Spatial Information Based OSort for Real-Time Spike Sorting Using FPGA

Laszlo Schäffer, Zoltan Nagy, Zoltan Kincses, Richard Fiáth, and Istvan Ulbert

***Abstract*— *Objective:* Spiking activity of individual neurons can be separated from the acquired multi-unit activity with spike sorting methods. Processing the recorded high-dimensional neural data can take a large amount of time when performed on general-purpose computers. *Methods:* In this paper, an FPGA-based real-time spike sorting system is presented which takes into account the spatial correlation between the electrical signals recorded with closely-packed recording sites to cluster multi-channel neural data. The system uses a spatial window-based version of the Online Sorting algorithm, which uses unsupervised template-matching for clustering. *Results:* The test results show that the proposed system can reach an average accuracy of 86% using simulated data (16–32 neurons, 4–10 dB Signal-to-Noise Ratio), while the single-channel clustering version achieves only 74% average accuracy in the same cases on a 128-channel electrode array. The developed system was also tested on in vivo cortical recordings obtained from an anesthetized rat. *Conclusion:* The proposed FPGA-based spike sorting system can process more than 11000 spikes/second, so it can be used during in vivo experiments providing real-time feedback on the location and electrophysiological properties of well-separable single units. *Significance:* The proposed spike sorting system could be used to reduce the positioning error of the closely-packed recording site during a neural measurement.**

***Index Terms*—FPGA, multi-channel, online spike sorting, real-time, spatial information.**

I. Introduction

THE brain is one of the most complex biological systems containing quadrillions of synapses and billions of neurons. To study this organ in humans or in animal models, and to examine elementary neuronal mechanisms or high-order brain functions, such as learning, sleeping, perception and memory, a widely used method can be applied: the extracellular measurement of the electrical activity in the brain [1]. During a typical *in vivo* electrophysiological experiment a single or multiple neural implants comprising dozens of small electrodes are inserted into the brain tissue for recording short, electrical impulses (usually referred to as action potentials or spikes) generated by neurons located close to the implanted devices [1]–[3]. The obtained signals contain the trains of action potentials fired by neurons located around the electrodes of the neural probe.

Spike trains of individual neurons (called single-unit activity) can be separated from the recorded multi-unit activity with a technique called spike sorting [4], [5]. In basic neuroscience research spike sorting is used during the offline analysis of the recorded neural data (e.g. investigating neural network dynamics [6]) as well as in real-time clinical applications (e.g. in brain-machine interfaces to control neuroprosthetic devices [7], [8]). However, a typical spike sorting algorithm may contain several computationally demanding steps (e.g. spike detection, feature extraction or clustering), which makes real-time processing of multi-channel neural data challenging and thus can greatly reduce the efficiency of clinical applications designed to provide rapid feedback.

High-channel-count neural probes [9]–[11] comprising over hundred electrodes are able to record the activity of hundreds of neurons from numerous individual brain positions simultaneously. Neural data acquired by such probes pose a great challenge for spike sorting algorithms applied on general-purpose computers. Therefore, future real-time clinical applications as well as electrophysiological experiments performed using high-density neural probes could get advantage of hardware-accelerated processing of neural recordings. Dedicated systems based on Field-Programmable Gate Arrays (FPGAs) or Application-Specific Integrated Circuits (ASICs) are especially suited for this task because they are small, wearable and use insignificant amount of energy compared to traditional Central Processing Units (CPUs) or Graphics Processing Unit (GPUs). Using a dedicated system the computation time to process the vast amount of data from a multi-channel (high-dimensional) neural recording can be significantly reduced [12]–[19]. Although ASIC chips

Manuscript received October 22, 2019; revised March 11, 2020; accepted May 14, 2020. Date of publication May 28, 2020; date of current version December 21, 2020. This work was supported by the Hungarian Brain Research Program under Grants KTIA_13_NAP-A-I/1, KTIA-13-NAP-A-IV/1-4,6 and 2017-1.2.1-NKP-2017-00002. The work of Richard Fiáth was supported by Hungarian National Research, Development and Innovation Office (postdoctoral excellence Programme, PD124175, Thematic Excellence Programme 2019). The work of Istvan Ulbert within project No. VEKOP-2.3.2-16-2017-00013 was supported by the European Union and the State of Hungary, co-financed by the European Regional Development Fund. Ministry of Human Capacities, Hungary under Grant 20391-3/2018/FEKUSTRAT is acknowledged. *(Corresponding author: Zoltan Nagy.)*

Laszlo Schäffer is with the Faculty of Science and Informatics, University of Szeged.

Zoltan Nagy is with the Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest H-1083, Hungary (e-mail: nagy.zoltan@itk.ppke.hu).

Zoltan Kincses is with the Faculty of Science and Informatics, University of Szeged.

Richard Fiáth and Istvan Ulbert are with the Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, and also with the Research Centre for Natural Sciences, Institute of Cognitive Neuroscience and Psychology.

Digital Object Identifier 10.1109/TBME.2020.2996281

might be smaller and consume less power compared to FPGAs, which is advantageous in electrophysiological experiments with freely behaving animals where wireless technology is used to transfer neural data from the brain to the recording system [20], they usually lack flexibility for changes. In contrast, design flexibility provided by FPGAs might be in many cases a more important factor than small chip area or low power consumption. For example, electrophysiological recording systems as well as algorithms used for spike sorting are subject to extensive research and development, therefore tend to change rapidly. Thus, FPGAs may be better suited for use in these fields than ASICs. In the literature several FPGA-based solutions can be found, which were developed for spike sorting not suited to process data recorded with high-channel-count neural probes [12]–[18], while [19], [21]–[23] capable of sorting action potentials recorded with high spatial resolution probes [24], [25].

High-density neural probes with closely-packed recording sites can detect the spikes of the same neuron simultaneously on multiple, adjacent sites [26], [27]. This high spatial sampling of spikes might allow a more reliable and accurate identification of neurons in case of template matching-based spike sorters. The reason behind this is that neurons located in different positions relative to the electrode array will have at least a slightly different multi-channel spike waveform (investigated for example in a spatial window incorporating 3×3 electrodes). Thus, the additional spatial information provided by high-density probes might be exploited to increase the accuracy of the spike sorting process [28], [29].

Using a high electrode density ($<20\ \mu m$ electrode-pitch) is also advantageous in multiple other ways. For instance, there is a higher chance that recording sites are physically located near individual neurons [30]. Furthermore, several studies showed that a higher single unit yield can be achieved with a higher electrode density [28], [29], and that a considerable number of pyramidal cells fire spikes with a larger spatial spread than interneurons [28]. Thus, the spikes of several small interneurons located in the vicinity of neural probes having a low electrode density might not be recorded, resulting in a biased pyramidal cell – interneuron ratio. High electrode coverage and increased electrode density might also provide other benefits, including, for instance, the compensation for electrode drift or a more accurate separation of overlapping spike waveforms [9]. However, using an electrode pitch below $4\ \mu m$ will not lead to any further improvement in terms of being close to neuronal signal sources [30].

