An Area-Efficient 128-Channel Spike Sorting Processor for Real-Time Neural Recording With 0.175 μW/Channel in 65-nm CMOS

Anh Tuan Do, *Member, IEEE*, Seyed Mohammad Ali Zeinolabedin, *Member, IEEE*, Dongsuk Jeon, *Student Member, IEEE*, Dennis Sylvester, *Fellow, IEEE*, and Tony Tae-Hyoung Kim, *Senior Member, IEEE*

***Abstract*—This paper presents a power- and area-efficient spike sorting processor (SSP) for real-time neural recordings. The proposed SSP includes novel detection, feature extraction, and improved K-means algorithms for better clustering accuracy, online clustering performance, and lower power and smaller area per channel. Time-multiplexed registers are utilized in the detector for dynamic power reduction. Finally, an ultralow-voltage 8T static random access memory (SRAM) is developed to reduce area and leakage consumption when compared to D flip-flop-based memory. The proposed SSP, fabricated in 65-nm CMOS process technology, consumes only 0.175 μW/channel when processing 128 input channels at 3.2 MHz and 0.54 V, which is the lowest among the compared state-of-the-art SSPs. The proposed SSP also occupies 0.003 mm^2/channel, which allows 333 channels/mm^2.**

***Index Terms*—Low power, neural recording, real-time recording, spike sorting.**

I. Introduction

MULTIELECTRODE intracranial recording technology offers exceptionally high spatial and temporal signal resolutions needed for neural prosthetic development and neuroscience research [1], [2], [23], [24]. Recorded brain signals are further analyzed to identify their source neurons. This process is called spike sorting and is a useful part for various applications such as brain–machine interfaces (BMI) and neural prostheses, especially when ultralow power feature is required [3]. Spike sorting processes usually consist of three main steps: 1) detection and alignment; 2) feature extraction (FE) and dimensionality reduction (DR); and 3) clustering as shown in Fig. 1(a). After sorting, every detected spike is assigned to a specific neuron ID. This neuron ID indicates the source from which the detected spike was fired.

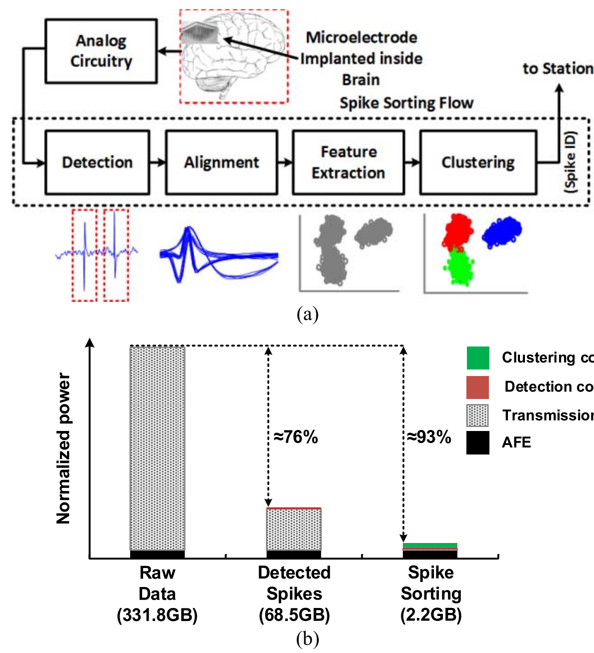


Fig. 1. (a) Typical brain signal recording system consisting of analog front end and spike sorting back end. (b) Sample estimated power and memory size for one-day recording using a 128-channel recording system. It is assumed that analog front end consumes 10 μW/channel, communication dissipates 1 nJ/bit, spike detection requires 2 μW/channel, and clustering spends 4.65 μW/channel [8].

Spike sorting has been generally run offline: raw digitized signals from multielectrode recording front end are transmitted to a nearby computer for further analysis. However, with the development of high-density microelectrodes arrays [4], this approach faces fundamental limitations because of the high data rate and high power of the multichannel recording systems. For instance, a 128-channel, 25-ksamples/s recording system using 8-bit analog-to-digital converters produces 25.6 Mb/s. When considering the power density requirement of 277 μW/mm^2 on implantable devices [5], transmitting this amount of data becomes extremely challenging.

Manuscript received March 5, 2018; revised July 9, 2018 and August 21, 2018; accepted September 25, 2018. *(Corresponding author: Tony Tae-Hyoung Kim.)*

A. T. Do was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with the Institute of Microelectronics, A*STAR, Singapore 138634 (e-mail: tonydo0701@gmail.com).

S. M. A. Zeinolabedin was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with Technische Universität Dresden, 01069 Dresden, Germany (e-mail: ali.zeinolabedin@tu-dresden.de).

D. Jeon is with Seoul National University, Seoul 08826, South Korea (e-mail: djeon1@snu.ac.kr).

D. Sylvester is with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: dmcs@umich.edu).

T. T.-H. Kim is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: thkim@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2018.2875934

Fig. 1(b) illustrates a sample power estimation showing that spike sorting processor (SSP) can significantly reduce the required memory, the data rate, and the power consumption. Therefore, BMI and implantable neural devices with a large number of channels become feasible.

Integrated SSPs have been utilized in various multichannel neural signal processing research [1]–[3]. Integrating the whole or a part of the spike sorting flow on-chip provides significant data reduction. Moreover, on-chip SSP provides real-time processing and lowers hardware system complexity, which allows SSP applicable to a wide range of operating scenarios.

Various implementations of SSPs have been reported [2]–[10]. In [2], only detection and FE included on the 128-channel design. The SSPs in [6] and [7] implemented detection, alignment, and FE for 64 channels. However, clustering requiring complex hardware is performed offline [2], [6], [7]. The first 16-channel SSP performing online unsupervised clustering algorithm is reported in [8]. However, the clustering algorithm processes the original spike data without FE, which requires a large memory and high power consumption. Therefore, the number of channels is relatively restricted. In [9], a complete SSP is designed for 32 channels. This SSP can support up to 43 channels/mm^2. However, this design is not verified by test chips. Another complete SSP inclusive of analog-to-digital conversion is introduced in [10] with the channel count of 128. However, the system is only verified in the form of a field-programmable gate array with several empirical algorithms.

In this paper, we present a 128-channel SSP with novel detection architecture, FE operators, and improved clustering. The proposed techniques improve the accuracy and reduce power consumption significantly. In addition, several low-power techniques such as ultralow voltage operation and ultralow power 8T SRAM are employed to further enhance the power and energy efficiency. This paper is an extended version of the conference paper presented in [13].

The rest of this paper is organized as follows. The proposed detection, FE, and clustering algorithms are explained in Section II. Section III discusses the test chip implementation. Measurement results are presented in Section IV, followed by conclusion in Section V.

II. Proposed Detector, Feature Extractor, and Improved K-Means Algorithm

A. Detection and Alignment

Spike detection is an essential step in the spike sorting flow for separating spikes from neural signal [12]. In general, spike detection techniques enhance the signal-to-noise ratios through preemphasizing and thresholding. After applying preemphasizing, a threshold value checks whether a spike is available or not. Here, the sampled waveform (e.g., 48 samples) of the detected spike is considered as the spike detection output. Most commonly used detection algorithms are absolute thresholding (AT), nonlinear energy operator (NEO) [14], and a hybrid algorithm [15]. There are tradeoffs between the detection accuracy (i.e., the probability of detection), the probability of false alarm, and the power consumption of the circuit [16]. Once a spike is detected, a window of the input signal is captured for alignment to mitigate sampling jitter and noise effects [12]. In general, detected spikes are aligned using maximum value, minimum value, or maximum slope.

