15.2 A 1024-Channel 0.00029mm²/ch 74nW/ch Online Spatial Spike-Sorting Chip with Event-Driven Spike Detection and Self-Organizing Map Clustering

Arash Akhoundi[1], Yawende Landbrug[1], Pumiao Yan[2], E. J. Chichilnisky[2], Boris Murmann[3], Dante Gabriel Muratore[1]

[1]Delft University of Technology, Delft, The Netherlands
[2]Stanford University, Stanford, CA
[3]University of Hawaii, Honolulu, HI

Next-generation brain-computer interfaces will enable motor and speech decoding in humans [1-3] and improve our understanding of brain function [4]. To achieve this requires high-density multi-electrode arrays (HD-MEA) [5,6]. This leads to massive amounts of raw data that must be reduced on-chip to enable wireless operation [7]. Spike sorting (SS) assigns spikes to putative neurons and can reduce the data rate substantially because only the neuron ID needs to be transmitted when a spike occurs. Prior art focuses on improving the scalability and power efficiency of on-chip SS [8-15]. However, they either require a large input buffer [12-14], use temporal features (TF) that do not scale well to multi-channel systems [8-13], access the entire clustering memory for every spike [11-13], or use high power [8-11] and area [8-10]. This work uses event-driven spike detection and spatial spike sorting to deal with these challenges and achieve 74nW/ch, 0.00029mm²/ch, and <50µs latency with 1024 channels. This represents a >10× improvement in power and area efficiency with 3× more channels than prior art (Fig. 15.2.6).

SS detects spikes in the raw data, projects them into a feature space for easier separation, and identifies clusters to assign spikes to putative neurons. Figure 15.2.1 shows the main challenges with SS: 1) The spike detector operates at the input rate and uses a large input buffer which dominates the system power consumption, but because spikes are sparse, it mostly processes noise samples. 2) Typically, SS relies on TFs of the distinct spike waveforms recorded from different neurons due to their unique distance to the recording electrode; however, their separability severely degrades in HD-MEA because the electrode-neuron distance is not necessarily unique for each neuron. Unfortunately, because previous approaches relied on single-channel datasets [16] or datasets with only 8 neurons [17], the extent of this problem was not apparent. Also, TFs are sensitive to electrode drift. 3) Conventional clustering algorithms need to retrieve the entire cluster memory for every spike because TFs carry no information about the cluster-ID. The impact of this issue can be limited with a geometry-aware algorithm [14], but it still used TFs for clustering and needed to retrieve 5% of the memory to account for drift.

We address these challenges with 3 techniques: 1) We use a wired-OR-based compressive ADC (wOR) and a spike pre-detector to perform event-driven spike detection and reduce the input memory size (Fig. 15.2.2). The wOR approach in [18] outputs a sample only for channels recording a unique value (channel event). Prior work has shown that this analog front-end can achieve ~100× compression with only 500nW/ch and minimal information loss [19,20]. Since most of the time channels record noise around the baseline, unique channels likely record spikes. Hence, spike detection can be performed by tracking the number of events in a given time for every channel. The valid counter spike pre-detector (VC-SPD) increases when there is a channel event and decreases otherwise. It flags a spike when it reaches a threshold TH_{SPD}=4 (fixed to minimize hardware cost). This event-driven SPD can achieve very high sensitivity but suffers from large false positives. A non-linear-operator-based spike detector (NEO-SD) solves this issue by removing false positives from the output of the VC-SPD. This approach achieves >96% accuracy over all tested datasets with ground truth (Fig. 15.2.2) while reducing the average activity rate of the VC-SPD and NEO-SD by 35× and 100× and only requiring a 4S/ch input memory (30× less than typical input buffers). Notably, the SD processes spike samples >75% of the time, compared to <8% when raw data is directly input. 2) We use spatial features (SF) that leverage redundant spike recordings from multiple electrodes to extract the neuron's location. For each spike, the largest recorded amplitude in the central channel (A_c) and the amplitudes of N neighboring channels (A_i) are used to calculate the magnitude ($|\Delta_{c,s}|$) and angle ($\angle\Delta_{c,s}$) of a vector from the central channel to the estimated neuron location. The estimated location is then referred to global coordinates $(x,y)_{SF}$ using the central channel coordinates (x_c,y_c) (equations in Fig. 15.2.3). Due to their different locations, two neurons recorded by the same central channel will generate different spike patterns across the HD-MEA, leading to separable SFs. Since SFs rely solely on the spike amplitude, they are more robust to noise than TFs requiring the entire waveform. Also, SFs scale to large channel counts compared to TFs when benchmarking with datasets with many neurons (Fig. 15.2.3 left). 3) A modified self-organizing map clustering algorithm (SOM-clust) uses the SF space to create a structural map of neuron locations. The cluster memory represents a grid of locations in the MEA, initialized to all electrode positions and increased by positions between the electrodes, adjusted based on the MEA density (4096 elements for 1024 channels in this work). During training, the goal is to reposition these grid elements to match the locations of the neurons and eliminate superfluous elements so that there is one element per neuron. SOM-clust calculates the Euclidean distance between the SF and the grid elements to find the best matching unit (BMU) and adjusts its location $(x,y)_{BMU}$ towards $(x,y)_{SF}$ using a weighted average. It keeps track of how often elements are selected as BMU using a frequency counter and removes those below a threshold T_{batch} in any given batch (t_{batch}=6s). T_{batch} is adaptively determined based on the number of detected spikes in one batch (N_{spike}) and the number of remaining active elements ($N_{element}$), assuming that the firing rates among neurons are relatively consistent [21] (equations in Fig. 15.2.3). Training happens on-chip and finishes when no elements are removed for two consecutive batches (typically after 5 batches, Fig. 15.2.3 right). During clustering, when a new spike is detected, the algorithm finds the BMU and updates its location. Notably, this approach does not require retrieving all elements from memory since the SF allows us to predict the most likely BMU. Also, since the cluster location is updated for every spike, it automatically tracks electrode drift.

