




Article

FPGA Design Integration of a 32-Microelectrodes Low-Latency Spike Detector in a Commercial System for Intracortical Recordings

Mattia Tambaro ^{1,2,*} , Marta Bisio ², Marta Maschietto ², Alessandro Leparulo ²  and Stefano Vassanelli ^{1,2} 

¹ Padova Neuroscience Center, University of Padua, 35131 Padua, Italy; stefano.vassanelli@unipd.it

² Department of Biomedical Sciences, University of Padua, 35131 Padua, Italy; marta.bisio@unipd.it (M.B.); marta.maschietto@unipd.it (M.M.); alessandro.leparulo@unipd.it (A.L.)

* Correspondence: Mattia.Tambaro@PhD.Unipd.it

Abstract: Numerous experiments require low latencies in the detection and processing of the neural brain activity to be feasible, in the order of a few milliseconds from action to reaction. In this paper, a design for sub-millisecond detection and communication of the spiking activity detected by an array of 32 intracortical microelectrodes is presented, exploiting the real-time processing provided by Field Programmable Gate Array (FPGA). The design is embedded in the commercially available RHS stimulation/recording controller from Intan Technologies, that allows recording intracortical signals and performing IntraCortical MicroStimulation (ICMS). The Spike Detector (SD) is based on the Smoothed Nonlinear Energy Operator (SNEO) and includes a novel approach to estimate an RMS-based firing-rate-independent threshold, that can be tuned to fine detect both the single Action Potential (AP) and Multi Unit Activity (MUA). A low-latency SD together with the ICMS capability, creates a powerful tool for Brain-Computer-Interface (BCI) closed-loop experiments relying on the neuronal activity-dependent stimulation. The design also includes: A third order Butterworth high-pass IIR filter and a Savitzky-Golay polynomial fitting; a privileged fast USB connection to stream the detected spikes to a host computer and a sub-milliseconds latency Universal Asynchronous Receiver-Transmitter (UART) protocol communication to send detections and receive ICMS triggers. The source code and the instruction of the project can be found on GitHub.

Keywords: digital signal processing; field programmable gate array; spike detection; action potential; multi-unit activity; multi electrode array; brain computer interface; neuroscience



Citation: Tambaro, M.; Bisio, M.; Maschietto, M.; Leparulo, A.; Vassanelli, S. FPGA Design Integration of a 32-Microelectrodes Low-Latency Spike Detector in a Commercial System for Intracortical Recordings. *Digital* **2021**, *1*, 34–53. <https://doi.org/10.3390/digital1010003>

Received: 20 December 2020

Accepted: 25 January 2021

Published: 30 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Intracortical microelectrode arrays enable the possibility to read the small amplitude signals—microvolts to millivolts range—of one of the most complex systems: The brain. Each electrode, which is only a few micrometers in width, allows reading the extracellular depolarization from a population of neurons in the proximity of its metallic surface [1–4]. Usually, multiple microelectrodes are arranged on a probe with one or more shanks, with a channel count increased up to hundreds or even thousands [5–7]. These microelectrodes can be also used to inject a variable current or voltage to evoke a physiological response from neurons in a sphere centered on the stimulating channels [8,9]. In order to process the information that relies behind the voltage fluctuations detected by the sensing spots on the probes, the acquired signal must be digitally sampled with a frequency high enough to discern the activity up to the fast depolarization of a single neuron, called Action Potential (AP) or spike, and Multi Unit Activity (MUA). MUA is the faster cumulative and overlapping signal propagated by multiple neurons close enough to the electrode to be sensed but too far to be singularly distinguished [10]. To accurately reconstruct this activity, as short as 1 ms, the samples must be collected at a sampling frequency that can easily go up to 30–40 kHz. Given the large number of channels featured by the most recent

probes [4,11,12], the number of samples generated per second can be difficult to process in real-time, precluding experiments where the latency in the activity events recognition is crucial. For example, when dealing with spike dependent closed-loop experiments for brain computer interfacing involving IntraCortical MicroStimulation (ICMS) [13,14], keeping the feedback timing in a biologically acceptable range can be a critical step [15–17]. Traditional processing of the information is subjected to the USB communication latency necessary to transfer the samples in chunks from the acquisition system to a computer, where the samples processing is in turn conditioned by the operating system scheduler, blocking calls and computer performances. Altogether, these constraints cause tens of milliseconds of delay from the sampling of the neural activity to its elaboration and interpretation.

This paper aims to overcome these limitations presenting a digital Spike Detector (SD) running in the recording system's FPGA, that elaborates the neural activity received directly from the headstage, which samples the signals acquired by the microelectrodes without intermediate steps. The headstage, mounted on the top of the probe, embeds up to two chips each managing 16 electrodes and including a Low Noise Amplifier (LNA), a 16-bit Analog-to-Digital Converter (ADC), high-pass and low-pass filters, and stimulation circuitry. Processing the data in-place on the recording system allows evaluating the digitalized signal point-by-point with the lowest latency from the ADC, and communicating the temporal and spatial firing information of each of the 32 channels in less than 1 ms from the maximum negative peak of the spikes. The project has been integrated in the open-source Hardware Description Language (HDL) code and acquisition software RhythmStim from Intan Technologies (Los Angeles, California, USA) and deployed on the RHS stimulation/recording controller (Figure 1) from the same company. The logic managing the sampling and data transmission has been added aside, creating a new dedicated pipeline to process the raw data stream acquired by the 32 channels headstage without reducing the original functionalities. The spike detection pipeline starts without any buffering system when receiving a new sample, guaranteeing the lowest latency from digitalization to processing. The SD presented in this work wants to improve the method included in the system by default, that works in detecting when the first order high-pass filtered signal crosses a threshold set by the user on a subset of a monitored channel, raising the signal from one of the mapped digital output of the system. This work aims to improve the detection accuracy of the system, despite the good noise performance at the high frequencies, to detect both APs and MUAs. Our experiments found the instrument noise at the high frequency of AP and MUA to be about 10 μV RMS during neuronal inactivity, that makes it trivial to detect APs that can reach amplitudes down to $-150 \mu\text{V}$ when neurons are close to the electrode surface (see Section 2.1.1 and Result). Unfortunately, in many experiments, these strong signals represent a minority, most of the signals originate from neurons that are far from the electrodes, leading to MUAs of about 15 μV RMS and a Signal-to-Noise Ratio (SNR) of 3.5 dB once the signal is filtered (see Results section). Hence, the need for a more powerful tool which is able to accurately detect both the AP and MUA events. The presented SD design includes:

- A third order Butterworth high-pass Infinite Input Response (IIR) filter with a cutoff frequency of 300 Hz and a Savitzky-Golay smoothing filter [18], fitting the high-pass filtered signal with a second degree polynomial.
- A point-by-point signal energy estimation using the SNEO [19], that aggregates both the frequency and the amplitude of the voltage fluctuations in a series of samples that can be compared with a threshold.
- A novel dynamic threshold per-channel estimation based on the Root Mean Square (RMS) of the SNEO output that accommodates any drift of the probe or changes in the signal quality, that rapidly converges to a firing-independent value close to an ideal threshold based only on the noise energy. The threshold can be adjusted by the user, to open the detection of both the single AP and MUA.

- A post-ICMS blind window of a user-defined duration, to prevent the detection of false positives following the high increase in the signal power due to the artifact induced by the injection of a current.
- The capability to stream the amplitude, the channel, and the timing of each detected spikes via the USB to the host computer, and, simultaneously, via the UART protocol to any other connected device with sub-milliseconds latency. By the UART command, it is also possible to trigger the ICMS set from the host application.



Figure 1. The Intan Technologies RHS stimulation/recording controller, where the Spike Detector (SD) has been deployed. It supports the simultaneous recording of up to 4 headstages of 32 channels each with a sampling frequency up to 30 kHz per channel. The IntraCortical MicroStimulation (ICMS) of different amplitudes and duration per phase can be applied.

The software application on the host computer used to manage the connection, the recording, and the setting of the stimulation/recording controller system has been extended to manage the new functionalities. A new dedicated window to manage the SD has been created to show a real-time raster plot of the spiking activity of each channel and to allow adjusting the threshold multiplier and the artifact blinding duration during the experiment to improve the SD performances. Through the application, the system can be also connected to the network to stream the spikes details to a specified IP address over the UDP protocol. Finally, channels can be excluded from the spike detection so that they are not sent via both UART and UDP. The system has been tested with *in vivo* experiments on a rat barrel cortex and it is currently used in our laboratory for closed-loop experiments of ICMS. Examples of the system output on APs and MUA detection are highlighted in the Results section.