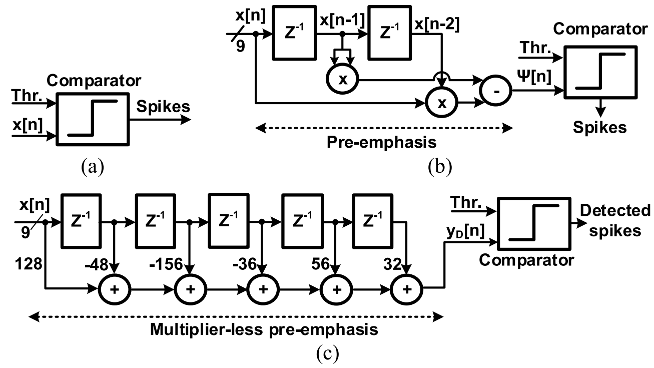


Fig. 2. Circuit diagrams of (a) AT, (b) NEO, and (c) proposed integer coefficient preemphasis.

Both AT and NEO are popular thanks to the simple circuit structures and the high detection accuracy [17], [18]. In AT, a spike is detected when the magnitude of the neural signal is larger than a threshold level [11]. The threshold level can define empirically. NEO takes into account the differences between neural signals ($x(n)$, $x(n-k)$, and $x(n+k)$) as given in

$$\psi[x(n)] = x^2(n) - x(n-k) \times x(n+k) \tag{1}$$

where $x(n)$ is the input data point from one channel at discrete time t_n (k is window size). This operator amplifies the spike signal while suppressing noise and other low-frequency components. Therefore, it provides better detection accuracy compared to AT at the cost of additional power. Like AT, the NEO ($\psi[x(n)]$) is compared with a threshold level (Thr_NEO) driven by

$$\text{Thr\_NEO} = 4\frac{1}{N}\sum_{1}^{N}\psi[x(n)] \tag{2}$$

to make a decision. Thr_NEO is also an empirical parameter based on the characteristics of the neural signal.

In this paper, we propose an integer coefficient preemphasis method as expressed in

$$\begin{aligned} y_D[n] = {} & 128x(n) - 48x(n-1) - 156x(n-2) \\ & - 36x(n-3) + 56x(n-4) + 32x(n-5) \end{aligned} \tag{3}$$

where $y_D[n]$ is the detection result and $x(n)$ is the input neural signal. The proposed filter acts as a short-window convolution to capture spike-like waveforms. Its parameters are empirically chosen to provide better accuracy performance compared to both NEO and AT, especially for noisy signals. The integer coefficients require only shift and addition without multiplication. Therefore, it can reduce power compared to NEO. The coefficients can be empirically updated if the characteristics of the input signal vary substantially. Fig. 2 illustrates the circuit diagrams of AT, NEO, and the proposed method.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

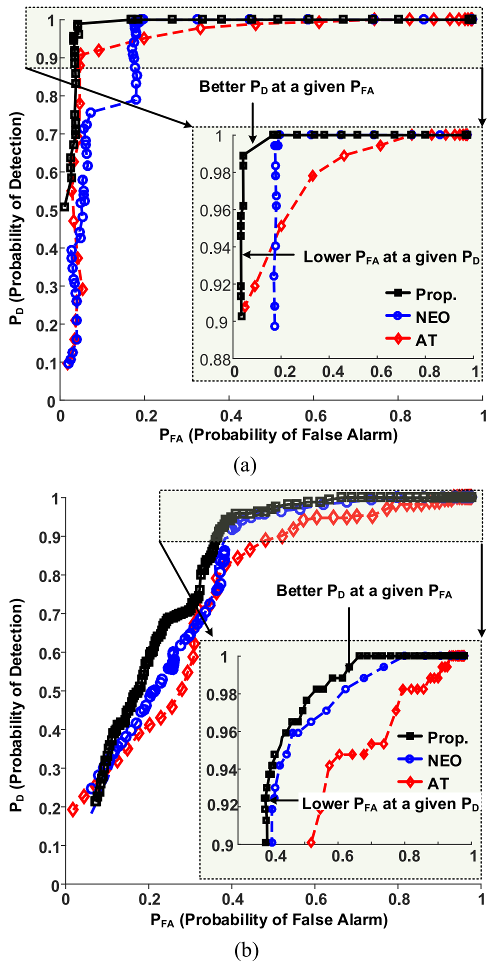


Fig. 3. Performance comparison of AT, NEO, and the proposed algorithm over two data sets with different noise levels. (a) Clean data set. (b) Noisy data set.

Fig. 3 demonstrates the receiver operating characteristic (ROC) of AT, NEO, and the proposed method using two different data sets with different noise levels from [11]. The ROC values are characterized by the detection probability (P_{D}) and the false alarm probability (P_{FA}) for each threshold value while it is swept from low to high. Apparently, the threshold should be determined carefully with a tradeoff between P_{D} and P_{FA}. The proposed method offers higher P_{D} at a given P_{FA}, and a lower P_{FA} at a given P_{D} compared to AT and NEO.

For spike alignment, we employ the maximum slope since it is biologically significant and results in superior clustering accuracy [12]. In this paper, every captured spike waveform with 48 samples is shifted so that index 11 has the maximum slope as shown in Fig. 4. Note that the location of the maximum can be chosen empirically as long as all the key features are captured in the alignment window. After alignment, each detected spike is stored in a memory for FE.

B. Feature Extraction and Dimensionality Reduction

FE along with DR extracts major components of the detected spikes and condenses them into several most important features for fast and accurate clustering. Prevalent FE methods are principal component analysis (PCA), discrete

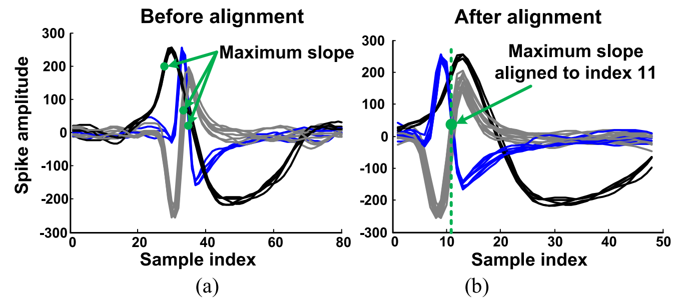


Fig. 4. Detected spikes are aligned to the maximum slope whose index is 11. (a) Before alignment. (b) After alignment.

wavelet transform (DWT), discrete derivatives (DDs), and integral transform (IT) [12]. Based on the study in [12] and [17], PCA is one of the most accurate approaches. However, it uses the whole waveform data, requiring large memories and high power. PCA and DWT compute the same number of coefficients as the detected spikes and then reduce them to several coefficients before clustering. However, they are not suitable for spike sorting systems with high channel count due to their high complexity and large memory requirements.

The DD approach in

$$dd_{\delta}(n) = x(n) - x(n-\delta) \tag{4}$$

is inspired by DWT and much more hardware-friendly. In (4), $x(n)$ is the spiked sample, δ is the time step, and n is the sample index. In [7], δ is set to 1, 3, and 7. Thus, for each raw sample $x(n)$, three features are generated: $dd_1(n)$, $dd_3(n)$, and $dd_7(n)$. The number of features is therefore three times of the spike samples. Since seven samples are uniformly selected for each spike, a total of 21 features are generated per spike.