In this paper, a window-based spike sorting hardware architecture using System-on-Chip (SoC) FPGA is presented for real-time processing of high-channel-count neural data recorded with a dense electrode array. The incoming spikes are observed on each channel in parallel for detection, therefore the spike detection is multi-channel. The sorting part of the proposed system is window-based, because it takes into consideration the spatial information in a window of the electrode array. To perform spike sorting in this window-based manner, the Online Sorting (OSort) algorithm [31] was modified. OSort is an unsupervised template matching algorithm, originally designed to process single-channel recordings. The functionality of the system was verified using a modified version of a hybrid ground truth signal generator [32] and in vivo measurement data [33].

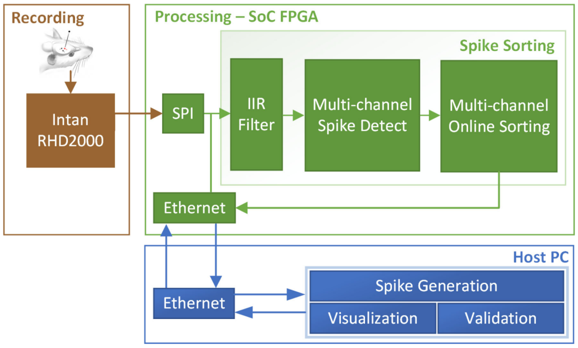


Fig. 1. Overview of the proposed system.

A preliminary version of this work has been reported in [34], which is extended in the presented paper with the algorithm and architecture of the spike detection, as well as the implementation details and validation results of the proposed spike sorting system.

II. Proposed System

The proposed system can be split into three main blocks, the *Recording*, the *Processing* and the *Host PC*, which can be seen in Fig. 1. The *Recording* block contains an *Intan RHD2000* electrophysiological recording system. The *Processing* block can be further divided into the *Ethernet* interface, *SPI* interface, Pre-processing *IIR Filter* module and the *Spike Sorting* module, which contains the *Multi-channel Spike Detect* and the *Window-based Online Sorting* cores. The *Host PC* block includes the *Ethernet* interface, the *Visualization* module, the *Spike Generation* module, and the *Validation* module.

The proposed system is able to process in vivo neural measurement acquired by the Intan RHD2000 (Intan Technologies, Los Angeles, CA, USA) board, as well as simulated neural recordings generated by the *Spike Generation* module on the *Host PC*. To switch between these two input modes only the bit file has to be changed on the FPGA board.

A. Electrophysiological Recording

The proposed system was designed to process neural data recorded with a 128-channel high-density silicon-based probe ($50\ \mu m \times 100\ \mu m$) comprising closely-packed electrodes ($20\ \mu m \times 20\ \mu m$) arranged in a 32×4 array with a center-to-center electrode distance of $22.5\ \mu m$ [33]. Measurements of wideband brain signals (0.1–7500 Hz) with this type of probe were obtained with the Intan RHD2000 electrophysiological recording system at 20 kHz sampling frequency/channel and with 16 bit resolution. These 128-channel recordings usually contain hundreds of thousands of spikes fired by dozens of neurons during a time period of one hour. The Intan RHD2000 uses the SPI protocol for communication purposes.

B. Pre-Processing IIR Filter

The high-density bioelectrical activity recorded from the brain tissue can be separated into local field potentials (below 500 Hz) and spiking activity (500–5000 Hz). Therefore the recorded wideband data should be filtered before spike sorting. In the proposed system a Butterworth Infinite Impulse Response (IIR) third-order zero-phase band-pass filter is used to extract the spiking activity.

C. Processing - SoC FPGA

The proposed system was implemented on a special Xilinx architecture called Zynq, which contains a traditional Programmable Logic (PL) and a Processing System (PS). The latter is an ARM-Cortex processor with various I/O interfaces to connect the system to the outside world. The Gigabit Ethernet interface is used to communicate with the *Host PC* and the SPI interface is used to receive data from the Intan RHD2000 device. The ARM PS communicates with the PL on AXI4 buses (AXI4-Lite, AXI4-Stream, AXI4).

D. Multi-Channel Spike Detection

The *Multi-channel Spike Detect* core is responsible for the real-time multi-channel spike detection. In the proposed system the Non-Linear Energy Operator (NEO) is used for spike detection, because it is the most efficient among the commonly used spike detection methods and it is easily implementable on FPGA devices [5], [12], [35]. The NEO signal can be calculated as follows.

$$\Psi[x(n)] = x(n)^2 - x(n+1) \cdot x(n-1), \tag{1}$$

where $x(n)$ is the nth sample in the input signal, and $\Psi[x(n)]$ is the nth sample in the resulting NEO signal. The NEO signal is very high in amplitude, when both the frequency and the power of the input signal is both high. The threshold of the detection T_N can be calculated as follows:

$$T_N = c_N \cdot \frac{1}{N} \sum_{n=1}^{N} \Psi[x(n)], \tag{2}$$

where c_N is the correction factor and N is the number of samples used. These parameters can be determined experimentally according to the actual signal processing task.

The flow diagram of the spike detection algorithm can be seen in Fig. 2. In our case one sample from each of the 128 channels is fed into the NEO and the standard deviation (STD) calculation block. The STD value is used later in the spatial window-based OSort module to calculate the automatic clustering and merging thresholds. When the $\Psi[x(n)]$ NEO value is larger than the automatically calculated T_N threshold, then the peak of a spike is detected. Afterwards, a temporal window around the detected peak on the actual channel is checked by the realign method, that the peak of the spike is certainly in the center of this temporal window. The correct alignment is crucial, because the comparison can not be done correctly with unaligned spikes, resulting in compromised classification.