2. Materials and Methods

2.1. Design Algorithms

The digital implementation of the SD relies on an algorithmic processing that can be split in six major steps: High-pass filtering, Savitzky-Golay fitting, SNEO, threshold estimation, detection, and local minimum finder. These core parts are here described, together with the parameters used for an implementation running with a sampling frequency of 25 kHz for each of the 32 channels. The input is assumed as a matrix where the first index indicates the electrode number, or channel, while the second index reports the sampled amplitude of the signal. The information to extract from the signals is given by the index where the spiking events are detected for each channel and the corresponding amplitude. The output then is a sparse matrix with the input shape, containing the negative amplitude of the filtered signal where a spike is detected. The parameters—length of the Savitzky-Golay window, k of SNEO and the threshold multiplier—are tuned both by simulation and experimentally to achieve the better accuracy. Synthetic datasets are created as shown in Figure 3, with spikes templates of different amplitudes and SNRs inserted in the known positions. In this case, the accuracy focuses on the single AP detection and is calculated from the ratio between the correctly detected events and the sum of the total events and the false positives. The results achieved varying the parameters of the algorithm are then compared to maximize the accuracy. The analysis is not shown here for the sake of brevity, but the method follows what was reported in a previous work [20]. Acquisitions from

experiments are then used to fine tune what has been found by the simulation, where it is possible to evaluate also the MUA detection.

2.1.1. High-Pass and Savitzky-Golay Filtering

When dealing with signals such as APs and MUA characterized by high frequency features, the first step of the processing chain is to high-pass filter the signal, to exclude the Local Field Potential (LFP) from the further analysis, to remove the predominant low frequency components of the signal below 300 Hz. Here, a third order IIR Butterworth filter is used, to remove the low frequency changing features of the LFP, offset, and off-band noise. The filter coefficients in Table 1 are computed using MATLAB and are used to implement the IIR filter with a cutoff frequency of 300 Hz. The filtering step is immediately followed by the smoothing Savitzky-Golay filter, used to fit the last seven points of the high-pass filtered signal with a second order polynomial at each iteration. The signal is convoluted by the seven coefficients mask shown in Table 1, providing a cutoff frequency of -3 dB at 3 kHz. This results in less sharp high frequency changes, increasing the SNEO performances as demonstrated in [21]. A fitting example of an extracellular signal is shown in Figure 2.

Table 1. Coefficients for the third order Infinite Input Response (IIR) filter (a and b), and for the Savitzky-Golay (SG) fitting.

Mask	0	1	2	3	4	5	6
a	0.9274	-2.7821	2.7821	-0.9274	\	\	\
b	1	-2.8492	2.7096	-0.8600	\	\	\
SG	-0.0952	0.1429	0.2857	0.3333	0.2857	0.1429	-0.0952

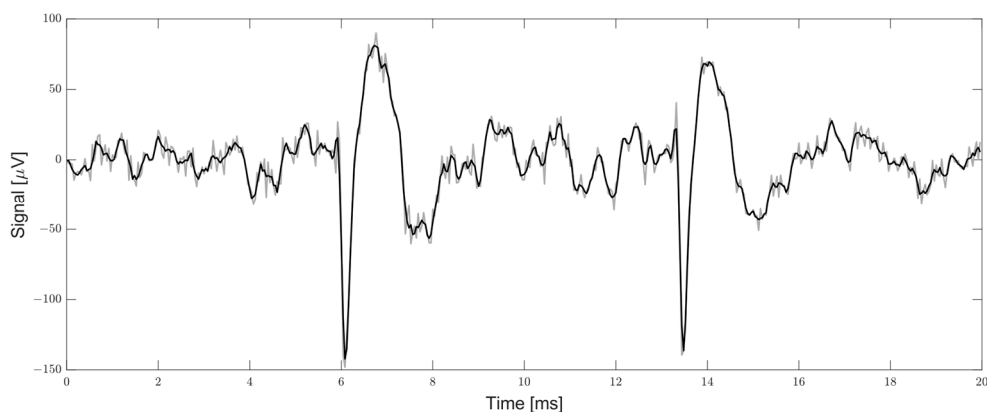


Figure 2. A 20 ms acquisition of extracellular signal from the rat's barrel cortex presenting two strong extracellular spikes recorded at 6 and 14 ms, propagated from a neuron near the recording electrode. In gray, the high-pass filtered signal with a cutoff frequency of 300 Hz. In black, the signal fitted with the Savitzky-Golay filter.

2.1.2. Smoothed Nonlinear Energy Operator and Threshold

The k NEO, on which the SNEO is based, gives an instantaneous estimation of the energy contained in a signal oscillation. Thanks to the energy dependent answer, the SNEO became widely used for the spike detection task and is still one of the preferred spike detector operators when going on real-time processing [22–25]. It fits very well to a real-time and low-latency application since it does not require any training phase and has a small footprint on the resource usage nevertheless the good performances, as proved and compared in [20]. This operator strictly relies on a parameter called k , which depends on the sampling frequency and the average spike duration. Given the sampling frequency of 25 kHz, the average spike duration, and the experimental tuning, the k parameter value achieving the best accuracy results in detection has been found to be 4. After computing

the nonlinear energy response of the filtered signal s of each channel ch , the SNEO makes it smoother with a Bartlett window BW of a size $4 \cdot k + 1$ (Equation (1)).

$$\begin{aligned} kNEO(ch, t) &= s(ch, t)^2 - s(ch, t+k) \cdot s(ch, t-k) \quad \forall t : t \in \{1, \dots, \text{signal length}\} \\ SNEO &= kNEO \otimes BW(4 \cdot k + 1) \end{aligned} \quad (1)$$

The result returned by the application of the SNEO is now compared with a pre-defined threshold which must be set to a value accurate enough to discern the physiological events from the signal noise. A fixed value for the threshold can sensibly reduce the detection performances, since each channel has its own Signal-to-Noise Ratio (SNR) level that can be also subjected to changes caused by probe drifts. A threshold value based on the SNEO output distribution can adaptively adjust the detection level of each individual channel for the entire duration of the recording. The median is the most reliable option to find an effective threshold using a scaling value, but unfortunately it does not fit well in a real-time hardware implementation pipeline. For this reason, this work relies on the RMS of the signal, multiplied by a scaling factor C to find an accurate threshold which is constantly updated over a window of samples—here called timeframe, tf —long enough to provide a correct and stable trend. However, when using the RMS to find the threshold, it must be considered that the SNEO is a nonlinear operator and the high SNEO energy response to a spike will highly affect the RMS value. Being the firing rate intrinsically non-constant, the RMS output can be sensibly affected with different intensities from timeframe to timeframe and therefore, it cannot be simply accounted in reducing the threshold multiplier C by a constant fraction. To mitigate this problem, this paper presents a new approach for the threshold estimation. The signal in each timeframe is compared with the threshold based on the RMS of the prior timeframe. Therefore, the new RMS value can be computed for every timeframe by replacing the samples exceeding the threshold with the RMS from the previous timeframe, to exclude the contribution of the spikes energy from the estimation. Using this approach, the RMS usually converges in a maximum of five iterations to a value close to the RMS value of the noise energy only, independently from the spike amplitude and firing rate, as shown in Figure 3. According to this, stimulation artifacts, which could increase the threshold far over the correct level, are also rejected from the RMS estimation. The threshold multiplier C , scaling the RMS value, is chosen by experimental results: A value about 5.5 is used for the MUA detection and a greater value up to 18 is used to detect a clear single AP.

$$Th(ch, tf) = C * RMS(s(ch, t)) \quad \forall t : t \in tf \quad (2)$$

2.1.3. Spike Detection and Local Minimum Finder

For each new sample, the SNEO output is computed and compared with the two previous samples searching for a local maximum, and with the threshold found in the previous RMS timeframe (Equation (3)). It must be considered that a peak in the SNEO output can be not precisely aligned with the peak in the filtered signal. In fact, depending also on the filtering [26], the positive rebound following the negative spike peak can reach a prominent amplitude in the transition from the negative and the positive phase of the spike, developing enough energy to shift the SNEO output peak (Figure 4). Therefore, if the output is a local maximum and is also greater than the threshold, the algorithm searches for the minimum value among the previous samples of the filtered signal, within a window wide as the Bartlett window, which is used to smooth the SNEO. Once the minimum is found, the index will represent the spike temporal information, while the value will be the spike amplitude (Equation (4)).

$$Peaks(ch, t) = True \text{ if } \begin{cases} SNEO(ch, t-1) \geq Th(ch, tf), \text{ where } tf \ni t \\ SNEO(ch, t) \leq SNEO(ch, t-1) > SNEO(ch, t-2) \end{cases} \quad (3)$$

$$Spike(ch, t') = \min(s(ch, t')) \quad \forall t' : t - 4 * k + 1 \leq t' \leq t \ \& \ Peaks(ch, t) = True \quad (4)$$

