electronics

Article

Low-Power Energy-Based Spike Detector ASIC for Implantable Multichannel BMIs

Gerardo Saggese * and Antonio Giuseppe Maria Strollo

Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Naples, Italy
* Correspondence: gerardo.saggese@unina.it

Abstract: Advances in microtechnology have enabled an exponential increase in the number of neurons that can be simultaneously recorded. To meet high-channel count and implantability demands, emerging applications require new methods for local real-time processing to reduce the data to transmit. Nonlinear energy operators are widely used to distinguish neural spikes from background noise featuring a good tradeoff between hardware resources and accuracy. However, they require an additional smoothing filter, which affects both area occupation and power dissipation. In this paper, we investigate a spike detector, based on a series of two nonlinear energy operators, and a simple and adaptive threshold, based on a three-point median operator. We show that our proposal provides good accuracy compared to other energy-based detectors on a synthetic dataset at different noise levels. Based on the proposed technique, a 1024-channel neural signal processor was designed in a 28 nm TSMC CMOS process by using latch-based static random-access memory (SRAM), demonstrating a total power consumption of 1.4 μW/ch and a silicon area occupation of 230 μm^2/ch. These features, together with a comparison with the state of the art, demonstrate that our proposal constitutes an alternative for the development of next-generation multichannel neural interfaces.

Keywords: BMI; action potentials; noise estimate; memory; VLSI; low power

Citation: Saggese, G.; Strollo, A.G.M. Low-Power Energy-Based Spike Detector ASIC for Implantable Multichannel BMIs. *Electronics* **2022**, *11*, 2943. https://doi.org/10.3390/electronics11182943

Academic Editors: Marcello Traiola, Elena-Ioana Vătăjelu and Angeliki Kritikakou

Received: 30 August 2022
Accepted: 14 September 2022
Published: 16 September 2022

1. Introduction

Brain-machine interfaces (BMIs) are nowadays considered one of the most effective tools to study brain activity, using the recording and the analysis of action potentials, described as a boost of instantaneous energy, or, briefly, spikes [1]. Breakthroughs in recording technology have increased the number of recorded neurons and provided better signal quality. Development of BMI-based devices, however, face several challenges: the integration of the entire system in a small-physical size chip, the high-speed functionality to support the real-time streaming of neural activities, the limited bandwidth offered by the wireless channel, the need to transmit a huge amount of data, and the limited power offered by battery or harvesting system [2]. On-implant signal processing becomes essential to reduce the bandwidth to make it possible for wireless transmission, and, for that, spike detection algorithms have received intensive attention from the field of neurophysiology, due to their capabilities of reducing the massive amount of data recorded by neural devices. In such a way, by sending only useful information, the communication bandwidth can be reduced from tens of MB/s to a few kB/s [3]. Then, an off-chip post-processing elaboration is exploited to verify and sort the extracted spikes without any power and performance constraints [4].

A standard workflow of spike detection includes a filtering stage to reduce out-of-band noise, a spike enhancement block, followed by thresholding for spike identification and extraction. The threshold can be either constant or adaptive, to ensure an acceptable compromise between false positives (noise wrongly detected as a spike) and false negatives (a spike being discarded) [5]. The filtering stage typically involves a low pass or

band pass filter to remove the residual noise after the analog-to-digital converter (ADC). Several methods and operators have been proposed to increase the signal-to-noise ratio (SNR), by enhancing the spikes with respect to the background noise. For instance, simple detection methods like amplitude thresholding can offset the detection of neurons with high amplitude spikes. This technique is attractive for real-time and low-power application given the little or no processing; however, its accuracy is greatly reduced when the data has a low SNR [6]. Other methods involve template matching and are particularly effective when the waveform of the spike is known beforehand, but in a real scenario, where the spike shapes are uncertain, their accuracy drastically drops [7,8]. More robust detection methods are based on Discrete Wavelet Transform and its variants and are particularly useful when the processing aims at both spike detection and sorting. Given their complexity, transform-based approaches are prohibitive for a limited resources detector and relegated to off-chip solutions [9]. A popular class of data compression techniques is based on non-linear operators, like Non-linear Energy Operator (*NEO*), Amplitude Slope Operator (*ASO*) and their variants, which are meant to be sensitive to the energy signal variations. By describing a spike as a short-lived burst with high amplitude and frequency, these operators increase the differentiation between spikes and noise, enhancing the SNR. This characteristic, together with a great trade-off between computational cost and accuracy, makes them more amenable for real-time data reduction goals [10,11]. Additionally, extracellularly recorded signals are inevitably corrupted by noise from several sources: the recording hardware, electromagnetic interference, the superimposed activity of multiple neurons and the spatially averaged activity of distant neurons. Importantly, the activity of distant neurons appears as noise which is highly correlated with the signal of interest (Local Field Potentials), described as a low-frequency oscillation. Further, the shape and amplitude of the signals of interest are highly variable. Thus, the design of the threshold constitutes another challenging task. It should adapt to different noise scenarios and to different spike firing rates over time [3]. A common approach is to evaluate the background noise by means of statistics like mean, root-mean square, median or more robust adaptive methods. Then, the threshold level is adjusted by a multiplicative constant which is determined through empirical tests [12].

In this paper, we extend our previous work [1], where we designed a hybrid version of Absolute Differential Operator (*ADO*) and Amplitude Slope Operator (*ASO*), together with a simple and fully adaptive threshold. A comparison with previous smoothed energy methods at different noise levels is provided to demonstrate its accuracy and suitability as a spike detector. The algorithms were initially developed in MATLAB using floating-point arithmetic with a synthetic dataset, then translated to fixed-point representation. Finally, by exploiting a latch-based SRAM, an ultra-low power very large-scale integration (VLSI) architecture is also presented to exhibit its hardware efficiency and electrical requirements, together with a comparison with the state of the art.

The remainder of this paper is organized as follows: Section 2 describes the metrics, the datasets, and the description and hardware implementation of the spike detector. Section 3 describes the software and hardware results, whereas Section 4 is devoted to the discussion and the comparison with the state of the art. Finally, in Section 5 we draw our conclusions.

2. Materials and Methods

The algorithm for adaptive spike detection was developed in two major phases. In the first one the algorithm was conceived, tested, and optimized, using MATLAB with floating point arithmetic, and compared to other energy-based detectors. In the second one, the algorithm was translated to a fixed-point representation and migrated to ASIC target, using hardware programming language.

A common approach to evaluate spike detection algorithms involves using a synthetic dataset for which the ground truth is known beforehand. We used a golden synthetic dataset provided by Quiroga et al. [13] to quantitatively evaluate our proposed algorithm and to make a fair comparison with previously published algorithms.

