]. State-of-the-art platforms and acquisition chips designed for processing of MEA-acquired signals show significant improvements with respect to the past, in terms of number of real-time processed parallel channels, sampling frequency, signal-to-noise ratio, latency and accuracy of the real-time algorithms, as described in [7] and [8]. However, MEA technology advances go even faster, resulting in devices able to sense signals coming from up to 65 000 electrodes [9]. Many of the currently available CMOS-MEA acquisition systems developed for *in vitro* electrophysiology, enable simultaneous sampling of around 1 to 4 thousands of channels, usually at 18-25 kSamples/s per channel with 12-bit precision, such as the 3·Brain BioCam X, the Multi Channel Systems CMOS-MEA5000-System or the Maxwell MaxOne Single-Well MEA. The real-time processing of the resulting data rate (around 1 Gbps) represents a challenging and computation-intensive task [10]. Moreover, the technological approach of adopting CMOS technology to realize MEAs with a large number of recording electrodes has been recently adopted for brain implantable probes [11], [12]. In this context, most promising studies about brain-machine interface systems adopt a closed-loop paradigm, which requires the system to be able to acquire data and to generate feedback stimuli with low and controlled latency [13]–[15].

However, dealing with closed-loop configurations using CMOS-MEAs introduces tighter constraints when designing pre-processing systems for a large number of recording electrodes. A key complex objective is to limit the latency associated with the execution of the overall loop, to improve responsiveness. Reaching this goal is not trivial, since the processing chain includes computation-intensive tasks, such as band-selective filtering of acquired signals and detection of meaningful neural spiking events. Moreover, the system needs also some degree of reconfigurability in order to target different setups and experiments. Nevertheless, the overall processing kernel exposes a significant degree of parallelism that

Giovanni Pietro Seu is with the Department of Informatics, Bioengineering, Robotics and System Engineering, University of Genoa, 16145 Genoa (Italy), and also with the Deparment of Electrical and Electronic Engineering (DIEE) at the University of Cagliari, Piazza D'Armi, 09123 Cagliari (Italy) (e-mail: giovanni.seu@diee.unica.it).

Gian Nicola Angotzi, Fabio Boi and Luca Berdondini are with the Neuroscience and Brain Technology (NBT) department at the Fondazione Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova (Italy) (e-mail: giannicola.angotzi@iit.it; fabio.boi@iit.it; luca.berdondini@iit.it).

Luigi Raffo and Paolo Meloni are with the Deparment of Electrical and Electronic Engineering (DIEE) at the University of Cagliari, Piazza D'Armi, 09123 Cagliari (Italy) (e-mail: raffo@unica.it; paolo.meloni@diee.unica.it).

[1] Equal contribution as principal investigators.

TABLE I: Comparison with state-of-the-art processing systems for MEAs

Reference Year	Hafizovich [23] 2007	Novellino [19] 2007	Venkatraman [28] 2009	Biffi [25] 2010	Wallach [21] 2011	Newman [22] 2013	Müller [26] 2013	Cong [30] 2014	Park [27] 2017	**This work** **2017**
Hardware	FPGA + PC	PC	PC	FPGA + DSP	PC	PC	FPGA	IC + μProcessor	FPGA	**AP-SoC**
Input ch	128	32	16	64	60	64	126	8	128	**4 096**
Sampling freq.	20 kHz	10 kHz	-	25 kHz	16 kHz	25 kHz	20 kHz	33 kHz	32.5 kHz	**18 kHz**
Resolution	8 bit	-	-	12 bit	12 bit	16 bit	8 bit	13 bit	16 bit	**12 bit**
Proc. tasks [2]	DF+SD	SD	DF+SD+SS	SD+DT+SS	SD	DF+SD+SS	DF+SD+DT	SA	DF+SD+SS	**DF+SD+DT**
Latency	2 ms	4 ms	15 ms	-	<1ms	7.1 ± 1.5 ms	<1 ms	-	-	**<2 ms**
Stimulating ch	128	8	-	-	2	64	42	32	8	**16**

can be exploited to achieve high throughput and low latency in the time-scale of a spiking event of few milliseconds.

To face this challenge, modern heterogeneous FPGAs integrating specialized hardware blocks, such as DSP slices, are a very promising target technology for applications with this kind of requirements. In this work, to implement a real-time processing system, we targeted the Xilinx Zynq® All Programmable SoC (AP SoC) family which integrates on the same chip the software programmability of an ARM-based processor with the hardware reconfigurability of an FPGA, to enable key analytics and hardware acceleration while integrating CPU, DSP, ASSP, and mixed signal functionality on a single device. The implemented system presents a tight coupling between the two subsystems available in the Zynq platform. The programmable logic is in charge of performing the computationally intensive data crunching tasks, such as band-selective filtering and detection of meaningful events on the acquired neural traces. The two ARM cores in the processing system are instead dedicated to respectively run a Linux-based operating system and a bare-metal program. The former permits an effective user interface and network connectivity while the later allows for implementation of closed-loop control algorithms. The elaboration system was implemented on a low cost Zynq device and tested in *ex vivo* experiments on the mouse retina by involving the analysis of the spiking activity recorded from thousands of retinal ganglion cells responding to light stimuli. As presented in this work, the reached performances in terms of processed channels, latency and degree of reconfigurability are a clear demonstration that AP SoCs are the perfect target technology to develop CMOS-MEA systems with real-time elaboration capabilities. This paper is organized as follows: the section II presents an overview of existing works for the elaboration of MEA-acquired signals. Our experimental setup and our processing system are instead presented in section III and IV respectively. The obtained performances are presented and discussed in section V before the conclusions.

II. Related work

Many different closed-loop technologies have been presented in literature to selectively perturb complex neuronal circuits with the aim of achieving fine and real-time control over their electrical activity. Such technologies find applications not only in basic neuroscience, to study how complex neural circuits are connected and how information is passed through complex networks [16], but are relevant also in the field of neural prosthesis for therapeutic interventions [17] and/or to restore sensory pathways [18]. Within this framework, previously presented works targeted different processing architectures as elaboration hardware, from simple software running on desktop PCs, to FPGA or integrated circuit implementations and there are also some works developed in analogue electronics. In the following, the most relevant published works are shortly presented.

For *in vitro* experiments, for many years the preferred target technology to develop closed-loop neuro-interfacing systems has been the desktop PC for its simplicity [19]–[22]. Novellino et all. in [19], for instance, presented a neuro-robotic system that connects a neural network with a mobile robot. For the purpose, neural data was recorded from 32 sensing electrodes and simultaneously processed by a desktop PC, which also took care of the control response sent through a home made 8-channel stimulus generator. Two more desktop PCs were used to control the other parts of the experimental setup. In [20] the desktop PC was used to run a Simulink based xPC target application. With this system authors were able to control a MEA featuring 60 electrodes which could be used both for recording and for stimulation. When a specific property of a spike train was detected a stimulus was sent with a minimum achieved latency under a ms, but only for predefined stimulating electrodes and waveforms. Instead, the system took about 10 ms to change the stimulating electrodes and waveform. In [21] they used this system to implement a closed-loop technique to control the response of neurons and to characterize their input-output relationships. The work presented in [22] was focused on a low cost approach. Authors developed the open-source system called NeuroRigther which allows to design sophisticated closed-loop experiments by maintaining a low input-output latency. Specifically, when targeting a 64 channels MEA the minimum achieved latency with this system was of 7.1 ± 1.5 ms.

Other approaches targeted hardware-embedded approaches using FPGAs in order to implement real-time elaboration and low-latency feedback control [23]–[27]. The work in [23] used both a FPGA and a desktop PC to achieve low-latency closed-loop capabilities and a CMOS-MEA chip with 128 bidirectional electrodes for acquisition and stimulation. The FPGA was used to simultaneously acquire data coming from all the 128 channels and to perform events detection. An event was detected when at least a spike was present in a segment of the input signal. Multiple spikes in the same segment form a single event. The PC, relieved from all the computational load needed for events detection, achieved a total latency of 2 ms to generate the closed-loop stimuli. Other works in [24] and [25] were instead focused on real-time processing of multielectrode array signals, but were not designed to achieve closed-loop capabilities. Featuring complex elaboration tasks they are able to simultaneously process in real-time 256 and 64 channels respectively. Even though these systems are implemented to work in real-time, no information was given about latency. The work in [26] is instead a good example of FPGA used to implement a closed-loop system. The FPGA here was in charge of spike detection and stimuli generation, it was able to process data coming from 126 channels simultaneously, while generating stimuli on 42 different output channels in less than 1 ms. Finally, a more recent closed-loop FPGA implementation, presented in [27], was shown to be capable not only of doing real-time spike detection but also spike sorting of 128 input channels simultaneously.