A 1024-channel spatial SS chip was designed with standard digital cells to process 8b wOR data from an FPGA-based emulator (Fig. 15.2.4). It operates in two alternate modes: (1) spike detection and (2) feature extraction and clustering. Each mode is clock-gated when inactive to save power. In mode (1), the amplitude and channel address of the wOR events are transmitted to the chip using 19 input pads: 10 pads for the channel address, 8 pads for the amplitude, and 1 pad for the event-valid signal. wOR events are sequentially stored in the 4kB input memory and processed by 1024 VC-SPDs. When a spike is pre-detected, a shared NEO-SD retrieves the last four valid samples from the input memory (x[n-3:n]) and checks the maximum location. NEO-SD runs when the maximum is aligned to x[n-2] to avoid double counting and ensure the maxima in the neighboring channels needed for the SFs are present in the shallow input memory. If a spike is detected, its channel address is stored in a rate adapter FIFO for the next stage. In mode (2), SFs from the channels stored in the FIFO are extracted only if they are central channels. $(x,y)_{SF}$ uses 6b for the coarse location (initial grid) and 5b for the fine location ($\Delta_{c,s}$) for each axis. The cluster memory stores the grid elements and has 2^{12} words of 18b (10b for (x,y) fine location, 7b for the frequency counter, 1b active flag). The MSBs in $(x,y)_{SF}$ address the closest element to the detected spike, avoiding the need to load the entire cluster memory. If this element was deactivated during training, the SOM-clust retrieves neighboring elements until it finds the BMU. On average, fewer than 4 elements (0.1% of cluster memory) need to be retrieved for each spike. If the coarse location of the BMU changes after the update, there are two scenarios: (1) it points to an active element, which means both elements should be merged, or (2) it points to a deactivated element, which means that the element has drifted significantly. In the latter case, the new word is activated, and the old one is deactivated to keep track of MEA drift. The chip uses 1 clock cycle per event and can process 256 wOR events in one sampling period in mode (1), assuming >4× wOR compression. Every spike uses 16 cycles for processing in mode (2), and it is assumed that at most 16 neurons are firing a spike simultaneously. Hence, for a 20kHz sampling frequency, the main clock is 10.24MHz. The chip can handle ~300spk/s/ch without overflow.

Our SS requires multi-channel datasets for validation since SFs cannot be extracted from a single channel. Figure 15.2.5 compares the similarity to Kilosort [25] results for *ex vivo* primate retina recordings where a ground truth does not exist (ExVivo), and the accuracy achieved by this work with (1) software SS algorithms [22] using the artificial datasets for which a ground truth exists (MEArec [23,24]) and (2) the only on-chip SS that used a multi-channel dataset (Neuropixel [17]). The achieved accuracy is competitive with other software SS and better than on-chip SS. The achieved compression in Fig. 15.2.5 for the wOR and the SS depends on the spike rate and the number of neurons and is always >1000×.