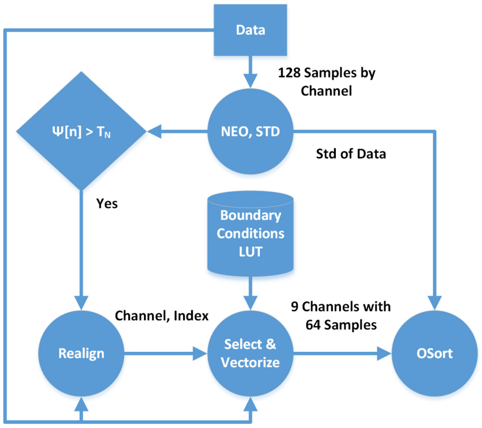
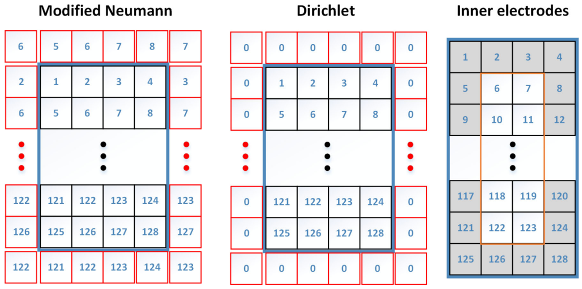


Fig. 2. Flow diagram of the NEO based spike detection algorithm.



Modified Neumann

6	5	6	7	8	7
2	1	2	3	4	3
6	5	6	7	8	7
⋮		⋮			⋮
122	121	122	123	124	123
126	125	126	127	128	127
122	121	122	123	124	123

Dirichlet

0	0	0	0	0	0
0	1	2	3	4	0
0	5	6	7	8	0
⋮		⋮			⋮
0	121	122	123	124	0
0	125	126	127	128	0
0	0	0	0	0	0

Inner electrodes

1	2	3	4
5	6	7	8
9	10	11	12
	⋮		
117	118	119	120
121	122	123	124
125	126	127	128

Fig. 3. Possible boundary conditions for the 32 × 4 electrode array configuration, which is stored in the LUT. The recording sites of the electrode array are within the blue rectangle.

In an electrode array with closely-spaced recording sites the spike waveform of a neuron spreads through adjacent sites, which also has to be considered for detection. After the realignment of the temporal spike window, a spatial window of 3×3 electrodes is examined around the firing channel at the aligned spike position to determine the source of the activity. The channel with a higher absolute amplitude in the neighbourhood will be the new center of origin, assuming that electrode is possibly located the closest to the soma of the neuron. Using this method the source channel of the unit activity can be found, eliminating the effect of the multiple detection of the same spike. In case of two or more neurons firing in the spatial vicinity of each other (8-connected or 8-neighbour) the detected spike matrix will be an overlapping spike waveform and will be removed in the merge phase of the classification process. Recording sites on the edges of the electrode array will also detect spikes. In this case an incomplete electrode window is selected, because there is no measurement beyond the edges of the electrode array. There are three possible solutions to deal with this issue, which can be seen in Fig. 3:

1) Modified Neumann: The boundary is filled with data of the neighbouring channels, considering a Gaussian distributional cross-talk between the channels.

2) Dirichlet: The boundary is filled with fixed constants, in this case zeros.

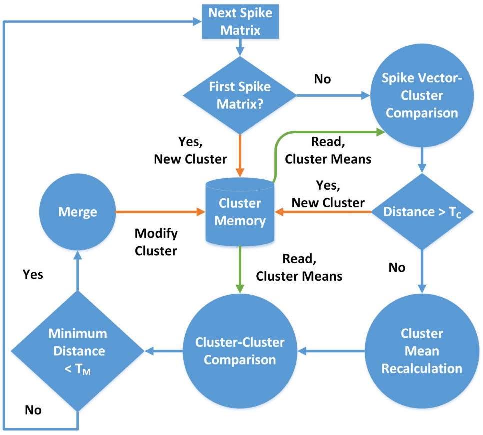


Fig. 4. Flow diagram of the spatial window-based Online Sorting algorithm.

3) Inner Electrodes: Only the inner electrodes are used for detection, therefore spike sorting is limited in this case.

The realign method specifies the origin of the activity and the precise alignment, so a spike matrix containing the data of 9 channels is selected from the electrode array and sent to classification. In the proposed system the *Modified Neumann* boundary condition is used.

E. Spatial Window-Based Online Sorting

The clustering and cluster merging thresholds (T_C and T_M respectively) are automatically determined and adaptively changed during the classification based on the STD value calculated in the spike detection part. The threshold value of one channel is calculated based on the standard deviation of a five seconds long moving window (determined experimentally) from the neural recording ($signal$) as follows.

$$T_C = T_M = std(signal)^2 \cdot c_C \cdot N_S, \tag{3}$$

where c_C is the clustering correction factor [31], and N_S is the number of sample points in a spike.

The flow diagram of the proposed algorithm can be seen in Fig. 4. This algorithm extends the original OSort algorithm with a cluster memory, which is required for the optimal FPGA implementation (Section III-C) and works as follows.

When the *Multi-Channel Spike Detect* core sends the first spike matrix containing 9 spikes from the 3 × 3 electrode window, it is stored as the first cluster in the cluster memory. The next spike matrix will be compared to the saved cluster mean using squared difference as the distance metric. If this spike matrix is similar to the already saved one, then the calculated distance is below T_C, so it is assigned to this cluster. If it is not similar, and the distance is above T_C, then the creation of a new cluster is required. This process is applied to the subsequent incoming spike matrices. After the assignment the mean of this cluster will be updated, because the composition of the cluster is changed. Furthermore, the cluster mean update changes the distance between cluster means, therefore a distance check between clusters is needed. If a distance is below T_M, then the updated and the closest cluster will be merged together forming a new, larger cluster. In this case the smaller cluster will be removed from the memory, and the spike matrices from the smaller are assigned to the larger cluster.

F. Host PC

The task of the Host PC (Intel Core i7-4770 CPU at 3.4 GHz, 8 GB DDR3 RAM) is to visualize the results of the spike sorting, when in vivo neural recordings are used, and to validate the sorting algorithm using hybrid ground truth neural data generation, when the signal generator is operating.

G. Spike Generation

The algorithm and the implementation of the Spike Generation module is fully discussed in [32]. Using ground truth data, the time corresponding to spikes of individual neurons is known and thereby it allows the quantitative assessment of the performance of the spatial window-based OSort spike sorting algorithm. Since we combined in vivo neural data (obtained by direct measurement) and synthetic data (background noise, position and firing time of neurons) to generate our dataset for validation, this is referred to as hybrid ground truth. Spike templates were created from *in vivo* electrophysiological data recorded with the previously mentioned, 128-channel silicon probes from the neocortex of anesthetized rats [33]. The spikes of well-separated units (n = 72) were averaged and the average spike waveforms were used for the construction of templates. Since the high-density probe could record the spikes of a particular unit on multiple, adjacent recording sites, only the recording channel on which the spike appeared with the largest peak-to-peak amplitude was used for template construction. A spike template was represented with 64 sample points/channel (3.2 ms). For a higher variability of spike templates, we selected spike waveforms corresponding to both putative neocortical principal cells (wide spikes, n = 58) and putative neocortical interneurons (narrow spikes, n = 13) [36].