The IT method computes the area under a spike curve both in positive and negative phases. It generates only two features per spike and can be easily implemented in hardware. However, it has limited accuracy and some parameters are required to be calculated through the training phase.

In this paper, we design the FE block using an integer-coefficient filter as described in the following equation for complexity and power reduction:

$$y_{\mathrm{FE}}[n] = 8x(n) - 2x(n-1) - 6x(n-2) - 4x(n-3). \tag{5}$$

Subsequently, four samples are selected from each spike to reduce the dimension. Since the sample selection greatly impacts on the clustering accuracy, the sampling method must be chosen carefully. In this paper, the indices of 8, 11, 18, and 25, and the filter coefficients are chosen through MATLAB simulations for the best clustering accuracy. Fig. 5 shows how the designed FE filter amplifies the key spike components and the significance of the selected indices. Note that the threshold crossing happens before the maximum slope of the spike, and therefore, the whole shape of the spike comes after the detection point. The selected indices are very close to the local peaks and index 11 corresponds to the maximum slope of the raw spike and the global peak of the filtered spike.

Fig. 6 shows how the proposed filter (y_{FE}) can improve the clustering accuracy in the feature spaces. In order to visualize it, index 11 ($x(11)$) and index 18 ($x(18)$) are selected for two

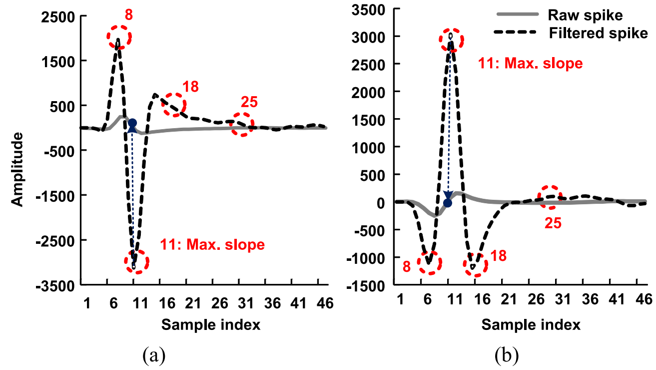


Fig. 5. Four selected features whose indexes are 8, 11, 18, and 25 on two sample spikes. (a) Sample 1. (b) Sample 2.

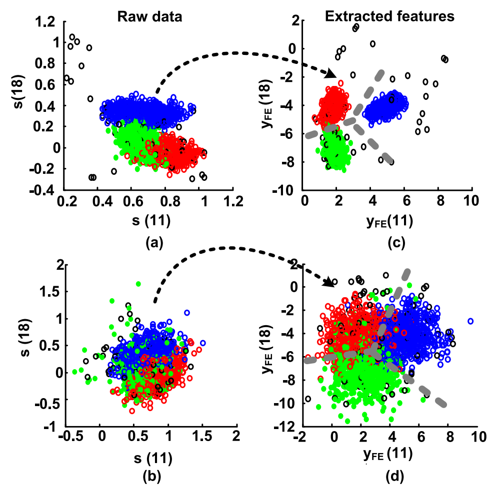


Fig. 6. (a) and (b) Original spike samples from two different data sets with different levels of noise. Data at indexes 11 and 18 are used for the scatter plot. (c) and (d) Outputs of y_{FE} whose indexes are 11 and 18. Green, red, and blue circles: actual neurons. Black ones: outliers.

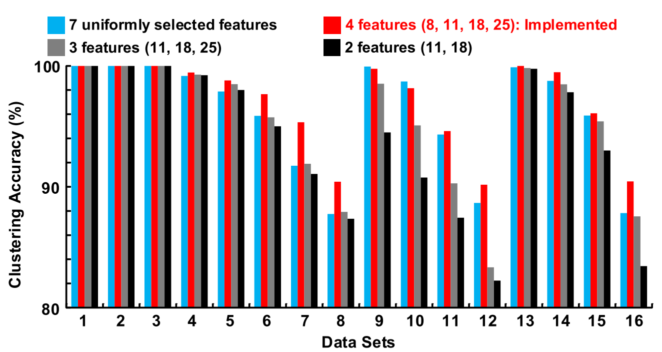


Fig. 7. Performance comparison of different choices of features across several data sets. MATLAB K-means was used for clustering in conjunction with our FE to compare the effectiveness of the choice of features.

different data sets. Compared to Fig. 6(a), the data points in Fig. 6(b) from different classes are hardly discriminated from one another. After going through the filter, they have much better feature separation as depicted in Fig. 6(c) and (d).

Fig. 7 shows the clustering accuracy of various feature selection scenarios using various data sets. Overall, selecting four features (indices of 8, 11, 18, and 25) leads to better clustering accuracy. Fig. 8 compares the accuracy of the proposed FE method using various numbers of features with

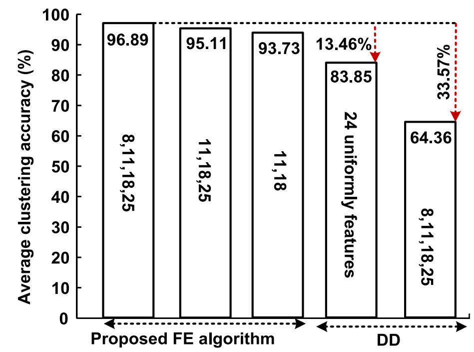


Fig. 8. Averaged clustering accuracy for different FEs.

DD whose δ is set to 1, 3, and 7. Note that DD is selected for comparison since it is the most suitable for the hardware implementation with good accuracy, and Gibson *et al.* [17] show that DD outperforms PCA, DWT, and IT in terms of clustering accuracy.

The clustering accuracy achieved by the proposed feature selection is also estimated by MATLAB. Note that the final classification is executed using the K-means algorithm. It is observed that the proposed FE method with four, three, and two features outperforms DD with 24 uniformly selected features and DD with the same four features. Furthermore, the proposed FE with four features requires much smaller memory compared to DD with 24 features. For instance, for a three-neuron input signal, the proposed four features require only 120 bits while DD needs 720 bits with 24 features. In addition, the number of features highly affects the clustering complexity and the amount of memory. For example, when no FE is implemented [8], a clustering algorithm requires the whole spike samples for the time-domain comparison. Consequently, the proposed FE significantly reduces the complexity of the clustering engine with better clustering accuracy.