The prototype chip in 40nm CMOS has a core area of 0.3mm² and 0.00029mm²/ch (Fig. 15.2.7). The area is dominated by the input and cluster memories (59%). The minimum supply voltage is 0.72V, which leads to a measured total power of 76µW and 74nW/ch while processing (Fig. 15.2.5), dominated by the SD (51%). Compared to prior art (Fig. 15.2.6), this work achieves the lowest area and power per channel and the largest number of channels and is the only work validated with multi-channel datasets with hundreds of neurons.

Acknowledgement:
The authors thank Z.Y. Chang for the technical support and L. Grosberg, C. Rhoades, A. Sher, and A. Litke for providing access to the experimental data. This publication is part of the project Dutch Brain Interface Initiative (DBI²) with project number 024.005.022 of the research program Gravitation, which is financed by the Dutch Ministry of Education, Culture and Science (OCW) via the Dutch Research Council (NWO).

2025 IEEE International Solid-State Circuits Conference (ISSCC) | 979-8-3315-4101-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISSCC49661.2025.10904780

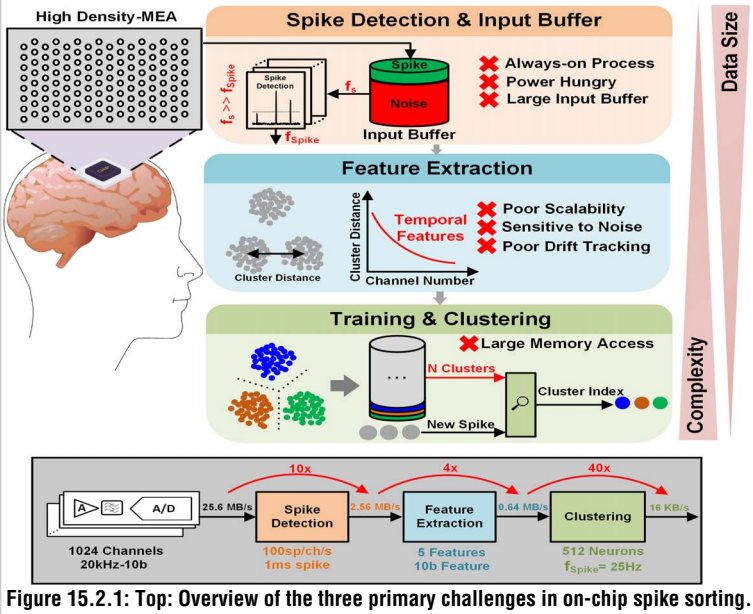


Figure 15.2.1: Top: Overview of the three primary challenges in on-chip spike sorting. Bottom: Typical data reduction rates within spike sorting steps.

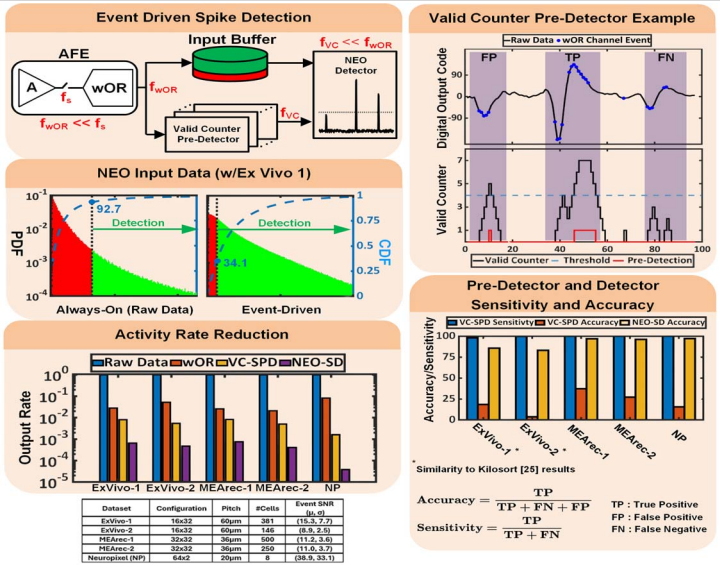


Figure 15.2.2: Event-driven spike detection with wired-OR, valid counter spike pre-detection (VC-SPD) and nonlinear energy operator (NEO). Activity rate, sensitivity and accuracy for different datasets (details in bottom left).