High-density neural probes provide high spatial resolution, that is, the spikes of neurons can be recorded on multiple electrodes simultaneously. The extent of spatial spreading of the spike waveform of a neuron depends on several factors (e.g. neuron-electrode distance, type of the neuron) but is usually a few tens of microns. Therefore, the maximal radius of spatial spreading of our template spike waveforms was six electrodes (132.5 μm) in each direction with the mean spike waveform located in the center.

To generate a more realistic neural dataset the spatial spread of the spike waveforms of 160 in vivo recorded cortical neurons were examined (obtained with the 128-channel silicon probe used in this study) to extract the spreading patterns. The Spike Generation module [32] was modified based on the found spatial patterns, so in the modified version each simulated neuron has an individual and asymmetrical spike waveform spreading.

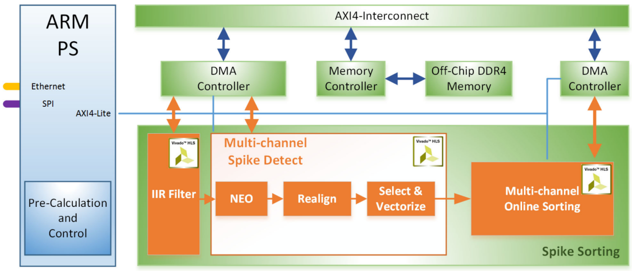


Fig. 5. The schematic diagram of Processing block system.

III. Implementation

The schematic diagram of the *Processing* block can be seen in Fig. 5. The system consists of the *ARM Processing System* (PS), the *Off-chip DDR4 Memory*, the *Memory Controller* core, the *DMA Controller* cores and the *Spike Sorting Module*. The *Spike Sorting* part is responsible for the spike detection and classification and consists of the *Multi-channel Spike Detect* core and the *Window-based Online Sorting* core [34].

The role of the *ARM PS* is to initialize the system with a pre-calculation of the required matrices, and to start and control the sorting process. The *ARM PS* and the cores in the FPGA are connected to each other through *AXI4-Lite* and *AXI4-Interconnect* buses, while the *ARM PS* communicates with a host PC through an Ethernet port.

The *Spike Sorting Module* consists of the *Window-based Online Sorting* core and the *Multi-channel Spike Detection* core. The original version of the OSort algorithm [31] works with a single-channel. To process data from a window of the electrode array, the algorithm was modified. The data flow and the structure of the original algorithm are completely redesigned. The neural data are processed in 3×3 spatial windows, so the clustering algorithm works on 9 selected channels at a time.

A. Pre-Processing IIR Filter

A Butterworth IIR third-order zero-phase pre-processing band-pass filter with 500–5000 Hz cut-off frequency is applied to the incoming signal from the Intan RHD2000 recording system for noise reduction and to eliminate low frequency content of the signal. The filter is implemented as two Finite Impulse Response (FIR) filters combined together based on Direct Form I structure with seven 18 bit coefficients in each filter. The FPGA resource requirement of the pre-processing IIR filter on 128-channels is only one 36 k BRAM and one DSP slice.

B. Multi-Channel Spike Detect

The *Multi-channel Spike Detect* architecture can be split into the STD, the NEO and the Realign computation parts, which can be seen in Fig. 6. The signal is sampled at 20 kHz with 128 channels in the proposed system.

The STD computational part is responsible for computing the standard deviation value of the incoming signal using a five seconds window, which contains 100000×128 samples. There is not enough memory for this on the FPGA, therefore the off-chip DDR4 memory is used to store the samples. The average computation can be done continuously in the AVG block, while the subtraction and power operations are done in the STD block. In the NEO computation part the spike detection with Eq. 1 and the threshold calculation using Eq. 2 is done. The NEO signal can be calculated parallel for each channel using the NEO BRAM1, BRAM2 and the Serializer as N-1, N and N+1 inputs.

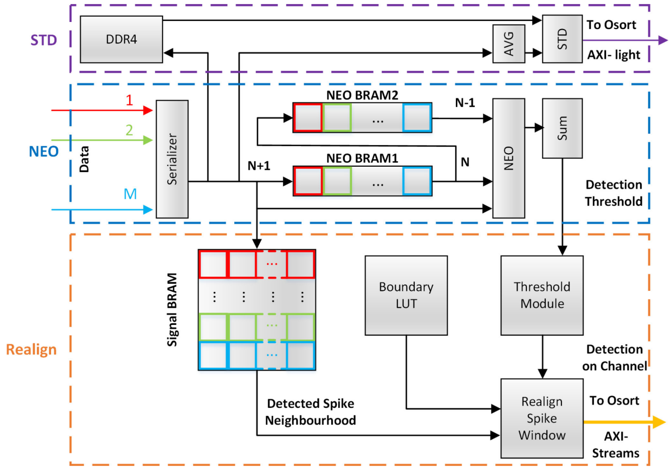


Fig. 6. The architecture diagram of the detection.

In this implementation the value of c_N and N are both 2 (determined experimentally), so the T_N NEO threshold value calculation can be reduced to a simple accumulation. At the first 2 samples the system is not operating, this can be considered as an initialization phase.

The Realign part of the *Multi-channel Spike Detect* is responsible for the following:

- Alignment of the detected spikes in the spike window;
- Determination of the channel containing the maximal amplitude spike in the neighbourhood around the detection;
- The appropriate selection of the 3×3 channels on the electrode array.

The Realign part uses the Signal BRAM, which can store 80 samples for each (128) channel. A spike is 64 samples wide, but the detection is not precise, so 16 more samples are stored, altogether 80. The Threshold Module compares the actual NEO value on each channel with this detection threshold value and when it is lower than the actual NEO value, then a spike is detected. Detection events for the channels are stored in a 128 wide vector. If a detection occurs, then the detection event flag for the actual channel is switched, which indicates that 40 samples later the Signal BRAM will contain the full spike for that channel and can be used for realignment. The Realign Spike Window block uses the absolute value of the spike in the temporal window from the Signal BRAM to find the maximal point for the alignment.

