High Throughput Neuromorphic Brain Interface with CuO_x Resistive Crossbars for Real-time Spike Sorting

Yuhan Shi[1], Akshay Ananthakrishnan[1], Sangheon Oh[1], Xin Liu[1], Gopabandhu Hota[2],Gert Cauwenberghs[2],Duygu Kuzum[1]
[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA, USA
[2]Department of Bioengineering, University of California San Diego, La Jolla, CA, USA. Email: dkuzum@ucsd.edu

***Abstract*—**Real-time spike sorting with large data throughput is essential for studying neural dynamics and brain-machine interfaces. Neural recordings from high-density multi-electrode arrays that consist of hundreds of electrodes impose stringent demands on spike sorting hardware regarding data transmission bandwidth and computation complexity. That leads to an urgent need for specialized hardware with high throughput, low power, and latency. Here, we present a real-time spike sorting processor that utilizes high-density BEOL-integrable CuO_x resistive crossbars to perform in-memory spike segregation. We experimentally demonstrate, for the first time, efficient hardware implementation of spike sorting from *in vivo* extracellular recordings with high accuracy. Our neuromorphic interface promises substantial performance gains (~1000× less area, ~200× less power, 4.8μs latency for sorting 100 channels) for *in vivo* real-time spike sorting.

I. Introduction

Spike sorting, the process of separating the firing activities of individual neurons from neural recordings, is crucial for studying neural circuits [1]. It has been widely used in various practical applications such as brain-machine interfaces (BMIs) [2] and neural prosthetics [3]. Conventionally, spike sorting is performed offline by transmitting raw digitized signals from neural recording to a nearby computer. However, transmitting vast amounts of neural recording data from advanced high-density microelectrode arrays (HDMEAs) leads to excessive power dissipation which poses a serious risk of damaging surrounding tissues [4]. A 100-channel microelectrode array with a 16-bit ADC operating at 30kHz sampling frequency generates 3MSamples/s and dissipates mW-level power to nearby tissues. More importantly, the high latencies associated with offline spike sorting make it impractical for closed-loop BMIs that interface with the brain via real-time feedback. An 8h recording experiment using a 100-channel microelectrode array would accumulate ~200GB of data [5], demanding at least a few hours to sort the recorded spikes in software [6]. These drawbacks highlight the need for low-power and high throughput hardware that can be integrated with HDMEAs to sort many neural spikes in real-time.

Although FPGA-based real-time spike sorting has been demonstrated, most implementations are inefficient in terms of area and power consumption as they require hundreds of block RAM and DSP units [7, 8]. In contrast, large-scale resistive switching random access memory (RRAM) arrays have enabled in-memory processing of machine learning algorithms [9], leading to parallelized and energy-efficient neuromorphic hardware implementations. However, to the best of our knowledge, no studies yet exist on the development of RRAM-based brain interfaces for real-time spike sorting.

In this work, we designed a high throughput neuromorphic brain interface based on CuO_x crossbar arrays that can sort neural spikes in real-time. First, we developed a low-temperature fabrication process for CMOS-integrable high-density CuO_x crossbars. We extensively studied the switching characteristics of CuO_x resistive memory devices. We then developed a spike sorting algorithm that can be directly mapped onto CuO_x crossbars for sorting multi-channel spikes. Low amplitude neural signals (few μVs) from an implanted neural probe were amplified and digitized before performing spike sorting based on template matching with the conductance-encoded templates stored in CuO_x crossbars (**Fig. 1**). We experimentally demonstrated our CuO_x crossbar arrays can sort simulated synthetic spikes as well as extracellular recordings from *in vivo* animal experiments with high accuracy close to ideal software implementation. Based on experimental results, we estimated that our approach can sort 100-channel recordings within 4.8μs with ~1000× reduction in chip area, ~200× reduction in power, and ~50× less energy per channel compared to state-of-the-art FPGA and microcontroller implementations.