2.1. Detection Metrics

The metrics used to evaluate the spike detection performance were the true positive ratio (*TPR*), false alarm rate (*FAR*) and accuracy (*ACC*), which are defined, respectively, in Equations (1)–(3).

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FAR = \frac{FP}{TP + FP} \tag{2}$$

$$ACC = \frac{TP}{NS + FP} \tag{3}$$

where *TP* indicates a true positive spike (a spike detected within its duration window), whereas false positive (*FP*) corresponds to a wrong detection (i.e., a spike that is signaled by the algorithm, while no spike was present). *FN*, instead, reflects the number of missed spikes while *NS* is the total number of spikes distributed over time.

High *TPR* means high sensitivity: most of the spikes are detected, while low *FAR* means high specificity: only a few of the detected spikes are incorrect. The accuracy provides an overall measure that represents the trade-off between *TPR* and *FAR*. For applications that further process the data after spike detection, e.g., spike sorting, high sensitivity is desirable (at the expense, possibly, of *FAR*) because any incorrectly detected spikes can be removed after subsequent classification.

2.2. Dataset

The simulated data for the evaluation and quantitative comparison were extracted from a well-known spike sorting tool *Wave_clus* provided by Quian Quiroga et al. in [13].

The dataset had 4 groups of synthetic recordings (*Easy1*, *Easy2*, *Difficult1*, *Difficult2*) submerged in background noise obtained by superimposing random spikes templates at random times and amplitudes to mimic the activity of distant neurons. The noise level ranged between 0.05 and 0.2 in relation to the normalized amplitude of the spikes. Each recording had a duration of 60 s and a sampling frequency of 24 kHz. Spikes were distributed with a mean firing rate of 20 Hz, whose arrivals followed a Poisson distribution. A part of 1 s of the easy1 group, with a noise level of 0.1, is shown in Figure 1, together with the arrival spike times. Please note that these datasets included different active neurons that explain the overlap in arrival spike times.

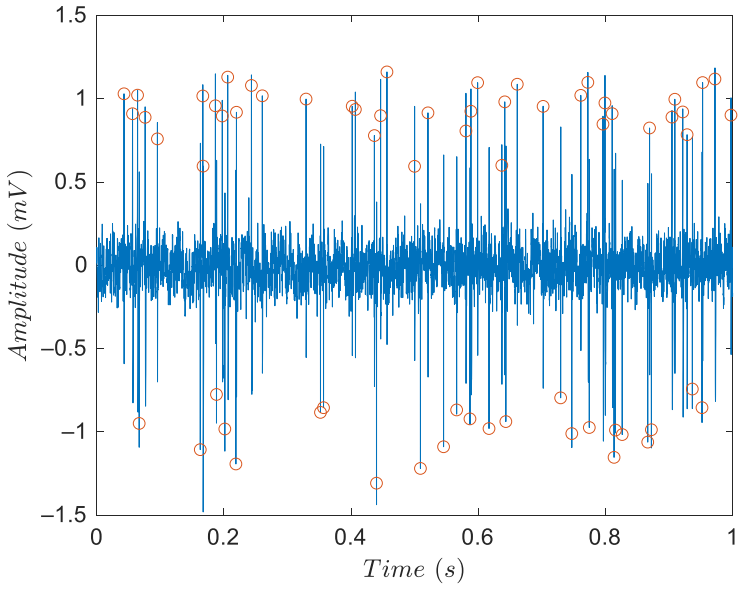


Figure 1. 1 s snapshot of synthetic data of easy1 group with the noise level of 0.1 with arrivals spike stamp highlighted with red circle mark.

2.3. Algorithm Design

Our proposal pipeline and data flow can be summarized in four steps, as depicted in Figure 2: (i) Filtering, (ii) differentiation block, (iii) threshold estimate and (iv) the timestamps of putative spikes.

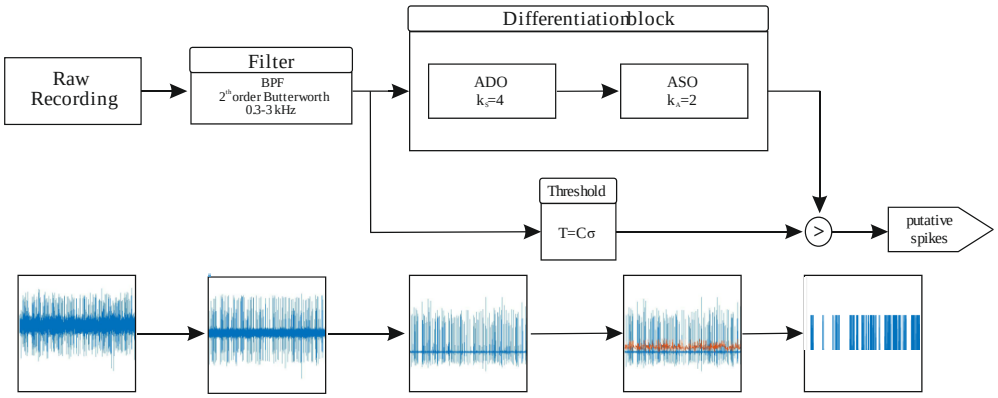


Figure 2. Workflow and dataflow of proposed spike detector.

Although in BMI systems the recording front-end has an analogue high pass or bandpass filter to remove broadband and LFP (i.e., Local Field Potential, low-frequency noise coming from neurons far from the recording site), the noise could still leak into the AP band of interest. Thus, an additional digital filter is required and for this purpose we used a 2nd IIR bandpass Butterworth filter with a bandwidth between 0.3 Hz–3 kHz to remove the residual background noise [3].

2.3.1. Differentiation Block: Nonlinear Energy Operators

As mentioned earlier, nonlinear energy operators are widely used in spike detector algorithms as they are sensitive to the high amplitude and fast variation of the signal. Furthermore, given their trade-off between computational cost and detection performance, they are considered very suitable for real-time and low-power applications, such as BMI. The most popular energy operator is the Non-linear Energy Operator (*NEO*). By denoting as $x(n)$ the filtered neural signal, the output of *NEO* is described by Equation (4), where a resolution or tuning parameter k_N is used to adjust the operator sensitiveness to a range of frequencies [6].

$$NEO(n) = x(n)^2 - x(n - k_N)x(n + k_N) \tag{4}$$

Another energy operator that relays on the sharp and fast transients characterizing the spikes is the Absolute Differential Operator (*ADO*). Equation (5) describes its behavior, where a tuning parameter k_S is introduced. The absolute value is used to fix the polarity of the operator, which can change depending on the position of the spike occurrence (peak or trough of the spike) as addressed in [14].

$$ADO(n) = |x(n) - x(n - k_S)| \tag{5}$$

By considering that a spike is described as a short-lived burst of energy characterized by high-frequency content, Zhang et al. [11] designed and proposed a novel operator called Amplitude Slope Operator (*ASO*) while a parametrized variant was proposed in [12]. The tuned *ASO* is formulated in Equation (6).

$$ASO(n) = x(n)[x(n) - x(n - k_A)] \tag{6}$$

This operator amplifies the signal in the intervals showing a large amplitude and slope.

