Article

A Low Power 1024-Channels Spike Detector Using Latch-Based RAM for Real-Time Brain Silicon Interfaces

Gerardo Saggese * and Antonio Giuseppe Maria StrolloDepartment of Electrical Engineering and Information Technology, University of Naples Federico II,
80125 Naples, Italy; antonio.strollo@unina.it

* Correspondence: gerardo.saggese@unina.it

Abstract: High-density microelectrode arrays allow the neuroscientist to study a wider neurons population, however, this causes an increase of communication bandwidth. Given the limited resources available for an implantable silicon interface, an on-fly data reduction is mandatory to stay within the power/area constraints. This can be accomplished by implementing a spike detector aiming at sending only the useful information about spikes. We show that the novel non-linear energy operator called ASO in combination with a simple but robust noise estimate, achieves a good trade-off between performance and consumption. The features of the investigated technique make it a good candidate for implantable BMIs. Our proposal is tested both on synthetic and real datasets providing a good sensibility at low SNR. We also provide a 1024-channels VLSI implementation using a Random-Access Memory composed by latches to reduce as much as possible the power consumptions. The final architecture occupies an area of 2.3 mm², dissipating 3.6 μ W per channels. The comparison with the state of art shows that our proposal finds a place among other methods presented in literature, certifying its suitability for BMIs.

Citation: 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. <https://doi.org/10.3390/electronics10243068>

Academic Editor: Heung-Il Suk

Received: 12 November 2021

Accepted: 7 December 2021

Published: 9 December 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/>).

Keywords: neuroscience; brain machine interface; digital signal processing; spike detection; low power; VLSI

1. Introduction

Brain-machine interfaces (BMIs) or Brain Silicon Interfaces (BSIs) enable the communication between the brain and the outside environment by conveying signals acquired from the brain to control actuators such as computers or robotic arms, with potential to restore motor function in individuals with disabilities [1,2]. Advances in microelectronics made it possible to use high-density microelectrode arrays (MEAs) which provide measurements of high accuracy and, consequently, the possibility to simultaneously record large neuronal populations at high spatiotemporal resolution with a good signal-to-noise ratio (SNR) [3]. The newer implantable neural interfaces can record up to 1000 s channels, however such a large number of channels poses major challenges for the communication link, for both wired and wireless systems [4,5]. Limited communication bandwidth and power limitations require data reduction to be performed on-chip before transmission, and spike detection is one way of accomplishing this. It is also essential that any spike detector must be low-power to prevent heat-related tissue damage and low-area to be implantable. On the other hand, requirements on the algorithms implemented in hardware are that they must be accurate, automatic, real-time, and computationally simple so as to stay within the power limitations [6].

Spike detection algorithms aim to correctly identify the action potentials (APs), or spikes, which encode important neural information, from the background noise. This allows the BMIs to transmit only the spike counts, their arrival time and the spike shapes instead of the entire raw signal [7,8]. A common design consists of a filter stage which extracts the only APs band of interest, an enhancement block that emphasizes the APs in presence of noise, improving the SNR, and a decision-making process, usually involving

a dynamic or static threshold with which the enhanced signal is compared [9]. The spike enhancement can be achieved by using methods consisting of classifying any event crossing a user-specified threshold as a putative spike, or others relying on template-matching [10,11]. Despite their simplicity and extensive use in literature, their performance is very sensitive to background noise as well as to the variation of occurrence-rate of spikes [8]. More robust detection methods are those based on wavelet transformation which are however computationally prohibitive for a real-time multichannel detector [12,13]. Other techniques, instead, lie on building a direct communication of the brain with a computer which elaborates the raw signal, extracts useful information and controls an external device, these methods are known as brain-machine interface (BCI). However, few of them involve on-fly pre-processing to reduce the transmission bandwidth [14,15].

The most popular and used spike detector methods are those based on a non-linear energy operator and its variations providing the best trade-off between resource requirements and detection performance, and are therefore more appropriate for an implantable solution [16,17]. These operators estimate the instantaneous frequency and amplitude of the signal to be processed. They fit the role of enhancing operator in spike detection, since an action potential can be described as an instantaneous high frequency energy variation [18]. After the enhancement process, a threshold is often used to find the action potential. Traditionally, the threshold is based on the statistics of the signal by providing an estimate of the noise standard deviation. Even if the literature provides different threshold techniques, they are often sensitive to noise and spikes variations that worsen the performance of the detector, regardless of their computational requirements. In our previous work we studied and compared different threshold methods identifying in the winsorisation (WA) the more amenable for BMIs [19].

In this paper, we investigate and propose a new spike detector based on the novel energy operator called ASO proposed by Zhang, Z et al. [20], and a threshold based on our noise estimate WA. The algorithm has initially been developed in MATLAB using a floating-point arithmetic and testing detection performance by using both synthetic and real data. A 1024 multi transistor arrays (MTAs) very large-scale integration (VLSI) architecture is also provided to demonstrate hardware efficiency, real-time capability, and power requirements. Setting aside the choice of the spike detector and the hardware approach (ASIC, embedded system, etc.), from a hardware implementation point of view, a crucial aspect that unites all the algorithms is the large use of flip-flops required to store and process the data, in particular the enhancement block, that might limit the increase of the channels to be processed because of the limited resource available. To overcome this limit, we implemented a low-power random access memory (RAM) by using latches; more details are given in Section 2 [21].