II. CuO_x Resistive Crossbars

We fabricated 16 × 16 Au/CuO_x/Au resistive crossbars (**Fig. 2a-c**) with 1μm line width (**Fig. 2d**) at the wafer-level using the CMOS-compatible process outlined in **Fig. 2e**. A 2μm pitch between the Au wordlines (WLs) and bitlines (BLs) achieved high areal densities. A 70nm-thick CuO_x switching layer was deposited using reactive sputtering and patterned with lithography. For long-term reliability, the arrays were passivated with a 300nm-thick SiO_2 layer. The Au/CuO_x/Au devices were reliably switched using V_{SET} = ~1.5V and V_{RESET} = ~ -0.7V (**Fig.3a**) with low cycle-to-cycle variations (**Fig.3b**). The high resistance (HRS) and low resistance (LRS) states of the device were ~1GΩ and ~100Ω, resulting in an ON/OFF ratio ~ 10^7 (**Fig. 3c**). Device-to-device variations in switching voltages and LRS/HRS resistances are shown in **Fig. 4a,b**. The measured SET and RESET latencies (**Fig. 5a,b**) and switching endurance meet the requirements for BMI applications as neuron templates do not require frequent updates. Both LRS and HRS show long retention (**Fig. 5c**), sufficient for storing templates for real-time spike sorting.

III. Mapping Spike Templates to Crossbar

To perform real-time spike sorting, we developed a template matching algorithm that sorts incoming spikes by comparing their shapes to stored neuron templates (**Fig. 6a**). Each neuron had a template matrix $T_n = [T_{n,1}, T_{n,2}, \ldots, T_{n,m}]$, where column $T_{i,j}$ represented the template for neuron i corresponding to channel j. The matrix was normalized by its Frobenius Norm ($T_n/\|T_n\|_F$). **Fig.6b** illustrates the method by showing a simplified example for the classification of two neurons (n=2) from three-channel recordings (m=3). The same

2021 IEEE International Electron Devices Meeting (IEDM) | 978-1-6654-2572-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEDM19574.2021.9720712

methodology can be used to classify a larger number of neurons recorded across hundreds of channels. We defined the neural signal $V(t) = [V_1(t), V_2(t)\ldots, V_m(t)]$, where $V_i(t)$ is the recorded signal from channel j. For each neuron i and channel j, $V_i(t)$ was convolved with the template ($T_{i,i}$) to produce convolution traces $C_{n,m}(t)=V_m(t)*T_{n,m}$. The traces corresponding to each neuron were summed up (**Fig.6c**) to give $C_n(t) = \sum_1^m C_{n,m}(t)$. To cluster the spikes, we applied a threshold and assigned the spikes to the neuron having the largest $C_n(t)$ (**Fig.6d**). For benchmarking using different neural implant technologies, we implemented our algorithm on two neural recordings: (1) a synthetic "NeuroNexus-32" data [10] containing extracellular recordings of twelve neurons from the NeuroNexus-32 probe (**Fig.7a**) with the ground truth, and (2) "real" spikes from *in vivo* animal experiments recorded with the NeuroFITM probe [11] (**Fig.8a**) where predictions from offline Kilosort algorithm [6] was considered as the ground truth. **Fig.7b,c,** and **Fig.8b,c** show representative templates and 30kHz recordings of multiple channels from these two probes. The templates for NeuroNexus-32 and NeuroFITM were obtained from biophysical simulations [10] and Kilosort. Representative single-channel recordings with clustered neuron spikes are shown for NeuroNexus-32 (**Fig.7d,e)** and NeuroFITM (**Fig.8d,e**). For each neuron, the shape of the clustered spike waveforms closely matched their respective original templates (**Fig.7b, Fig.8b**). The sorting performance of our algorithm was quantified using the F1 score (in %) given by 2TP/ (2TP+FP+FN), where TP, FP, and FN denote the true positive, false positive, and false-negative outcomes. The spike predictions from our algorithm were in great agreement with the ground truth, enabling us to classify eleven out of twelve neurons in the NeuroNexus-32 neural data with an F1 score > 90%, as well as the neurons in the NeuroFITM recordings (**Fig.9**). This performance could be retained in hardware by quantizing the templates to at least 4-bit resolution (**Fig.10**). To process hundreds of spikes per second, it would be necessary to adopt a multi-core architecture (**Fig.11a**) where each core consists of a crossbar that stores the templates for a specific set of neurons (**Fig. 11b**). By convolving the voltage spike inputs on WLs (V_{WLi}) with the templates stored as crosspoint conductances (G_{ij}), each crossbar core can perform template matching (BL currents $I_{BLi}=\sum G_{ij}V_{WLi}$) in parallel (**Fig. 11c**). The classification result is obtained by adding the BL currents for each neuron i.e., $I_n = \sum_1^m I_{BLn,j}$ from all m channels and then assigning the spike to the neuron with the maximum I_n.