[2] Table I also summarizes processing tasks executed by each system: DF=Digital Filter, SD=Spike Detection, DT=Dynamic Threshold, SS=Spike Sorting, SA=Spectrum Analysis

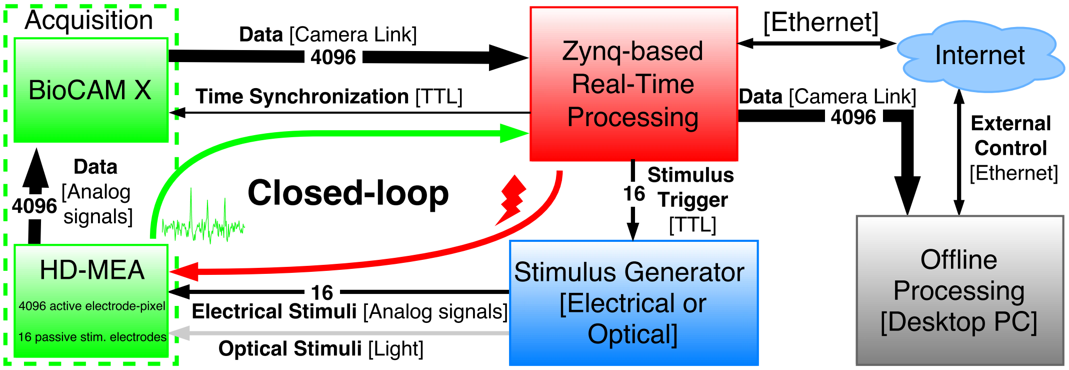

Acquisition
BioCAM X
Data [Camera Link]
4096
Time Synchronization [TTL]
Zynq-based Real-Time Processing
[Ethernet]
Internet
Data [Camera Link]
4096
External Control [Ethernet]
Data [Analog signals]
4096
Closed-loop
Stimulus Trigger [TTL]
16
HD-MEA
4096 active electrode-pixel
16 passive stim. electrodes
16
Electrical Stimuli [Analog signals]
Optical Stimuli [Light]
Stimulus Generator [Electrical or Optical]
Offline Processing [Desktop PC]

Fig. 1: Schematic block diagram of the experimental setup.

Other recent works were instead focused on *in vivo* experiments due to the increasing interest in adopting closed-loop system to study and interface neural circuits [28]–[31]. In [28] the processing was implemented on a desktop PC. The system was able to generate feedback stimulation in response to the neural activity recorded from 16 electrodes implanted in barrel cortex of an awake rat within 15 ms. Similarly, the work described in [29] exploited a microcontroller for *in vivo* experiments on behaving small laboratory animals aiming at closing the loop from 8 recording channels and as many as possible stimulating sites. The works in [30], [31] presented more elaborated processing architectures. In the former, the processing was handled by an integrated circuit and in the latter by an analog circuit. In both works the system control and closed-loop decision was done using a microprocessor. They were able to process in real-time 8 and 16 channels, respectively. Finally, as reviewed in [32],a number of works exploiting optogenetic stimulation have been recently introduced, thus indicating the need to close the loop not only with stimuli delivered by on-chip electrodes but also with externally triggered light-stimulation systems.

To the best of our knowledge, all the previously presented systems were developed for recording devices with a small number of electrodes and are thus limited to the processing of a low number of sensing electrodes. As a result they can only be used to target very local and confined neural circuits. By targeting the use of large-scale CMOS-MEA devices capable of simultaneously record from thousands of electrodes, our work, conversely, is conceived to advance the limitations of the state-of-the-art in the field of processing systems connected to MEAs. In Table I, we present a direct comparison of our work with other implementations presented in literature.

Summarizing, our system:

- is the first effort that exploits the heterogeneous processing architecture of modern All-Programmable SoCs, to improve flexibility while keeping up with very high data-rates;
- increases by more than one order of magnitude the number of parallel recording channels processed in real time, while guaranteeing a closed-loop latency lower than 2 ms;
- provides a high degree of reconfigurability/adaptability to target different acquisition systems and different stimulus generators.

To assess the system functionalities, we built an experimental setup targeting data acquisition from a commercially available high resolution neural recording platform, the BioCam X from 3Brain AG. By using active CMOS-MEA devices with 4 096 recording sites, such platform can indeed be used to map neural activity on large-field of views of several square millimeters, with submillisecond resolution (sampling up to 18kHz), in different *in vitro* and *ex vivo* experimental models.

III. System Architecture Overview

Fig.1 shows the experimental setup that we built in order to validate the performance of the proposed real-time processing system. Four main modules can be identified:

- Acquisition unit: it consists of a commercial, large-scale sensing, CMOS based Microelectrode Array platform for label-free *in vitro* electrophysiology;
- Stimulation unit: it can be any commercially available or custom made stimuli generator that accepts Transistor-Transistor Logic (TTL) digital inputs to deliver preloaded stimulus.
- Real-time Processing unit: it is the core of this work and will be described in detail in section IV. Briefly, it is implemented on a ZedBoard^{TM} development kit and, by exploiting the embedded Xilinx Zynq®-7000 All Programmable SoC, it permits real-time processing of the acquired neural data as well as implementation of user programmable closed-loop algorithms while keeping timing consistency among all subsystems;
- Offline Processing unit: consists of a desktop workstation that is used for real-time data visualization and storage for further offline processing.

The two different data flows forming the closed-loop system are detailed hereafter.

A. Forward data flow

In the proposed system neural data are acquired by a BioCAM X platform that permits whole array recordings with submillisecond resolution (sampling rate up to 18kHz) from 4 096 active sites integrated on CMOS-MEA planar devices. The chip types used are the 3·Brain HD-MEA Stimulo when electrical stimulation is involved, otherwise the HD-MEA Arena. The amplified and digitalized signals (12-bit of resolution) are continuously sent through a Camera Link communication protocol to the Zynq-based Processing unit for real-time elaboration of the 4 096 raw neural traces and extraction of features that might be meaningful for closed-loop experiments. For each of such neural traces, the system implements a bank of programmable filters that allow to separate spiking signal events (from 300 Hz to 3 400 Hz) from low-frequency signal oscillation (up to 300 Hz). The resulting data traces are processed by an event detection module whose output can be finally used for closing the loop by means of user defined decision algorithms running on one of the two ARM processors that are available on the Xilinx Zynq®-7000 All Programmable SoC. For verification purposes, the Real-time Processing unit also produces a time synchronization signal that is sampled by the BioCAM X and forwarded together with the neural signals through the Camera Link interface. Such signal provides a timing reference that can be used to compare the spikes detected online on the FPGA with those detected offline. By using another Camera Link connection, the raw data is sent to the Offline Processing unit where a data acquisition software, the BrainWave X (from 3Brain AG), running on a PC allows for online visualization and storage of the acquired raw data. Finally, the Zynq-based Real-time Processing unit also features Ethernet-based connectivity; thus, it can be connected to a local LAN or to the Internet in order to be remotely re-programmed at any time, to change the HW/SW configuration or to adjust experimental parameters. The Ethernet-based connectivity is also used to communicate the online results to the Offline Processing unit.

B. *Feedback data flow*

The proposed system can be used by neuroscientists in different experimental paradigms requiring the use of either electrical [33] or optical stimuli [34] for closing the loop. As previously stated, in fact, any commercial or custom made stimulus generator can be part of our architecture, provided that it accepts TTL digital input triggers to deliver preloaded stimuli. In current implementation, electrical stimuli can be delivered using 16 channels of the commercial Plexon PlexStim Electrical Stimulator System. Such electrical stimulator, in fact, permits to selectively send preloaded electrical stimuli of arbitrarily user-defined waveforms to a selected channel, upon detection of a TTL trigger on the correspondent digital input, within 1μs input-output latency. The so produced stimuli are delivered to biological tissues through the 16 stimulation electrodes available in the HDMEA Stimulo. Similarly, visual stimulation of the retina can be achieved by delivering light-stimulation images by using a DLP system, in our case the Texas Instrument DLP® LightCrafter$^{\text{TM}}$ Evaluation Module which can display high-speed light pattern sequences.