When comparing the expression of Equations (4)–(6), the *NEO* would require two multiplication and one subtraction, *ADO* would exploit one subtraction and absolute

operator, while *ASO* one multiplication and one subtraction. Additionally, the use of *NEO* calls for $2k_N$ memory registers against k_S/k_A registers for *ADO*/*ASO*.

Despite the operator used, a current practice is to smooth the energized signal with a Hamming or Bartlett window to enhance the SNR, providing benefit to spike detection. It has been demonstrated that the optimum window length can be related to the tuning parameter and was found to be $4k + 1$ [10] (where k is one of k_N, k_S or k_A).

Unfortunately, as the channel count increases, this additional smoothing process affects both area occupation and power consumption, with limited benefit to the detector performance [10]. Inspired by that, and considering the power-limited application budget, we decided to further investigate a spike enhancement technique proposed in our previous work [1], which is composed of the cascade of two energy stages involving *ADO* and *ASO*. Owing to the use of the cascade of two energy operators, which is different to previous approaches [11,12,14], there is no need to use a smoothing window to improve detection accuracy. This minimizes resource usage with benefits on power dissipation and area occupation, as shown in the following sections. Figure 3 shows the effectiveness of the proposed technique in enhancing the spikes with respect to the background noise.

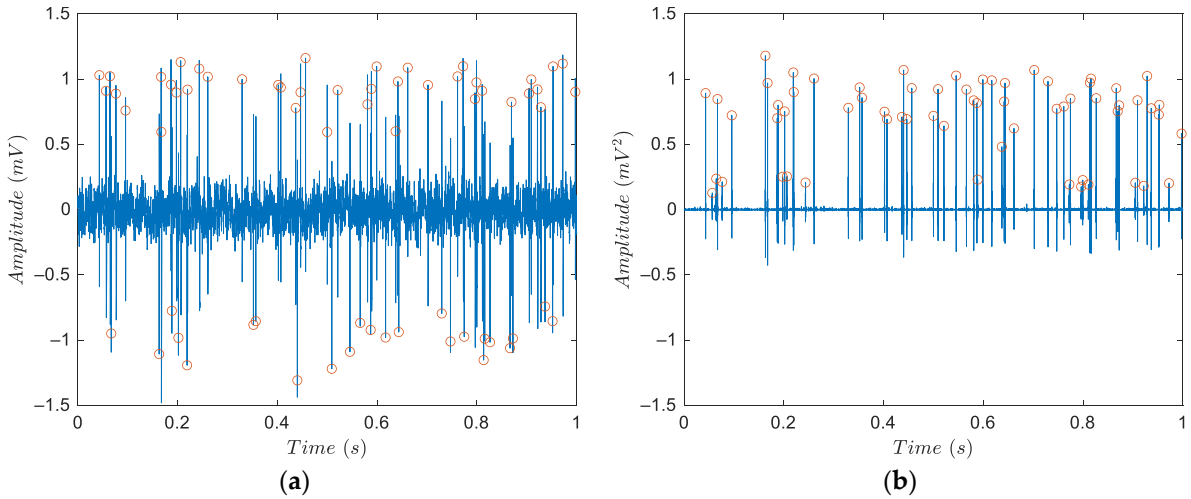


Figure 3. (**a**) The raw signal and (**b**) the enhanced signal after applying the proposed cascade of energy operators. It can be observed how the spikes are enhanced with respect to background noise, that is reduced. The red circle marks indicate the spikes position over the time.

Please note that tuning parameters k_S and k_A were evaluated experimentally by maximizing the *TPR* values over all the datasets, and they were found to be, for the proposed system, 4 and 2 for *ADO* and *ASO*, respectively.

2.3.2. Threshold

The threshold in any spike detector method is related to estimation of the background noise level. The estimate should show both resilience to outliers represented by neural spikes and adaptiveness to the noise variation. Although there are many robust estimates in the literature, such as root mean square, or even more complex methods, like median operator (MAD) [13], most of them are not suitable for an implantable and real-time application and are constrained to be used off-chip, due to their complexity posing significant problems in hardware implementation. Inspired by that, we propose a simple method that can be summarized as follows:

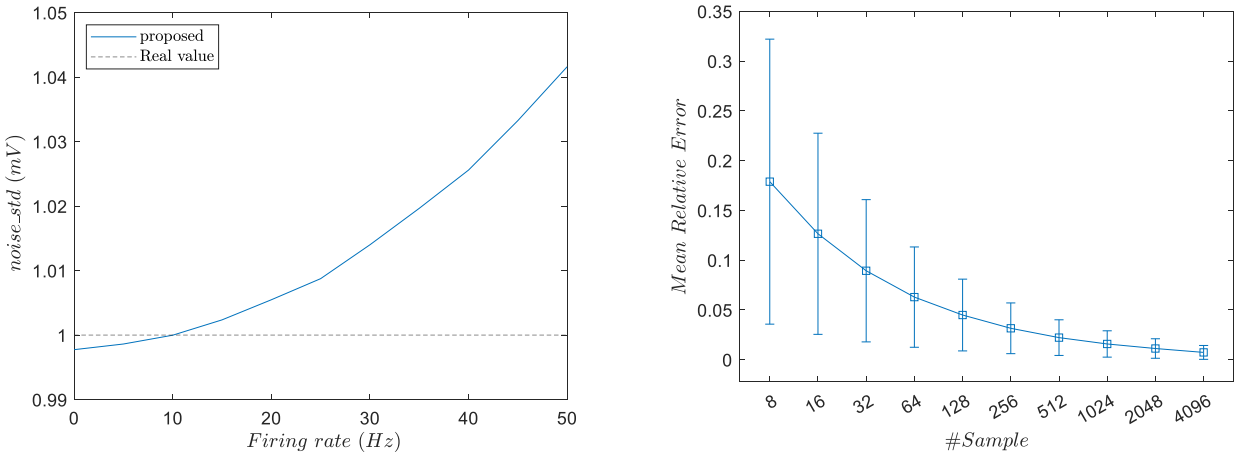
The mean of the absolute value of three batches of M consecutive bandpass filtered signal samples is cyclically evaluated.
A simple three-point moving median operator is used to estimate the noise level, to discard the contribution of outliers,

The estimated noise level is then multiplied by a constant factor C to adjust the detector sensitivity.