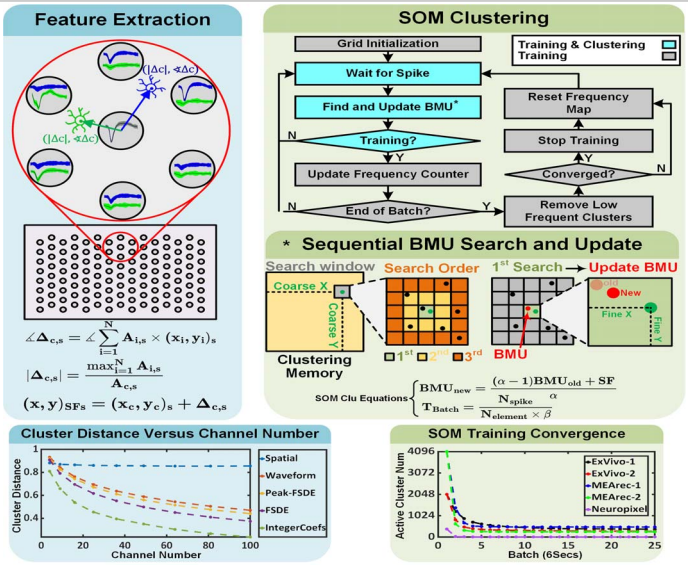


Figure 15.2.3: Left: Spatial features from redundant spike recordings and a scalability comparison with temporal features. Right: SOM clustering algorithm flow and its convergence for different datasets.

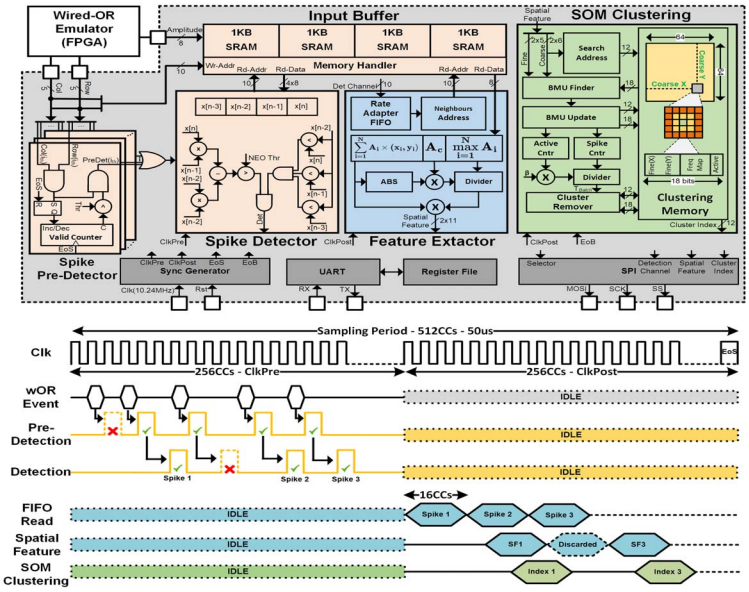


Figure 15.2.4: System architecture and timing diagram of the spike sorting chip.

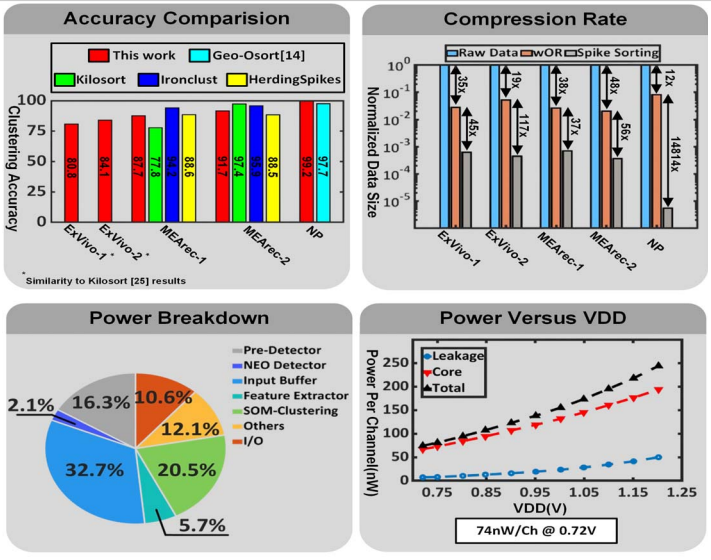
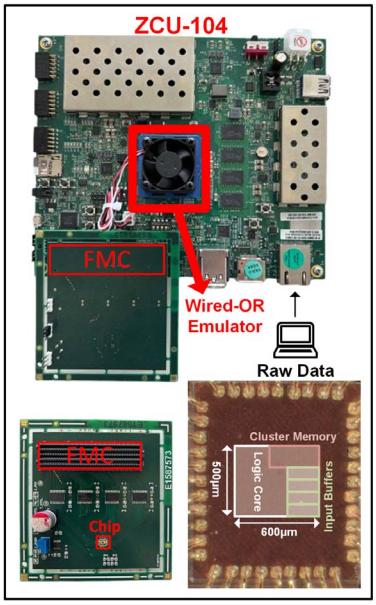


Figure 15.2.5: Top: SS accuracy and compression for different multichannel datasets. Bottom: Power breakdown and power versus VDD of the fabricated SS chip (power of FPGA wired-OR emulator is not included).