The rest of the paper is organized as follows. Section 2 presents merits, datasets, the details of algorithm and design implementation. Section 3 provides the results whereas Section 4 is devoted to discussion and comparison with the state of art. Finally, in Section 5 we draw our conclusions.

2. Materials and Methods

2.1. Evaluation Metrics

In order to evaluate the generality of the proposed method, the detection is evaluated by measuring the accuracy (ACC), sensitivity (TPR) and the false-alarm rate (FAR), given by Equations (1)–(3).

$$ACC(\%) = \frac{TP}{TP + FP + FN} \times 100 \quad (1)$$

$$TPR(\%) = \frac{TP}{TP + FN} \times 100 \quad (2)$$

$$FAR(\%) = \frac{FP}{TP + FP} \times 100 \quad (3)$$

where TP , FP and FN represent the number of spikes rightly detected, the number of false spikes (noise detected as putative spike) and the number of missed spikes [19].

Equation (4) represents the relative error (RE) used to provide the goodness of threshold WA whereas the root mean square error ($RMSE$) is used to estimate the error between the floating point and fixed point method [22].

$$RE(\%) = \frac{|x - \hat{x}|}{x} \times 100 \quad (4)$$

where x is the exact value, $\hat{x}$ represents the computed value.

2.2. Dataset

The evaluation of spike detection can be done by using either synthetic or real data. The main advantage with synthetic data, is that the ground truth is known. Although synthetic recordings are used to quantitatively evaluate the method, they often are not representative of a real recording since variations of noise across different electrodes and changes of neurons population are not considered. Conversely, real data provide realistic behavior, but the ground truth is unknown making them useless to measure the detector performance. For this reason, we decide to use both types of datasets: the synthetic to optimize the algorithm while real recordings are employed to qualitatively show its performance.

Synthetic data are obtained by using an extracellular generator called NEUROCUBE provided by Camuñas et al. [23]. This MATLAB toolbox allows to generate extracellular recording for different electrodes configuration (single or tetrode) by superimposing a detailed compartmental model for the target neurons. We generated a 30 s track of a tetrode configuration (i.e., 4 recording channels) with a sampling frequency of 10 kHz and an active neurons rate of 10% in a 1 mm³ of virtual brain tissue. To ensure a more realistic recording the noise source is modelled by linearly adding to the noiseless track a white noise, emulating the electronic noise, while the electric potential in the extracellular space around the recording site (LFP) is modeled according to Mondragón et al. [24]. Additionally, for the synthetic dataset, the detection metrics specified previously are provided as average on 10 repetitions at different SNR values (0–15 dB). The different SNR levels are obtained by varying the noise standard deviation.

Real data (with an annotated ground truth) are, instead, collected by Mizuseki et al. [25] and made public through Collaborative Research in Computational Neuroscience (CRCNS) data sharing platform. The dataset contains multichannel simultaneous recordings made from layer CA1 (hippocampus region) of three subjects during open field tasks. The experiment duration ranged from about 17 s to more than 1 h. The wideband (raw) data was recorded at 20 kHz, simultaneously from 31 to 64 channels (4 or 8 shanks). The data are then post-processed to extract the time stamp and clusters of the putative spikes by using a “gold standard” spike sorter tool Klusters [26]. For the sake of simplicity, we extracted a frame of 10 min from an 8-shanks record (64 channels) and we estimated the SNR by subtracting the waveforms of the sorted spikes from the raw frame. According to our analysis the mean value of SNR across all the recording matrix (8 × 8) is about 10 dB. Figure 1a,b show a 1 s frame of both synthetic and real recording from 1 channel.

2.3. Algorithm Design

Figure 2 shows the workflow and the data flow of our proposal that can be summarized in four steps: band-pass filter with a 2nd order Butterworth filter, an enhancement block by using the amplitude slope operator (ASO) in combination with a smoothing window on an averaged pixel, threshold estimation and highlight of putative spike.

Figure 1. (a) A 1 s frame of synthetic data generated with NEUROCUBE toolbox. The spike train follows an exponential distribution with a firing rate of 100 Hz. This explains the higher number of spikes compared to the real data (b) where the firing rate on average is few Hz. The SNR for both the tracks is around 10 dB.

Figure 2. The workflow summarizes the followed processing steps of the proposed method while the dataflow shows the effect of each block on the input signal coming from the recording matrix up to detection. It worth noticing that the threshold is evaluated by the smoothed signal as variance. Then a constant factor is used to adjust the detection rate.

It is worth noticing that the effect and analysis of a mean operation over the recording matrix has already been studied in our previous work [19] and it is not shown here. Briefly, we can assume that if the mechanical properties of the electrodes (pitch, dimensions) are about few tens of μm it is possible to further reduce the channels to be processed by working on an average signal.

2.3.1. Band-Pass Filter

A common first approach when dealing with neuronal recording is to select a specific signal band that is typically for extracellular action potentials within 300 to 5 kHz so as to reduce both low and high frequency noise [27]. Here, as mentioned, a 2nd IIR Butterworth filters is used with a bandwidth of 300 Hz–3 kHz, to remove the low frequency noise

characterized by LFP, the offset and the off-band noise. The effect of band-pass filter on the raw synthetic track is shown in Figure 3.