Afterwards, the neighbourhood of the detected spike at the alignment point is extracted from the Signal BRAM, and searched for the maximal value in that 3×3 electrode window. Using this method the channel with the maximal amplitude will be in the center of the electrode window, which is crucial for the

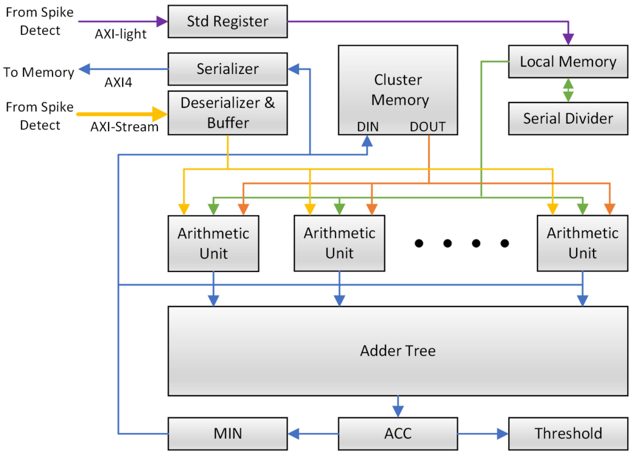


Fig. 7. Architecture of *Window-based Online Sorting* core (modified from [34]).

effectiveness of the comparison performed in the *Window-based Online Sorting* core later. If the channel is at the border of the electrode array, and the spatial window is out of this border, then with the help of the Boundary Look Up Table (BLUT), the 3×3 spatial window can be prepared and sent to classification using AXI-Streams, 9 streams parallel for each channel.

For different electrode array configurations different pre-calculated Boundary LUTs could be used, which needs different bit files on the FPGA. The appropriate one must be chosen for the corresponding electrode array and programmed it to the FPGA.

C. Window-Based Online Sorting

In the *Window-based Online Sorting* core a cluster mean consists of $3 \times 3 \times 64$ data points, which are represented as 16 bit integers. The clusters are represented as fixed point type numbers, all of them are 18 bit with 2 bit fractional part. The single-channel OSort processes spikes with 64 samples using double precision. The memory required for cluster mean storage are less if integer values are used. The T_C classification threshold values are in the range of 10^5, therefore rounding the values does not change the result of the comparison.

In this implementation for the calculation of the T_C classification threshold the c_C clustering correction factor was determined experimentally and set to 0.4 for all simulated data and 0.1 for the in vivo recording, while the number of sample points N_S were 64 for both cases.

In our case a 18 bit wide and 2048 element deep configuration of Xilinx 7 series BRAMs are used, therefore, 32 clusters can be stored in 9 BRAMs. The number of clusters in the proposed system (32×4 electrode array) are expected to be around 100. Therefore the maximal number of clusters is 128 using 36 BRAMs. The computation of the *Window-based Online Sorting* core uses an array of processing elements with 9 (3×3) Arithmetic Units.

The architecture of *Window-Based Online Sorting* core is shown in Fig. 7. Spike data is received from the *Multi-channel Spike Detect* core via AXI-Stream buses and go through a deserializer and a 14 spike matrices deep buffer to handle multiple detections. During a 0.1 ms period (the processing and classification time for one spike matrix is 85.17 μs, see Section IV) only one neuron will fire on average in the vicinity of a recording site (or near a 3×3 window of sites). Therefore, on 128 channels 14 neurons can fire in average in a 0.1 ms period, so a system capable of storing 14 spike matrices will cover most of the cases, even when multiple neurons fire in a short time frame [37].

There are 4 stages, which function is to load the ($3 \times 3 \times 64$ samples) spike matrix and compare it to the cluster means in the Cluster Memory (1), update the chosen cluster mean or create a new cluster (2), compare the cluster means to each other (3) and finally merge clusters (4). The details can be seen in [34].

TABLE I
AREA REQUIREMENTS AND DEVICE UTILIZATION OF THE SYNTHESIZED OVERALL SYSTEM

	Device Utilization			
	FF	LUT	DSP	BRAM
M. Spike Detect Core	10915	42692	6	7
W. OSort Core	6569	8982	54	91
Summarized	17484	51674	60	98
Utilization(%)	3.79%	22.42%	3.47%	15.74%

IV. RESULTS

The proposed *Spike Sorting* and *Processing* blocks were developed using Vivado HLS 2018.3. The prototype MATLAB algorithms are translated to a High Level Synthesis (HLS) based C/C++ solution taken into consideration the architecture described in Section III. The *Processing* block was implemented on a Xilinx ZCU106 SoC FPGA board, which contains a Zynq UltraScale+ XCZU7EV FPGA as PL and a quad-core ARM Cortex-A53 processor as PS.

During the overall validation process, the system was tested with varying number of neurons and noise levels using the hybrid ground truth generator and in vivo cortical measurements. The results were evaluated and visualized on the host computer in MATLAB.

A. Implementation Results

In case of the *Window-based Online Sorting* core, the spike matrix input is loaded trough 9 AXI-Stream buses, the output for the clustering result are mapped to an AXI-Stream bus, while the standard deviation value used in the calculation of the clustering threshold can be updated using an AXI-Light connection.

The available resources on the XCZU7EV FPGA and the resource requirements of the cores can be seen in Table I, which shows that the main resource consumption is the memory requirements (LUT, BRAM). The LUT resource utilization can be reduced if more BRAMs are used instead of LUTRAMs or URAMs can be utilized.

The results of the test measurements on the XCZU7EV FPGA showed that spike matrices can be clustered in 18,005 clock cycles. The latencies of the synthesized cores, which are capable

TABLE II
DETAILED LATENCY OF THE SPIKE SORTING ARCHITECTURE

	Latency (clock cycles)	Trip Count (clock cycles)
Spike Sorting Module		
Load Spike	193	192
Stages 1-4	16712	16704
Read Merge Table	129	128
Total	17034	17024
Spike Detection Module		
Load Channel	129	128
NEO	131	128
Realign	163	154
Total	423	410
STD	12.8 M	12.8 M

TABLE III
CLASSIFICATION ACCURACY (%) OF THE PROPOSED SYSTEM

Neurons	Average SNR (dB)				
	10	8	6	4	3
4	88 ± 9	87 ± 10	59 ± 26	27 ± 43	16 ± 23
8	94 ± 3	93 ± 3	84 ± 13	57 ± 38	28 ± 28
12	92 ± 2	92 ± 1	87 ± 2	59 ± 17	21 ± 18
16	92 ± 3	91 ± 2	88 ± 2	69 ± 9	21 ± 14
20	89 ± 3	90 ± 3	87 ± 3	81 ± 5	39 ± 27
24	90 ± 1	90 ± 1	88 ± 2	80 ± 3	55 ± 12
28	89 ± 3	89 ± 2	87 ± 2	82 ± 7	62 ± 16
32	86 ± 3	86 ± 3	85 ± 2	81 ± 6	64 ± 8

of operating on 200 MHz clock frequency can be seen in Table II. The latency is given in clock cycles, while trip count is the minimum number of times a loop executes. The latencies are grouped by the *Multi-channel Spike Detect* and *Window-based Online Sorting* cores. The STD block is shown separately, because it can run independently of the detection or the classification.