IV. Hardware Demonstration of Spike Sorting

A custom PCB board was used to address the WLs and BLs of the wire-bonded CuO_x crossbar (**Fig. 12a,b**). The as-fabricated devices had initial resistances greater than 500kΩ (**Fig.12c**). Neuron templates were quantized and mapped onto crossbar columns by programming the devices using a $V_{dd}/2$ write scheme where the selected WL and BL were biased to $V_{dd}/2$ and $-V_{dd}/2$ and all other unselected lines were grounded. **Fig.12d,e** shows four representative templates implemented in the crossbar. When all WLs were biased high (V_{WLs}=0.25V), the weighted-sum BL currents (I_{sum}) increased proportionately with the number of LRS devices in the columns, thereby validating the accuracy of crossbar convolutions.

Using the programmed templates, we evaluated the sorting performance on NeuroNexus-32 and NeuroFITM recordings. Neural recordings encoded as 8-bit voltage pulse trains were fed into the WLs and I_{sum} were obtained on the BLs. **Fig.13a,c** show the NeuroNexus-32 and NeuroFITM recordings and the hardware spike sorting results implemented to sort representative three neurons from the NeuroNexus-32 data and two neurons from the NeuroFITM data. **Fig.13b,d** present convolution traces generated by the CuO_x crossbar for spikes highlighted with rectangular boxes in **Fig.13a,c**. For each spike, the neuron with the highest peak in the convolution trace was assigned to the spike. The shapes of convolution traces produced by the CuO_x crossbars matched closely with software.

Based on the hardware spike sorting results obtained over a 100ms time window (**Fig.13**), we calculated F1 scores on the entire 30s-wide recordings in both neural data and compared them with software predictions. **Fig.14** shows neurons could be sorted with high mean accuracy (~92.5% for NeuroNexus-32, ~94.6% for NeuroFITM). Overall, our crossbar-based spike sorting hardware can achieve ~1000× smaller (area/channel) [12] and consume ~200× less power [7] compared to state-of-the-art spike sorting hardware (**Table I**). Unlike previous works that rely on sequential processing, our crossbars can process multi-channel electrode recordings in a highly parallelized manner. We estimate that twelve CuO_x crossbar (256×256) cores can process recordings from 100-channel within 4.8µs, consuming ~30-50× less energy [7,12]. These performance gains make real time spike sorting possible using our crossbars for high throughput BMI applications.

V. Conclusion

We demonstrated a high throughput neuromorphic brain interface for real-time spike sorting based on CuO_x resistive crossbars. These crossbars were fabricated using a low-temperature reactive sputtering process, enabling BEOL-integration with CMOS-based spike detection circuits. Hardware implementation of template matching using CuO_x crossbars accurately classified spikes from individual neurons recorded *in vivo*, offering substantial performance gains in area, power, latency, and energy for neural probes with high channel counts. Our work will pave the way towards in-memory computing based real-time spike processors for next-generation closed-loop brain interfaces.

Acknowledgment

This work was supported by the Office of Naval Research (N000142012405) and the National Science Foundation (ECCS-1752241, ECCS-2024776).

References

[1] L. Luo, et al., Neuron, vol. 98, no. 2, pp. 256-281, 2018. [2] U. Chaudhary, et al., Nat.Rev. Neurol, vol. 12, no. 9, pp. 513-525, 2016. [3] J. L. Collinger et al., The Lancet, vol. 381, no. 9866, pp. 557-564, 2013. [4] S. Kim, et al., IEEE T Neur Sys Reh, vol. 15, no. 4, pp. 493-501, 2007. [5] S. Gibson, et al., J. Neurosci. Methods, vol. 215, no. 1, pp. 1-11, 2013. [6] M. Pachitariu, et al., NIPS, vol. 29, pp. 4448-4456, 2016. [7] P. K. Wang et al., PLoS One, vol. 14, no. 11, p. e0225138, 2019. [8] L. Schäffer et al., IEEE. Trans. Biomed, vol. 68, no. 1, pp. 99-108, 2020. [9] D. Ielmini et al., Nat. Electron., vol. 1, no. 6, pp. 333-343, 2018. [10] A. P. Buccino et al., Neuroinformatics, vol. 19, no. 1, pp. 185-204, 2021. [11] X. Liu et al., Nat. Neurosci., vol. 24, no. 6, pp. 886-896, 2021. [12] S. Luan et al., J. Neural Eng., vol. 15, no. 4, p. 046014, 2018.