IV. Zynq-based Real-time Processing unit

The core of the proposed system is the Real-time Processing unit, whose schematic block diagram is depicted in Fig.2. It consists of three major parts: *I*) Programmable Logic & Hardwired Blocks, *II*) a Zynq Processing System and, *III*) a DDR Memory. These parts are tightly coupled and their synergy is used to achieve high performance and low latency, exploiting parallel processing, and flexibility through interaction between software and reconfigurable hardware.

A. *Programmable Logic & Hardwired Blocks*

The programmable logic and the hardwired blocks (DSPs, Block RAMs) are used to implement different functional Intellectual Property (IP) cores, connected to each other in a dataflow fashion. The input data, received from the BioCAM X through a Camera Link interface, is de-serialized and interpreted in order to extract the incoming samples, based on the rules of the proprietary communication protocol used by the 3·Brain acquisition platform. In the subsequent blocks, the raw data is filtered by a bank of programmable digital filters and then a spike detection algorithm based on either hard or dynamic thresholding is performed to detect neural activity, as detailed respectively in section IV-A2 and IV-A3. The filtered data, together with the information about the detected spikes (time stamp and channel identifier), are finally temporarily stored in the DDR memory where they can be accessed by the Zynq Processing System. Furthermore, the programmable logic also re-routes the raw input samples to the Offline Processing unit using a Camera Link protocol, for real-time data visualization, permanent storage and, possibly, offline processing.

1) I/O Data Interface

This module is used to interface the Zynq board with the BioCAM X and with the PC. At each clock cycle, the 28-bit data words received from the BioCAM X are de-serialized and decoded according to the Camera Link communication protocol: 4bits are used as control signals to discriminate when data are valid and when a full frame of 4 096 samples is acquired; the 24 remaining bits represent data samples (12-bit each) from two of the 4 096 available recordings channels. More in detail, as shown in Fig.3, the input interface is composed of four serial data links and a synchronization clock. The serial data links have a serialization factor of 7 which means that for each clock cycle 7 data bits are transfered for each link. The synchronization clock is used by the *Clock Manager* block to generate 2 different clocks, phase aligned to each other and aligned to the input clock. One of such clocks (7× the input clock frequency) serves the *Serdes* blocks which are used to sample the input data while the other is used to sample the output 28bit data resulting from the de-serialization process. Given the very high input bandwidth (about 1Gbit/s), which results in very fast data transitions, synchronization with the Camera Link clock (operating at 50MHz) is mandatory for correct data recovery. For this reason, with the aim of implementing data sampling in the middle of the time interval between two data transitions, the module also integrates a *Calibration* module, which permits variable time shifting of the input clock, and a *Deskew* module integrated in each data line, which keeps adjusting the individual data delay to perform runtime data alignment and compensate the skew among different data lines. Finally, from the so produced 28-bit data words, the module extracts the two 12-bit data samples belonging to as many different channels among the 4 096 supported. The so produced data is routed into a bank of 32 stream channels, thus exploiting the full parallelism offered by the proposed platform. For validation purpose and for possible further offline signal processing, the setup also permits to store all the acquired raw data. For the purpose, a Camera Link Serializer integrated in the module converts the parallelized data in its original form and sends it to the output Camera Link interface connected to the Offline Processing unit.

2) Band-pass Filters (BPFs)

To preserve the fast spiking activity of single neurons while removing line noise and low frequency signal components resulting from the concurrent activity of a large neuronal population (known as Local Field Potentials - LFP), the incoming raw data is processed by a bank of digital filters, each implementing a finite-impulse response (FIR) band-pass filter. In current implementation the filters are designed with cutoff frequencies respectively placed at 300 Hz and 3 400 Hz

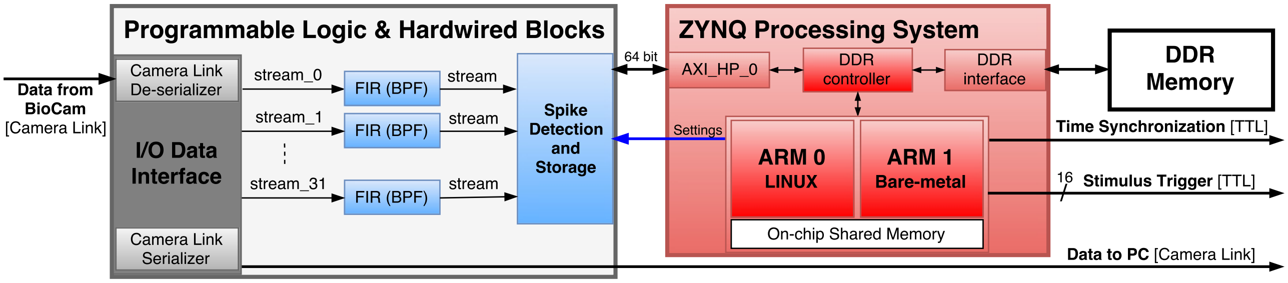


Fig. 2: Schematic block diagram of the Zynq-based Real-time Processing unit consisting of three major parts: Programmable Logic & Hardwired Blocks, a Zynq Processing System and, a DDR Memory.

as suggested [35]. Also, to improve spike detection, the FIR blocks feature linear phase response, thus avoiding signal distortions and preserving the shape of the spikes. To save FPGA resources, each block operates in a time division multiplexing fashion. More in detail, the architecture integrates 32 FIR blocks, allowing to filter 4 096 input channels, partitioned in groups of 128 channels for each block. We selected this specific partitioning configuration among many options in order to minimize resource usage, as detailed in [36]. Along with the partitioning scheme, another relevant architectural parameter of the FIR stage is the order of the filter (N): a higher N indeed permits a better band-pass selectivity, but reflects in higher resource usage and latency. Given the application constraints, we chose a filter order of N = 63, which revealed to be the best design point of our design space exploration [36]. By using symmetric coefficients the number of multiplications needed can be halved. The resultant run-time workload required for filtering 4 096 input channels, with a maximum sampling frequency f_s of 18 kHz is then:

$$WL_{FIR} = 4096 \cdot 18\,kHz \cdot \frac{(N+1)}{2}\ MAC/s \approx 2.359 \cdot 10^9\ MAC/s$$

expressed in multiply and accumulate (MAC) operations per second.

3) Spike Detection and Storage

The filtered samples are processed by a single block, which performs the spike detection and storage tasks, as described hereafter.

a) Spike Detection: Spikes are detected on the filtered neural traces using an algorithm based on amplitude threshold [37]. The choice for such threshold, however, is of mandatory importance. The use of high thresholds, in fact, would lead to a large number of missed spikes while false detections would originate from noise crossing low thresholds. As proposed in [38], a proper solution for this problem consists in computing the amplitude of the threshold by estimating the standard deviation of the noise $\sigma_n = median\,|x|/0.6745$, with x being the filtered signal, and multiplying it by a constant α, typically between 3 and 6, resulting in $Thr = \alpha \cdot \sigma_n$.

Furthermore, a convenient way to take into account multiple signal and noise amplitudes which may change over time among the recording channels, is to dynamically and automatically adjust the threshold by using a sliding window mechanism which considers only a given number of recent samples. Implementing such dynamic approach with the solution proposed by Quiroga, however, has practical limitations since the calculation of the median is computationally intensive; thus, in our implementation, the threshold is evaluated using as σ_n the exact value of the standard deviation of the signal.

$$\sigma_n = \sqrt{\left(\sum_{k=0}^{M-1} x_{i-k}^2 - \left(\sum_{k=0}^{M-1} x_{i-k}\right)^2 \cdot \frac{1}{M}\right) \cdot \frac{1}{M}} \qquad (1)$$

In Eq.1 M is the size of the sliding window (4 096 samples in current implementation, corresponding to a time interval of about 228 ms), and x_i is the i-th sample of the filtered signal. To reduce the computational complexity of the proposed algorithm, multiple optimizations were applied. More in detail:

- M is a power of 2, in this way the divisions are implemented as bit-shifts;
- the sum terms in the equation 1 are updated considering only the new acquired sample and the oldest one stored in memory;
- σ_n^2 is used instead of σ_n to avoid calculating the square root: a square value of the threshold is compared with a square value of the the samples (already available).