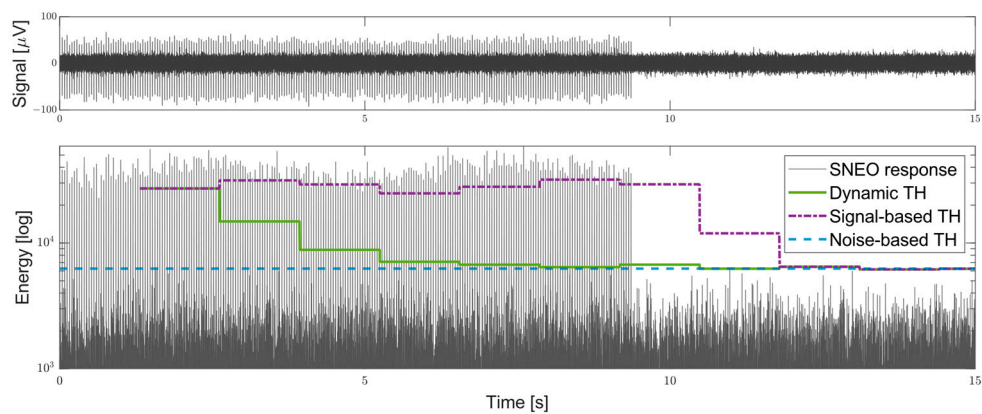


Figure 3. SD on 15 s synthetic dataset created from an ex vivo extracellular recording. Top: The ex vivo recording is high-pass filtered at 300 Hz. It does not contain any neuronal activity, thus well representing the instrumental noise floor. A spike template with a negative amplitude of $\sim 80 \mu\text{V}$ is generated from the average spikes shape propagated by a neuron nearby the electrode from a filtered in vivo experiment. Twenty-five templates per second are added with a uniform distribution to the ex vivo noise, for a duration of 9.4 s. Bottom: The energy response of the Smoothed Nonlinear Energy Operator (SNEO) is shown in gray. The dashed blue line is the threshold based on the Root Mean Square (RMS) of the filtered ex-vivo energy, representing the value that a spike-independent threshold must ideally have. The solid green line represents the dynamic threshold presented in this work. In comparison, the dash-dotted purple line is the threshold computed on the same timeframes but from the RMS of the entire signal: It is evident how it is largely affected by the firing rate. After the first timeframe where the threshold is not yet computed, a first value is estimated and after four steps the threshold converges to a stable value close to the ideal given by the dash-dotted line.

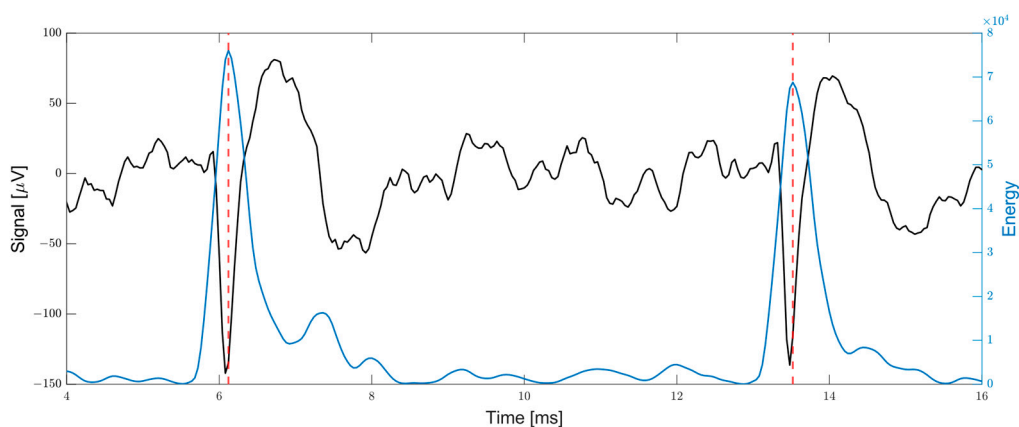


Figure 4. An example of two action potentials (APs) recorded from an in vivo experiment. In black, the signal is fitted by the Savitzky-Golay filter. In blue, the energy is computed by the SNEO. The two vertical red dashed lines highlight the peaks shift between the spike and the energy.

2.2. Design Implementation

The algorithms presented in Section 2.1 have been described in an HDL language to be implemented on the Opal Kelly XEM6010-LX45 board mounting the Xilinx Spartan-6 XC6SLX45-2 FPGA. This board is mounted by default by the Intan RHS stimulation/recording controller and together with the RhythmStim design provides the interface between a host computer and the headstages (up to 4). Each headstage is connected through a Serial Peripheral Interface (SPI) cable to the recording system, used to communicate simultaneously two digitalized samples. The RhythmStim allows acquiring the data for both visualization and recording on the host computer application. The application is used to set the filters on the headstages and to program the latter to stimulate a channel with a user-defined current shape based on different external trigger sources connected to the

recording system. The RhythmStim hardware part manages the samples acquisition from each channel at a constant frequency of 20, 25, or 30 kHz and the electrical ICMS when it is triggered. In a high-level overview, the system continuously loops over 20 action slots: The last four slots are used to start and stop the ICMS and to set the parameters of the headstages, while in the other first 16 slots, a 32-bit command is sent via SPI to sample a different pair of channels. The headstage answers after a delay of two slots with a pair of values of 32-bit, each containing in the 16 Most Significant Bits (MSBs) the sampled value, stored with a positive offset of 32,768, where the least significant bit of the value corresponds to 0.195 μV . The pair is written in the RAM of the Opal Kelly board (Portland, Oregon, USA), which is used as a buffer for the asynchronous USB communication with the host application. The design has been modified to write the 16 MSBs of both the samples pair, received from the headstage connected to the system port D, also in a 64-bit width First In First Out (FIFO), filling the other 32 bits with a system timestamp representing the number of samples acquired. The FIFO crosses the domain clock and provides to the SD the values at the working frequency of 100 MHz. Here, the SD pipeline begins, whose main steps of the process are described in Figure 5.

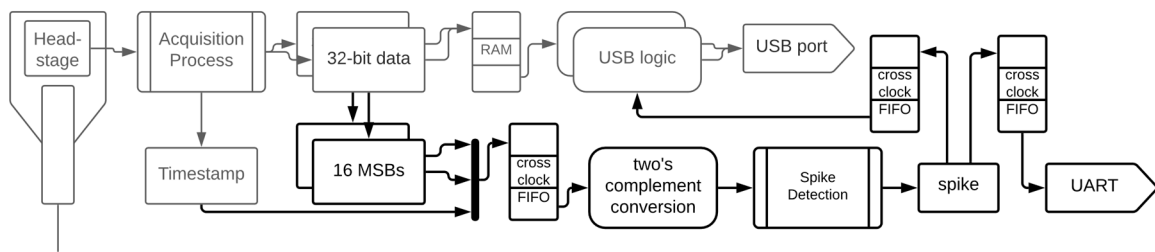


Figure 5. Block diagram of the system; in gray the existing parts, in black the pipeline added for the Spike Detector (SD).

The values are read as soon as the FIFO is not empty, and the two samples are converted in two's complement removing the positive offset of 32,768. The SD design is split into different HDL components described in the next subsections. These components are orchestrated by a main design that allows the various output to flow from one component to the next, keeping the channel and the samples count. The samples pair is processed sequentially through the pipeline shown in Figure 6, one sample immediately after the other. The processing is tuned for a sampling frequency of 25 kHz. The SD can work also with a 16 channels headstage, since the second value of each samples pair will simply have a constant value of 0. As the main design rules, the operations are performed in the integer domain. Where a division is required, it is constrained to a power of two, to be performed as a bit-shifting operation—a 0.5 rounding precision is achieved performing the n bit-shifting operation on the value summed to 2^{n-1} . All the components work in time division multiplexing over the 32 channels, storing each channel state. The HDL part of the project has been developed on the Xilinx ISE Design Suite 14.7. The pseudocode of the components can be found in Appendix A.

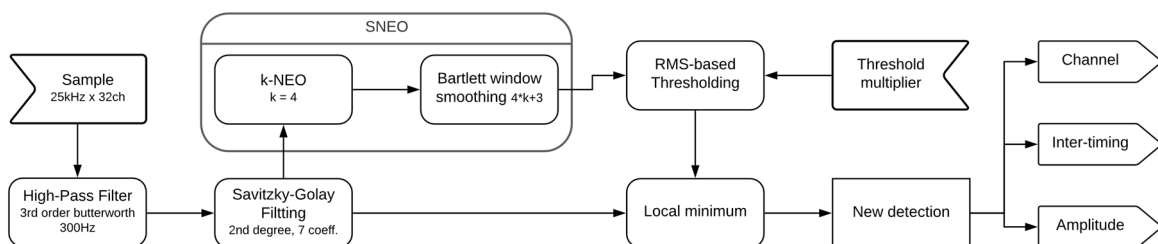


Figure 6. Spike detection pipeline.