Figure 3. A 15 ms frame of synthetic extracellular signal. In grey the input signal, while the filtered signal is highlighted in blue. One can notice the decrease of noise contribution as well as a light but negligible shift on the right because of non-linear phase of the IIR.

2.3.2. Enhancing Block: ASO

As mentioned in the introduction, the non-linear energy operators are considered more amenable for an implantable and low-power application as BMI. The first operator used in spike detection is the Teager-Kaiser energy operator (TKEO), also known as NEO [28], which is able to enhance the spike from the noisy signal because of its sensibility to high frequency and fast variation that typically characterized a spike. A parametrized version of NEO, called k NEO, is also provided by introducing the tuning parameter k that aims to modify the operator sensitiveness to a range of frequencies [6,29]. The Equation (5) describes the time discrete form of k NEO.

$$k\text{NEO}(n) = x(n)^2 - x(n-k)x(n+k) \quad (5)$$

It is common practice to use a smoothing window (Hamming or Bartlett) on the enhanced signal to smooth the outcoming energy signal, increasing further the SNR as well as providing benefit to the spike detection [8]. It has been studied and shown that the optimum length of the smoothing window can be expressed as function of the tuning parameter k : $4k + 1$ [30].

By observing Equation (5) it can be noted that the energy operator will be high when the signal is large in power ($x^2(n)$) and high in frequency (high amplitude and small slope). By considering this behavior and the fact that the spike is a sudden amplitude change, Zhang et al. [20] described and proposed a novel operator called ASO (here referred to as k ASO), it is formulated in Equation (6).

$$k\text{ASO}(n) = x(n) \cdot [x(n) - x(n-k)] \quad (6)$$

Intuitively, the k ASO amplifies the signal intervals that show a higher amplitude and a more significant slope while suppressing other segments that do not match these conditions. When comparing the two expression of Equations (5) and (6) the computation is reduced by half since the k ASO requires one multiplication and one subtraction. Additionally, looking at the number of registers, the use of ASO necessitates of k registers compared to the $2k$ needed for k NEO. This makes ASO more appropriate for hardware implementation. Moreover, in this case one can use a smoothing window. The tuning parameter was evaluated

experimentally by considering the sampling frequency, the average spike duration as well as either detection performance or resource requirements. It has been found that good performances are achieved for $k = 4$.

The enhancement effect of both operators above described is shown in Figure 4.

Figure 4. A 15 ms frame of synthetic extracellular signal after applying the energy operators: k ASO (blue) and k NEO (red). The tuning parameter is 4. It can be seen that the spike at 25.5 ms is more enhanced by the ASO operator. However, with both operators the energy signal provides for the same spike two peaks that might cause a double detection. This is another reason to use a smoothing window, not only to smooth the spurious noise but also to reduce the multiple detection of the same spike.

2.3.3. Threshold: WA

Following the idea that a spike shows higher amplitude to the noise, we decided to investigate a new approach to estimate the noise considering the limited resource available for a brain machine interface. The method, that we briefly addressed as WA in [19], performs a clipping or truncation of the signal with the aim of discarding the spikes from the distribution under analysis as we want to evaluate the noise distribution. In this way a more precise standard deviation of noise can be obtained. Concisely, the approach can be summarized as follows:

- a preliminary evaluation of standard deviation is performed. It can be done by using average or root mean square. We used the one referred as AA in [19].
- the clipping procedure consists of comparing the previous calculated standard deviation with the absolute value of the new incoming sample and replacing this latter if it is higher than the standard deviation.
- finally, the clipped signal is used to update the threshold. The new estimate is then used for the clipping. The evaluation of the threshold is described in Equation (7).

$$\sigma = 1.58 \frac{1}{M} \sum x_c(n) \quad (7)$$

x_c stands for the clipped signal while M indicates the number of samples employed to update the threshold. The correction factor 1.58 is chosen to get the same standard deviation of Gaussian distribution.

2.4. Algorithm Implementation

The algorithm described in Section 2.3 has been described in VHDL language to be implemented as VLSI architecture. We assume to use a recording 32×32 square matrix with a total amount of 1024 pixels (electrodes). The sampling frequency is 10 kHz for pixel. When dealing with neuronal recording, the bit width of the front-end (ADC) is typically within 8–16 bit. It is believed that 10 bits are sufficient to retain neuronal information [27,31,32]. Therefore, considering a matrix acquired in time division multiplexing row by row, the total amount of samples to store depends only on the number of electrodes in a row and in our case is 32 pixels. Hence, the main sampling frequency is 320 kHz.

The samples are computed through the pipeline scheme shown in Figure 5. The incoming samples from the ADCs feed a block performing the mean operation of 4 neighbor pixels over all the initial matrix, this allows the reduction of the initial matrix from 32×32 to 31×31 . This means that the other blocks process the averaged information of the tetrode configuration. A time division multiplexing block is implemented to serialize the reduced matrix by column, this also explains the sampling frequency of 9.6 MHz. The averaged samples are then processed by the enhancement blocks which store them in a dedicated RAM before computing them. Finally, after a latency depending on the number of samples and channels to be processed, the smoothed samples are compared with their respective threshold.