The processing flow implemented in hardware to compute $Thr^2 = \alpha^2 \cdot \sigma_n^2$, where σ_n represents now the exact value of the standard deviation of the signal, is reported in Fig. 4. By doing such optimizations we achieved a workload of only 4 MAC operations per channel per input sample, resulting in a total workload associated to the spike detection stage of about:

$$WL_{detect} = 4096 \cdot 18\ kHz \cdot 4\,MAC/s \approx 295 \cdot 10^6\ MAC/s$$

b) Storage: The filtered data samples as well as information about detected spikes, including the time stamp and the channel identifier, are saved in the DDR memory. Filtered samples are used by the hardware module for dynamic and automatic adjustment of the threshold, as previously described, while detected spikes are used by the closed-loop decision

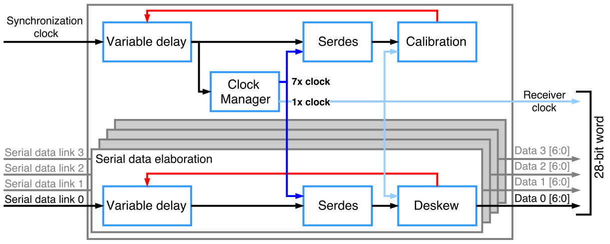


Fig. 3: Schematic of the Camera Link De-serializer.

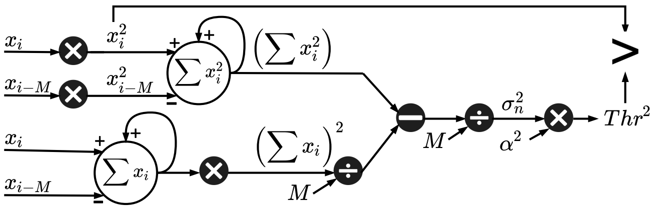


Fig. 4: HW implementation of the spike detection algorithm.

algorithm running on the ARM processor. In perspective, also the filtered sample can be easily accessed by any user defined software running on the ARM processor to perform further analysis and operations. To better match with the processor architecture, the filtered samples are stored as 16-bit wide data, while each spike detection is saved as an 8-bit data flag. Thus, the required input and output bandwidth can be evaluated taking into account that, for each data sample, the system requires two 16-bit data accesses (to update the threshold Thr^2 as described previously) and one 8-bit data access (to write the spike detection flag):

$$BW = 4096 \cdot 18\ kHz \cdot (16 + 16 + 8)\, bits \approx 3\, Gbit/s$$

As shown in Fig. 2, the storage module is connected to one of the four AXI High Performance (HP) ports that are available in the Zynq Processing System (PS) which permit direct access to the DDR memory through a dedicated controller, without any processor intervention. Being the HP ports 64-bit wide, and considering a working frequency of 80 MHz, the bus allows for a maximum bandwidth of around 5 Gbit/s, which is adequate to meet the bandwidth requirements of the storage block.

B. ZYNQ processing system

The Zynq processing system integrates a dual-core ARM Cortex^{TM}-A9 processor.

Our implementation adopts an Asymmetric Multi Processing configuration of the two available ARM cores in order to exploit the processing system for multiple functionalities. Specifically, one core runs a Linux-based operating system, devoted to housekeeping and interfacing with the external environment while the second available core runs a bare-metal application (i.e. without any operating system), which performs the final elaboration steps required to implement the closed-loop decision task. Both cores run a lightweight shared-memory based API framework that allows for remote control, configuration, synchronization and inter-process communication between softwares running on the two independent cores.

1) Linux Core

The operating system running on the first core allows the system to feature a high-level user interface and network connectivity. This core is also responsible of system initialization, programmable logic clock configuration and management, and of performing the bare-metal core management/communication tasks. Thanks to the networking support, the platform allows to remotely update and manage both the programmable logic hardware and the bare-metal program. Thus a partial or full reconfiguration can be done at any time in order to handle different experimental configurations.

2) Bare-metal Core

The bare-metal core has easy and real-time access to the DDR memory, where filtered data and the occurrences of detected spikes are saved. Thus, it is used to run user defined decision algorithms implementing specific closed-loop experimental paradigms. Interestingly, since such algorithms are described using general-purpose programming languages (such as C), they can be easily modified and adjusted by the final user, thus making the proposed system architecture extremely reusable and versatile. To respect real-time constraints, however, the bare-metal core must run the closed-loop algorithm in a time frame of $\frac{1}{f_s} = 55.56\,\mu s$. As the ARM cores work at 666 MHz frequency, the decision algorithm must be completed within 37,000 cycles.

TABLE II: Hardware utilization report

Task	FF	LUT	BRAM	DSP
Filtering	4 480	4 928	96	32
Spike Detection	11 431	15 163	14	28
Other	8 083	6 869	6	0
Whole System	23 994 (23%)	26 960 (51%)	116 (83%)	60 (27%)

Finally the hardware is equipped with 512 MB of DDR3 memory, which is virtually shared by the two cores. Notice that the amount of memory dedicated to the bare-metal core is dependent on the specific workload to be executed within the closed-loop decision algorithm and can be easily changed during software development. In our experiments, we did not notice any significant mutual interference between the memory accesses performed by the two cores, impacting on the closed-loop performance of the system.

V. Results

To assess the possibility of the proposed programmable SoCs solution to be employed as the real-time processing core for closed-loop experimental paradigms, we implemented it on a ZedBoard^{TM} provided by Avnet and used it in *ex vivo* experiments on mouse retina.

A. Hardware utilization

The target ZedBoard^{TM} board integrates a Xilinx Zynq$^{®}$ Z-7020 All-Programmable SoC featuring 106 400 Flip-Flops (FFs), 53 200 Look-Up Tables (LUTs), 140 Blocks RAM for a total of 4.9 Mb of on-chip memory and 220 hardwired DSP48E1 slices. Each DSP48E1 contain a 25x18 multiplier and a 48-bit accumulator, meaning that running at 80 MHz it can perform $80 \cdot 10^6 MAC/s$. Thanks to the hardware optimization focused on minimizing resource usage the whole system fits the targeted FPGA device with the occupancy rate summarized in Table II. More in detail, thanks to time division multiplexing implemented in the FIR blocks, the whole workload of the filtering stage, $2.359 \cdot 10^9 MAC/s$, takes just 32 DSPs with each one used at 92% of its performance. For the threshold update and the spike detection, with the relative low workload calculated in the previous section, $295 \cdot 10^6 MA/s$, the system uses instead 28 DSPs. Comparing this with the performance offered by a DSP48E1, results in a very low utilization factor of each DSP. This is due to two major reasons. The first is that some multiplications

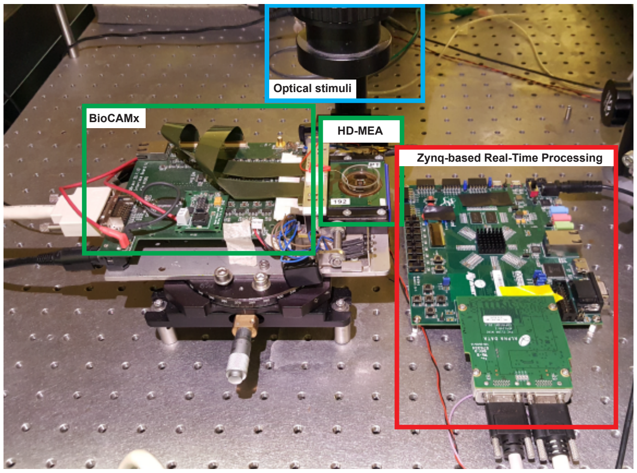


Fig. 5: Experimental setup used for closed-loop *ex vivo* experiments on mouse retina.