The approach is formulated in Equation (7):

$$Th = C\sigma; \text{ where} : \sigma = median_3\left\{\frac{1}{M}\sum_{i=0}^{M-1}|x_i(n)|\right\} \quad (7)$$

Note that using statistics of the underlying signal might lead to very high threshold values, especially in cases of high firing rates and large spike amplitude. Figure 4a shows that our estimate remains close to the "real" noise level over different firing rates, resulting in a robust noise estimate, while requiring fewer hardware resources, as is shown in Section 3. Figure 4b shows the mean relative error in estimating the noise level over different window lengths M. An accurate evaluation of noise level would require a high number of samples (in the order of thousands); however, a window length in the order of 100 is enough to limit the relative error in the order of 10%. We used M = 64 as a reasonable trade-off between accuracy and hardware complexity.

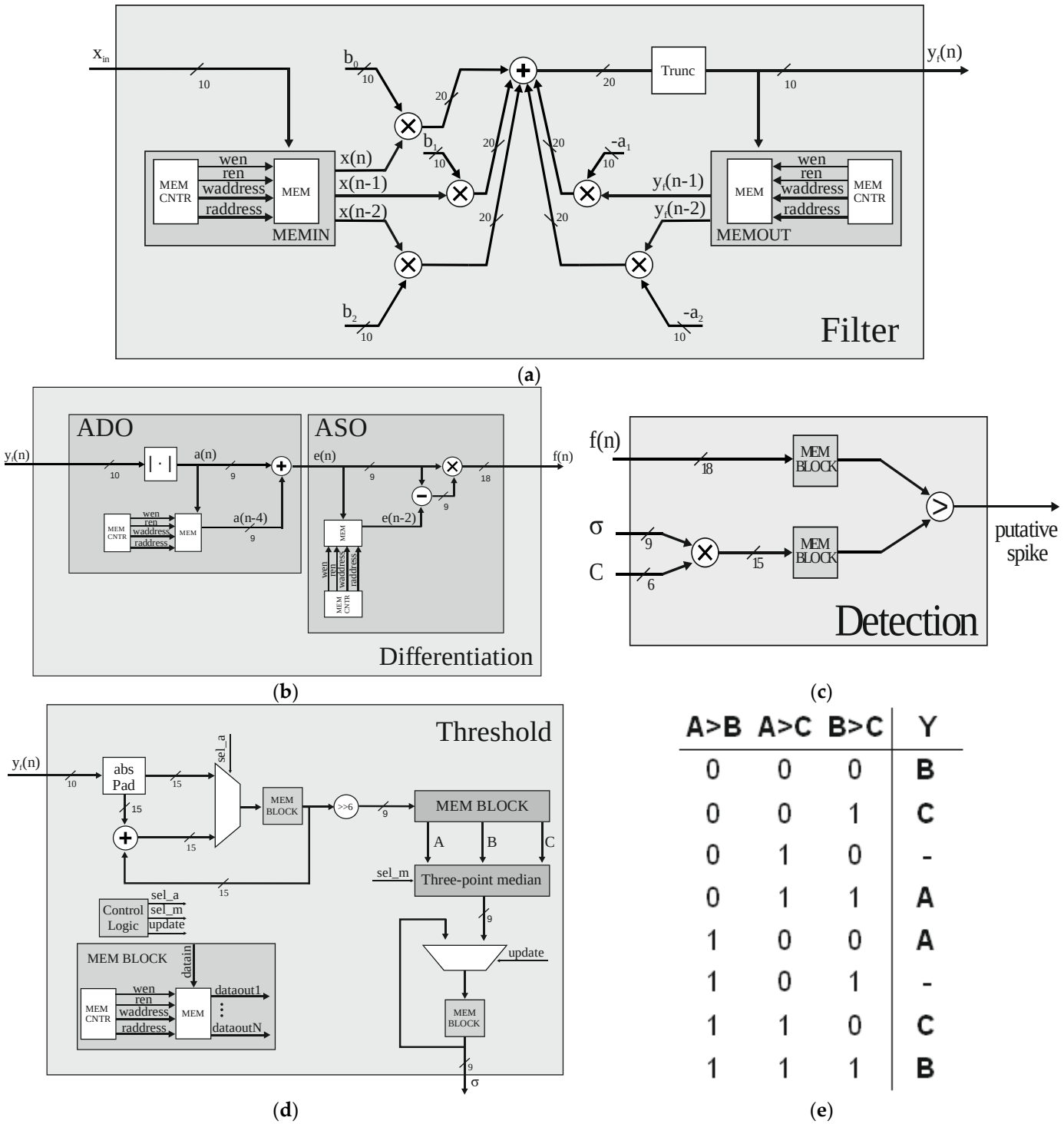


(**a**) (**b**)

Figure 4. (**a**) Estimation of the noise level. Note how our proposal remained close to the real value (**b**) mean relative error bar plot when determining the noise level over different window length M.

2.4. Hardware Implementation

The algorithm described in Section 2.3 was described in VHDL language to be implemented as VLSI architecture and adopted for a multichannel application. We assumed 10 bits width for the raw data coming from the analog-to-digital converter (ADC) [12]. The block diagram in Figure 5 shows the proposed architecture. The 2nd IIR bandpass filter was implemented according to direct form I, as depicted in Figure 5a, with coefficients represented by 10 bits. The filter coefficients were properly scaled to prevent overflow. Figure 5b shows the block that implements the energy operators, composed of memory elements and a simple adder/multiplier. Figure 5c shows the detection block, consisting of a multiplier to adjust the noise level and a comparator to extract the putative spikes. The noise estimator hardware block is depicted in Figure 5d. The mean block was based on a multiplier-accumulator (MAC) and a finite state machine to generate control signals. A 15-bits register was used for the accumulation while the division by M = 64 was obtained by discarding the six less-significant bits of the accumulator. The three-point median operator was obtained by using three comparators and a multiplexer driven by a simple selection logic to compute the output value. The selection logic follows the truth table depicted in Figure 5e.



A>B	A>C	B>C	Y
0	0	0	B
0	0	1	C
0	1	0	-
0	1	1	A
1	0	0	A
1	0	1	-
1	1	0	C
1	1	1	B

Figure 5. Hardware block scheme of the proposed spike detector. (**a**) IIR 2nd filter based on the direct form I with 10-bit width coefficients. The numerator coefficients were scaled to prevent overflow. The fixed-point filter maintains its stability. (**b**) Series of energy operator ADO and ASO while (**c**) the detection block, together with the multiplication of the noise estimate with the scale factor C. (**d**) Noise estimator based on absolute mean and three-point median operator whose outputs were chosen according to (**e**) the truth table.