2.2.1. Filter

Direct and direct-transposed from I and II have been simulated for the FPGA integer-based implementation of the third order IIR filter. Using the direct-transposed form I (Figure 7), the state variables of the filter did not saturate, and the filter remained stable despite the rounding error on the output, avoiding using a cascade of second order filters. Being the coefficients real numbers close to the 0 value, the error introduced by the rounding would be disastrous. To overcome this problem working in the integer domain, the coefficients are scaled by a factor big enough to reduce the rounding error to a negligible one. This scaling factor is chosen as a power of 2, to be then removed from the filter output with a bit-shift operation. The filter form fits well to the DSP48A1 multiplier/adder of the Spartan-6, that provides a signed multiplication of 18-bit per operand, followed by a 48-bit adder. Two DSPs are used to create a 35×18 -bit fully pipelined multiplier with a 48-bit adder after the multiplication stage. The six filter state variables of the 32 channels are stored in a 48-bit width BRAM, where the state variables of each channel are mapped. A scaling factor of 2^{15} is chosen to scale the filter coefficients to fit the 18-bit signed input of the multiplier being sure to not exceed the $\pm 2^{17}$ limit. In Table 2, the integer coefficients are shown with the percentage rounding error. For each new sample to filter, the sum of the 16-bit value of the new sample with the state $s_4(n)$ is tied in the input on the 35-bit multiplicand port. This value is reduced by the scaling factor and is resized to 35 bits without losing any information. The filter coefficients are given sequentially on the 18-bit multiplier port and the corresponding channel state variable is given on the 48-bit sum port to compute all the new state values. The filter output, reduced by the scaling factor, can be safely resized to 16 bits. Even if it is smaller on average compared to the input, the output bit width is kept the same since an eventual stimulation artifact can cause the signal to span on the entire range even after the filtering.

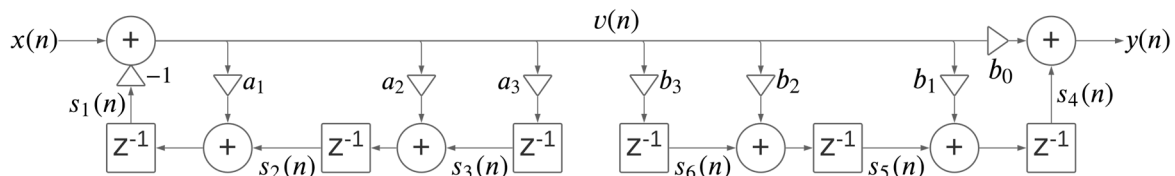


Figure 7. IIR filter direct-transposed form I structure.

Table 2. Coefficients for the third order IIR filter scaled by the scaling factor of 2^{15} and the percentage error introduced from the integer rounding.

	Coefficients				Rounding Error (%)			
	0	1	2	3	0	1	2	3
a	30.388	-91.163	91.163	-30.388	0.0021	0.0010	0.0010	0.0021
b	32.768	-93.364	88.789	-28.180	0	0.0009	0.0001	0.0008

2.2.2. Savitzky-Golay Fitting

The Savitzky-Golay fitting consists in a convolution of a mask of seven coefficients with the last seven samples of the channel. Similarly to the high-pass filter, the seven coefficients of the second grade polynomial fitting are multiplied by a scaling factor. The coefficients can safely be scaled by a factor of 2^{18} , without exceeding the signed range of an 18-bit word. The last seven values of each channel are stored in a 16-bit width BRAM. The last six samples are ordered in BRAM by channel (i.e., ch1-sample1, . . . , ch1-sample6, . . . , ch32-sample1, ch32-sample 6) to be linearly indexed. The convolution exploits the DSP48A1, configured in an 18×18 -bit multiply and accumulate mode. Each sample is extracted sequentially from the BRAM and is multiplied by its respective coefficient with the DSP multiplier, the result is then accumulated in the 48-bit register. Each sample, when read from the BRAM, is re-inserted as one position delayed—except for the sixth which is

discarded—and the new sample is inserted first, to achieve the behavior of a shift register. The accumulation register is reset to a value of 2^{17} before receiving any new samples, to account for the 0.5 rounding of the bit shift division used to remove the scale factor from the convolution output. The average error introduced by the rounding after the scaling is 0.0004%.

2.2.3. Smoothed Nonlinear Energy Operator

The operations of the SNEO component involve two DSP48A1 configured in a 35×18 -bit fully pipelined multiply and accumulate mode and a 35-bit width BRAM. The component receives three samples depending on the SNEO k parameter, to compute the new signal energy value of a 35-bit width as shown in the first row of Equation (1). Both the BRAM organization and convolution with the Bartlett window are implemented as in the Savitzky-Golay component, but with a convolution length of 17. The multiplier input of 35-bit is required due to the energy value width. The Bartlett window is scaled by a factor of 2^{16} , introducing an average error of 0.0025%. Therefore, the accumulation register starts from a value of 2^{15} for the bit shift 0.5 rounding.

2.2.4. Energy RMS and Threshold

This component provides a threshold based on the RMS estimation of the SNEO output to each channel. The RMS is computed incrementally, accumulating the square of the 32-bit energy values of a timeframe. To compute the square of the value, a 35×35 -bit multiplier based on a single DSP48A1 slice is used. The cumulative sum of the 32 channels is stored in an 85-bit width BRAM. The last thresholds found for each channel and the last 32 RMS values from the previous timeframe are also stored in two 35-bit width BRAM. This allows comparing the energy value with the channel threshold before accumulating it and, in case it is greater, to replace its value with the last RMS value. A counter keeps the number of samples processed and after a timeframe of 2^{15} samples, the square root of the accumulated squared sum is computed with the abacus algorithm, returning an integer approximation of the solution in a variable number of iterations. The result is multiplied by the user-selected multiplier and is stored in the thresholds BRAM.

2.2.5. Local Minimum Finder and Spike Output

Whenever a spike is detected, the minimum is found cycling through the last 17 samples of the filtered signal of the channel, stored in a 16-bit width BRAM, to find the maximal negative amplitude. The final timestamp is computed as the difference from the position of the minimum value and the current timestamp, while the amplitude is the minimum value. The timestamp, the amplitude, and the channel of the detected spike are inserted in two different FIFOs. The first is used to cross the clock domain to match the USB communication frequency. Amplitude (16-bit), channel number (8-bit), threshold multiplier (8-bit), and timestamp (32-bit) are written concatenated in the FIFO, which keeps the data count and, thanks to the different aspect-ratio, provides the data in a 16-bit word to the logic managing the USB data transfer. The data count helps a fast transfer of the data to the host PC, as explained in the next subsection. The second FIFO has an 8-bit width, and the timestamp (27-bit) channel number (5-bit), and amplitude (16-bit) of the spike are written in little-endian byte order. This FIFO is used to manage the data transfer via the UART protocol from a GPIO on the high-speed port on the back of the recording system. Data is read and transferred at 8-bit at a BAUD rate that can be customized in the design.

2.2.6. Software Integration

The open-source software of the RhythmStim for the host computer has been modified (with Qt 5.15.1) to support the new features introduced in the acquisition system. By default, the software manages the data transfer from the acquisition system to the computer thanks to the USB communication provided by the PipeOut construct, of the Opal Kelly board and manages the transfer of data via USB. On the HDL side, the PipeOut is instantiated

on a specific endpoint, an address, and it is interfaced with the read-side of a 16-bit width FIFO. At this point, the data contained in the FIFO can be retrieved with an API call on the host side, specifying the endpoint and the length of the transfer in bytes. A PipeOut is instantiated in the RhythmStim design connected to the Opal Kelly RAM to manage and buffer the recording transfer. A second PipeOut has been added to bypass the RAM buffering and to provide the spikes information with a lower latency. While the first PipeOut continues to stream the buffered data in big chunks, the software runs a secondary thread polling the new PipeOut for new data every millisecond. The FIFO connected to the PipeOut provides to the software the count of words that it contains. Whenever the software notices that this count is greater than 0, it reads from the PipeOut all the spikes contained in the FIFO. The words counter is given to the host by the WireOut construct of the Opal Kelly, that thanks to an endpoint exposes FPGA variables of 32 bits via USB through an API. The spikes acquired by the software can be streamed via the UDP protocol to any user-defined IP address. The packet is formed by four integers of 4 bytes in Big-Endian order which are, ordered in an array position from 0 to 3: Zero, timestamp, amplitude, and channel. If the acquisition system is saving data on the disk, the detected spikes are also stored in a file separated from the main record. Thanks to the WireIn constructs, used to show variables from the software to the FPGA, the user can control the digital SD through the software to run or stop it, to set the threshold multiplier (in 0.5 step), the blind window length (to stop the detection for a few ms after a stimulation command), and the channels whose outputs have to be active. These parameters are exposed to the user with a dedicated application window reachable from the main window of the RhythmStim application, also showing a real time raster plot of the spiking activity (Figure 8).