C. Issues in Conventional K-Means

K-means is a popular clustering algorithm that classifies the input data based on their distance to the existing trained cluster means [12]. During training, cluster means are initiated using randomly chosen initial data points and adjusted by looping through the whole training data sets. One undesirable feature of K-means is the requirement to store the whole data set for

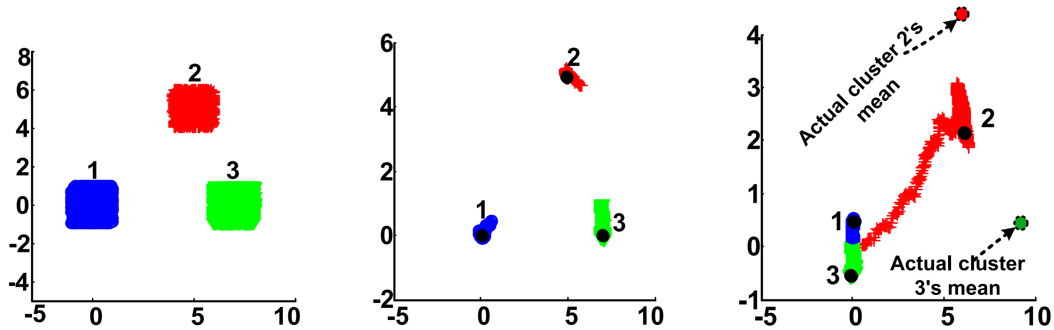


Fig. 9. (a) Artificial data set with three clusters. (b) K-means properly converges when initial means are chosen properly. Colors: traces of cluster means during training. Black dots: Final mean values. (c) K-means does not converge properly when all the means are initiated from cluster 1.

training, which is not practical for hardware implementation. Furthermore, if the initial cluster means are assigned wrongly, K-means will have difficulty in converging to the correct means. For example, consider a data set including two classes, A and B. Assume that both initial means (M_1 and M_2) are randomly chosen from data points in class A (i.e., $M_1 = A_i$ and $M_2 = A_j$). M_1 and M_2 are the dynamic cluster means which are updated through training. In this case, any data points including those in class B will be assigned to either M_1 or M_2 depending upon the actual distance regardless of their actual classes. As a result, the cluster means will never converge to the actual ones. Fig. 9 illustrates this situation with three artificial clusters. Fig. 9(a) is the original data with three cluster clearly separated. Fig. 9(b) illustrates how K-means assign and correctly adjust the cluster means as more data points pass through the algorithm. However, as depicted in Fig. 9(c), if the cluster means are all initiated from the same cluster, they do not converge to the correct values. This situation can easily occur because initial means are randomly chosen or the first data points are assigned as means, regardless of their actual clusters. Therefore, K-means is usually repeated several times to find the best converged means. Note that this can only be done offline when the whole data set is available in the memory and can be reused and reshuffled which is not feasible in real-time hardware implementation. Many of the currently available clustering algorithms are run on software with large memory capacity and powerful processors (WaveClus [11], KlustaKwik [19], and Osort [20]). In this paper, we propose an improved K-means to improve both clustering accuracy and hardware efficiency, which will be discussed in Section II-D.

D. Proposed Improved K-Means

The proposed improved K-means addresses the convergence issue in the conventional K-means. The number of cluster and their initial values need no initialization. However, the upper bound in the number of clusters needs to be specified to limit the generated intermediate clusters. Unlike the conventional K-means, the proposed K-means algorithm allows new data to form a new cluster instead of forcing it to be assigned to one of the existing means. In addition to the distances between new data and the existing means, the distances between the existing means are also calculated. If the distances between the new data and the existing means are significantly larger than the distances between the existing means, it is obvious that the new data does not belong to the existing means and thus deserves to form a new cluster. In this case, two existing means with the smallest distance are merged to maintain the same number of clusters. Extensive simulations have demonstrated that the proposed K-means algorithm converges even though initial means are purposely assigned wrongly.

The detailed implementation of the proposed improved K-means algorithm is as follows. Our cluster means' features are denoted as C_{ik} ($i = 0$ to 5 represents six clusters and $k = 0$ to 3 indicates four features). The number of clusters is selected as six because usually less than six neurons are associated with a channel [8], [19]. The upper bound is not the actual number of clusters. If the number of clusters is three, only the first three clusters are meaningful. Initially, C_{ik} is filled with the first 24 features from the first six detected spikes. Then, the next four features of the seventh spike are stored in the temporary cluster (seventh cluster) whose means are C_{6k}. Two sets of distances represented by $d(C_{ik}, C_{jk})$ and $d(C_{ik}, C_{6k})$ are calculated as

$$d(C_{ik}, C_{jk})_{i,j=0:5;i\neq j} = \sum_{k=1}^{4} |C_{ik} - C_{jk}| \tag{6a}$$

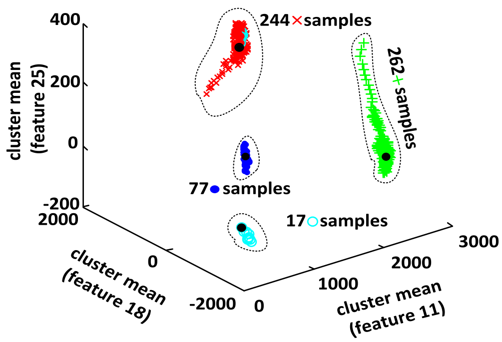
$$d(C_{ik}, C_{6k})_{i=0:5} = \sum_{k=1}^{4} |C_{ik} - C_{6k}|. \tag{6b}$$

l_1-distance (i.e., Manhattan distance) is used because of its good clustering accuracy and simple circuit implementation compare to l_2-distance (i.e., Euclidean distance) where $d(C_{ik}, C_{jk})_{i,j=0:5}$ represents the distances between existing clusters, while $d(C_{ik}, C_{6k})_{i=0:5}$ represents the distance between new data (i.e., the temporary cluster) and the existing clusters. After calculating all $d(C_{ik}, C_{jk})$, they are weighted as given

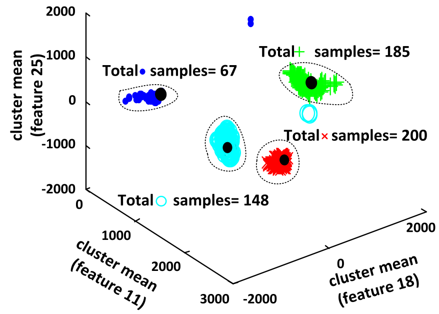
$$d_u(C_{ik}, C_{jk})_{i,j=0:5,i\neq j} = 1.5 * d(C_{ik}, C_{jk}) \tag{7}$$

so that the existing clusters are less subjected to being merged. The minimum values among $d_u(C_{ik}, C_{jk})$ and $d(C_{ik}, C_{6k})$ are selected and the two corresponding clusters are merged together. Once two clusters i and j are merged in cluster i, the cluster i mean is updated using

$$C_{ik,\text{new}} = \frac{15 * C_{ik} + C_{jk}}{16}. \tag{8}$$

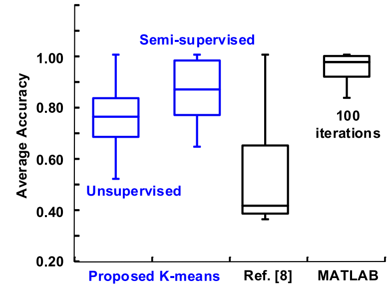


(a)

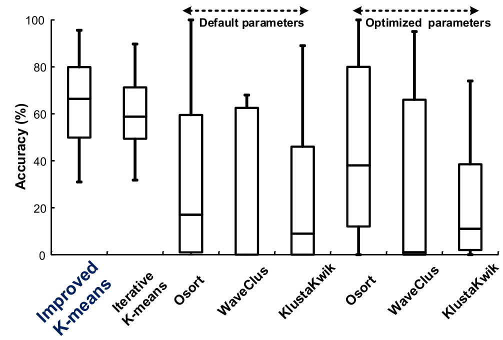


(b)

Fig. 10. Cluster means convergence during the training phase using two different data sets (a) and (b) randomly chosen from the database. Three features are used to plot.



(a)



(b)

Fig. 11. (a) Clustering accuracy of modified K-means compared to [8] and original K-means using "Replicates" option set to 100. The Performance of [8] is reproduced using our data. (b) Clustering accuracy of the proposed method compared to state-of-the-art spike sorting algorithms using data sets in [21].