Figure 5. Hardware pipeline scheme.

As side note, the clock will be provided by the analogue front-end through a crystal oscillator while a piezoelectric generators and ultrasound waves block will be used as power supply and communication link [33].

Please notice that every choice made have been based on hardware consideration with the purpose of achieving the best trade-off between consumptions and precision of fixed-point representation. For the sake of clarity, we provide in the following an insight of few blocks (filters and latch-based RAM), the ones that require a more detailed explanation while for the others blocks a brief description is offered as simple logic and arithmetic operations are required.

2.4.1. Filters: BPF and Smoothing

The 2nd order Butterworth filter with a bandwidth of 0.3–3 kHz is implemented by using the direct form I depicted in Figure 6. The coefficients are first properly scaled to ensure an overflow-free condition and expressed in integer domain with 10 bits. The filtered signal is represented with the same bit width of the input (10 bits) by scaling the filter denominator coefficients. In this way we act on the magnitude response of the filter, working with zeros, leaving unaltered the phase response. The scaled coefficients with their respective rounding errors are listed in Table 1.

Figure 6. IIR filter direct form I structure.

Table 1. IIR coefficients scaled by 2^9 and the percentage error introduced by integer rounding.

	Coefficients			Rounding Error (%)		
	0	1	2	0	1	2
a	1	−0.720	−0.125	0	0.061	0.414
b	0.291	0	−0.291	0.027	0	0.027

The input samples are first stored in a circular buffer, then after a latency of $2 \times 32 + 1$ clock cycle the first filtered sample of first pixel is ready. Another buffer is also required to store the output of the filtered.

The second filter included in this implementation is a finite impulse response (FIR) filter known as smoothing, in our analysis we considered a hamming window. As described previously, the optimum length of the window is about 4 times the tuning parameter, that in our case is 4, hence the hamming window length is 17. This means that for a channel with 961 pixels in series the number of registers is $16,337 \times W$, where W refers to the depth of memory (i.e., bits width). Such a large number of registers would have a negative impact on the area and power of the system. A solution we investigated is the design of an equivalent IIR filter, as IIR filters require less registers than their FIR counterparts. The analysis focused on finding an equivalent IIR providing the best mean-square approximation to the impulse response of the hamming window. The analysis was carried out by using the MATLAB function *prony* which follows Prony's method [34]. We varied the order of the equivalent filter and finally choose a 4th order IIR with a root-mean-square error (RMSE) of 3.7%. This has enabled Hamming FIR filter resources and latency to be reduced by half. Both the impulse response of the analyzed filter and the scaled coefficients with their respective rounding error are shown in Figure 7 and Table 2.

2.4.2. Latch-Based RAM

Regardless the spike detector and the hardware system, the increase on channels count leads to an increase of the memory dimension too. Static Random-Access Memories (SRAM) can be used to limit system area. However, memories implemented by using standard cells represent an interesting alternative to conventional SRAM arrays when implementing embedded memories in the range of tens of kb. These memories store information in arrays of latches, to minimize area, and can be operated at much lower voltages than conventional SRAMs, to minimize power [21]. The latch-based RAM is composed by three blocks: read, write and storage. As shown in Figure 8, the storage unit is composed by R rows and C columns, where R refers to the number of C -bits sample to store. Let us assume, for example $C = R = 2$, and the case when the incoming sample has to be written at the memory location #0, corresponding to the first row. The write enable is high ($we = 1$), the write address ($waddr = 0$) goes in the Write Address Decoder (WAD) which activates the Clock Gate Latch (CGL) of the corresponding row. Then, the CGL produces a clock pulse to store the data in the C latches. Now, let us assume the case when the data just written has to be read. The read enable is asserted ($ren = 1$) and the read

address ($raddr = 0$) is decoded by the Local Read Address Decoders (RADs) to select the bit (row) of each column. The selected bits are stored in FFs to construct the data to be read.

Figure 7. The impulse response of hamming filter with window of length 17 (red) and the 4th equivalent IIR (blue) filter. It is worth noticing that the impulse response of the equivalent filter presents a small stopband ripple (around 20th sample) which can be considered negligible.

Table 2. Equivalent 4th IIR coefficients scaled by 2^8 and the percentage error introduced by integer rounding.

Coefficients						Rounding Error (%)				
	0	1	2	3	4	0	1	2	3	4
a	1	−1.839	0.492	0.812	−0.437	0	0.195	0.013	−0.189	−0.049
b	0.007	−0.003	−0.003	0.001	0.003	0.172	0.008	0.118	−0.089	0.026

As side note, the FFs are positive-edge triggered and the latches are transparent-high to store the bits. Additionally, the CGL is used to reduce as much as possible the switching activity of the clock and avoiding unnecessary switching since the clock is known as major source of power dissipation in digital circuits.