Figure 8. Graphical user interface. The window on the left shows the real time raster plot and allows tuning the parameter of the SD. The window on the right is the standard application window that shows the raw recording and manages the interface with the recording system.

2.3. Animals and Surgical Procedure

The experiments shown in the Results section are recorded from the barrel cortex of anesthetized rats belonging to the Wistar strain, which are maintained under standard environmental conditions in the animal research facility of the Department of Biomedical Sciences of the University of Padua. All the procedures are approved by the local Animal Care Committee (OPBA) and the Italian Ministry of Health (authorization num-

ber 522/2018-PR). Young adult P28–P32 rats (being P0 the day of birth) of both genders and with body weight between 90 and 120 g are anesthetized with an intra-peritoneal induction dose of Urethane (0.15/100 g body weight), followed by a single additional dose (0.015/100 g body weight). Each animal is positioned on a stereotaxic instrument and the head is fixed by teeth- and ear-bars. After a sub-cutaneous injection of Carprofen painkiller (Rimadyl; 10 μ L/100 g body weight), an anterior-posterior opening in the skin is made in the center of the head and a dedicated window in the skull is drilled over the somatosensory barrel cortex at stereotaxic coordinates $-1 \div -4$ AP, $+4 \div +8$ ML referred to as bregma [27,28], and the linear probe (E32+R-65-S1-L6 NT IrOx; Atlas Neuroengineering, Figure 9) is then inserted orthogonal to the cortical surface in the middle of the window at coordinates -2.5 AP, $+6$ ML. As a reference for the recording/stimulation, the depth is set at 0 when the electrode proximal to the chip tip touches the cortical surface. The neuronal activity is recorded by the 32 electrodes from the entire barrel cortex (from 0 to -1750 μ m), which is constantly bathed in Krebs' solution (in mM: NaCl 120, KCl 1.99, NaHCO₃ 25.56, KH₂PO₄ 136.09, CaCl₂ 2, MgSO₄ 1.2, glucose 11). An Ag/AgCl electrode bathed in the extracellular solution in proximity of the probe is used as a reference.

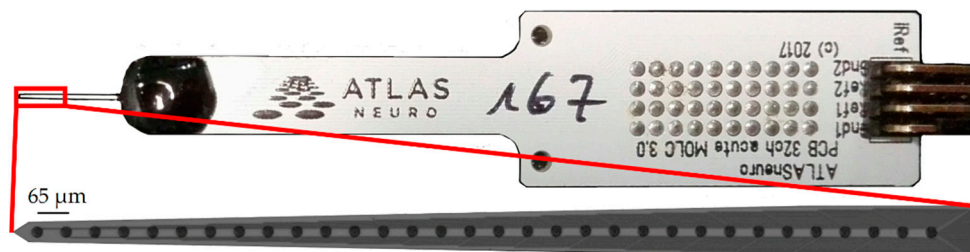


Figure 9. Atlas neuroengineering E32+R-65-S1-L6 NT probe. The magnification of the red box allows observing the 32 iridium oxide microelectrodes.

3. Results

The SD has been tested on both simulations and different in vivo experiments collected in our laboratory from the barrel cortex of six different rats. The experimental protocol consists of a 5 min recording where 28 cathodic to anodic (negative to positive) ICMSs of 50 μ A amplitude and 200 μ s duration per phase have been delivered every 10 s from the fifteenth electrode. Figure 10 shows the performances of the SD on 15 s around one of the given ICMS of a channel recording a strong spiking signal. The threshold multiplier is set to a value of 18 to focus on the detection of the single unit AP. Note that the first spikes are lost due to the threshold initialization. The threshold fluctuation depends on the MUAs energy around the APs, that being undetected (as expected), influences the RMS level. Anyway, the fluctuation is small compared to the energy developed by the spikes and the threshold remains closer to what would be the noise-based value, allowing for an accuracy greater than that achievable with the firing-dependent threshold. The accuracy achieved on the entire acquisition of 5 min is 92%, with 1369 spikes detected out of 1429 and 58 false negatives. The accuracy, once again, has been defined as the ratio between the correct detection and the sum of the total spikes and the false positives.

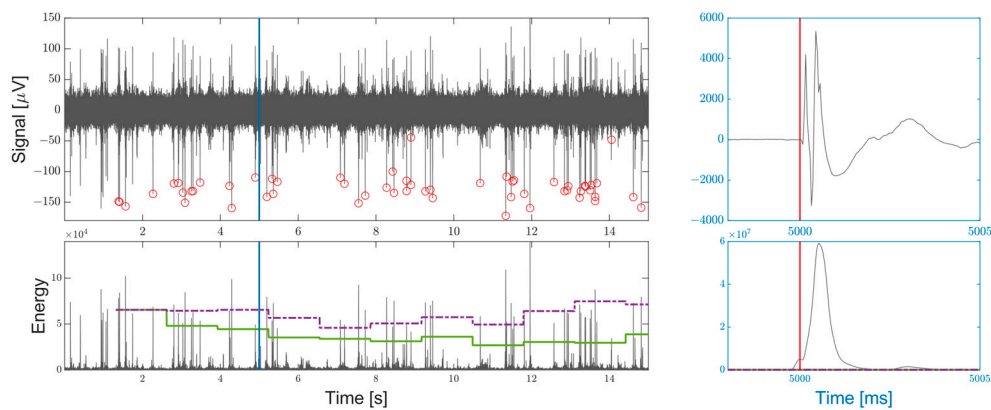


Figure 10. Spike detection of a single unit Action Potential (AP) activity on 15 s of acquisition. The vertical blue line corresponds to a cathodic to anodic ICMS of $\pm 50 \mu\text{A}$ of $200 \mu\text{s}$ per phase injected by a channel $520 \mu\text{m}$ distant. Top: Spikes generated by a neuron nearby the recording electrode surface are clearly visible. Positions and amplitudes corresponding to events detected by the SD are circled in orange. The ICMS artifact, recorded as a fast oscillation ranging from -3.5 to 5.5 mV , is magnified in the blue box on the right, where the vertical red bar indicates the stimulation instant. Bottom: The energy output from the SNEO (in gray), is compared with the dynamic threshold in solid green to detect the spikes. In comparison, the threshold altered by the firing-rate based on the whole signal RMS computed in the same timeframes is shown as the dashed purple line. The ICMS energy answer is magnified in the blue box on the right and is several orders of magnitude bigger than the values observed in the graph ($\sim 10^7$). Despite this, the threshold estimation is not altered, and the stimulation artifact does not cause false events detection.

The off-line analysis performed on our experiments showed that APs were found on a reduced subset of channels. Since the probe used in this study is not a high-density one ($65 \mu\text{m}$ pitch), the probability to get close to multiple neurons with the electrodes surface is low, resulting in only one or two channels acquiring strong APs. On the six experiments, spikes with a strong negative peak below $-70 \mu\text{V}$ were not present in any channels in two experiments, on two channels in three experiments, and only on a single channel in a single experiment. This is the reason behind the need of an accurate MUA detection. Unfortunately, the SD performances for MUA detection are difficult to properly evaluate on simulation. MUAs are generated by a simultaneous activity of large populations of neurons spatially distributed and interconnected, a behavior which is difficult to emulate reliably to create a synthetic dataset. Therefore, here MUAs are quantified on the information provided on *in vivo* experiments. A MUA detection example from a channel without neurons close enough to the electrode to allow distinguishing single unit APs can be seen in Figure 11. The RMS multiplier used to find the threshold must be set to a value significantly lower than that used for the single AP detection and here is set to 5.5. A more stable threshold compared to that of the single AP detection can be observed, which is due to the fact that the MUA contribution is excluded from the RMS estimation, while MUA fluctuations in the AP detection were treated as noise. The results are a smoother threshold change between the different timeframes and the capability to also accurately detect the smaller fluctuation at the beginning and end of MUAs.

The ICMS experiments allowed evoking a response both in the LFP and in the high frequency domain and thus, to perform a comparison with the MUA detection. A stereotyped response, as shown in Figure 12, has been evoked and therefore, detected reliably in the six *in vivo* experiments for 12, 20, 26, 27, 27, and 28 trials out of the 28 given stimuli. On all the trials where a response was visible in the LFP domain, the SD highlighted MUAs correspond to the LFP negative peaks. Figure 12 shows an example of a stereotyped evoked response where MUAs are detected in the channels near the stimulation site after a silent period of 170 ms from the ICMS and further expands rhythmically in repetitions spreading across multiple channels of the linear probe. Before the stimulation, in comparison, the detected activity shows some spurious unsynchronized spiking activity.