Design		TVLSI 2019 [12]	TBCAS 2021 [10]	TVLSI 2022 [13]	TBCAS 2022 [11]	JSSC 2023 [14]	This Work
Algorithm	Spike Detection	Integer Coefficients	Absolute Thresholding	Absolute Thresholding	NEO	NEO	Valid Counter and NEO
	Feature Extraction	Integer Coefficients	FSDE	Waveform	Adaptive Filter	Peak-FSDE	Spatial Features
	Clustering	Modified K-means	Perturbed K-means	Cross Correlation	Configurable	Geo-Osort	Modified SOM
Number of Channels		128	4	64	16	384	1024
Training		On-chip	On-chip	Offline	On-chip	On-chip	On-chip
Feature Space		Temporal	Temporal	Temporal	Temporal	Temporal Ch-Sel	Spatial
Dataset	Name	Quiroga	Quiroga	Quiroga	Quiroga	Quiroga, NP	NP, MEArec, ExVivo
	Type	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic, Real
	# Cells	<5	<5	<5	<5	<10	<10, 250~500, 146~381
	# Channels	1	1	1	1	1, 128	128, 1024, 512
Clustering Latency		>1.84ms	1ms	>1.36ms	>66.7µs	<401µs	<50µs
Clustering Accuracy (%)	Quiroga	72	93.2	85	94.1	89.5	-
	NP	-	-	-	-	97.7	99.7
	MEArec	-	-	-	-	-	87.7, 91.7
	ExVivo	-	-	-	-	-	80.8, 84.1*
Technology		65nm	180nm	180nm	22nm	22nm	40nm
Core Voltage		0.54V	1.5V	1.8V	0.5V	0.59V	0.72V
Sampling Frequency		25KHz	24KHz	25KHz	25KHz	30KHz	20KHz
Input Resolution		9-bits	Analog	8-bits	9-bits	12-bits	8-bits
Power per Channel		0.175 µW	4.68 µW	1.74 µW	2.79 µW	1.78 µW	0.074 µW
Area per Channel		0.003 mm²	1.032 mm²	0.047 mm²	0.014 mm²	0.0013 mm²	0.00029 mm²

*Similarity to Kilosort [25] results

Figure 15.2.6: Performance comparison with recent state-of-the-art online SS chips.



Technology		40nm
Area	Chip	1.4 mm x 1.2 mm
	Core	500 µm x 600 µm
#Channels		1024
Core V_{DD}		0.72 V
Operating Freq.		10.24 MHz
On-Chip SRAM		14 kB
Power		76.3 µW
Latency		<50 µs

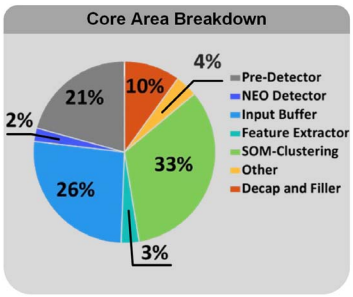


Figure 15.2.7: Test setup, die photo, and chip core area breakdown.