A modified version of this architecture was included both in ASO and in Smoothing blocks. Specifically, the ASO as stated by Equation (6) requires the availability of 4 previous incoming 10-bits samples. So, the embedded memory was modified to allow the reading of 4 data at the same time by replicating 4 times the READ block. On the other hand, the equivalent smoothing filter requires two memory blocks to store either the 20-bits input energy sample coming from ASO or the 20-bits smoothed sample. Moreover, in this case both memories were implemented to provide 4 outputs at the same time.

2.4.3. Others

The Mean block consists of 10 bits register array disposed as a 32×3 matrix. The mean operation is performed by a logic right shift of 2 position (division by 4) since we assumed a tetrode configuration of the recording matrix. After a latency of 3 clock cycle (here 320 kHz), the 31 averaged pixel are provided to the time division multiplexer (TDM) which serializes them working with a clock frequency of 9.6 MHz.

Figure 8. The embedded memory array is composed of $R \times C$ latches ($R = C = 2$ in this figure). The data (w_{din}) stored in the FFs register is written to the respective row by asserting its write enable (w_{en}) coming from the Write Address Decoder (WAD). The read process involves Local Read Address Decoder (LRAD) to select the C latches composing the word. The grey box shows the implementation of the clock gate cell which let the clock signal pass if the enable is asserted.

The amplitude slope operator latency is four times the number of serialized channels, 4×961 cycle clock (9.6 MHz). It is implemented by 10 bits signed adder and 20 bits signed multiplier.

The WA circuit performs the estimate of noise standard deviation of the output of smoothed energy operator for each channel. As Equation (7) states, the WA is computed incrementally, accumulating the M 20-bit energy values of smoothed signal. After a latency of M cycle clock the first estimate is ready and compared with the $M + 1$ input sample. If this latter is higher, it is replaced by the estimate and used in the accumulation process while the constant factor is changed from 1.25 to 1.58. However, since the architecture serialized all the averaged pixels, the latency is $961 \times M$ clock cycle (9.6 MHz) and two buffers to store the 961 accumulated results and the 961 noise estimates were implemented.

As side note, the window sample M of threshold was evaluated experimentally with different window lengths. The analysis, not shown here, was carried out estimating the standard deviation of a white noise 100 times per window lengths. It showed that a length of 64 samples provides just a relative error of 20%. For this reason, we believe it is a good compromise between performance and resources.

3. Results

This section presents the results of our work divided in two subsections: software and hardware evaluation.

In the former, we compare the two energy operators NEO and ASO on a synthetic dataset generated as previously described. Here either the Hamming window or the same scaled true noise standard deviation are considered to provide a fair comparison. Then, we

focus on ASO and its performance when the Hamming filter is replaced by the equivalent IIR filter. The software section ends presenting the detection performance of our proposal either on synthetic or real dataset to finally certify its rightness in a real scenario.

In the hardware subsection we provide the results of consumptions of our architectures by synthesizing the VHDL modules in a TSMC 28 nm CMOS technology to prove its suitability for implantable BMIs.

It is worth mentioning that detection performance on synthetic dataset were averaged on 10 repetitions at each SNR level.

3.1. Software Evaluation

By using the true noise power as threshold, the non-linear energy operator (NEO) and the amplitude slope operator were compared with a SNR ranging within 0–15 dB. The comparison was made by means of detection performance described in Section 2.1 and shown in Figure 9a. One can notice that on average the ASO performs better than NEO, even at lower SNR levels while it shows an increase in the number of false positives (FAR). The overall effect of replacing the Hamming smoothing filter with our equivalent IIR filter is then shown in Figure 9b. As expected, the replacement of the equivalent IIR does not affect much the detection, in fact the performance trends are almost the same. Given that result provided by this analysis, we concentrated our attention on ASO, replacing both the Hamming window and true known noise power with the equivalent IIR and WA. The results were carried out by considering both synthetic and real data as well described previously. However, it is worth remarking that for the synthetic dataset we considered just 4 electrodes (tetrode), averaging the results at SNR range of 0–15 dB, while for the real data we applied the detector over all the recording matrix consisting of 64 channels grouped in 16 tetrodes at the estimated SNR of 10 dB. Concerning the real dataset a wider neurons population was analyzed leading to a higher false positive rate because of the multi-unit activity of the targets leading to multiple spikes overlapping. For this reason, instead of focusing on maximizing all detection performance, a good practice is to increase the number of spikes rightly detected (TPR) over FAR and Accuracy since an off-chip sorting algorithm will be responsible for extracting the spike features and discarding all the false positives. However, the detection performance of our proposal on synthetic and real dataset are respectively shown in Figure 10a,b.

3.2. Hardware Evaluation

The tetrode architecture proposed and described in the previous Section was adapted for a 1024-channels and synthesized in TSMC 28 nm CMOS technology by means Cadence Genus, from which we obtained the occupied area and power consumption. The main clock period is 9.6 MHz for the serialized blocks (TDM, enhancement, Threshold) while for the parallel blocks (BPFs and mean) it is 320 kHz. The power dissipation was evaluated by simulating the synthesized circuit with path delays annotated in standard delay format (SDF) file, whereas the switching activity is described by toggle count format (TCF). The area was also obtained by means of post-synthesis analysis. Three 1024-channels versions were synthesized:

- proposed1: an architecture based on flip-flops as storage elements (without latch-based RAM) and with Hamming smoothing windows.
- proposed2: like proposed1 but using latch-based RAM as a storage element.
- proposed3: like proposed2, replacing the Hamming window with the equivalent IIR filter.