As a side note, the architecture was designed to process either a single channel or more channels by sharing the same control logic units. For a multichannel solution we added a time division multiplexer block that helped to serialize all the recording matrix which

was assumed to be 32 × 32 as in [12]. Furthermore, we considered 24 kHz for the single channel architecture, whereas a 10 kHz per channel was assumed for the multichannel system, to perform a fair comparison with previous work. The down sample of the clock frequency did not affect the detection results, with the only exception being that of the filter coefficients, which was in agreement with what was suggested in [12].

3. Results

The following section presents both software and hardware results, divided respectively into two subsections.

The software evaluation contains the comparison of our proposed detector with two smoothed energy-based detectors addressed as Smoothed *NEO* (*SNEO*) and Smoothed *ASO* (*SASO*) with k_N and k_A equal to 4. A Hamming window of length 17 was used to smooth the output in both *SNEO* and *SASO*.

In the hardware subsection, we first compared the hardware requirement of our noise estimate with the simple technique based on computing the mean of the absolute values (MAD), to demonstrate the light resource footprint offered. Then, we compared single-channel versions of smoothed energy operators with our work to address its suitability for implantable BMIs. Finally, to put everything in perspective, we implemented two versions of a 1024 channel detector. The first one, named *proposedFF*, implemented memory blocks with flip-flops, whereas in the second one, *proposedLL*, we implemented memory elements with latch-based SRAM [15] to reduce power and area consumptions as much as possible. The investigation was performed by synthesizing the VHDL modules targeting a TSMC 28 nm complementary metal-oxide semiconductor (CMOS) technology.

The multiplicative factors C were experimentally chosen by taking into account accuracy and they were found to be 5, 7, and 17 for *SNEO*, *SASO*, and our proposal, respectively.

3.1. Software Evaluation

As mentioned before, sixteen synthetic tracks were used for performance testing, divided into four groups, each containing four different noise levels. By considering the same noise estimate for all the spike detectors, the accuracy curves over the four groups, previously described, are shown in Figure 6, while the sensitivity (*TPR*) and *FAR* are reported in Table 1.

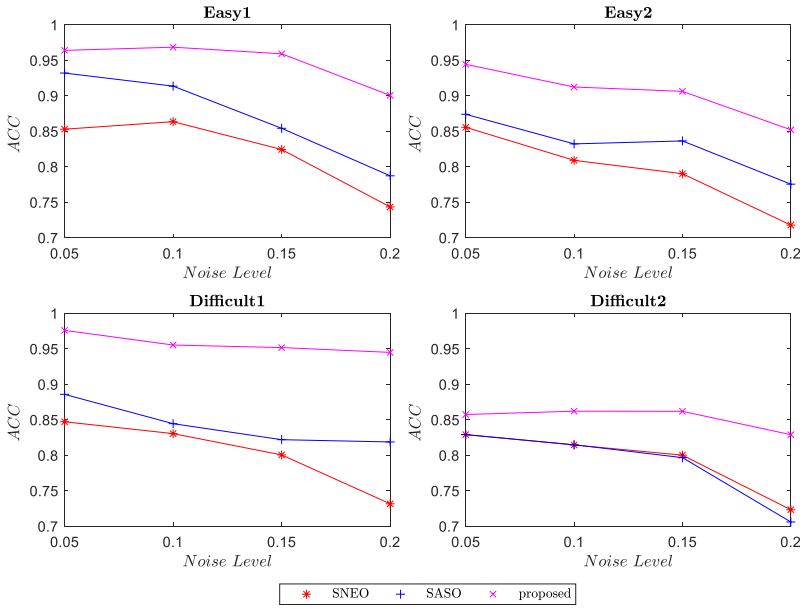


Figure 6. Accuracy results of energy operators on four synthetic datasets over four different noise levels: *SNEO*, *SASO*, and our proposal. The smoothed versions were considered with a tuning parameter of 4 and a Hamming window of length 17.

Table 1. Sensitivity and False Alarm Rate of studied spike detectors on a synthetic dataset.

Dataset	Noise Level	SNEO		SASO		Proposed	
		TPR	*FPR*	*TPR*	*FPR*	*TPR*	*FPR*
Easy1	0.05	0.98	0.13	1	0.06	0.98	0.01
	0.1	1	0.13	0.98	0.06	0.98	0.01
	0.15	0.95	0.14	0.92	0.07	0.97	0.01
	0.2	0.86	0.16	0.85	0.09	0.91	0.01
Easy2	0.05	0.99	0.14	0.93	0.06	0.95	0.01
	0.1	0.94	0.15	0.89	0.08	0.93	0.02
	0.15	0.93	0.16	0.90	0.08	0.92	0.02
	0.2	0.85	0.17	0.84	0.09	0.86	0.01
Difficult1	0.05	1	0.15	0.95	0.07	0.99	0.01
	0.1	0.96	0.14	0.90	0.07	0.97	0.01
	0.15	0.93	0.15	0.89	0.09	0.97	0.02
	0.2	0.86	0.17	0.88	0.08	0.96	0.01
Difficult2	0.05	0.98	0.15	0.90	0.09	0.87	0.02
	0.1	0.94	0.14	0.87	0.08	0.87	0.01
	0.15	0.93	0.15	0.86	0.08	0.88	0.02
	0.2	0.85	0.17	0.76	0.09	0.84	0.02
Averaged		0.93	0.15	0.89	0.08	0.93	0.01

It can be noticed that our proposal, on average, provided better accuracy values than *SNEO* and *SASO*, while these remained close over different groups. Although our approach offered an accuracy drop when using a more challenging dataset like *Difficult2*, it was still acceptable if one considers the reduction of resources by not using an additional smoothing process. An insight into the algorithms' behavior can be noticed by looking at the *TPR* and *FAR* results. Here, once again, for the first three groups (*Easy1*, *Easy2*, and *Difficult1*) all the considered detectors offered a high level of sensitivity, on average more than 90%, with the *SNEO* providing a higher number of false positives compared to the others approaches. On the other hand, for the most challenging dataset, *Difficult2*, the *SNEO* still offered the highest level of sensitivity, as well as the highest *FAR* values as shown in Figure 7. This is challenging for a limited transmission bandwidth and multichannel solution. Nevertheless, by considering the average results, our solution provided the same sensitivity value of *SNEO*, while providing the lowest *FAR* level, also compared to the *SASO* approach. This makes our algorithm a valid candidate and alternative to *NEO* and *ASO*-based methods, when it comes to finding a good trade-off between detections and electrical performance, as addressed in the following section.