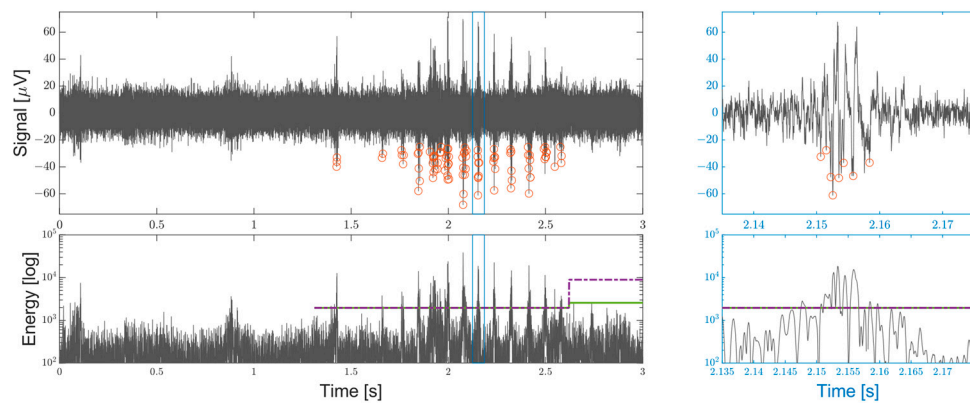


Figure 11. Multi Unit Activities (MUAs) on 3 s of acquisition. Top: Activity recorded by an electrode without any clear source of APs near the surface. The MUAs oscillations detected are circled in red and provide a firing profile of the neuronal population. On the right, the detection of the highlighted single MUA is magnified. Bottom: The energy output from the SNEO operator (in gray, reported in a logarithmic scale), is compared with the dynamic threshold in solid black to detect the oscillations. The threshold based on the RMS of the entire signal is reported with the dashed line for comparison. On the right, the energy response of the highlighted MUA is shown.

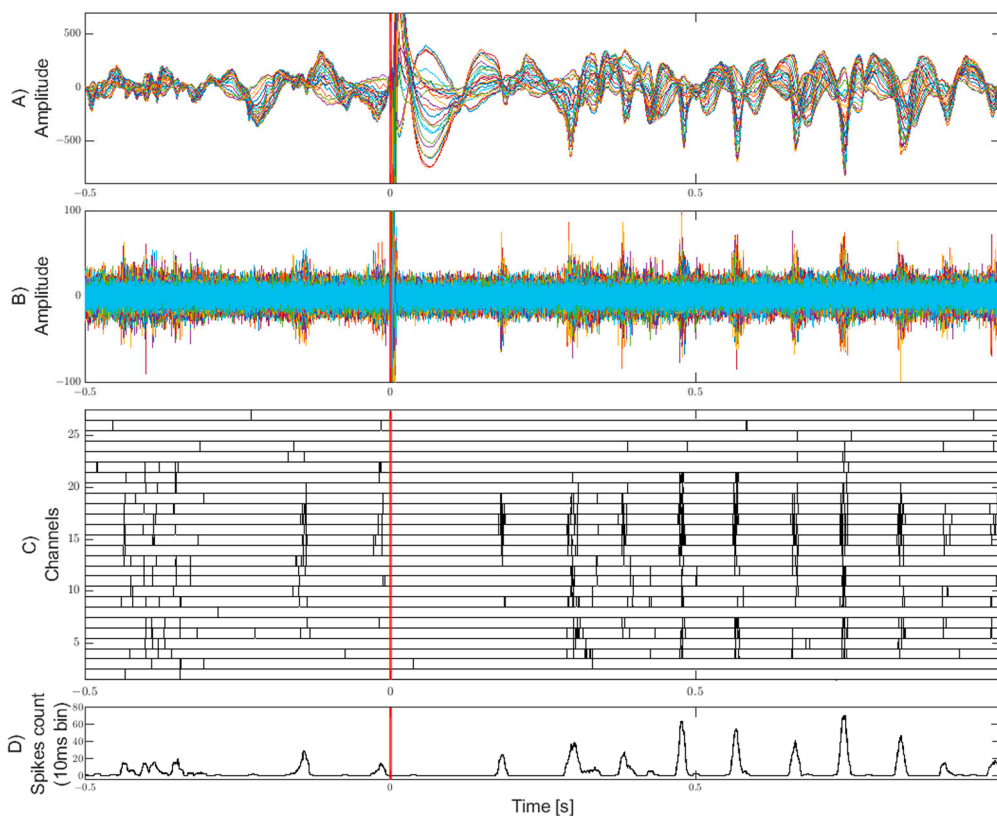


Figure 12. ICMS evoked response in a 1.5 s window. The red bar at time 0 corresponds to a cathodic to anodic ICMS of $\pm 50 \mu\text{A}$ of 200 μs per phase injected by the fifteenth channel every 10 s. A: LFP evoked response enhanced by band-pass filtering the signal between 5 and 200 Hz. B: Spiking activity enhanced by high-pass filtering the signal with a cutoff frequency of 300 Hz. C: Raster plot of MUA detection on data recorded by the linear probe. Channel 1 is the deepest in the barrel cortex, channels 28 to 32 are excluded being outside the cortex. The electrode of channel 8 has a high impedance and is not recorded properly. Before the stimulation, a random firing pattern can be observed from -400 to -300 ms and a spontaneous synchronous activation at -140 ms. After the stimulation, a synchronous activation of a neuronal population is observed spreading across the different channels. D: Cumulative spiking activity (instantaneous firing rate) of the channels in a moving sum window of 10 ms.

The clock cycles required for each component of the design to output the result are reported in Table 3, where the latency of each pipeline component is evaluated, from the samples acquisition of the main RhythmStim loop to the writing of the results in the output FIFOs. The cycles required are intrinsically dependent on the parameters k of the SNEO, that in this design is assumed to be 4. Table 4 compares the FPGA logic consumed by the Intan RhythmStim design by default without any addition and the logic used after the modified implementation, highlighting the amount of space left available on the FPGA for eventual further development of other functionalities. This led to a power consumption reported by the Xilinx XPower Analyzer of the design modified with the SD of 1.002 W, with an increment of 0.096 W over the default design consumption of 0.906 W.

Table 3. Clock cycles required by the design components to provide the output.

Component	Clock Cycles to Output	Notes
Samples preprocessing	7	cross-clock and samples format
High-pass filter	11	complete the filter state after 17
Savitzky-Golay fitting	15	8 + mask length
SNEO	34	$18 + 4 * k$ ($k = 4$)
Thresholding	5	
Local minimum	20	$4 * k + 4$ ($k = 4$)
Samples postprocessing	4	writing to output FIFOs
Threshold estimation ¹	11	the square root and eight more cycles are required every 2^{15} samples
Total	96	One hundred and twenty-five cycles are available between the two samples

¹ Threshold estimation does not influence the total cycles count since no other components wait for its output.

Table 4. Logic utilization of designs.

Logic	Default	Custom	Increment	Available
Slice register	8883	13,770	4887	54,576
Slice LUT	17,775	23,065	5290	27,288
BRAM16	69	76	7	116
BRAM8	0	15	15	232
DSP48A1	8	19	11	58

4. Discussion

The first design constraint of the system is the sub-millisecond latency between a neuronal spike peak in the biological domain and its detection and digital communication. The latency of the system can be calculated from Table 3. Each of the 16 pairs of samples is acquired at 25 kHz, thus the total system sampling rate is 400 kHz. Considering the SD working frequency of 100 MHz, each pair disposes of 250 clock cycles to be elaborated, hence 125 cycles for each sample. The design requires 96 clock cycles to process a single sample, that results in about 1 μ s of processing time. The detection can be algorithmically found after only 14 samples from a spike peak, since the k NEO output evaluates the energy of the central sample in the $2k + 1$ window and in turn the smoothing refers to the central energy value of the Bartlett window of $4k + 1$, where $k = 4$. Considering the processing time and the channels sampling period of 40 μ s, the time required to detect a spike is about half-millisecond. The communication latency depends on the UART baud rate: Assuming a rate of 230,400, 6 bytes can be transferred in less than 240 μ s. This satisfies the initial constraint of the sub-millisecond processing time from a spike peak to its propagation for a fast and reliable closed-loop system design. Obviously, the time required to transfer the spikes information via the USB to the host computer is affected by different factors as mentioned in the introduction, but considering the privileged USB

communication and the 1 ms polling by the host application, the spike reception latency will be of a few milliseconds at most. The second constraint was to have a higher system accuracy compared to the SD included in the default RhythmStim design. The SD accuracy of this work allows detecting the single unit APs with a great precision up to 92%, with a reduced number of false positives and without false detections during the post-ICMS artifact. Furthermore, the SD can be used to provide a precise firing profile of MUAs, that due to the low SNR are difficult to correctly recognize with a standard threshold crossing approach. The firing-independent threshold helps in this by providing a stable threshold value, especially when detecting MUAs, and it has the valuable advantage to avoid the hand-tuning of the threshold for each channel, an important limitation of the default SD. As already mentioned, our recording from the rat barrel cortex showed strong APs—with a similar shape—only on few channels per records, while MUAs were recorded by most of the channels. For this reason, a spike sorter has not been included in the design.