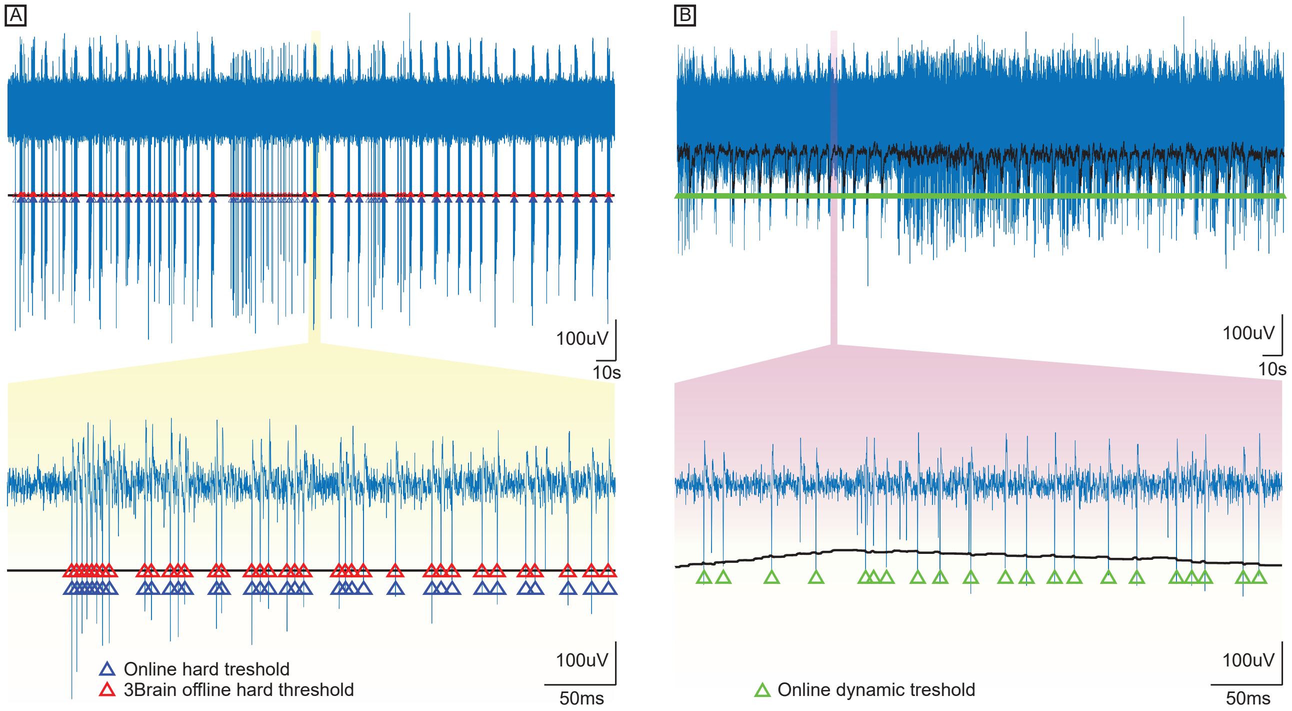


Fig. 6: Representative raw data recorded by a single electrode of the CMOS-MEA from a retinal ganglion cell (top) and closed up view showing a narrower time window (bottom). A) Black line represents the hard threshold used by both the online and offline detection methods. Blue and red triangles respectively represent the spike occurrences detected online by the presented system and offline by the Brainwave X detection tool. B) Black line represents the dynamic threshold that is derived from the current noise standard deviation of the signal. In this case spike occurrences are marked as green triangles.

performed for the threshold update involve numbers that don't fit the 25x18 multiplier, resulting in the need of more DSP for the same operations. The second reason is that, being the DSPs the less used resource, we decided to waste DSP cycles in order to minimize the routing and the control that otherwise would have take much more LUTs and BRAMs. The unused hardware resources can be perspectively used for further processing in the future to increase performance and/or to add functionalities. The scalability of the approach was evaluated in a preliminary exploratory work in [36]. As we can see from the table II, the performance of the configuration presented here are limited by the amount of BRAM blocks that are necessary to implement FIR filters and to buffer the input samples, while only a small portion of DSP blocks are utilized. It is worth to point out that the amount of BRAMs needed is independent on the input sampling frequency, as long as the filter order is the same. Thus, only the number of DSPs are affected by an increasing of the input frequency, meaning that the exact same architecture configuration presented here can process the 4 096 channels with an input frequency up to 30 kHz, using 124 DSPs.

The resource that saturates at 30 kHz is the bandwidth to the DDR memory fixed at around 5 Gbit/s by the AXI HP port. However this limitation can be easily overcome by using multiple blocks and thus multiple ports to access the DDR, up to 4 ports are available. We find out that by using two spike detection modules we can reach an input frequency up to 40 kHz before saturating the hardware resources available in the current device. If an even greater input sampling frequency is needed or for increased number of input channels or when a major filter order is required for a better accuracy, a bigger device must be used, with more available BRAM blocks and LUT. Similar bigger AP-SoCs are available on the market at prices around 1 Keuro.

B. *Experimental validation on ex vivo mice retina*

With the aim of validating the implemented hardware, we performed experiments using retina whole-mounts on 4096 CMOS-MEAs subjected to controlled visual stimulation. Specifically, we assessed all functionalities of the system, from signal acquisition to stimulus triggers generation. By following previously reported protocols [39]. Due to unavailability of the last model of the BioCAM X, the experiments have been executed with a previous version capable of sampling each of the 4 096 electrodes at around 7 kHz. We dissected and isolated adult mouse retinas after sacrificing the anesthetized animal by cervical dislocation. All the experiments were performed in accordance with the guidelines established by the European Community Council (Directive 2010/63/EU of 22 September 2010). Experimental protocols were approved by the Italian Ministry of Health. Once isolated, we faced the retina down onto the HD-MEA device putting the retinal ganglion layer in contact with its surface and leaving the photoreceptor layer exposed. In order to maintain the tissue during the experiments, a perfusion line, supplied by a peristaltic pump (1 ml/min), ensured a constant flow of a media composed by AMESs medium (Sigma - Merck KGaA, Darmstadt, Germany) with 1.9g/L of sodium bicarbonate equilibrated with carboxigen ($95\%O_2$ and $5\%CO_2$).

1) *Open-loop system evaluation*

To assess the capability of the system of detecting spikes in real-time and to extract their waveforms, we recorded the spontaneous neural activity of retinal ganglion cells (RGCs) from a mouse retinal whole mount. In such experiment we

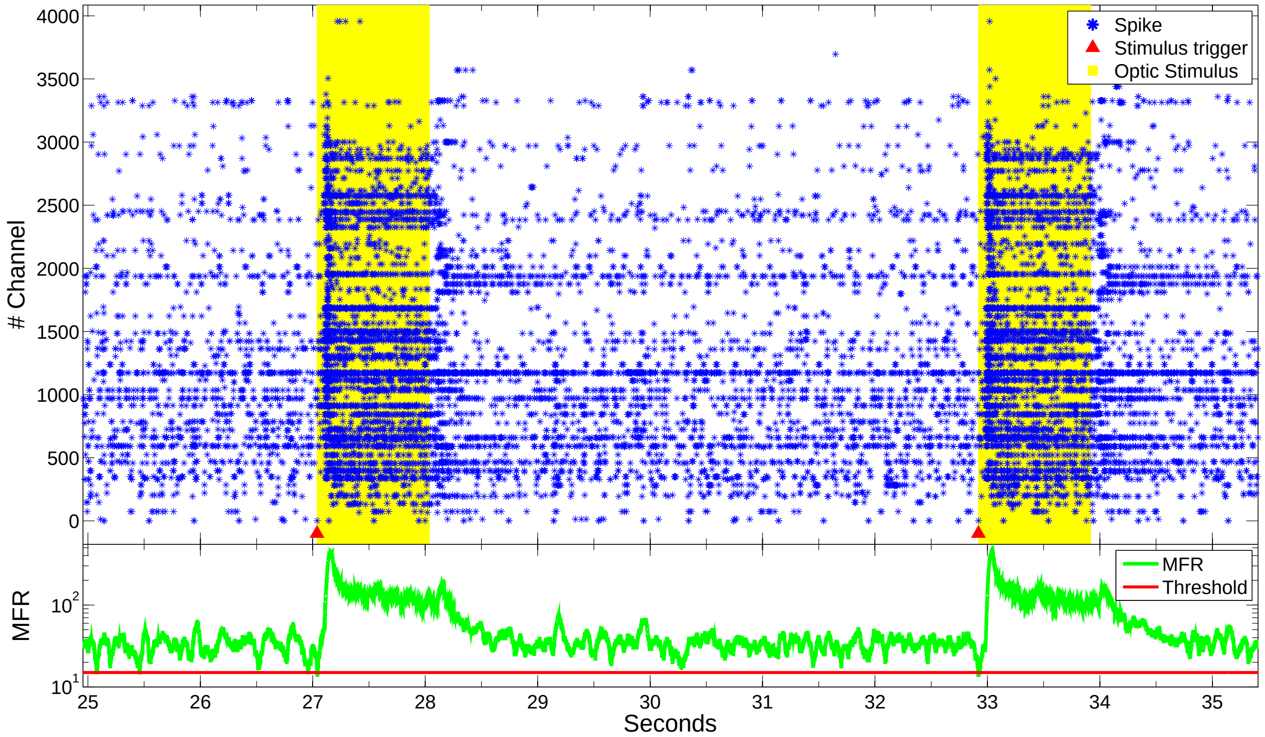


Fig. 7: Results from a closed-Loop Experiment involving recordings from a mouse retinal whole mount and optical stimulation delivered according to the mean firing rate measured on a time window of 50ms.