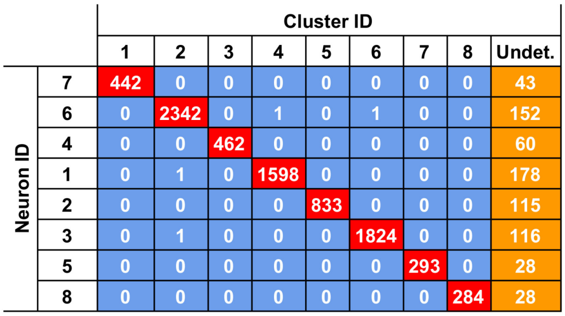
Due to the relatively low operand bit width, the system can operate on 200 MHz clock frequency, so a spike can be detected, realigned and selected in 423 clock cycles, which is 2.12 μs. One spike matrix ($3 \times 3 \times 64$ data points) can be clustered in 17034 clock cycles in the worst case, resulting in 85.17 μs clustering time. Altogether the detection and classification can be done in 87.29 μs, therefore the proposed system can process up to 11456 spike/s.

The spike sorting was also tested offline with the same dataset on the Host PC in MATLAB and the results showed that the average computation time is 7202 μs. Therefore the XCZU7EV FPGA-based system is around 80 times faster.

B. Validation Results

The verification of the classification performance of the proposed system contains multiple datasets with different average Signal-to-Noise Ratios (SNR with 3–10 dB) and different numbers of neurons (4–32), since using the 128-channel neural probe an average of 30 neurons can be separated from a single cortical recording location [37]. The average SNR is calculated using the same method as in [31]. The duration of a dataset was 300 seconds with 20,000 samples/second/channel on a 32×4 electrode array (128 channels). Spike templates corresponding to different neurons were added to arbitrarily chosen channels at random time points. Spike templates were selected randomly from the template database [32] with 80–20 ratio between pyramidal cells and interneurons.

To measure the accuracy of the clustering, spike times of the template neurons and the spike times of the created clusters are matched. Only the spikes placed in the appropriate (best match) cluster are taken into account.



Neuron ID	Cluster ID								
	1	2	3	4	5	6	7	8	Undet.
7	442	0	0	0	0	0	0	0	43
6	0	2342	0	1	0	1	0	0	152
4	0	0	462	0	0	0	0	0	60
1	0	1	0	1598	0	0	0	0	178
2	0	0	0	0	833	0	0	0	115
3	0	1	0	0	0	1824	0	0	116
5	0	0	0	0	0	0	293	0	28
8	0	0	0	0	0	0	0	284	28

Fig. 8. Matching matrix for 8 simulated neurons using the proposed system.

The average and standard deviation of the classification efficiency can be seen in Table III, in case of 128 channels using 300 seconds neural dataset with varying SNR and neuron numbers. In most cases the accuracy is between 80%–90%, except for low neuron numbers and low SNR, where the proposed method has low accuracy. Increasing the number of neurons the accuracy is also increasing for all SNRs.

In case of low neuron numbers on a high-channel count electrode array (128 channels) only a small part of the channels contain spikes and these channels are also used in the STD calculation making the clustering threshold lower, so in case of multi-channel clustering the spike matrices do not meet the clustering condition and falsely new clusters are created.

To visualize the performance of the proposed system a specific table layout is used in which each row represents a spike template of a neuron, while each column represents a sorted cluster and the fields of the matrix contain the number of spikes. A matching matrix for 8 simulated neurons can be seen in Fig. 8, where, for example, the cluster identified with *1* has 442 spikes from spike template *7* . The column named *undetermined* includes those (usually noisy) spikes, which were in clusters discarded due to low number of spikes (<10).

The accuracy of the proposed system using multi-channel spike detect with only single-channel clustering was evaluated. In Table IV the results can be seen, where gray cells denote lower accuracy compared to the proposed system. The results show that the proposed system achieves better performance in most cases, except some low neuron and SNR configurations.

TABLE IV
CLASSIFICATION ACCURACY (%) USING MULTI-CHANNEL SPIKE DETECT AND SINGLE-CHANNEL CLUSTERING

Neurons	Average SNR (dB)				
	10	8	6	4	3
4	64 ± 4	64 ± 4	66 ± 5	64 ± 6	62 ± 5
8	79 ± 4	79 ± 3	78 ± 2	75 ± 2	70 ± 4
12	78 ± 5	77 ± 5	77 ± 4	71 ± 4	69 ± 6
16	78 ± 6	78 ± 6	77 ± 6	74 ± 6	67 ± 5
20	76 ± 7	78 ± 4	75 ± 7	69 ± 7	60 ± 9
24	75 ± 4	75 ± 3	71 ± 4	65 ± 7	54 ± 8
28	80 ± 6	80 ± 5	77 ± 6	70 ± 3	61 ± 5
32	75 ± 3	73 ± 3	72 ± 1	65 ± 2	57 ± 4

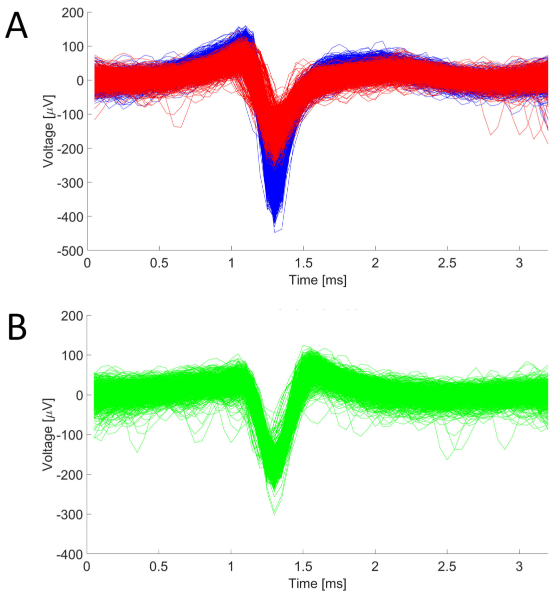


Fig. 9. Partial results of the proposed system on the 5-minute-long in vivo cortical dataset from an anesthetized rat [33]. (A) shows 2 clusters on channel 12. (B) shows a cluster on channel 109.

Recordings with the 128-channel probe typically contains neural activity from 30-40 neurons in average [37], therefore using the proposed system with higher neuron numbers (e.g. 32), a better performance can be achieved than single-channel clustering. In cases above 3 dB SNR the difference between the proposed method and the single-channel version is significant (Student's t-test, $p = 0.002$).