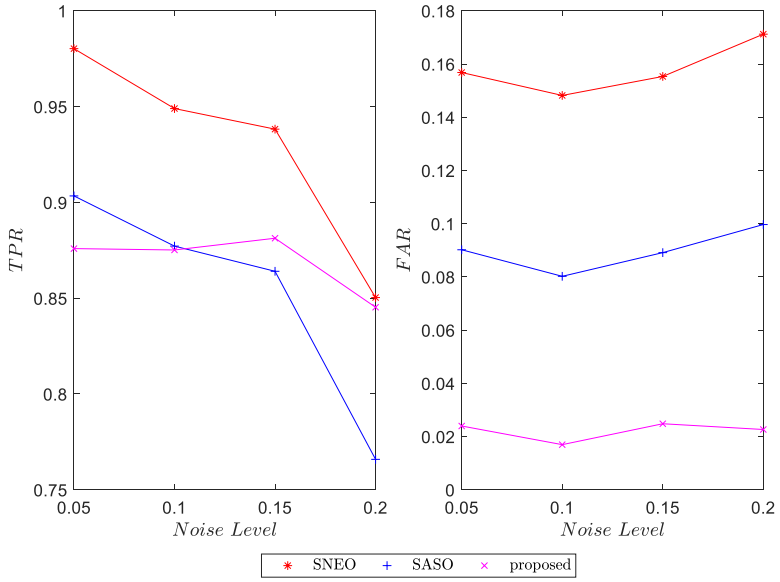


Figure 7. *TPR* and *FAR* on *Difficult2* synthetic dataset over the four different noise levels: *SNEO*, *SASO*, and our proposal.

It is worth mentioning that a possible solution to increase the sensitivity of a spike detector is to lower the scale factor, hence lowering the threshold level. By doing this, one can enhance the *TPR* values but increase the *FAR* value too. This option must be carefully pursued as it strictly depends on transmission bandwidth constraints.

3.2. Hardware Evaluation

As said before, a robust statistical method to extract a good estimate of the background noise was based on the median operator that, because of its computational costs, was limited to an offline approach. Although there are many sorting approaches to implementing the median operator, we used the histogram method which was found to be less computationally prohibitive than other methods [16]. This method works by constructing a cumulative histogram of input data by keeping a count of occurrences of each possible input value. The hardware block diagram is shown in Figure 8a. Each value of the input data corresponded to a bin which was incremented when a new input value was received. However, for a cumulative histogram, the increment was done on the corresponding bin and all the subsequent bins. A ROM was used to enable the increment of the corresponding bin. The bin structure is shown in Figure 8b. It consisted of a register to keep a count of the input value occurrence, a multiplexer controlled by output ROM, and a comparator. The median value was extracted as soon as a bin reached the median index, by means of a priority encoder. In this case, given an even length (M = 64) we used two indices and obtained the median as the result of the mean of two middle values. Given the 10 bits input sample, whose absolute value allows a reduction of one bit, there were 512 bins to keep the count of each input value. After M cycle clock, the median was extracted and still represented with 9 bits. All the hardware implementations were synthetized in TSMC 28 nm CMOS technology by means of Cadence Genus, from which we obtained the occupied area and power consumption. The power dissipation was evaluated by simulating the synthetized circuit with path delays annotated in standard delay format (SDF) file, whereas the switching activity was described by toggle count format (TCF). The area was also obtained by means of post-physical synthesis analysis. The synthesis results of median and our noise estimate are reported in Table 2.

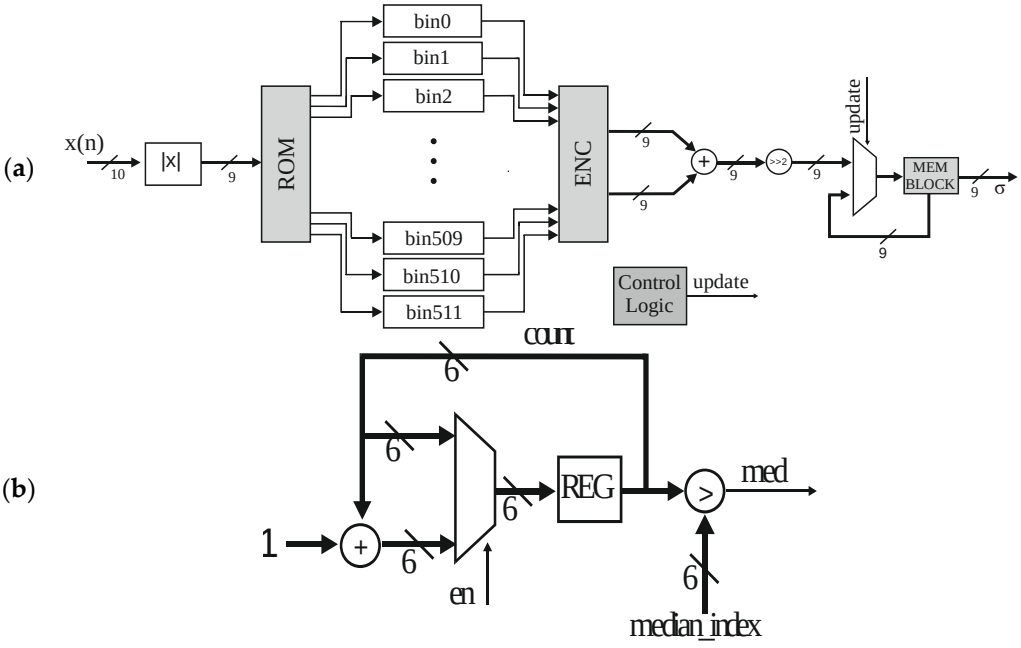


Figure 8. MAD structure: (**a**) Cumulative histogram of the median operator and (**b**) bin structure.

Table 2. Post-physical synthesis consumptions of noise approaches candidates: MAD and our three-point median based method.

Consumptions	Median	Our Noise Estimate
Power (nW)	1513	30
Area (μm^2)	19,628	362

As expected, the median operator constituted a complex and power-hungrier solution compared to our noise estimate. In fact, it would require a silicon occupation and a power consumption more than 50× larger than our approach. This demonstrated that the median operator is limited to application comprising few channels and with more relaxed silicon area/power constraints. Nowadays, where thousands of channels must be processed, this method is hardly useful for real-time and low power applications. The previous proposed spike detectors were also implemented as a single channel version together with our circuit. The electrical performances (area and power) are listed in Table 3.

Table 3. Electrical consumptions of single channel spike detectors: SNEO, SASO and our work.