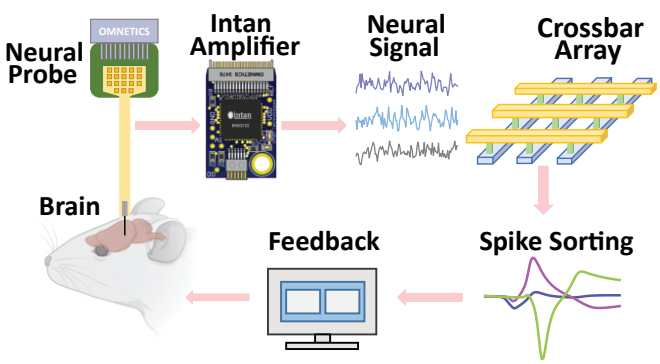

Fig.1 Proposed neuromorphic brain interface based on CuO_x crossbar array for spike sorting. Neural signals recorded by multichannel neural probe are amplified and digitized using an Intan amplifier and ADC respectively. CuO_x crossbar array performs spike sorting in real-time and relays relevant information to the brain via feedback.

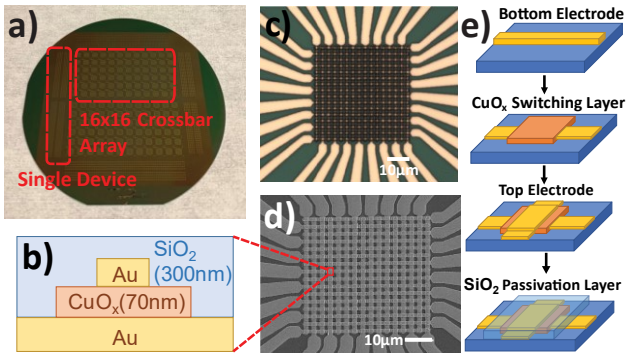

Fig.2 a) Wafer containing the fabricated 16×16 CuO_x crossbar arrays and single devices. b) Device cross-section highlighting the 70nm CuO_x resistive switching layer sandwiched between 100nm Au electrodes. A 300nm SiO_2 passivation layer is deposited on top of the stack. c) Optical and d) SEM images of 16×16 crossbar with 4μm^2 cross point. d) Fabrication process for CuO_x -based single devices and 16×16 crossbar.

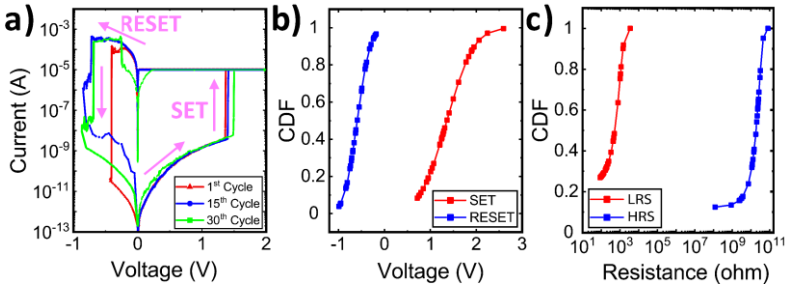

Fig.3 a) DC switching characteristics of single devices for 30 cycles. b) Cumulative distribution function (CDF) of SET (1V to 2.5V) and RESET (-1V to -0.2V) voltages. c) CDF of HRS (100MΩ to 100GΩ) and LRS (100Ω - 1kΩ) resistances.