Figure 9. (a) The non-linear energy operator (blue) and amplitude slope operator (red) detection results with $k = 4$ with known standard noise. (b) ASO with hamming window of 17 (blue) and 4th equivalent IIR (red). The multiplier factor maximizing the performance was found to be 5.

The area and power dissipations are reported in Table 3. The proposed1 dissipates 8.3 mW. More than 90% of power dissipation is due to the switching activity of flip-flops; the area of 7.5 mm² is mainly occupied by registers. The second architecture (proposed2) provides a reduction of about 30% on both power dissipation and area occupation owing to the use of latch-based RAM. The final version, proposed3, shows a further reduction in consumption: a 50% in power dissipation and 60% in dimensions compared to proposed2.

Table 3. Post-synthesis consumption of multichannel architecture with flip-flops-based memory with hamming smoothing (proposed1), a latches-based memory of proposed1 (proposed2) and our final proposal including a RAM latches-based with the equivalent IIR (proposed3).

Consumption	Proposed1	Proposed2	Proposed3
Power (mW)	8.3	6.4	3.6
Area (mm ²)	7.5	5.4	2.3

Figure 10. (a) Accuracy, TPR and FAR of ASO algorithm with WA and IIR as threshold and smoothing filter over different SNR levels. (b) the sensitivity matrix of the detector on real dataset.

4. Discussion

From the detection metrics shown in Section 3, it is visible that the ASO method performs better than NEO in detecting the spikes at lower SNR. This trend also highlights in Figure 11a where the ratio between the energy operators' detection metrics is plotted under the same conditions. With SNR levels lower than 5 dB, the ASO outperforms its counterpart with a factor of almost $2\times$ while at higher SNR values there is no visible difference between them. On average, the ASO shows an increase of 10% in accuracy and TPR whereas a worse behavior in detecting noise as spikes (FPR) is observed. However, an average increase of 30% of FPR can be considered acceptable as the main goal of the spike detector is to rightly detect as many true spikes as possible, accepting a light deterioration of the false positive. This is true since an off-chip spike sorter algorithm will be applied on the putative spikes which will be discarded the useless information. In addition, by considering the reduction offered by ASO in terms of arithmetic operations and memory registers, it can be considered a good alternative of the well-known NEO. The further analysis carried out by replacing the Hamming FIR with an equivalent 4th IIR shows a light improvement on Accuracy and TPR while providing a small average increase of about 10% (see Figure 11b) which is mainly caused by the distortion provided.

Figure 11. (a) Detection differences between ASO and NEO with Hamming FIR filter and noise ground true; (b) Detection differences between of ASO with Hamming FIR filter and its equivalent IIR version, both with WA as noise estimate.

On real dataset our proposal provides a discrete sensitivity value (*TPR*) of around 60%. Nevertheless, looking at power/area consumptions of Table 3, the additional resource savings offered by choosing this alternative among the others make it a valid solution for the implantable and limited power budget BMIs. Finally, we evaluated the *RMSE* introduced by deploying our proposed in a fixed-point arithmetic which was found to be 20% compared to its floating-point version. The main error sources were found in the quantization processes of the filter's coefficients (Butterworth and Smoothing). Although, a reduction can be pursued by increasing the number of bits used to represent the filter coefficients at risk of worsening the power dissipation and the area occupied, our study showed no significant influence on performance.

The literature offers several works on spike detection; however, the majority focuses on computational methods for offline analysis while few are implemented in hardware. Among them, there is always a challenge in collecting and providing a fair comparison due to the diverse hardware approaches (e.g., ASIC, embedded, FPGA, etc.). However, we selected few representative works for qualitative comparison on different hardware methods: FPGA, embedded processor, and analogue implementation.

In Wang et al. [35], an FPGA closed loop application with both spike detector and spike sorting is presented. The detection is based on non-linear energy operator (NEO) with a unitary tuning parameter and without introducing smoothing process. The threshold was evaluated by estimating the RMS of the signal. Therefore, they also introduced a sorting process which explains the high-power dissipation achieved in the FPGA as shown in Table 4. Although their proposal provides great performance, considering a real time analysis, the power consumption does not allow an in-vivo implantation. An analogue solution is, instead, presented by Dwivedi et Gogoi [8]. Their work is based on an approximated version of NEO, named energy-of-derivative (ED), which computes the square of the derivative of the signal. However, the detection analysis was based just on synthetic dataset which might challenge their work reliability. Another interesting work is provided by Zhang et al. [20] where the authors used an embedded processor to show the hardware consumption of their solution based on ASO and an adaptive threshold technique which is closer to our proposal. But we provided a simpler and less power greedy threshold technique which was also demonstrated to be less sensitive to the different firing rates of neuron population [19]. In conclusion, our work, providing a good trade-off between detection and complexity, can be considered a good candidate for a real time implantable solution as demonstrates the overall comparison in Table 4.

Table 4. Comparison of our proposal with other works present in literature.