Fig. 10 illustrates how the cluster means converge to their final values for four clusters in two different data sets. In each data set, four cluster means are initially selected from the first four spikes. After training, each cluster mean converges to its final value shown by the black circle. Note that in this simulation, the number of clusters is specified as four but there are only three actual clusters. Thus, the actual clusters correspond to three various neurons and the fourth cluster is the outlier one. The algorithm will assign a few neurons to the fourth cluster but this does not affect the convergence of the real cluster means.

Fig. 11(a) evaluates the proposed clustering algorithm applied to the whole database in [11] with 21 data sets. The accuracy of the proposed clustering algorithm is 86% when the number of clusters is specified in advance while it becomes 72% without specified cluster count. The proposed algorithm outperforms the original K-means in both semisupervised (i.e., the number of clusters specified) and unsupervised modes (i.e., the number of cluster not specified). It shows better accuracy in the semisupervised mode. The original K-means running in MATLAB with 100 iterations gives the best result. It is because the mean values are computed 100 times with random initial values to get the best possible clustering accuracy, which cannot be implemented in real-time clustering. Fig. 11(b) shows the comparison between the proposed algorithm and state-of-the-art ones using the data set in [21]. More than 30 data sets with various numbers of spikes and noise levels are introduced in [21]. Note that Reference [8] in Fig. 11(a) is a simplified version of Osort in Fig. 11(b).

III. Proposed Spike Sorting Processor Implementation

A. Architecture and Operation

The block diagram of the proposed SSP, illustrated in Fig. 12, consists of input control and detection, FE and alignment, and clustering functions. Input data from 128 channels operating at 25 kS/s (CLKS) are serialized in the frequency of 3.2 MHz (CLK). The frequency of CLK is that of CLKS multiplied by the number of channels (i.e., 128 in this paper). Detection has 128 shift register (SR) banks (i.e., SR_{1-128}), each containing five 9-bit registers to store five consecutive inputs from the corresponding channel. Clk_i is the gated clock attributed to the ith channel. An array of five 128-to-1 multiplexers (MUX) selects correct input samples from these SRs using the channel select (i.e., Ch. Select) signal generated by the control circuit.

The processor operates in two modes: the training mode and the classifying mode. The training mode runs once in a while to update the cluster means. One channel is trained in each time. The training period for each channel is controlled by the control circuit and can vary from one channel to another depending upon the number of neurons associated in each channel and the activity of each neuron. During the training mode, once a spike is detected, features are calculated and passed to the training engine for calculating cluster means (*Clus. Means (trained)* in Fig. 12). The final cluster means are stored in the SRAM to be used by the classifier in the classifying mode.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

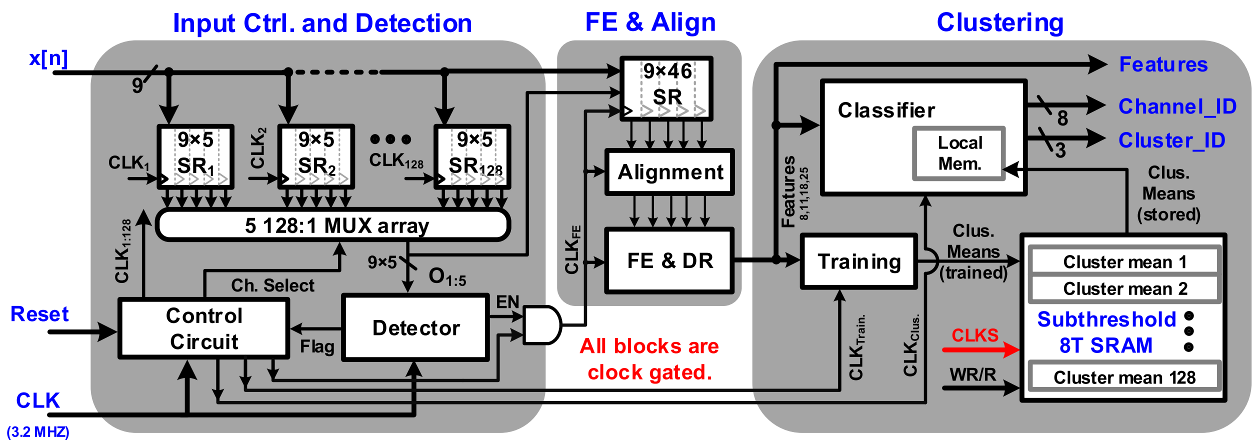


Fig. 12. Functional block diagram of the proposed SSP.

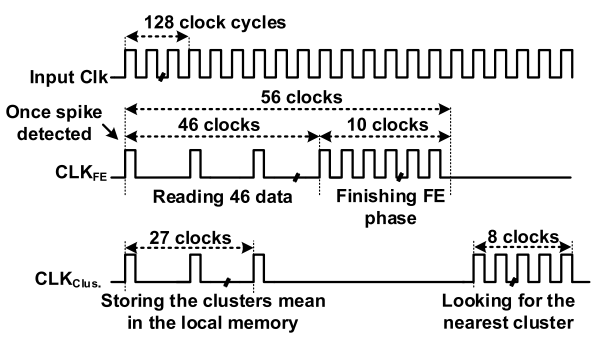


Fig. 13. Timing and scheduling diagram of “detector” and “FE and DR,” and clustering when a spike is detected.

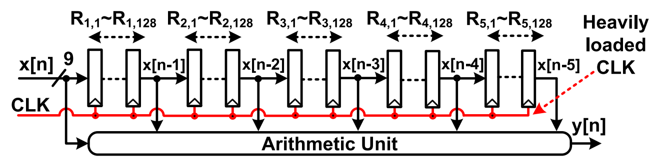


Fig. 14. Conventional simplified interleaved architecture for 128-channel detection [7].

During the classifying mode, once a spike is detected, the classifier read the previously trained cluster means of the corresponding channel from the SRAM and temporarily stores them in a local memory inside the classifier. This operation is executed in parallel with FE, which helps to reduce the overall processing time. Furthermore, it allows the SRAM to operate using the slower clock (CLKS). This allows us to employ higher threshold voltage (HVT) devices in the SRAM and suppress the leakage. After the FE, the classifier computes the nearest cluster using l_1-distance to assign it within eight clock cycles. Once the classification is done, the classifier is clock gated and reset for the next classifying operation. Note that all the major blocks such as the local memory, the FE and alignment block, the classifier, and the training are clock gated to save power. In addition, the detector, the FE, and the Training blocks are shared by two operation modes to save hardware resources.

Fig. 13 describes the timing and scheduling of the proposed SSP. After a spike is detected, the detector is fully clock gated by the control circuit. It means that once a spike is detected in any input channel, the detector is temporarily halted for 56 clock cycles. This is one of the limitations of the current design. To accommodate multiple spikes from different channels, one of the solutions is to reduce the number of channels per processing engine so that multiple engines can handle different channels concurrently.

After detection, the alignment and the FE and DR read the next 40 samples from the same channel using $\mathrm{CLK_{FE}}$ in Fig. 13. These 40 samples are combined with the original six samples from the detectors as a complete raw spike waveform for FE. Simultaneously, FE and DR calculates the maximum slope of the spike waveform and the features and stores them in the local memory. Because the input data from 128 channels are serialized with 3.2-MHz clock, the data from a detected channel are only available once every 128 cycles of input. Therefore, the FE and DR is also clocked once every 128 clock cycles (i.e., using CLKS) to reduce unnecessary switching activity. However, after reading the necessary data samples, the alignment and the FE and DR run with the fast clock (i.e., CLK) to finish the computation of maximum slope and features, and align the extracted features to the computed maximum slope. As a result, the alignment and the FE and DR blocks consume $5898(46 \times 128 + 10)$ CLK cycles. Note that 5898 clock cycles represent $\sim$1.84 ms, which is almost the same as a single spike window. Then, the extracted features are transferred either to the training or to the classifier depending on the operating mode as depicted in Fig. 12. Finally, the classifier or the training block computes the nearest cluster over eight CLK cycles executing the proposed algorithm explained in Section II.