Consumptions	SNEO	SASO	Our Work
Power (nW)	538	438	143
Area (μm^2)	7868	7370	1618

One can observe the cascade of energy operators provided a silicon area and power reduction of 5× and 3× compared to its counterparts, where the smoothing block was observed to occupy more than 30% of the silicon area. Finally, to assess the amenability of our proposal for implantable neural interface, we implemented and synthetized two 1024 channels versions of the architecture. The first one, named *proposedFF*, implemented memory blocks with flip-flops, whereas in the second one, *proposedLL*, we implemented memory elements with latch-based SRAM [15].

Area and power consumption of the two implementations are reported in Table 4. The *proposedFF* dissipated in total 3.2 mW, of which more than 90% was due to dynamic power, (mostly related to switching activity of the registers); the silicon area occupation was of 0.39 mm^2, mainly occupied by the memory blocks. The second architecture, *proposedLL*, provided a power reduction of 46% and an area saving of 60% compared to the previous solution. As expected, the use of latch-based SRAM helped to reduce both the area and the power consumptions (owing to the use of clock gating technique).

Table 4. Post-physical synthesis consumptions of 1024-channels version: a flip-flop-based memory implementation (*proposedFF*) and a RAM latch-based proposal (*proposedLL*).

Consumptions	proposedFF	proposedLL
Power (mW)	3.2	1.49
Area (mm^2)	0.39	0.24

4. Discussion

The previous section shows that our floating-point implementation showed interesting detection performance, providing higher sensitivity and lower FAR, demonstrating also good adaptivity and robustness over different noise levels and datasets. In addition, we also studied the impact of the fixed-point representation, based on the hardware implementation (complexity, power), and compared our results together with either a popular offline spike detector, reported in [13], or a deep learning approach, based on one dimensional convolutional neural networks (1D-CNNs), proposed in [17]. The offline sorter relies on a simple thresholding, where the noise level is evaluated by applying the absolute median operator over all the track duration. On the other hand, the convolutional model contains four convolutional layers and two pooling layers, followed by a fully connected

neural network and a softmax classifier. Figure 9 shows the average accuracy over the four groups, for conciseness, of floating-point and fixed-point implementations, together with the offline and convolutional models. The comparison between the two representations revealed that the degradation in performance, due to the fixed-point implementation, was relatively minor, within 1–3% reduction in accuracy. The offline sorter method provided a mere 2% increase, on average, compared to our implementations, whereas the 1D-CNN model offered the highest accuracy level, reaching more than 99%. As mentioned, a key objective in the hardware implementation of a spike detection algorithm is to minimize its power requirements. For an implantable application (e.g., BMI) the reason is twofold: firstly, the energy budget is severely limited and secondly, any power consumed is ultimately dissipated as heat that can lead to damage of neural tissue. On the other hand, it is also desirable to reduce complexity to allow for high integration density. The *proposedLL* satisfies these requirements, with a silicon occupation of 230 μm^2 per channel, while dissipating 1.43 $\mu W/ch$ (well within the tissue-safe requirements [18]). Although there are several examples of hardware implementation of spike detectors in literature, there is always a challenge in gathering a reasonable and exhaustive comparison, due to the diverse hardware approaches (e.g., ASIC implementation, reconfigurable logic or embedded processor). together with the difficulty of replicating them. Thus, for a qualitative comparison, we selected a few representative works, the electrical performances of which are summarized in Table 5. The spike detector proposed in [19] is based on an approximation of the conventional non-linear energy operator, energy-of-derivative ED), which basically measures the square of the signal derivative. They implemented an analogue spike detector, which shows high sensitivity value over a wide change of firing rates. A more complex solution is, instead, provided by Yang et al. in [20] based on a modified smoothed version of *NEO*, together with a principal component analysis algorithm, to extract high quality information on the spikes (positive and negative detected spike peaks). On the other hand, a much simpler algorithm was designed by Zhang et al. [3]. They used the *ADO* as enhancement operator and an adaptive threshold based on the estimate of the firing-rate of the spikes. In our previous work [10], we designed a 1024 channel architecture, based on *ASO* and an IIR filter, for the smoothing process, implemented with latch-based SRAM. The *proposed LL* architecture of this paper occupies a mere 10% of the silicon area that would use our previous smoothed *ASO*-based detector, while dissipating half the power. In conclusion, although our work does not embed a sorter system, it provides a good compromise between detection and hardware resources, which makes it a suitable candidate for real-time implementation.

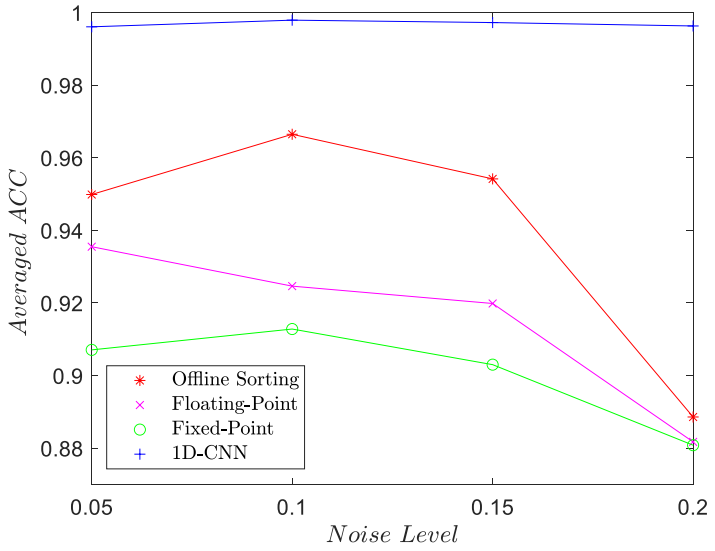


Figure 9. Comparison of floating-point and fixed-point implementation of our algorithm together with an offline sorting tool provided by [13] and a 1D-CNN model [17].

Table 5. Comparison with the state of the art.

	Dwivedi et al. [1] **[19]**	**Yang et al.** **[20]** [1]	**Zhang et al.** **[3]**	**Saggese et al.** **[10]** [2]	**proposedLL** [3]
N. of channels	1	32	128	1024	1024
Domain	Analog	Digital	FPGA	Digital	Digital
Technology	180 nm	130 nm	-	28 nm	28 nm
Frequency (kHz)	-	160	7	$\approx 1 \times 10^4$	$\approx 1 \times 10^4$
Core Voltage (V)	0.8	1.2	-	0.8	0.8
Power Density (μW/ch)	5.1	0.75	0.29	3.6	1.45
Area Density (mm^2/ch)	0.018	0.023	-	0.0022	0.00024
Spike Detector Feature	ED	NEO	ADO	ASO	ADO-ASO

[1] post-layout results. [2] ASO based algorithm where latch-based SRAM is used only in enhancement block for ASO and smoothing filter. Post-physical results. [3] By serializing all the 32×32 recording matrix, the main frequency increases by a factor about of 1000.