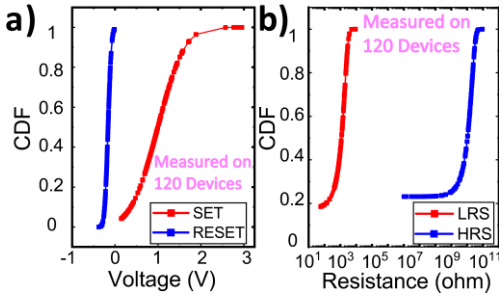

Fig.4 CDF of the switching voltages and HRS/LRS resistances measured across 120 devices on the wafer.

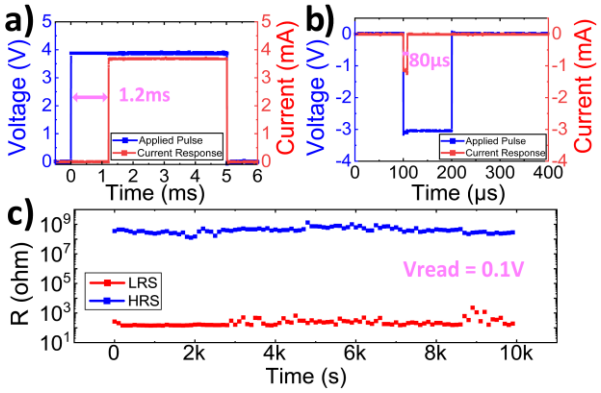

Fig.5 Applied voltage pulses and transient current responses for a) SET and b) RESET operations. c) Retention characteristics. Device resistance was monitored intermittently using 0.1V read pulses.

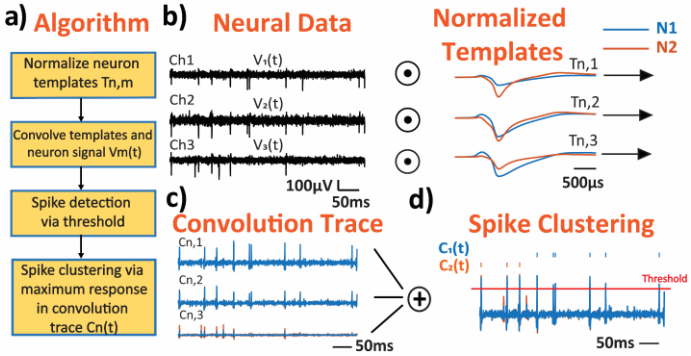

Fig.6 a) Overview of the proposed template matching based spike sorting algorithm. b) Convolving neural recordings i.e., voltage traces with normalized templates N1 and N2. c) Summing the convolution traces ($C_{n,m}(t)$) corresponding to each neuron . d) Thresholding and assigning spikes to neurons N1 or N2 based on whether $C_1(t) > C_2(t)$ (assign to N1) or $C_1(t) < C_2(t)$ (assign to N2).

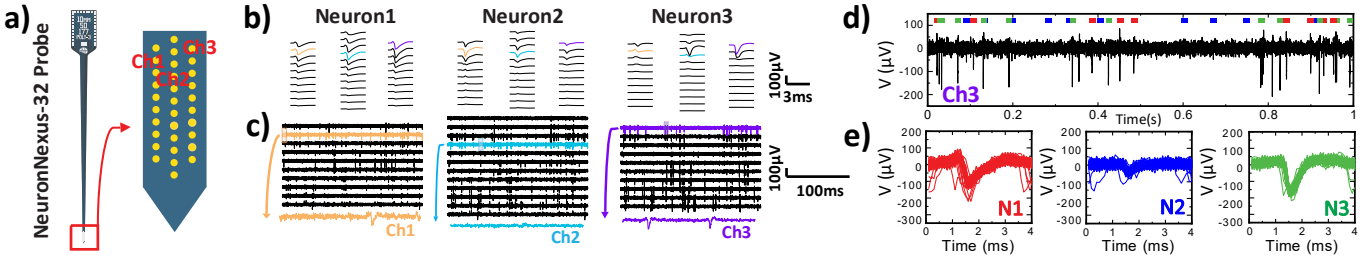

Fig.7 a) Schematic of 32-channel NeuroNexus-32 probe. b) Representative templates for three neurons. c) Representative 300ms-recordings from 32 channels of the NeuroNexus-32 probe, showing Ch1,2,3 used for sorting N1,N2,N3. d) Example recordings from Ch3 with predicted spike train marked in colored squares and e) Clustered spikes for N1 to N3 for Ch3.