B. Register Architecture for Dynamic Power Reduction

In SSPs, detectors and the associated registers are the main sources of dynamic power consumption since they are active all the time. The clock frequency and the amount of memory in the detectors linearly increase with the number of channels. Thus, their dynamic power also augments significantly with the number of channels. Other blocks for alignment, FE and DR, training, and classification are shared by multiple channels and activated only when they are requested. Their power consumption is much smaller than that of the detectors

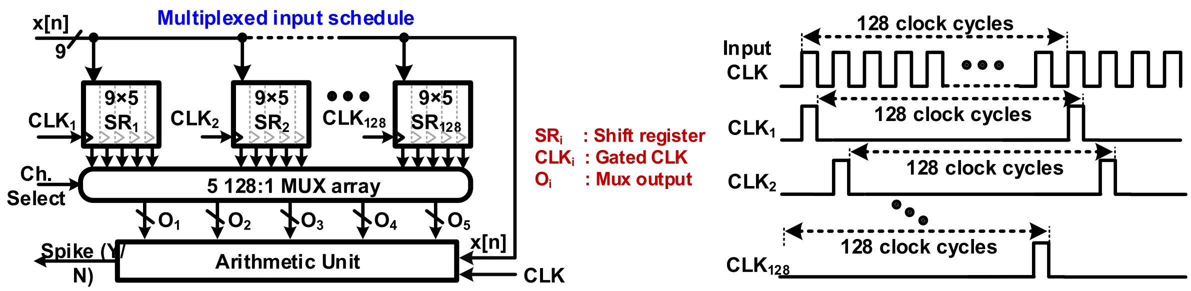


Fig. 15. (a) Time-multiplexed implementation of the SR for the 128-channel detection. SR_i blocks: SR banks for each channel. Each channel is controlled by its specified clock signal CLK_i. (b) Timing diagram of CLK_i.

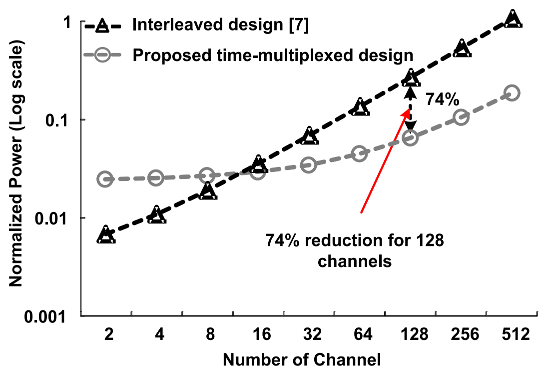


Fig. 16. Normalized power comparison between interleaved and time-multiplex architecture.

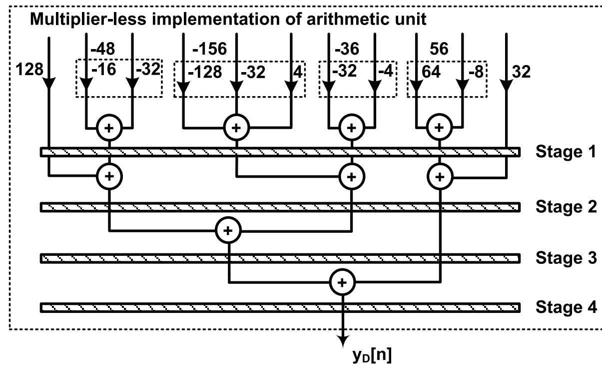


Fig. 17. Arithmetic unit implemented by converting filter coefficients to power-of-two values and using left shift as a multiplier. $y_D[n] = 128x(n) - 48x(n-1) - 156x(n-2) - 36x(n-3) + 56x(n-4) + 32x(n-5)$.

and the registers. In this section, we discuss a novel register architecture for reducing the dynamic power of the input registers and the detector of the proposed SSP.

Conventionally, an interleaved architecture as shown in Fig. 14 is used to schedule input samples from multiple channels in SSPs [7]. It leads to a huge dynamic power because all the registers for 128 channels are connected in series and their values change at each clock cycle as depicted in Fig. 14. To avoid excessive data transitions in the registers, we propose time-multiplexed register architecture (Fig. 15), which reduces both the clocking frequency and loading. The control circuit in Fig. 12 generates 128 clock signals (i.e., CLK_{1-128}) as depicted in Fig. 15(b) so that each SR_i is clocked only when its input arrives. In Fig. 15, SR_i represents the SR bank for each channel. The MUX array selects five data of the same channel and stores them in the input registers of the arithmetic unit. Fig. 16 compares the power of two architectures varying the numbers of channels. Note that the proposed register architecture is more power efficient when the channel count is higher (>16), including the input driver buffers. The proposed architecture reduces power by 74% for 128 channels. It is worth noticing again that proposed multichannel detection architecture runs 1 SR out of 128 per clock cycle whereas in interleaved architecture all 128 SRs are running with the fast clock. For eight or fewer channels, the conventional interleaved implementation outperforms the proposed time-multiplexed architecture due to the overhead caused by the clock gating.

To further reduce the computational complexity of the arithmetic unit in the detector, all multiplications introduced by the integer filter coefficients are converted to shift- and addition-based operations. Power-of-two coefficients such as 128 and 32 are easily realized by the shift operations. Other coefficients such as -48, -156, -36, and 56 are first converted to the summation of power-of-two values and then implemented by the shift operations as described in Fig. 17. Four pipeline stages are incorporated to improve the performance in the near-threshold voltage where the latency of the adder chains becomes critical and limits the overall design performance.

C. Ultralow Voltage 8T SRAM for Cluster Means Storage

One of the key blocks for the clustering is a memory storing all cluster means. In the proposed clustering, a memory density of 39 kbits is required to support 128 channels. Implementing this memory with flip-flops will consume huge dynamic power and silicon area. To address this, we implemented a 39-kbit 8T SRAM (Fig. 18). The 8T SRAM cell in [22] was chosen for low-voltage operation. Note that the memory is written once after training and read only when a spike is detected. Therefore, its active power is less significant than its leakage. As a result, HVT devices are used for the 6T structure to minimize the leakage, while standard threshold voltage devices are used at the read port and peripheral circuits for higher read speed. Simulation results show that the maximum operating frequencies at 0.5 and 0.4 V are 4 MHz and 500 kHz, respectively. This provides safe timing margins as the target f_{CLKS} is only 25 kHz. To ensure the cell stability under the half-select condition [22], Monte Carlo simulations