connected the system similarly to the configuration depicted in Fig.1 but in an open-loop mode (i.e. without connection to the stimulus generator). For each detected spike, we configured the system to store on a text file the data samples within a time window of 4ms centered around the threshold crossing. In order to assess the capability of our system to effectively perform real-time spike detection, we used as ground truth the offline hard threshold algorithm available into the BrainWave X acquisition software. In detail, we run the offline algorithm on the raw data acquired by our acquisition system with the same parameters used by our system for the online spike detection (hard threshold set to -200μV, 2ms refractory period). We obtained over 99% of coincidence (with a tolerance of $\pm 500\mu$s) between real-time and off-line detected spikes that were recorded from the 4096 MEA channels. A representative example of the results obtained with the online and offline hard threshold detection on a single raw trace is shown in Fig. 6.A. We also tested the real-time spike detection based on the dynamic threshold approach described in section IV-A3 (setting for each channel $Thr^2 = 26 \cdot \sigma_n^2$). Results confirmed that also in this case the system correctly detects the spiking events (see Fig. 6.B. for a representative example).

2) *Closed-loop system evaluation*

A second experiment was designed to assess the closed-loop performances of the system. The experiment was performed on a mouse retinal whole mount by closing the loop to control the optical stimuli generated by a DLP® LightCrafter™ Evaluation Module. To validate the system we programmed a closed-loop algorithm running in the bare-metal core. First, we computed the mean firing rate (MFR) as the number of occurrences of spikes detected in a block of N recording channels, within a sliding time window of user defined duration (WINDOW_SIZE) that is shifted for every new filtered sample written in the DDR.

Second we provided a 1 s light stimuli, with a programmable delay (STIM_DELAY), each time the computed MFR was below a user defined threshold MFR_THR. Finally, for verification purpose only, we stored temporal and spatial information of each detected spike and triggered stimulus.

In Listing 1 we report the pseudo-code for the closed-loop algorithm. The algorithm is executed for each new input sample, without violating real-time constraints, since its execution time requires less than 37,000 cycles on the ARM core. The execution time does not depend on the sliding window size, thus any MFR variation determined by any new acquired sample may be monitored. Notice that any other algorithm respecting the real-time constraint may be used, enabling monitoring of variables changing much faster than MFR. For example, if we reduce sliding window size to one sample frame, the algorithm monitors Instantaneous Firing

Listing 1: Closed-loop code

```
for each block
{
  spikes = 0
  for each channel in block
  {
    if there is a spike
    {
      store channel_id and time
      spikes++
    }
  }
  MFR = MFR + spikes - spikes_memory[addr]
  spikes_memory[addr] = spikes

  if(MFR < MFR_THR)
  {
      store block_id and time
      send stimulus
  }
}
if(addr == (WINDOW_SIZE-1))
  addr = 0
else
  addr++
```

Rate (IFR).

In the current proof-of-concept implementation, the algorithm manages up to 16 different stimuli, each one triggered by the neural activity recorded from a separate block of 256 recording sites.

On the top side of Fig. 7 we show the raster plot resulting from an experimental session in which we calculated the MFR on a single block comprising N=4096 recording channels, considering a sliding window of WINDOW_SIZE=50 ms. We delivered 1s of light stimulus (in yellow) when the MFR (green line on the bottom side of the figure) was below the value of MFR_THR=15 spikes (represented by the horizontal red line), without any delay (STIM_DELAY=0ms). A strong response is registered after around 100 ms from the onset of the optical stimulus, and for the whole duration of the stimulation [40].

C. Assessment of real-time processing and latency

After the validation of the whole system, we evaluated the responsiveness of the proposed solution, intended as the time required to deliver an output stimulus in response to a specific input event. Fig. 8 shows a timing diagram representing the scheduling of the tasks executed by different modules. The time duration of each task was first measured from HDL simulation and successively verified after implementation by using a counter connected to the processing system to evaluate the execution time in terms of clock cycles. The system implements a pipeline, synchronized with the sampling period of the system, 0.056 ms ($1/f_s$ with $f_s = 18$ kHz). Each task lasts the time required for processing the data coming from a whole input frame, which comprises 4096 channels.

The filtering is the first task to start as soon as a sample is received. Even if such processing is very fast, thanks to the exploited parallelism, all the filtered samples include an intrinsic delay due to the phase response of the filter. In detail, since we used a linear phase FIR filter, the group delay is constant for all the frequencies and is equal to $N/(2 \cdot f_s) = 1.75\ ms$ (where N = 63 is the filter order). This is the major contribution to the overall system latency and, if necessary, can be reduced by lowering the FIR order. Filtered data are subsequently processed for spike detection and any detected event has to be stored. For the purpose, the spike detection task requires the hardwired blocks to access the DDR to read the last sample (in order to update the dynamic threshold); however, it can start only after the block has completed saving any information about the spikes detected in the previous frame (*WS task* in the figure). Interestingly, the spike detection task is so fast that its pace is determined by the production of filtered samples rather than by its processing time, thus its execution is completed shortly after the filtering of the last channel. Once the detection is performed on all the channels, the communication interface is used to write into the DDR the information about any detected spike.

Finally, the closed-loop algorithm presented earlier is executed by the bare-metal core. The specific values of the user programmable parameters in the algorithm do not affect significantly the execution time. However, two different cases requiring very different execution times can be identified, as reported in Fig. 8. For the first version, the system does not require to store information about detected spikes, resulting in the execution time to be independent by the number of detected spikes. In this particular version the total measured input-output latency is of 1.861 ms. The second version, conversely, stores the spatio-temporal information (i.e. the time stamp and the channel identifier) of each detected spike for further offline processing and verification. The execution time of this second approach largely depends on the number of detected spikes due to the overhead introduced for saving the information in the external memory. In the figure, *Case A* represents the situation with the maximum number of spikes that can be processed in an iteration without violating the real-time constraints. *Case B*, instead, considers the case when a spike is simultaneously detected in each of the 4096 available channels. Although the measured latency for this later case is below 2 ms, the time required for the processing is larger than the sampling period, resulting in real-time violation. Such worst case scenario, however, is very unlikely and, given the refractory period of a neuron being about 2 ms [41], it can happen at most once every many iteration. From the *Case A*, we can calculate the maximum firing rate that our application can support. By being able to process 150 spikes/iteration, in fact, this means that 2.7 millions spikes per second can be processed while still meeting the real-time constraints. This corresponds to all 4096 channels having a firing rate of around 660 spikes/s, which is larger than the maximum firing rate for a neuron that is about 400-500 spikes/s [42]. Thus, in the long run, the real-time constraints are always satisfied, even if an iteration may take more time than the sampling period. Since the closed-loop control executed on the ARM uses almost all the available execution time, more complex algorithms may require support from the programmable logic to comply with the real-time constraints.

VI. Conclusions

This work presents a state-of-the-art acquisition and processing platform that can be used in neuroscience to implement closed-loop experiments involving neural recordings from high-density CMOS-MEA devices with up to 4096 simultaneously recording electrodes. For such purpose the platform was designed and implemented targeting full exploitation of a low cost Xilinx Zynq device that integrates two ARM cores and FPGA logic. The system implementation effectively exploits the synergy between programmable logic and processing system in the Zynq, and utilizes less resources than those available on the target mid-to-low end device. Thus, also considering the size of other devices of the same family, the proposed hardware architecture allows for scalability and adaptivity, as the number of input channels, input sampling frequency and filters' order can be tuned according to a wide range of use cases.

We presented results from the full system validation through *ex vivo* experiments performed on mouse retina. Specifically, we demonstrate that the platform is able of real-time processing data coming from a 4 096-electrode MEA and permits closed-loop stimulation and recordings with a maximum latency of 1.86ms.