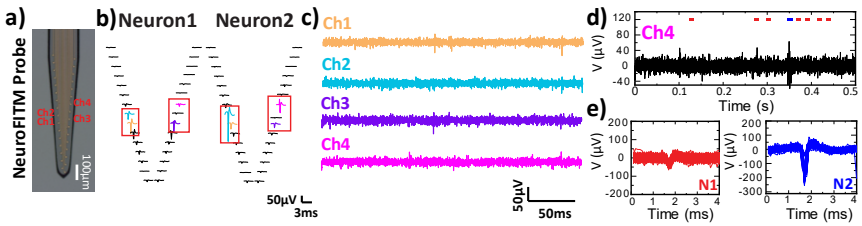

Fig.8 a) Image of a 32-channel NeuroFITM probe. b) Representative templates for the two neurons from Ch1 to Ch4. c) Representative 300ms-recordings from four channels. d) Example recordings from Ch4 with predicted spike train marked in colored squares and e) Clustered spikes for N1 and N2 for Ch4.

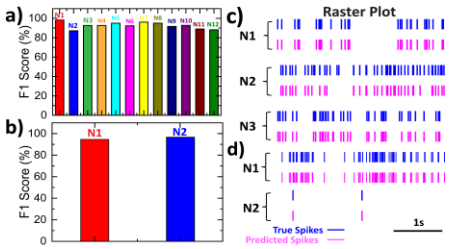

Fig.9 F1 scores and spike assigment for NeuroNexus-32 (a,c), NeuroFITM (b,d) neural data.

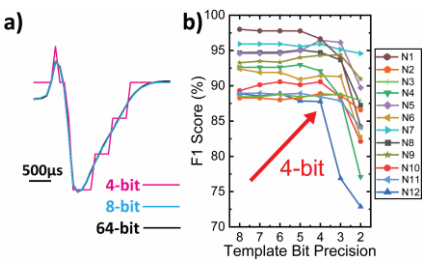

Fig.10 a) Template precision: full (64-bit) vs. quantized (8-bit and 4bit). b) F1 score (%) as a function of template precision. 4-bit quantized templates are used in hardware experiments.

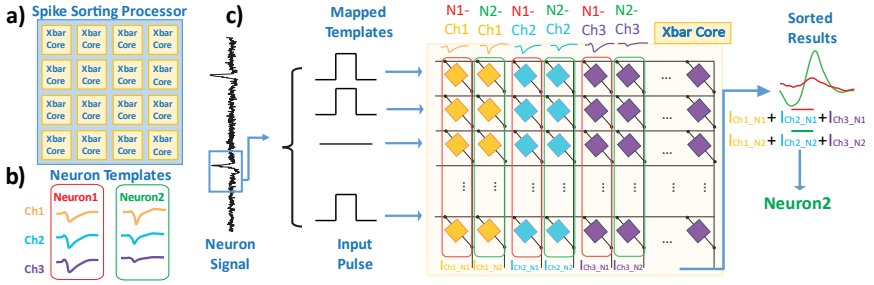

Fig.11 a) Real-time spike sorting processor with multiple crossbar cores. b) Representative templates of two neurons. c) Crossbar spike sorting: each crossbar column stores a neuron template. 8-bit digitized neural signals are provided as voltage inputs and weighted-sum currents from convolutions are obtained on the BLs. Neuron-wise aggregation of channel currents determines the sorting result.