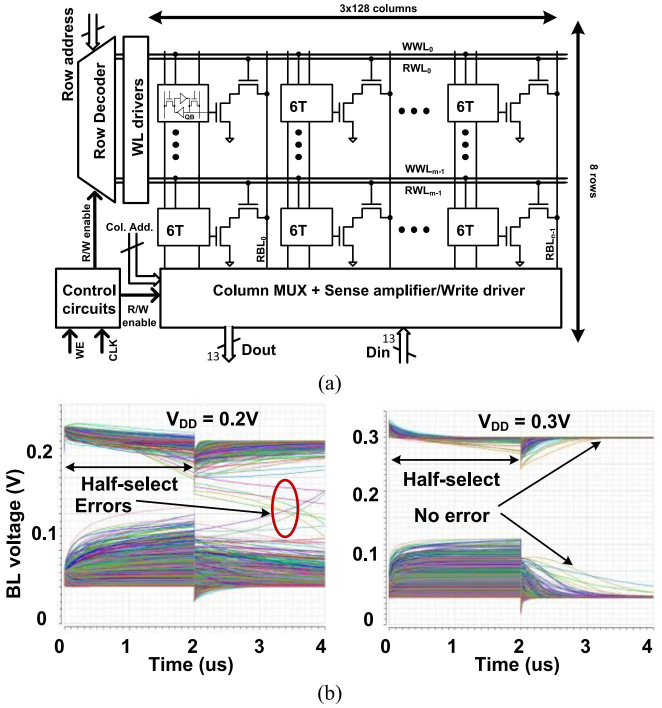


Fig. 18. (a) Block diagram of the customized 8T SRAM (b) 1000-iteration Monte Carlo simulation of the memory cell to verify its stability under half-select condition.

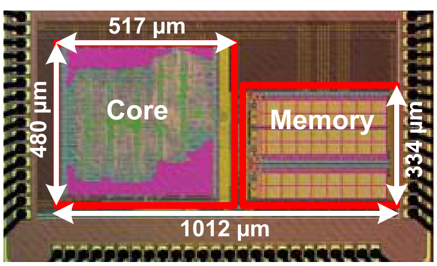


Fig. 19. Implemented 128-channel SSP micrograph.

were executed. It is noticeable that cell flipping occurs at $V_{DD} = 0.2$ V. No flipping was observed when V_{DD} is raised to 0.3 V. As the target operating supply voltage is around 0.5 V, the implemented 8T SRAM is stable enough. Furthermore, our measurement results of the SRAM alone ensure that no cell failure happened during the testing of the design.

IV. Test Chip Measurement Results

The proposed SSP was fabricated in a 65-nm CMOS process technology. Its microphotograph is depicted in Fig. 19. Table I summarizes the test chip specifications. The typical sampling frequency for one-channel neural spike signal recording is 25 kHz [6]–[8], and therefore, the operating frequency of the proposed 128-channel SSP is set to 128 × 25 kHz = 3.2 MHz. A Xilinx ZYNQ-7000 SoC kit and a Xilinx FMC XM105 debug card were used to test the implemented SSP as shown in Fig. 20. We use the synthetic data [11] to evaluate the classification accuracy of the design compared with prior arts. Comprehensive algorithm comparison is done using MATLAB and fabricated chip functionality/performance is verified using

TABLE I
Test Chip Specifications

Technology	65 nm
Total Area (mm^2)	0.414
Core Voltage	0.54
Number of Ch.	128
Total Power (μW)	22.4
Input Frequency (MHz)	3.2
Spike Data Reduction	257

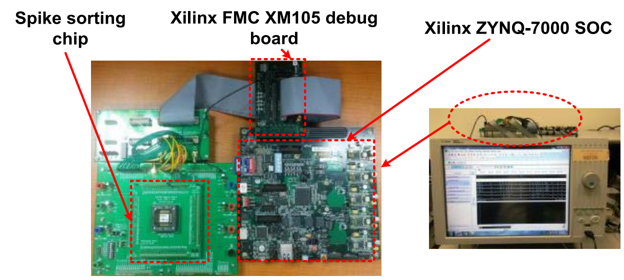


Fig. 20. Testing setup using a ZYNQ-7000 SoC kit with a Xilinx FMC XM105 debug card.

MATLAB results as 1-to-1 reference. This database has 21 data sets; each contains 1 440 000 data points representing a digitized version of continuous neural recording. Since the typical neural signal sampling rate is 20∼30 kS/s [12], each data set is equivalent to a waveform of 60-s recording time. The synthetic data also provide true spike locations and cluster IDs are also available. Each waveform has a different number of spikes (1636–1787), spike locations, noise levels, as well as the number of spike clusters. A logic analyzer was used to capture the output data and compare them with the known cluster IDs. The designed chip can be configured to operate in three different modes: 1) detection only; 2) detection and FE; and 3) classification. If none of the modes is activated, the processor will transmit raw waveforms. This is to accommodate different testing scenarios and to enable the testers to validate the sorting quality before performing long-term recording with the proposed SSP.

Fig. 21 presents sample measured waveforms during the aforementioned modes. In the detection only mode, if a spike is detected, a detection flag is enabled accordingly. Similarly, in the detection and FE modes, an FE flag is also enabled in addition to a detection flag, showing that features are calculated. Finally, using the classification mode, the processor generates neuron IDs corresponding to each detected spike.

The minimum operating voltage for 3.2 MHz is 0.54 V. The SSP still operates below 0.54 V but with lower clock frequency. Fig. 22 summarizes the power consumption per channel at different supply voltages. Unlike other designs [7], [8] where leakage power is dominant, the proposed SSP demonstrates significantly suppressed leakage thanks to the ultralow voltage SRAM. As illustrated in Fig. 23(a), the dynamic power constitutes 83% of the total power. Fig. 23(b) illustrates the power distribution of major functional blocks. The detector block contributes 64% of the total dynamic power because it is always active. Table II compares the proposed SSP

TABLE II
PERFORMANCE COMPARISON

Reference	[2]	[6]	[7]	[8]	[9]	**This work**
No. of Chs.	128	64	64	16	32	**128**
Detection	Y	Y	Y	Y	Y	**Y**
Feature Extraction	Y	Y	Y	N	Y	**Y**
Clustering (Accuracy)	N	N	N	Y (~75%)	Y (60~80%)	**Y (72~86%)**
Core Voltage (V)	3	0.25	0.55	0.27	1.2	**0.54**
Power (μW/Ch)	75	0.46	2.03	4.68	0.75	**0.175**
Area (mm^2/Ch)	0.11	0.03	0.06	0.07	0.023	**0.003**
Process (nm)	500	65	90	65	130	**65**

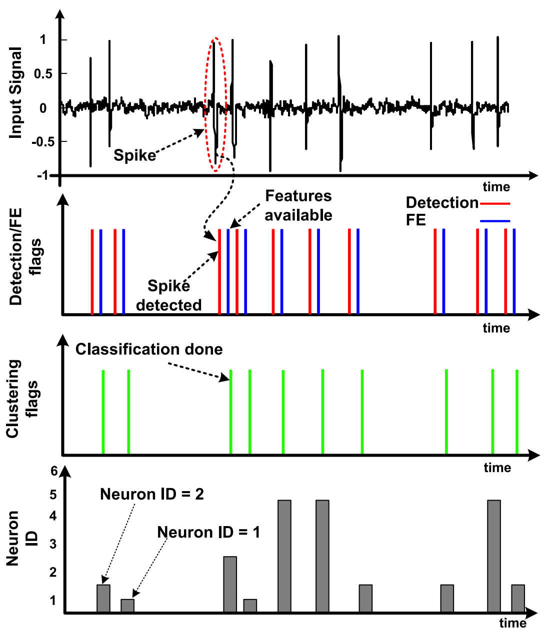


Fig. 21. Main outputs of the 128-channel spike sorting chip for a sampled input.