A 5-minute-long in vivo cortical dataset obtained with a 128-channel silicon probe from an anesthetized rat was used for validation. After using the kiloSort [38] spike sorting algorithm offline the resulting clusters were manually curated defining 26 well-separated single unit clusters [33]. Running the proposed system on this cortical recording, 32 clusters were created, which showed 80% similarity using cross-correlation between the results obtained with the method described above and with the proposed system.

The results of the spike sorting can be seen in Fig. 9, which contains some example clusters from two distinct channels.

V. DISCUSSION

Besides the spatial information based online spike sorting, the proposed FPGA-assisted system can support *in vivo* experiments by determining the position of electrodes containing spiking activity in real-time, even shortly after the implantation or the relocation of high-density neural probes. Furthermore, by obtaining various firing properties (autocorrelogram, firing rate, spike width, spike shape, location on the electrode, etc.) and the waveform of the single units extracted in real-time, neural activity in brain areas under examination could be surveyed in a short time, which presumably would be a useful aid for neuroscientists.

The proposed system is capable of clustering more than 11,000 spikes in a second, therefore real-time spike sorting is possible. Based on this and on the resource utilization (Table I), even 1024 clusters and channels (8 times more) could be used.

A. Comparison to Other Systems

The original OSort algorithm [31] was compared to our system using the hybrid ground truth signal generator [32]. Altogether six neurons were simulated on channels (two neurons on each) 33, 42, and 43. The original OSort algorithm was running in parallel on 128 channels to be comparable to our system. The proposed system created only 2 clusters on each appropriate channel (F-Score 100% [39]), while the original OSort (with the default parameters) also found the same 6 clusters on the appropriate channels, but made another 735 false clusters too (F-Score 0.8%) from noisy, overlapping spikes and due to the crosstalk between channels.

In addition to the original software solution described above many FPGA-based spike sorting algorithms can be found in the literature [12]–[19], [21]–[23] but only three of them [12], [21], [22] are using a version of the original OSort, and only one OSort-based solution is capable of multi-channel classification [22].

In [12] the original OSort algorithm was implemented on FPGA, but it cannot be used for multi-channel spike sorting, and the system can only process single-channel neural recordings containing only 3 neurons with the accuracy of 89%.

A real-time unsupervised FPGA-based spike sorting system is presented in [21], which applies NEO for spike detection and a parallel architecture based on the OSort algorithm for classification. The presented system works only on single-channel measurements and was tested on the Easy1_noise01, Easy2_noise005, and the Difficult1_noise005 data containing only 3 neurons from the WaveClus dataset [40] achieving F-Score accuracy of 94.93%, 96.94%, and 91.50% respectively.

In [22] a real-time template matching multi-channel spike sorting system is presented. In this system also a NEO-based spike detection is applied, but the OSort algorithm is used only to define the most commonly occurring waveforms offline. These predetermined waveforms (only three different types) are stored in a template memory. During the classification the detected spikes are compared to these templates, so this solution is not fully unsupervised. Furthermore in this spike sorting system the multi-channel operation means only that each channel processed individually and does not take into account the spatial correlation between the channels. The Easy1_Noise and Difficult1_Noise005 data containing only 3 neurons were used from the WaveClus dataset [40] for single-channel simulations to test this system achieving F-Score accuracy of 93.3% and 93.9% respectively. Our system is processing only multi-channel recordings, therefore applying single-channel simulations from the WaveClus dataset is pointless.

B. Scalability and Limitations

The scalability of the system depends on the number of channels in the electrode array (in this case 128) and on the spatial window size used in the classification. In our system the spatial window size is fixed (3 × 3), therefore only the number of channels can be changed, which impacts the maximum number of clusters. The processable number of clusters can be further increased using more BRAMs or URAMs. Creating more clusters increases the processing time, therefore more window-based OSort Cores can be utilized in parallel to maintain the same classification speed. Based on the resource requirement of the system even 4 window-based OSort Core can be implemented on the XCZU7EV FPGA (Table I).

In the actual state of the proposed system every 3 × 3 spike matrix is compared with every cluster mean, even if it comes from a completely different electrode channel. To address this issue in the future the electrode coordinates can be checked during the classification, which requires only an additional 1-byte information (channel number) per cluster.

VI. Conclusion

In this paper an FPGA-based implementation of the OSort algorithm for unsupervised online window-based spike sorting system is presented. The results show that our architecture can be implemented on a mid-range SoC FPGA device running on 200 MHz, which is capable of the detection and classification of the incoming spikes on 3 × 3 spatial windows in 87.29 μs, or more than 11000 spikes/s above 80% classification accuracy for 32 simulated neurons with 4–10 dB SNR. Furthermore it can be concluded that the neural spike sorting on the FPGA is 80 times faster than the identical algorithm running offline with the same dataset on the Host PC in MATLAB.

The proposed architecture is currently capable of sorting 128 channels of neural data in 3 × 3 spatial windows in real-time using the SoC FPGA. Based on the validation results it can be concluded that the clustering efficiency of the system can reach an average accuracy of 86% for high neuron numbers (16–32) above 3 dB SNR, while the single-channel clustering version achieves 74% average accuracy in the same cases.

The proposed system was tested with generated neural data, and real data recorded with the Intan measurement system. In the future our plan is to test the system *in vivo*.

Acknowledgment

Ministry of Human Capacities, Hungary is acknowledged. Also Thematic Excellence Programme 2019 by the National Research, Development and Innovation Office is acknowledged.

References

[1] G. Buzsaki, “Large-scale recording of neuronal ensembles,” *Nature Neurosci.*, vol. 7, no. 5, pp. 446–451, May 2004.

[2] J. Csicsvari *et al.*, “Massively parallel recording of unit and local field potentials with silicon-based electrodes,” *J. Neurophysiol.*, vol. 90, no. 2, pp. 1314–1323, Aug. 2003.

[3] A. Berenyi *et al.*, “Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals,” *J. Neurophysiol.*, vol. 111, no. 5, pp. 1132–1149, Mar. 2014.

[4] M. S. Lewicki, “A review of methods for spike sorting: The detection and classification of neural action potentials,” *Network*, vol. 9, no. 4, pp. R53–78, Nov. 1998.

[5] H. G. Rey *et al.*, “Past, present and future of spike sorting techniques,” *Brain Res. Bulletin*, vol. 119, no. Pt B, pp. 106–117, Oct. 2015.

[6] S. Fujisawa *et al.*, “Behavior-dependent Short-term assembly dynamics in the medial prefrontal cortex,” *Nature Neurosci.*, vol. 11, no. 7, pp. 823–833, Jul. 2008.

[7] L. R. Hochberg *et al.*, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” *Nature*, vol. 442, pp. 164–171, Jul. 2006.