References:
[1] S.L. Metzger, et al., "A high-performance neuroprosthesis for speech decoding and avatar control," *Nature*, vol. 620, no. 7976, pp. 1037-1046, Aug. 2023.
[2] F.R. Willett, et al., "A high-performance speech neuroprosthesis," *Nature*, vol. 620, no. 7976, pp. 1031-1036, Aug. 2023.
[3] L.R. Hochberg, et al., "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372-375, May 2012.
[4] A.C. Paulk, et al., "Large-scale neural recordings with single neuron resolution using Neuropixels probes in human cortex," *Nat. Neurosci.*, vol. 25, no. 2, pp. 252-263, Feb. 2022.
[5] N.A. Steinmetz, et al., "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings," *Science*, vol. 372, no. 6539, Apr. 2021.
[6] Y. Wang, et al., "Implantable intracortical microelectrodes: reviewing the present with a focus on the future," *Microsyst. Nanoeng.*, vol. 9, p. 7, Jan. 2023.
[7] N. Even-Chen, et al., "Power-saving design opportunities for wireless intracortical brain-computer interfaces," *Nature Biomedical Engineering*, Aug. 2020.
[8] D. Valencia and A. Alimohammad, "Partially binarized neural networks for efficient spike sorting," *Biomed. Eng. Lett.*, vol. 13, no. 1, pp. 73-83, Feb. 2023.
[9] C. Seong, et al., "A Multi-Channel Spike Sorting Processor With Accurate Clustering Algorithm Using Convolutional Autoencoder," *IEEE TBioCAS*, vol. 15, no. 6, pp. 1441-1453, Dec. 2021.
[10] H. Hao, et al., "A 10.8 µW neural signal recorder and processor with unsupervised analog classifier for spike sorting," *IEEE TBioCAS*, vol. 15, no. 2, pp. 351-364, Apr. 2021.
[11] S.M.A. Zeinolabedin, et al., "A 16-Channel Fully Configurable Neural SoC With 1.52 µW/Ch Signal Acquisition, 2.79 µW/Ch Real-Time Spike Classifier, and 1.79 TOPS/W Deep Neural Network Accelerator in 22 nm FDSOI," *IEEE TBioCAS*, vol. 16, no. 1, pp. 94-107, Feb. 2022.
[12] A.T. Do, et al., "An Area-Efficient 128-Channel Spike Sorting Processor for Real-Time Neural Recording With 0.175 µW/Channel in 65-nm CMOS," *IEEE TVLSI*, vol. 27, no. 1, pp. 126-137, Jan. 2019.
[13] F. Kalantari, et al., "Hardware-Efficient, On-the-Fly, On-Implant Spike Sorter Dedicated to Brain-Implantable Microsystems," *IEEE TVLSI*, vol. 30, no. 8, pp. 1098-1106, Aug. 2022.
[14] Y. Chen, et al., "An Online-Spike-Sorting IC Using Unsupervised Geometry-Aware OSort Clustering for Efficient Embedded Neural-Signal Processing," *IEEE JSSC*, vol. 58, no. 11, pp. 2990-3002, Nov. 2023.
[15] Y. Chen, et al., "A 384-Channel Online-Spike-Sorting IC Using Unsupervised Geo-OSort Clustering and Achieving 0.0013mm2/Ch and 1.78 µW/ch" *ISSCC*, pp. 486-487, 2023.
[16] R.Q. Quiroga, et al., "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661-1687, Aug. 2004.
[17] "Sorting Comparison Results," Neuropixels Datasets, 26-Feb-2016. [Online]. Available: http://phy.cortexlab.net/data/sortingComparison/. [Accessed: 10-May-2023].
[18] D.G. Muratore, et al., "A Data-Compressive Wired-OR Readout for Massively Parallel Neural Recording," *IEEE TBioCAS*, vol. 13, no. 6, pp. 1128-1140, Dec. 2019.
[19] M. Jang, et al., "A 1024-Channel 268 nW/pixel 36x36 µm2/ch Data-Compressive Neural Recording IC for High-Bandwidth Brain-Computer Interfaces," *IEEE Symp. VLSI Technology and Circuits, 2023.*
[20] P. Yan, et al., *"Data Compression Versus Signal Fidelity Tradeoff in Wired-OR Analog-to-Digital Compressive Arrays for Neural Recording," IEEE Trans. Biomed. Circuits Syst.*, vol. PP, July 2023.
[21] Z. Zhang and T. G. Constandinou, "Firing-rate-modulated spike detection and neural decoding co-design," *J. Neural Eng.*, vol. 20, no. 3, May 2023.
[22] J. Magland, et al., "SpikeForest, reproducible web-facing ground-truth validation of automated neural spike sorters," *Elife*, vol. 9, p. e55167, May 2020.
[23] A.P. Buccino and G. T. Einevoll, "MEArec: A fast and customizable testbench simulator for ground-truth extracellular spiking activity," *Neuroinformatics*, vol. 19, no. 1, pp. 185-204, Jan. 2021.
[24] A.P. Buccino, et al., "SpikeInterface, a unified framework for spike sorting," *Elife*, vol. 9, Nov. 2020.
[25] M. Pachitariu, et al., "Spike sorting with Kilosort4," *Nat. Methods*, vol. 21, no. 5, pp. 914-921, May 2024.

979-8-3315-4101-9/25/$31.00 ©2025 IEEE