	This Work	[35]	[20]	[8]
N. of channels	1024	1	1	1
Sensibility (TPR %) ¹	>80%	70–95%	>80%	>80%
Adaptive Threshold	Y	N	Y	Y
Feature	ASO	NEO-ED/CM ²	ASO	ED
Domain	Digital	Digital	Digital	Analog
Technology	CMOS 28 nm	FPGA-28 nm	Embedded	CMOS 180 nm
Power Density ($\mu\text{W}/\text{channel}$)	3.6	460×10^3	120	5.1
Area Density ($\text{mm}^2/\text{channel}$)	0.0022	N. A	N. A	0.0018

¹ Results on synthetic dataset; ² Includes spike sorting too.

5. Conclusions

A 1024-channel VLSI architecture based on RAM made of latches is presented. The detector relies on a simple and robust threshold technique that combined with the novel non-linear energy operator ASO make it amenable for an implantable solution. Our results show that the detector works well under different noise scenarios both on synthetic and real recordings. The use of IIR filters (instead of a FIR) does not affect detection performance and brings the advantage of reducing complexity and memory requirements. Three versions of algorithms have been implemented in an ultra-scaled CMOS integrated circuit. The total area occupied by the circuit is 3.2 mm^2 with a 0.0022 mm^2 per channel. Power dissipation is 3.6 mW, corresponding to less than $3.6 \mu\text{W}/\text{channel}$. Hence, the power dissipated per channel remains well below the tolerable limit of $800 \mu\text{W}/\text{mm}^2$ [36,37]. We also have shown that the performances of the proposed circuit are competitive with other similar works presented in literature.

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 Brain28 nm PRIN 2017 (Prot. 20177MEZ7T) project, founded by MIUR.

Data Availability Statement: The MATLAB scripts and VHDL modules are available on request to the corresponding author.

Acknowledgments: The authors would like to thank Mizuseki K. and Gyorgy Buzsàski lab for providing through CRCNS the dataset used herein.

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

References

1. You, A.; Zippi, E.L.; Carmenta, J.M. Large-Scale Neural Consolidation in BMI Learning. In Proceedings of the 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER), San Francisco, CA, USA, 20–23 March 2019; pp. 603–606. [\[CrossRef\]](#)
2. Vaidya, M.; Flint, R.D.; Wang, P.T.; Barry, A.; Li, Y.; Ghassemi, M.; Tomic, G.; Yao, J.; Carmona, C.; Mugler, E.M.; et al. Hemispherectomy in Traumatic Brain Injury: A Noninvasive Platform to Investigate High Gamma Activity for Brain Machine Interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 1467–1472. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Barzan, H.; Ichim, A.M.; Muresan, R.C. Machine Learning-Assisted Detection of Action Potentials in Extracellular Multi-Unit Recordings. In Proceedings of the 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 21–23 May 2020. [\[CrossRef\]](#)
4. Musk, E. An Integrated Brain-Machine Interface Platform With Thousands of Channels. *J. Med. Internet Res.* **2019**, *21*, e16194. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Soontornpipit, P. Design and Development of a Dual-band PIFA Antenna for Brain Interface Applications. In Proceedings of the 2019 7th International Electrical Engineering Congress (iEECON), Hua Hin, Thailand, 6–8 March 2019; pp. 1–4.
6. Tambaro, M.; Vallicelli, E.A.; Saggese, G.; Strollo, A.; Baschirotto, A.; Vassanelli, S. Evaluation of In Vivo Spike Detection Algorithms for Implantable MTA Brain—Silicon Interfaces. *J. Low Power Electron. Appl.* **2020**, *10*, 26. [\[CrossRef\]](#)