Table 5 shows a comparison with other similar recent works. Work from Murphy, M. at al. [29] is the most similar, being developed on the same recording system, while works from Wang, P.K. et al. [24] and Luan, S. et al. [30] use only the same headstage. Particularly, the latter is a battery-powered standalone platform that logs data on a SD card and therefore, could be considered belonging to another category of devices.

Table 5. Comparison of this work with other works including the default spike detector of the Intan RHS stimulation/recording system.

	This Work	Default	[29]	[24]	[30]
AP detection accuracy (%)	92	N/A	90.7	80–96 ¹	N/A
Latency (ms)	~0.5	0.2	0.3–0.8	1.96	0.1
Supported channels	32	8	8	1	32
Automatic threshold	✓	✗	✗	✓	✓
Stimulation artifact dealing	✓	✗	✓	✗	✗
MUA detection	✓	~	✓	✗	✗
Sorting	✗	✗	✗	✓ ²	✓ ²
UDP spike communication	✓	✗	✗	✗	✗
Use Intan RHS	✓	✓	✓	✓ ³	✓ ³
Code availability	✓	✓	✓	✗	✓

¹ On spike sorting; ² off-line clustering with template matching; ³ headstage only.

5. Conclusions

In this work, a design for an embedded low latency reliable SD for the Intan Technologies RHS stimulation/recording controller, able to detect AP and MUA on up to 32 channels, has been presented. The implementation consists of a third order high-pass digital IIR filter, a Savitzky-Golay polynomial fitting, a SNEO, and a continuously updated firing-rate-independent threshold and allows dealing with the stimulation artifact, avoiding the detection of false positives. Particularly, the threshold computation is based on an RMS estimation that rejects the spikes energy, approximating the recording real noise. This allowed reaching an accuracy of the 92% for the single AP detection and a precise firing profiling of MUAs. The design requires a total of less than half-millisecond from a spike peak for the detection and the communication of the spike details via the UART protocol. Simultaneously, the detections details are also sent via a privileged USB connection to the RHS Stimulation/Recording application, where a newly developed interface allows configuring the spike detector to show the real-time raster plot of the activity detected and to send through the network the detection information via the UDP protocol.

This design is currently used in our laboratory for the purpose of the SYNCH project. This project aims to create a hybrid system where a neural network in the rat brain is interconnected by neuromorphic synapses to an on-chip silicon neural network of spiking neurons, to enable the co-evolution of connectivity and the co-processing of information by the two. Therefore, the modified system presented in this work is used to recognize the spiking activity of the neurons population to be given in the input to the artificial

network. This processes the information and interacts and influences the biological network through ICMS.

The design works out-of-the-box with the Intan RHS stimulation/recording controller, but it has been encapsulated and parametrized to make it easily adapted to other platforms, and it can be modified and scaled both in the capabilities and number of supported channels. The elaboration throughput, if required, can be increased by running the different components of the pipeline in parallel, allowing to start the processing of a new sample before the previous has been completed. The source code of the project can be found on GitHub (www.github.com/Tiax93/RhythmStim-SNEO).

Author Contributions: Conceptualization, M.T. and S.V.; methodology, M.T.; software, M.T.; validation, M.T. and M.B.; formal analysis, M.T. and M.B.; investigation, M.M. and A.L.; resources, S.V.; data curation, M.T.; writing—original draft preparation, M.T.; writing—review and editing, M.B., M.M., S.V., and A.L.; visualization, M.T. and M.B.; supervision, S.V.; project administration, S.V.; funding acquisition, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the SYNCH project, founded by Horizon 2020—Future and Emerging Technologies.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Institutional Ethics Committee O.P.B.A. of the University of Padova and the Italian Ministry of Health (protocol code 522/2018-PR of 04 July 2018).

Data Availability Statement: The source code of the project and the instructions are available online at www.github.com/Tiax93/RhythmStim-SNEO.

Acknowledgments: We thank the Intan Technology company for providing the open-source code and in particular, Harrison R. for his helpfulness. We thank the members of the other SYNCH project groups, especially George R. for his support. Finally, thanks to Cecchetto C. for the critical reading of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The pseudocode of the main algorithms used to create the SD design can be found below. The pseudocode ignores the time-division-multiplexing depending on the channel in the sake of simplicity. The states variables are independent from the procedure life and keep their value between multiple procedure calls. Where the initialization value of a variable is relevant, the value is specified.

The IIR high-pass filtering. The new sample acquired is given in the input. The a and b coefficients are scaled by the 2^{15} factor and rounded, the s variables are the filter internal states.

```
state variables a1, a2, a3 : integer := filter 'a' coefficients * 2^15
state variables b0, b1, b2, b3 : integer := filter 'b' coefficients * 2^15
state variables s1, s2, s3, s4, s5, s6 : integer := 0

procedure filter(unfiltered_sample : integer)

    variable filtered_sample : integer
    variable v : integer

    v := unfiltered_sample - s1 / 2^15
    filtered_sample := (v * b0 + s4) / 2^15
    s1 := v * a1 + s2
    s2 := v * a2 + s3
    s3 := v * a3 + 2^14
    s4 := v * b1 + s5
    s5 := v * b2 + s6
    s6 := v * b3 + 2^14

return filtered_sample
```


Savitzky-Golay fitting. The output of the high-pass filter is given in the input. The mask coefficients are hard coded scaled by the 2^{18} factor and rounded, the *samples* array is used to keep the last seven samples.

```
state variable mask : integer[7] := Savitzky-Golay coefficients * 2^18
state variable samples : integer[7] := [0]

procedure Savitzky-Golay(filtered_sample : integer)

    variable fitted_sample : integer := 2^17

    for i=7 downto 2
        samples[i] := samples[i-1]
    end for
    samples[1] := filtered_sample
    for i=1 to 7
        fitted_sample := fitted_sample + samples[i] * mask[i]
    end for
    fitted_sample := fitted_sample / 2^18

return fitted_sample
```

SNEO. The last value output from the Savitzky-Golay fitting is taken in the input. The Bartlett windows are hard coded scaled by the 2^{16} factor and rounded, the *samples* and *energies* arrays are used to keep the last 17 samples and 17 *k*NEO output.

```
state variable bartlett_window : integer[17] := Bartlett coefficients * 2^16
state variable energies : integer[17] := [0]
state variable samples : integer[17] := [0]

procedure SNEO(fitted_samples : integer)

    variable smoothed_energy : integer := 2^15

    for i=9 downto 2
        samples[i] := samples[i-1]
    end for
    samples[1] := fitted_samples
    for i=17 downto 2
        energies[i] := energies[i-1]
    end for
    energies[1] := samples[5]^2 - samples[1] * samples[9]
    for i=1 to 17
        smoothed_energy := smoothed_energy + energies[i] * bartlett_window[i]
    end for
    smoothed_energy := smoothed_energy / 2^16

return smoothed_energy
```

Threshold estimation. The SNEO output and the current threshold value are in the input.

```

state variable counter : integer := 0
state variable cumulative_squared : integer := 2^14
state variable rms : integer := 0
state variable threshold : integer := inf

procedure threshold_estimate(smoothed_energy : integer,
                           threshold_multiplier : integer)

  if smoothed_energy < threshold
    cumulative_squared := cumulative_squared + smoothed_energy^2
  else
    cumulative_squared := cumulative_squared + rms^2
  end if
  if counter < 2^15
    counter := counter + 1
  else
    rms := squared_root(cumulative_squared / 2^15)
    threshold := rms * threshold_multiplier
    counter := 0
    cumulative_squared := 2^14
  end if

return threshold : integer

```

Thresholding. The current threshold and the new SNEO output are required in the input. The last three SNEO output values are kept in the *energies* array, which can be merged with the SNEO *energies* array.

```

state variable energies : integer[3] := [0]

procedure thresholding(smoothed_energy : integer,
                     threshold : integer)

  variable spike_detection : boolean

  for i=3 downto 2
    energies[i] := energies[i-1]
  end for
  energies[1] := smoothed_energy
  if energies[2] >= energies[1] & energies[2] > energies[3]
    spike_detection := true
  else
    spike_detection := false
  end if

return spike_detection