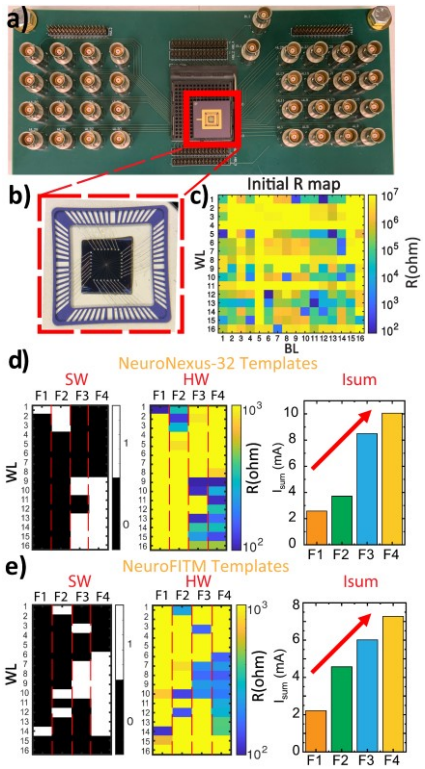

Fig.12 a) Custom PCB board to access individual WLs and BLs of the CuO_x crossbar for the write and read operations. b)16×16 crossbar wire-bonded onto a PGA package. c) Initial resistance map of a 16×16 CuO_x crossbar. Left column: four representative binarized (black=0 and white=1) filters (F1-F4) from d) NeuroNexus-32 and e) NeuroFITM. middle column: programmed crossbar columns implementing these filters. right column: I_{sum} at V_{WLs}=0.25V for four filters.

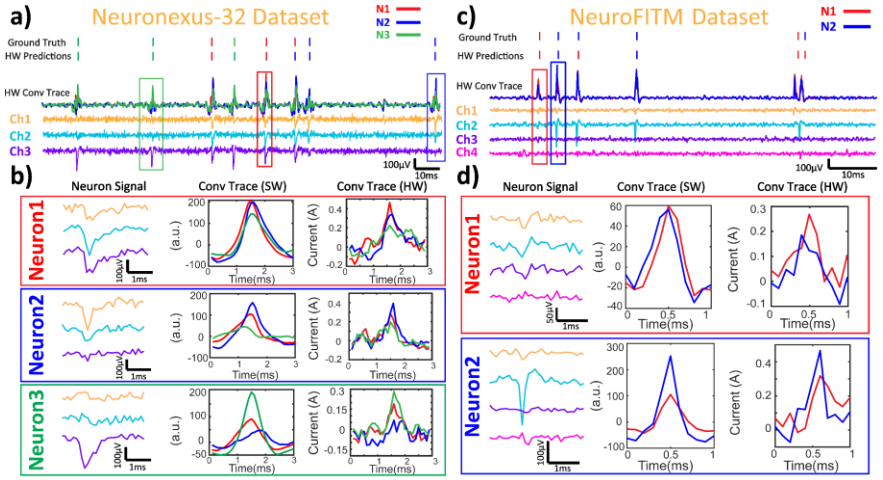

Fig.13 a) NeuroNexus-32: Ch1,2,3 are used to classify neurons N1, N2, N3. A segment of recordings from Ch1 to Ch3 and predicted hardware (HW) convolution (conv) traces for three neurons. b) Representative spike sorting results for N1-N3 showing convolution implemented in HW agrees with the software (SW) implementation. c) NeuroFITM: Ch1,2,3,4 are used to classify neurons N1, N2. Segments of recordings from Ch1 to Ch4 and predicted HW conv traces. d) Representative spike sorting results for N1, N2 implemented in HW agrees with the SW implementation.

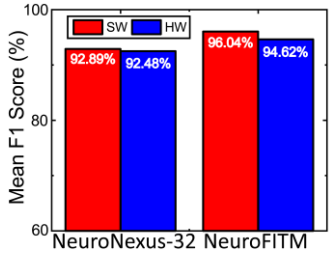

Fig.14 Accuracy obtained on the NeuroNexus-32 and NeuroFITM data from software (SW) and hardware (HW) experiments.

Table I Performance Benchmarking

Reference	This Work	[7]	[12]
Hardware	Crossbar	FPGA	Microcontroller
Recording Data	Simulated, in-vivo experiments	in-vivo experiments	in-vivo experiments
No. of channel	32	1	32
Area/Channel (mm²)	8e-4	> 10	0.78
Power/Channel (mW)	2.15	460	3.11
Sorting Latency (μS)	4.8 per 100 channel	0.72 per channel	169 per channel
Energy/Channel (nJ)	10.3	331.2	525.6

Table I. Benchmarking our results against previous works [7], [12] in terms of hardware type, recording data used in the studies, channel count, area/channel, power/channel, sorting latency and energy/channel.