Interestingly, the system can also be used in experimental paradigms that use of electrical instead of optical stimuli. For now the system trigger preloaded stimuli from an external device but in future implementations stimuli might be generated by the real-time hardware itself. Furthermore, the possibility to easily modify the decision algorithms running on the bare-metal core makes the proposed system an extremely powerful tool for neuroscience studies and closed-loop experiments that

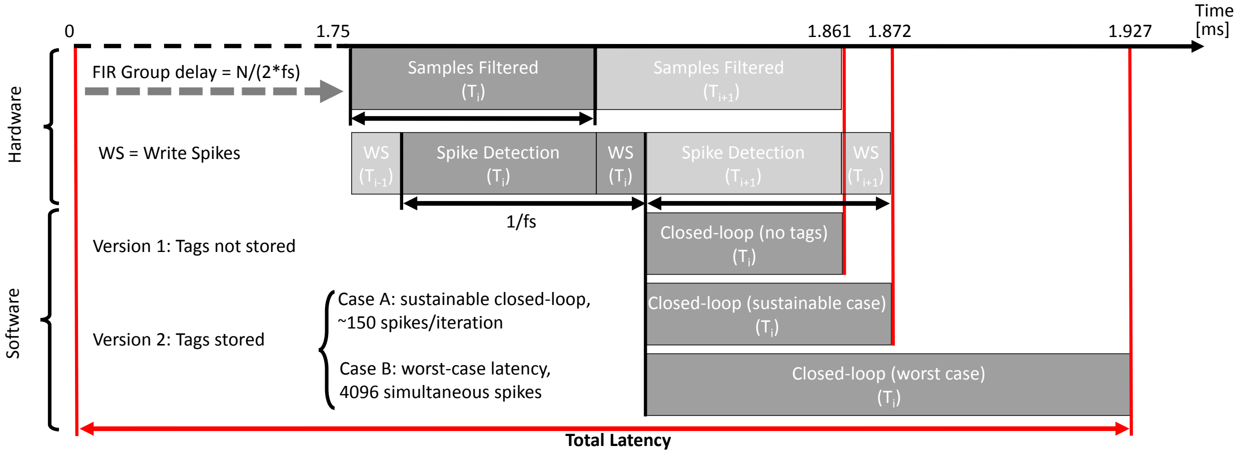

Time [ms]
0
1.75
1.861
1.872
1.927
Hardware
FIR Group delay = N/(2*fs)
Samples Filtered (T_i)
Samples Filtered (T_{i+1})
WS = Write Spikes
WS (T_{i-1})
Spike Detection (T_i)
WS (T_i)
Spike Detection (T_{i+1})
WS (T_{i+1})
1/fs
Software
Version 1: Tags not stored
Closed-loop (no tags) (T_i)
Version 2: Tags stored
Case A: sustainable closed-loop, ~150 spikes/iteration
Case B: worst-case latency, 4096 simultaneous spikes
Closed-loop (sustainable case) (T_i)
Closed-loop (worst case) (T_i)
Total Latency

Fig. 8: Timing diagram representing the scheduling of the processing tasks. The time duration of each task was derived from HDL simulations and successively confirmed from direct measurements on a system prototype.

exploit emerging generations of planar or implantable CMOS-MEAs with a large number of recording electrodes.

References

[1] J. Pine, "A history of mea development," *Advances in network electrophysiology*, pp. 3–23, 2006.

[2] M. E. J. Obien, W. Gong, U. Frey, and D. J. Bakkum, *CMOS-Based High-Density Microelectrode Arrays: Technology and Applications*. Singapore: Springer Singapore, 2017, pp. 3–39.

[3] H. Oka, K. Shimono, R. Ogawa, H. Sugihara, and M. Taketani, "A new planar multielectrode array for extracellular recording: application to hippocampal acute slice," *Journal of neuroscience methods*, vol. 93, no. 1, pp. 61–67, 1999.

[4] C. Sekirnjak, P. Hottowy, A. Sher, W. Dabrowski, A. Litke, and E. Chichilnisky, "Electrical stimulation of mammalian retinal ganglion cells with multielectrode arrays," *Journal of neurophysiology*, vol. 95, no. 6, pp. 3311–3327, 2006.

[5] D. A. Wagenaar, J. Pine, and S. M. Potter, "Effective parameters for stimulation of dissociated cultures using multi-electrode arrays," *Journal of neuroscience methods*, vol. 138, no. 1, pp. 27–37, 2004.

[6] S. M. Potter, A. El Hady, and E. E. Fetz, "Closed-loop neuroscience and neuroengineering," *Frontiers in neural circuits*, vol. 8, 2014.

[7] F. Franke, D. Jckel, J. Dragas, J. Müller, M. Radivojevic, D. Bakkum, and A. Hierlemann, "High-density microelectrode array recordings and real-time spike sorting for closed-loop experiments: an emerging technology to study neural plasticity," *Frontiers in Neural Circuits*, vol. 6, p. 105, 2012.

[8] M. E. J. Obien, K. Deligkaris, T. Bullmann, D. J. Bakkum, and U. Frey, "Revealing neuronal function through microelectrode array recordings," *Frontiers in Neuroscience*, vol. 8, p. 423, 2015.

[9] D. Tsai, E. John, T. Chari, R. Yuste, and K. Shepard, "High-channel-count, high-density microelectrode array for closed-loop investigation of neuronal networks," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 7510–7513.

[10] A. Maccione, M. Gandolfo, S. Zordan, H. Amin, S. D. Marco, T. Nieus, G. N. Angotzi, and L. Berdondini, "Microelectronics, bioinformatics and neurocomputation for massive neuronal recordings in brain circuits with large scale multielectrode array probes," *Brain Research Bulletin*, vol. 119, Part B, pp. 118 – 126, 2015, advances in electrophysiological data analysis.

[11] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, p. nature24636, 2017.

[12] G. N. Angotzi, M. Malerba, S. Zucca, and L. Berdondini, "A 512-channels, whole array readout, cmos implantable probe for acute recordings from the brain," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 877–880.

[13] C. E. Bouton, A. Shaikhouni, N. V. Annetta, M. A. Bockbrader, D. A. Friedenberg, D. M. Nielson, G. Sharma, P. B. Sederberg, B. C. Glenn, W. J. Mysiw *et al.*, "Restoring cortical control of functional movement in a human with quadriplegia," *Nature*, vol. 533, no. 7602, pp. 247–250, 2016.

[14] F. Boi, M. Semprini, F. A. M. Ivaldi, S. Panzeri, and A. Vato, "A bidirectional brain-machine interface connecting alert rodents to a dynamical system," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 51–54.

[15] J. E. O'Doherty, M. Lebedev, T. L. Hanson, N. Fitzsimmons, and M. A. Nicolelis, "A brain-machine interface instructed by direct intracortical microstimulation," *Frontiers in integrative neuroscience*, vol. 3, p. 20, 2009.

[16] R. C. Froemke and Y. Dan, "Spike-timing-dependent synaptic modification induced by natural spike trains," *Nature*, vol. 416, no. 6879, pp. 433–438, 2002.

[17] H. Kassiri, S. Tonekaboni, M. T. Salam, N. Soltani, K. Abdelhalim, J. L. P. Velazquez, and R. Genov, "Closed-loop neurostimulators: A survey and a seizure-predicting design example for intractable epilepsy treatment," *IEEE Transactions on Biomedical Circuits and Systems*, 2017.

[18] J. Wright, V. G. Macefield, A. van Schaik, and J. C. Tapson, "A review of control strategies in closed-loop neuroprosthetic systems," *Frontiers in neuroscience*, vol. 10, 2016.

[19] A. Novellino, P. D'Angelo, L. Cozzi, M. Chiappalone, V. Sanguineti, and S. Martinoia, "Connecting neurons to a mobile robot: an in vitro bidirectional neural interface," *Computational Intelligence and Neuroscience*, vol. 2007, 2007.

[20] C. Zrenner, D. Eytan, A. Wallach, P. Thier, and S. Marom, "A generic framework for real-time multi-channel neuronal signal analysis, telemetry control, and sub-millisecond latency feedback generation," *Frontiers in neuroscience*, vol. 4, 2010.

[21] A. Wallach, D. Eytan, A. Gal, C. Zrenner, and S. Marom, "Neuronal response clamp," *Frontiers in neuroengineering*, vol. 4, 2011.

[22] J. P. Newman, R. Zeller-Townson, M.-F. Fong, S. Arcot Desai, R. E. Gross, and S. M. Potter, "Closed-loop, multichannel experimentation using the open-source neurorighter electrophysiology platform," *Frontiers in neural circuits*, vol. 6, p. 98, 2013.

[23] S. Hafizovic, F. Heer, T. Ugniwenko, U. Frey, A. Blau, C. Ziegler, and A. Hierlemann, "A cmos-based microelectrode array for interaction with neuronal cultures," *Journal of Neuroscience Methods*, vol. 164, no. 1, pp. 93 – 106, 2007.