5. Conclusions

In this paper we investigated a spike detector, based on a series of two nonlinear energy operators, and a simple and adaptive threshold, based on a three-point median operator. The proposed threshold technique demonstrated itself to be effective in evaluating noise level, even in the presence of large spikes rate, while remaining simple and, hence, amenable for hardware implementation. The analysis with the synthetic dataset revealed that our proposed detector is accurate, while being simpler than previous implementations, reaching more than 90% of sensitivity with a very low false alarm rate. The proposed algorithm was implemented in a compact digital CMOS integrated circuit, in a multichannel architecture able to process 1024 channels. The 1024-channel architecture, using latch-based SRAM, was found to occupy a total silicon area of 0.24 mm^2 with a dissipated power of 1.49 mW. The present design operates at a low supply voltage of 0.8 V and occupies a small silicon area per channel. Overall, the proposed spike detector presents a high detection performance, a moderate power consumption and is area-efficient and is, hence, well suited for multichannel implantable neural data acquisition systems.

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

Funding: This research was supported by the Brain28nm PRIN 2017 (Prot. 20177MEZ7T) project, founded by MIUR.

Data Availability Statement: The data the support the findings of this study are available upon reasonable request from the corresponding author.

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

References

1. Saggese, G.; Zacharelos, E.; Strollo AG, M. Low Power Spike Detector for Brain-Silicon Interface using Differential Amplitude Slope Operator. In Proceedings of the 17th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), Villasimius, Italy, 12–15 July 2022; pp. 301–304. [CrossRef]
2. Schäffer, L.; Zagy, Z.; Kineses, Z.; Fiáth, R. FPGA-Based Neural Probe Positioning to Improve Spike Sorting with OSort Algorithm. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4. [CrossRef]
3. Zhang, Z.; Savolainen, O.W.; Constandinou, T.G. Algorithm and Hardware Considerations for Real-Time Neural Signal on-Implant Processing. *J. Neural Eng.* **2022**, *19*, 016029. [CrossRef] [PubMed]
4. Schaffer, L.; Pletl, S.; Kincses, S. Spike Detection Using Cross-Correlation Based Method. In Proceedings of the IEEE 23rd International Conference on Intelligent Engineering Systems (INES), Gödöllő, Hungary, 25–27 April 2019; pp. 000175–000178. [CrossRef]

5. Mirzaei, S.; Hosseini-Nejad, H.; Sodagar, A.M. Spike Detection Technique Based on Spike Augmentation with Low Computational and Hardware Complexity. In Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 894–897. [CrossRef]
6. Tariq, T.; Satti, M.H.; Saeed, M.; Kamboh, A.M. Low SNR Neural Spike Detection Using Scaled Energy Operators for Implantable Brain Circuits. In Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Korea, 11–15 July 2017; pp. 1074–1077. [CrossRef]
7. Jiang, T.; Wu, D.; Gao, F.; Cao, J.; Dai, S.; Liu, J.; Li, Y. Improved Spike Detection Algorithm Based on Multi-Template Matching and Feature Extraction. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 249–253. [CrossRef]
8. Hao, H.; Chen, J.; Richardson, A.; Van der Spiegel, J.; Aflatouni, F. A 10.8 MW Neural Signal Recorder and Processor With Unsupervised Analog Classifier for Spike Sorting. *IEEE Trans. Biomed. Circuits Syst.* **2021**, *15*, 351–364. [CrossRef]
9. Shaeri, M.A.; Sodagar, A.M. A Method for Compression of Intra-Cortically-Recorded Neural Signals Dedicated to Implantable Brain–Machine Interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2015**, *23*, 485–497. [CrossRef] [PubMed]
10. Saggese, G.; Tambaro, M.; Vallicelli, E.A.; Strollo, A.G.M.; Vassanelli, S.; Baschirotto, A.; Matteis, M.D. Comparison of Sneo-Based Neural Spike Detection Algorithms for Implantable Multi-Transistor Array Biosensors. *Electronics* **2021**, *10*, 410. [CrossRef]
11. Zhang, Z.; Constandinou, T.G. Adaptive Spike Detection and Hardware Optimization towards Autonomous, High-Channel-Count BMIs. *J. Neurosci. Methods* **2021**, *354*, 109103. [CrossRef] [PubMed]
12. Saggese, G.; Strollo, A.G.M. A Low Power 1024-Channels Spike Detector Using Latch-Based RAM for Real-Time Brain Silicon Interfaces. *Electronics* **2021**, *10*, 3068. [CrossRef]
13. Quiroga, R.Q.; Nadasdy, Z.; Ben-Shaul, Y. Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. *Neural Comput.* **2004**, *16*, 1661–1687. [CrossRef] [PubMed]
14. Liu, Z.; Sun, Z.; Shi, G.; Wu, J.; Xie, X. A Novel Algorithm for Online Spike Detection. In Proceedings of the Materials Science, Engineering and Chemistry Conference (MATEC), Nanjing, China, 24–26 May 2018; Volume 173, p. 02017. [CrossRef]
15. Esposito, E.; Strollo, A.G.M.; Alioto, M. Power-Precision Scalable Latch Memories. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4. [CrossRef]
16. Fahmy, S.A.; Cheung, P.Y.K.; Luk, W. Novel FPGA-Based Implementation of Median and Weighted Median Filters for Image Processing. In Proceedings of the International Conference on Field Programmable Logic and Applications, Tampere, Finland, 24–26 August 2005; pp. 142–147. [CrossRef]
17. Li, Z.; Wang, Y.; Zhang, N.; Li, X. An Accurate and Robust Method for Spike Sorting Based on Convolutional Neural Networks. *Brain Sci.* **2020**, *10*, 835. [CrossRef] [PubMed]
18. Navajas, J.; Barsakcioglu, D.Y.; Eftekhar, A.; Jackson, A.; Constandinou, T.G.; Quian Quiroga, R. Minimum Requirements for Accurate and Efficient Real-Time on-Chip Spike Sorting. *J. Neurosci. Methods* **2014**, *230*, 51–64. [CrossRef] [PubMed]
19. Dwivedi, S.; Gogoi, A.K. A Novel Adaptive Real-Time Detection Algorithm for an Area-Efficient CMOS Spike Detector Circuit. *AEU Int. J. Electron. Commun.* **2018**, *88*, 87–97. [CrossRef]
20. Yang, Y.; Boling, S.; Mason, A.J. A Hardware-Efficient Scalable Spike Sorting Neural Signal Processor Module for Implantable High-Channel-Count Brain Machine Interfaces. *IEEE Trans. Biomed. Circuits Syst.* **2017**, *11*, 743–754. [CrossRef] [PubMed]