7. Semmaoui, H.; Drolet, J.; Lakhssassi, A.; Sawan, M.; Zhou, Y.; Wu, T.; Rastegarnia, A.; Guan, C.; Keefer, E.; Yang, Z.; et al. Setting adaptive spike detection threshold for smoothed TEO based on robust statistics theory. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 474–482. [[CrossRef](#)] [[PubMed](#)]
8. 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](#)]
9. Wang, Z.; Wu, D.; Dong, F.; Cao, J.; Jiang, T.; Liu, J. A Novel Spike Detection Algorithm Based on Multi-Channel of BECT EEG Signals. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 3592–3596. [[CrossRef](#)]
10. 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 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 894–897. [[CrossRef](#)]
11. 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* **2021**, 7747. [[CrossRef](#)]
12. Huang, L.; Ling, B.W.K.; Cai, R.; Zeng, Y.; He, J.; Chen, Y. WMSorting: Wavelet Packets' Decomposition and Mutual Information-Based Spike Sorting Method. *IEEE Trans. Nanobiosci.* **2019**, *18*, 283–295. [[CrossRef](#)] [[PubMed](#)]
13. Schaffer, L.; Nagy, Z.; Kineses, Z.; Fiath, R. FPGA-based neural probe positioning to improve spike sorting with OSort algorithm. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 12–15. [[CrossRef](#)]
14. Katona, J.; Ujbanyi, T.; Sziladi, G.; Kovari, A. Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based brain-computer interface. In Proceedings of the 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, Poland, 16–18 October 2016; pp. 251–256.
15. Katona, J.; Ujbanyi, T.; Sziladi, G.; Kovari, A. Electroencephalogram-Based Brain-Computer Interface for Internet of Robotic Things. In *Cognitive Infocommunications, Theory and Applications*; Klempous, R., Nikodem, J., Baranyi, P.Z., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 253–275, ISBN 978-3-319-95996-2.
16. 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 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jeju, Korea, 11–15 July 2017; pp. 1074–1077.
17. 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](#)] [[PubMed](#)]
18. Abd El-Samie, F.E.; Alotaiby, T.N.; Khalid, M.I.; Alshebeili, S.A.; Aldosari, S.A. A Review of EEG and MEG Epileptic Spike Detection Algorithms. *IEEE Access* **2018**, *6*, 60673–60688. [[CrossRef](#)]
19. Saggese, G.; Tambaro, M.; Vallicelli, E.A.; Strollo, A.G.M.; Vassanelli, S.; Baschiroto, A.; Matteis, M. De Comparison of Sneo-Based Neural Spike Detection Algorithms for Implantable Multi-Transistor Array Biosensors. *Electronics* **2021**, *10*, 410. [[CrossRef](#)]
20. 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](#)]
21. Esposito, D.; Strollo, A.G.M.M.; Alioto, M. Power-precision scalable latch memories. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 7–10. [[CrossRef](#)]
22. Di Meo, G.; De Caro, D.; Saggese, G.; Napoli, E.; Petra, N.; Strollo, A.G. A Novel Module-Sign Low-Power Implementation for the DLMS Adaptive Filter With Low Steady-State Error. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, 1–12. [[CrossRef](#)]
23. Camuñas-Mesa, L.A.; Quiroga, R.Q. A Detailed and Fast Model of Extracellular Recordings. *Neural Comput.* **2013**, *25*, 1191–1212. [[CrossRef](#)]
24. Mondragón-González, S.L.; Burguière, E. Bio-inspired benchmark generator for extracellular multi-unit recordings. *Sci. Rep.* **2017**, *7*, 43253. [[CrossRef](#)] [[PubMed](#)]
25. Mizuseki, K.; Sirota, A.; Pastalkova, E.; Buzsáki, G. Theta Oscillations Provide Temporal Windows for Local Circuit Computation in the Entorhinal-Hippocampal Loop. *Neuron* **2009**, *64*, 267–280. [[CrossRef](#)] [[PubMed](#)]
26. Hazan, L.; Zugaro, M.; Buzsáki, G. Klusters, NeuroScope, NDManager: A free software suite for neurophysiological data processing and visualization. *J. Neurosci. Methods* **2006**, *155*, 207–216. [[CrossRef](#)] [[PubMed](#)]
27. Even-Chen, N.; Muratore, D.G.; Stavisky, S.D.; Hochberg, L.R.; Henderson, J.M.; Murmann, B.; Shenoy, K.V. Power-saving design opportunities for wireless intracortical brain-computer interfaces. *Nat. Biomed. Eng.* **2020**, *4*, 984–996. [[CrossRef](#)] [[PubMed](#)]
28. Jabloun, M. A new generalization of the discrete Teager-Kaiser energy operator-application to biomedical signals. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 4153–4157.
29. Schaffer, L.; Pletl, S.; Kincses, Z. Spike Detection Using Cross-Correlation Based Method. In Proceedings of the 2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES), Gödöllő, Hungary, 25–27 April 2019; pp. 000175–000178.
30. 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](#)]
31. Osipov, D.; Paul, S.; Stemmann, H.; Kreiter, A.K. Energy-Efficient Architecture for Neural Spikes Acquisition. In Proceedings of the 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), Cleveland, OH, USA, 17–19 October 2018; pp. 4–7. [[CrossRef](#)]

32. Xu, J.; Nguyen, A.T.; Wu, T.; Zhao, W.; Luu, D.K.; Yang, Z. A Wide Dynamic Range Neural Data Acquisition System With High-Precision Delta-Sigma ADC and On-Chip EC-PC Spike Processor. *IEEE Trans. Biomed. Circuits Syst.* **2020**, *14*, 425–440. [[CrossRef](#)]
33. Ballo, A.; Grasso, A.D.; Privitera, M. An Efficient AC-DC Converter in 28nm Si-Bulk CMOS Technology for Piezo-Powered Medical Implanted Devices. In Proceedings of the 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Lansing, MI, USA, 9–11 August 2021; pp. 344–347.
34. Liu, J.; Isufi, E.; Leus, G. Filter Design for Autoregressive Moving Average Graph Filters. *IEEE Trans. Signal Inf. Process. Networks* **2019**, *5*, 47–60. [[CrossRef](#)]
35. 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*, e0225138. [[CrossRef](#)] [[PubMed](#)]
36. Seese, T.M.; Harasaki, H.; Saidel, G.M.; Davies, C.R. Characterization of tissue morphology, angiogenesis, and temperature in the adaptive response of muscle tissue to chronic heating. *Lab. Investig.* **1998**, *78*, 1553–1562. [[PubMed](#)]
37. Karkare, V.; Gibson, S.; Marković, D. A 130- μ W, 64-Channel Neural Spike-Sorting DSP Chip. *IEEE J. Solid-State Circuits* **2011**, *46*, 1214–1222. [[CrossRef](#)]