```

Minimum finder. The last high-pass filtered sample is taken in the input, as well as the current timestamp and the thresholding output. It returns the index and the amplitude of the minimum in the last 17 samples. The *samples* array can be merged with that in the Savitzky-Golay pseudocode.

```

state variable samples : integer[17] := [0]

procedure local_minimum_finder(filtered_sample : integer,
                              spike_detection : boolean,
                              timestamp : integer)

  variables amplitude, sample_index : integer := 0
  variable position : integer

  for i=17 downto 2
    samples[i] := samples[i-1]
  end for
  samples[1] := filtered_sample
  if spike_detection = true
    for i=1 to 17
      if samples[i] < amplitude
        amplitude = samples[i]
        sample_index = i
      end if
    end for
  end if
  position := timestamp - sample_index

return [amplitude, position]

```

References

- Kim, G.H.; Kim, K.; Lee, E.; An, T.; Choi, W.; Lim, G.; Shin, J.H. Recent Progress on Microelectrodes in Neural Interfaces. *Materials* **2018**, *11*, 1995. [[CrossRef](#)] [[PubMed](#)]
- Obien, M.E.J.; Frey, U. Large-Scale, High-Resolution Microelectrode Arrays for Interrogation of Neurons and Networks. In *In Vitro Neuronal Networks; Advances in Neurobiology*; Chiappalone, M., Pasquale, V., Frega, M., Eds.; Springer: Cham, Switzerland, 2019; Volume 22. [[CrossRef](#)]
- Spira, M.E.; Hai, A. Multi-electrode array technologies for neuroscience and cardiology. *Nat. Nanotechnol.* **2013**, *8*, 83–94. [[CrossRef](#)] [[PubMed](#)]
- Nam, Y.; Wheeler, B.C. In vitro microelectrode array technology and neural recordings. *Crit. Rev. Biomed. Eng.* **2011**, *39*, 45–61. [[CrossRef](#)] [[PubMed](#)]
- Jun, J.J.; Steinmetz, N.A.; Siegle, J.H.; Denman, D.J.; Bauza, M.; Barbarits, B.; Lee, A.K.; Anastassiou, C.A.; Andrei, A.; Aydın, Ç.; et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature* **2017**, *551*, 232–236. [[CrossRef](#)] [[PubMed](#)]
- Scholvin, J.; Kinney, J.P.; Bernstein, J.G.; Moore-Kochlacs, C.; Kopell, N.; Fonstad, C.G.; Boyden, E.S. Close-Packed Silicon Microelectrodes for Scalable Spatially Oversampled Neural Recording. *IEEE Trans. Biomed. Eng.* **2016**, *63*, 120–130. [[CrossRef](#)]
- Berényi, A.; Somogyvári, Z.; Nagy, A.J.; Roux, L.; Long, J.D.; Fujisawa, S.; Stark, E.; Leonardo, A.; Harris, T.D.; Buzsáki, G. Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals. *J. Neurophysiol.* **2014**, *111*, 1132–1149. [[CrossRef](#)]
- Butovas, S.; Schwarz, C. Spatiotemporal effects of microstimulation in rat neocortex: A parametric study using multielectrode recordings. *J. Neurophysiol.* **2003**, *90*, 3024–3039. [[CrossRef](#)]
- Tehovnik, E.J. Electrical stimulation of neural tissue to evoke behavioral responses. *J. Neurosci. Methods* **1996**, *65*, 1–17. [[CrossRef](#)]
- Harrison, B.J.; Pantelis, C. Multiunit Activity. In *Encyclopedia of Psychopharmacology*; Stolerman, I.P., Ed.; Springer: Berlin/Heidelberg, Germany, 2010. [[CrossRef](#)]
- Angotzi, G.N.; Boi, F.; Lecomte, A.; Miele, E.; Malerba, M.; Zucca, S.; Casile, A.; Berdondini, L. SiNAPS: An implantable active pixel sensor CMOS-probe for simultaneous large-scale neural recordings. *Biosens. Bioelectron.* **2019**, *126*, 355–364. [[CrossRef](#)]
- Raducanu, B.C.; Yazicioglu, R.F.; Lopez, C.M.; Ballini, M.; Putzeys, J.; Wang, S.; Andrei, A.; Rochus, V.; Welkenhuysen, M.; Helleputte, N.V.; et al. Time Multiplexed Active Neural Probe with 1356 Parallel Recording Sites. *Sensors* **2017**, *17*, 2388. [[CrossRef](#)]
- Rolston, J.D.; Gross, R.E.; Potter, S.M. A low-cost multielectrode system for data acquisition enabling real-time closed-loop processing with rapid recovery from stimulation artifacts. *Front. Neuroeng.* **2009**, *2*, 12. [[CrossRef](#)] [[PubMed](#)]
- Venkatraman, S.; Elkabany, K.; Long, J.D.; Yao, Y.; Carmena, J.M. A System for Neural Recording and Closed-Loop Intracortical Microstimulation in Awake Rodents. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 15–22. [[CrossRef](#)] [[PubMed](#)]
- Bi, G.Q.; Poo, M.M. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **1998**, *18*, 10464–10472. [[CrossRef](#)] [[PubMed](#)]
- Grosenick, L.; Marshel, J.H.; Deisseroth, K. Closed-loop and activity-guided optogenetic control. *Neuron* **2015**, *86*, 106–139. [[CrossRef](#)] [[PubMed](#)]
- Newman, J.P.; Fong, M.F.; Millard, D.C.; Whitmire, C.J.; Stanley, G.B.; Potter, S.M. Optogenetic feedback control of neural activity. *Elife* **2015**. [[CrossRef](#)] [[PubMed](#)]

18. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627–1639. [[CrossRef](#)]
19. Mukhopadhyay, S.; Ray, G.C. A new interpretation of nonlinear energy operator and its efficacy in spike detection. *IEEE Trans. Biomed. Eng.* **1998**, *45*, 180–187. [[CrossRef](#)]
20. Tambaro, M.; Vallicelli, E.A.; Saggese, G.; Strollo, A.; Baschiroto, A.; Vassanelli, S. Evaluation of In Vivo Spike Detection Algorithms for Implantable MTA Brain—Silicon Interfaces. *J. Low Power Electron. Appl.* **2020**, *10*, 26. [[CrossRef](#)]
21. Azami, H.; Sanei, S. Spike detection approaches for noisy neuronal data: Assessment and comparison. *Neurocomputing* **2014**, *133*, 491–506. [[CrossRef](#)]
22. Yu, B.; Mak, T.; Li, X.; Xia, F.; Yakovlev, A.; Sun, Y.; Poon, C.S. Real-Time FPGA-Based Multichannel Spike Sorting Using Hebbian Eigenfilters. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2011**, *1*, 502–515. [[CrossRef](#)]
23. Tariq, T.; Satti, M.H.; Saeed, M.; Kamboh, A.M. Low SNR neural spike detection using scaled energy operators for implantable brain circuits. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* **2017**, 1074–1077. [[CrossRef](#)]
24. Wang, P.K.; Pun, S.H.; Chen, C.H.; McCullagh, E.A.; Klug, A.; Li, A.; Vai, M.I.; Mak, P.U.; Lei, T.C. Low-latency single channel real-time neural spike sorting system based on template matching. *PLoS ONE* **2019**, *14*. [[CrossRef](#)] [[PubMed](#)]
25. Valencia, D.; Alimohammad, A. A Real-Time Spike Sorting System Using Parallel OSort Clustering. *IEEE Trans. Biomed. Circuits Syst.* **2019**, *13*, 1700–1713. [[CrossRef](#)] [[PubMed](#)]
26. Yael, D.; Bar-Gad, I. Filter based phase distortions in extracellular spikes. *PLoS ONE* **2017**, *12*. [[CrossRef](#)] [[PubMed](#)]
27. Shimegi, S.; Akasaki, T.; Ichikawa, T.; Sato, H. Physiological and anatomical organization of multiwhisker response interactions in the barrel cortex of rats. *J. Neurosci.* **2000**, *20*, 6241–6248. [[CrossRef](#)]
28. Swanson, L.W. Brain maps 4.0-Structure of the rat brain: An open access atlas with global nervous system nomenclature ontology and flatmaps. *J. Comp. Neurol.* **2018**, *526*, 935–943. [[CrossRef](#)]
29. Murphy, M.; Buccelli, S.; Bornat, Y.; Bundy, D.; Nudo, R.; Guggenmos, D.; Chiappalone, M. Improving an open-source commercial system to reliably perform activity-dependent stimulation. *J. Neural Eng.* **2019**, *16*. [[CrossRef](#)]
30. Luan, S.; Williams, I.; Maslik, M.; Liu, Y.; De Carvalho, F.; Jackson, A.; Quiroga, R.Q.; Constandinou, T.G. Compact standalone platform for neural recording with real-time spike sorting and data logging. *J. Neural Eng.* **2018**, *15*. [[CrossRef](#)]