[24] K. Imfeld, A. Maccione, M. Gandolfo, S. Martinoia, P.-A. Farine, M. Koudelka-Hep, and L. Berdondini, "Real-time signal processing for high-density microelectrode array systems," *International Journal of Adaptive Control and Signal Processing*, vol. 23, no. 11, pp. 983–998, 2009.

[25] E. Biffi, D. Ghezzi, A. Pedrocchi, and G. Ferrigno, "Development and validation of a spike detection and classification algorithm aimed at implementation on hardware devices," *Computational Intelligence and Neuroscience*, vol. 2010, 2010.

[26] J. Müller, D. Bakkum, and A. Hierlemann, "Sub-millisecond closed-loop feedback stimulation between arbitrary sets of individual neurons," *Frontiers in Neural Circuits*, vol. 6, p. 121, 2013.

[27] J. Park, G. Kim, and S.-D. Jung, "A 128-channel fpga based real-time spike-sorting bidirectional closed-loop neural interface system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2017.

[28] S. Venkatraman, K. Elkabany, J. D. Long, Y. Yao, and J. M. Carmena, "A system for neural recording and closed-loop intracortical microstimulation in awake rodents," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 1, pp. 15–22, 2009.

[29] G. N. Angotzi, F. Boi, S. Zordan, A. Bonfanti, and A. Vato, "A

programmable closed-loop recording and stimulating wireless system for behaving small laboratory animals," *Scientific reports*, vol. 4, 2014.

[30] P. Cong, P. Karande, J. Landes, R. Corey, S. Stanslaski, W. Santa, R. Jensen, F. Pape, D. Moran, and T. Denison, "A 32-channel modular bi-directional neural interface system with embedded dsp for closed-loop operation," in *European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014-40th*. IEEE, 2014, pp. 99–102.

[31] X. Liu, M. Zhang, A. G. Richardson, T. H. Lucas, and J. Van der Spiegel, "Design of a closed-loop, bidirectional brain machine interface system with energy efficient neural feature extraction and pid control," *IEEE transactions on biomedical circuits and systems*, 2017.

[32] L. Grosenick, J. H. Marshel, and K. Deisseroth, "Closed-loop and activity-guided optogenetic control," *Neuron*, vol. 86, no. 1, pp. 106–139, 2015.

[33] C. DiMattina and K. Zhang, "Adaptive stimulus optimization for sensory systems neuroscience," *Frontiers in neural circuits*, vol. 7, 2013.

[34] T. Gollisch and A. V. Herz, "The iso-response method: measuring neuronal stimulus integration with closed-loop experiments," *Frontiers in neural circuits*, vol. 6, 2012.

[35] S. Gibson, J. W. Judy, and D. Marković, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal processing magazine*, vol. 29, no. 1, pp. 124–143, 2012.

[36] G. P. Seu, G. N. Angotzi, G. Tuveri, L. Raffo, L. Berdondini, A. Maccione, and P. Meloni, "On-fpga real-time processing of biological signals from high-density meas: a design space exploration," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*. IEEE, 2017, pp. 175–183.

[37] H. G. Rey, C. Pedreira, and R. Q. Quiroga, "Past, present and future of spike sorting techniques," *Brain research bulletin*, vol. 119, pp. 106–117, 2015.

[38] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.

[39] G. Hilgen, S. Pirmoradian, D. Pamplona, P. Kornprobst, B. Cessac, M. H. Hennig, and E. Sernagor, "Pan-retinal characterisation of light responses from ganglion cells in the developing mouse retina," *Scientific reports*, vol. 7, p. 42330, 2017.

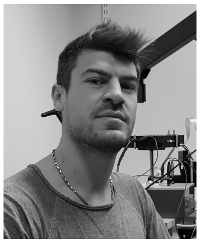
[40] J.-J. Pang, F. Gao, and S. M. Wu, "Light-evoked excitatory and inhibitory synaptic inputs to on and off α ganglion cells in the mouse retina," *Journal of Neuroscience*, vol. 23, no. 14, pp. 6063–6073, 2003.

[41] J. P. Seymour, F. Wu, K. D. Wise, and E. Yoon, "State-of-the-art mems and microsystem tools for brain research," *Microsystems & Nanoengineering*, vol. 3, p. 16066, 2017.

[42] J. J. Harris, R. Jolivet, and D. Attwell, "Synaptic energy use and supply," *Neuron*, vol. 75, no. 5, pp. 762–777, 2012.

Giovanni Pietro Seu received the B.Sc. and M.Sc. degree in electronics engineering from the University of Cagliari, Italy, in 2013 and 2015, respectively. In 2015 he started the joint-Ph.D. program in Bioengineering and Bioelectronics from the University of Genoa, Italy and University of Cagliari, Italy. His research interests mainly involve the development of real-time elaboration systems for biomedical applications. For his work he is currently collaborating with the Istituto Italiano di Tecnologia (IIT), Genoa, Italy and 3Brain AG, Wädenswil, Switzerland.

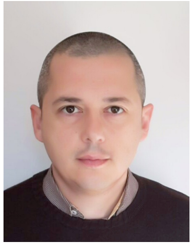
Gian Nicola Angotzi received the M.Sc. degree in electronic engineering at the University of Cagliari, in 2003 and the Ph.D. degree in electronic and computer science in 2007 from the same University. He is now researcher technologist at the Istituto Italiano di Tecnologia. His research is focused on the design of electronic front-ends for neural recording systems. He has a strong background on the conception, design, realization and validation of Very Large Scale Integration (VLSI) Circuits.



Fabio Boi received the M.Sc. degree in Robotics Engineering at the University of Genoa, in 2012 and the Ph.D. degree in Robotics, Cognition and Interaction Technologies in 2016 from the same University. He is currently a post-doc at Istituto Italiano di Tecnologia. His research activity is focused on neural recordings and data analysis of signals recorded from the central nervous system. He has a strong background on programming languages, electrophysiological signals and surgical procedures.

Luigi Raffo is full professor of Electronics at the Department of Electrical and Electronic Engineering - University of Cagliari (ITALY). He received the laurea degree in Electronic Engineering at University of Genoa (ITALY) in 1989, the PhD degree in Electronics and Computer Science at the same university in 1994. In 1994 he joined the Department of Electrical and Electronic Engineering of University of Cagliari (ITALY) as assistant professor, in 1998 as associate professor and from 2006 as full professor of electronics. He teaches courses on system/digital and analog electronic design and processor architectures for the Courses of studies in Electronic and Biomedical Engineering. He was coordinator of the project EU IST- FET IST-2001-39266 BEST and he was unit coordinator of the project EU IST-FET - SHAPES Scalable Software Hardware Architecture Platform for Embedded Systems. He has been local coordinator of industrial projects in the field (among others: ST-Microelectronics - Extension of ST200 architecture for ARM binary compatibility, ST-Microelectronics - Network on chip). He is responsible for cooperation programs in the field of embedded systems with several other European Universities. He was coordinator of the MADNESS EU Project (FP7/2007-2013) and local coordinator in the ASAM (ARTEMIS-JU) and ALBA projects (national founded project) and RPCT (regional founded project).

Luca Berdondini is currently a Senior Researcher leading the IIT-NetS3 laboratory at the Dpt. of Neuroscience and Brain Technologies of the Istituto Italiano di Tecnologia. He received in 1999 a M.Sc. degree in microengineering from the Swiss Federal Institute of Technology of Lausanne (EPFL) with a Master Thesis at Caltech (USA) and in 2003 a PhD on nano-/micro-fabricated interfaces and CMOS devices for electrophysiology (Samlab, EPFL, Switzerland). His research team focuses on the development and application of innovative neuroelectronic solutions to measure and analyze electrical activity at cellular resolution in large neuronal assemblies and brain circuits. He is among the pioneers of CMOS-based multielectrode arrays for electrophysiology, and cofounder of 3Brain GmbH (Landquart, Switzerland).



Paolo Meloni is currently assistant professor at the Department of Electrical and Electronic Engineering (DIEE) in the University of Cagliari. In October 2007 he received a Ph.D. in Electronic Engineering and Computer Science, presenting the thesis Design and optimization techniques for VLSI network on chip architectures. His research activity is mainly focused on the development of advanced digital systems, with special emphasis on the application-driven design of multi-core on-chip architectures. He is author of a significant record of international research papers and tutor of many bachelor and master students thesis in Electronic Engineering. He is teaching the course of Embedded Systems at University of Cagliari.