[8] D. M. Taylor *et al.*, “Direct cortical control of 3D neuroprosthetic devices,” *Science*, vol. 296, no. 5574, pp. 1829–1832, Jun. 2002.

[9] J. J. Jun *et al.*, “Fully integrated silicon probes for high-density recording of neural activity,” *Nature*, vol. 551, no. 7679, pp. 232–236, Nov. 2017.

[10] R. Fiath *et al.*, “Large-scale recording of thalamocortical circuits: In vivo electrophysiology with the two-dimensional electronic depth control silicon probe,” *J. Neurophysiol.*, vol. 116, no. 5, pp. 2312–2330, Nov. 2016.

[11] B. C. Raducanu *et al.*, “Time multiplexed active neural probe with 1356 parallel recording sites,” *Sensors (Basel)*, vol. 17, no. 10, Oct. 2017, Art. no. 2388.

[12] S. Gibson *et al.*, “An FPGA-based platform for accelerated offline spike sorting,” *J. Neurosci. Methods*, vol. 215, no. 1, pp. 1–11, Apr. 2013, doi: 10.1016/j.jneumeth.2013.01.026.

[13] B. Yu *et al.*, “Real-time FPGA-based multichannel spike sorting using Hebbian eigenfilters,” *IEEE J. Emer. Sel. Topics Circuits Syst.*, vol. 1, no. 4, pp. 502–515, Dec. 2011.

[14] W. J. Hwang *et al.*, “Efficient architecture for spike sorting in reconfigurable hardware,” *Sensors (Basel)*, vol. 13, no. 11, pp. 14860–14887, Nov. 2013.

[15] Y. L. Chen *et al.*, “An efficient VLSI architecture for multi-channel spike sorting using a generalized hebbian algorithm,” *Sensors (Basel)*, vol. 15, no. 8, pp. 19830–19851, Aug. 2015.

[16] H. Y. Chen *et al.*, “An efficient hardware circuit for spike sorting based on competitive learning networks,” *Sensors (Basel)*, vol. 17, no. 10, Sep. 2017, Art. no. 2232.

[17] V. Karkare, S. Gibson, and D. Marković, “A 75-mu W, 16-channel neural spike-sorting processor with unsupervised clustering,” *IEEE J. Solid-State Circuits*, vol. 48, no. 9, pp. 2230–2238, Sep. 2013.

[18] S. Luan *et al.*, “Compact standalone platform for neural recording with real-time spike sorting and data logging,” *J. Neural Eng.*, vol. 15, no. 4, 2018, Art. no. 046014.

[19] J. Park, G. Kim, and S. Jung, “A 128-channel FPGA based real-time spike-sorting bidirectional closed-loop neural interface system,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 12, pp. 2227–2238, Dec. 2017.

[20] R. E. Hampson *et al.*, “A wireless recording system that utilizes bluetooth technology to transmit neural activity in freely moving animals,” *J. Neurosci. Methods*, vol. 182, no. 2, pp. 195–204, Sep. 2009.

[21] D. Valencia and A. Alimohammad, “A real-time spike sorting system using parallel osort clustering,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1700–1713, Dec. 2019

[22] D. Valencia and A. Alimohammad, “An efficient hardware architecture for template matching-based spike sorting,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 3, pp. 481–492, Jun. 2019.

[23] J. Dragas *et al.*, “Complexity optimization and high-throughput low-latency hardware implementation of a multi-electrode spike-sorting algorithm,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 149–158, Mar. 2015.

[24] C. M. Lopez *et al.*, “A neural probe with up to 966 electrodes and up to 384 configurable channels in 0.13 mu m SOI CMOS,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 3, pp. 510–522, Jun. 2017.

[25] J. Scholvin *et al.*, “Close-packed silicon microelectrodes for scalable spatially oversampled neural recording,” *IEEE Trans. Biomed. Eng.*, vol. 63, no. 1, pp. 120–130, Jan. 2016.

[26] T. J. Blanche *et al.*, “Polytrodes: High-density silicon electrode arrays for large-scale multiunit recording,” *J. Neurophysiol.*, vol. 93, pp. 2987–3000, 2005

[27] C. Gold *et al.*, “On the origin of the extracellular action potential waveform: A modeling study,” *J. Neurophysiol.*, vol. 95, pp. 3113–3128, 2006

[28] R. Fiath *et al.*, “Fine-scale mapping of cortical laminar activity during sleep slow oscillations using high-density linear silicon probes.” *J. Neurosci. Methods*, vol. 316, pp. 58–70, 2019

[29] G. Dimitriadis *et al.*, “Why not record from every channel with a CMOS scanning probe?” 2018, *bioRxiv*, doi: 10.1101/275818.

[30] V. Viswam *et al.*, “Optimal electrode size for multi-scale extracellular-potential recording from neuronal assemblies,” *Frontiers Neurosci.*, vol. 13, 2019, Art. no. 385.

[31] U. Rutishauser *et al.*, “Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo,” *J. Neurosci. Methods*, vol. 154, no. 1–2, pp. 204–224, Jun. 2006.

[32] L. Schaffer *et al.*, “FPGA-based real-time multi-channel neural dataset generation,” in *Proc. Eur. Conf. Circuit Theory Des.*, Sep. 2017, pp. 1–4, doi: 10.1109/ECCTD.2017.8093235.

[33] R. Fiath *et al.*, “A silicon-based neural probe with densely-packed low-impedance titanium nitride microelectrodes for ultrahigh-resolution in vivo recordings,” *Biosensors Bioelectron.*, vol. 106, pp. 86–92, Jan. 2018.

[34] L. Schaffer *et al.*, “FPGA-based neural probe positioning to improve spike sorting with OSort algorithm,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2017, pp. 1375–1378.

[35] S. Gibson, J. W. Judy, and D. Markovic, “Comparison of spike-sorting algorithms for future hardware implementation,” in *Proc. IEEE EMBS Conf.*, 2008, pp. 5015–5020.

[36] P. Bartho *et al.*, “Characterization of neocortical principal cells and interneurons by network interactions and extracellular features,” *J. Neurophysiol.*, vol. 92, no. 1, pp. 600–608, Jul. 2004.

[37] R. Fiath *et al.*, “Slow insertion of silicon probes improves the quality of acute neuronal recordings,” *Scientific Rep.*, vol. 9, no. 1, 2019, Art. no. 111.

[38] M. Pachitariu *et al.*, “Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels,” 2016, *bioRxiv*, doi: 10.1101/061481.

[39] C. J. V. Rijsbergen, *Information Retrieval*. 2nd ed., London, U.K.: Butterworth, 1979.

[40] R. Quiroga *et al.*, “Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering, “*Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, 2004.