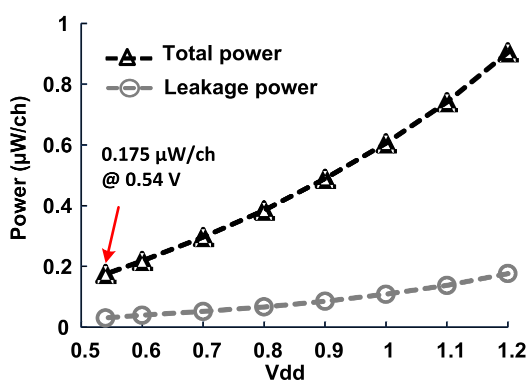


Fig. 22. Power consumption per channel.

with other state-of-the-art SSPs. Compared to the first SSP including clustering for 16 channels in [8], the proposed SSP improves the power and the area efficiencies per channel by

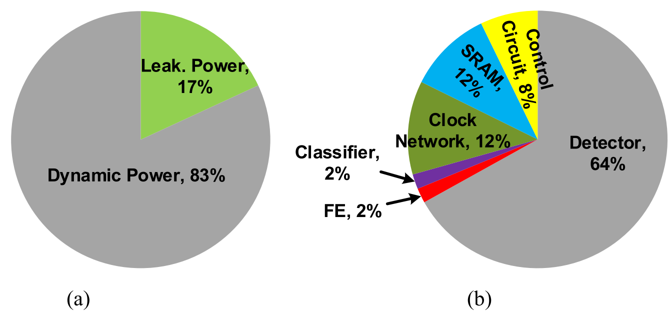


Fig. 23. Power breakdown of the SSP at 0.54 V and 3.2 MHz. (a) Measured dynamic power versus leakage power. (b) Simulated power breakdown between functional blocks.

26× and 23×, respectively. This allows 333 channels/mm^2 with the power density of $333 \times 0.175\ \mu$W $= 58.3\ \mu$W, which is much smaller than the power density requirement of 277 μW/mm^2 on implantable devices [5].

V. CONCLUSION

This paper has presented a 128-channel SSP to analyze recorded brain activities. This design proposes several online spike sorting algorithms to improve the clustering accuracy and lower the power and the area per channel. The integer coefficient filters used in both detection and FE improves signal quality by simplifying arithmetic computation without multipliers. In addition, the proposed improved K-means algorithm addresses the issues of means convergence in the conventional K-means. The averaged clustering accuracy is between 72% and 86%. Finally, the time-multiplexed registers in the detection block and the low leakage ultralow voltage SRAM reduce the overall area, the dynamic power, and the leakage power. Measurement results demonstrate that the proposed design operates down to 0.54 V, consumes 0.175 μW/channel, and occupies 0.003 mm^2/channel. This improves the power and the area efficiencies by 26× and 23×, respectively, compared to the processor in [8]. Therefore, the proposed SSP is applicable to the next-generation ultralow power spike sorting systems with a high channel count.

REFERENCES

[1] A. M. Kamboh and A. J. Mason, "Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 1, pp. 1–9, Jan. 2013.

[2] M. S. Chae, Z. Yang, M. R. Yuce, L. Hoang, and W. Liu, "A 128-channel 6 mW wireless neural recording IC with spike feature extraction and UWB transmitter," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 4, pp. 312–321, Aug. 2009.

[3] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain–computer interface," *Nature*, vol. 442, no. 7099, pp. 195–198, Jul. 2006.

[4] A. M. Sodagar, G. E. Perlin, Y. Yao, K. Najafi, and K. D. Wise, "An implantable 64-channel wireless microsystem for single-unit neural recording," *IEEE J. Solid-State Circuits*, vol. 44, no. 9, pp. 2591–2604, Sep. 2009.

[5] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher, "Thermal impact of an active 3-D microelectrode array implanted in the brain," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 493–501, Dec. 2007.

[6] T.-T. Liu and J. M. Rabaey, "A 0.25 V 460 nW asynchronous neural signal processor with inherent leakage suppression," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 897–906, Apr. 2013.

[7] V. Karkare, S. Gibson, and D. Marković, "A 130-μW, 64-channel neural spike-sorting DSP chip," *IEEE J. Solid-State Circuits*, vol. 46, no. 5, pp. 1214–1222, May 2011.

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

[9] Y. Yang, S. Boling, and A. J. Mason, "A hardware-efficient scalable spike sorting neural signal processor module for implantable high-channel-count brain machine interfaces," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 4, pp. 743–754, Aug. 2017.

[10] J. Park, G. Kim, and S.-D. 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.

[11] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[12] S. Gibson, J. W. Judy, and D. Marković, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal Process. Mag.*, vol. 29, no. 1, pp. 124–143, Jan. 2012.

[13] S. M. A. Zeinolabedin, A. T. Do, D. Jeon, D. Sylvester, and T. T.-H. Kim, "A 128-channel spike sorting processor featuring 0.175 μW and 0.0033 mm^2 per channel in 65-nm CMOS," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2016, pp. 32–33.

[14] I. Obeid and P. D. Wolf, "Evaluation of spike-detection algorithms fora brain-machine interface application," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 905–911, Jun. 2004.

[15] A. T. Do and K. S. Yeo, "A hybrid NEO-based spike detection algorithm for implantable brain-IC interface applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 2393–2396.

[16] S. Mukhopadhyay and G. C. Ray, "A new interpretation of nonlinear energy operator and its efficacy in spike detection," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 2, pp. 180–187, Feb. 1998.

[17] S. Gibson, J. W. Judy, and D. Marković, "Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 5, pp. 469–478, Oct. 2010.

[18] S. Gibson, J. W. Judy, and D. Marković, "Comparison of spike-sorting algorithms for future hardware implementation," in *Proc. IEEE Eng. Med. Biol. Conf.*, Aug. 2008, pp. 5015–5020.

[19] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsáki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *J. Neurophysiol.*, vol. 84, no. 1, pp. 401–414, 2000.

[20] U. Rutishauser, E. M. Schuman, and A. N. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo*," *J. Neurosci. Methods*, vol. 154, nos. 1–2, pp. 204–224, 2006.

[21] J. Wild, Z. Prekopcsak, T. Sieger, D. Novak, and R. Jech, "Performance comparison of extracellular spike sorting algorithms for single-channel recordings," *J. Neurosci. Methods*, vol. 203, no. 2, pp. 369–376, Jan. 2012.

[22] A. T. Do, Z. C. Lee, B. Wang, I.-J. Chang, and T. T.-H. Kim, "0.2 V 8T SRAM with PVT-aware bitline sensing and column-based data randomization," *IEEE J. Solid-State Circuits*, vol. 51, no. 6, pp. 1487–1498, Jun. 2016.

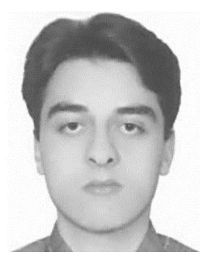
[23] D. Khodagholy *et al.*, "NeuroGrid: Recording action potentials from the surface of the brain," *Nature Neurosci.*, vol. 18, pp. 310–315, Dec. 2014.

[24] J. J. Jun *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, pp. 232–236, Nov. 2017.

Anh Tuan Do (M'11) received the B.S. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 2007 and 2010, respectively.

From 2010 to 2015, he was a Research Fellow with VIRTUS, IC Design Centre of Excellence, NTU. In 2015, he joined the Digital IC Design Group, Institute of Microelectronics, A∗STAR, Singapore. He has authored or coauthored more than 45 journal and conference papers. His current research interests include biomedical circuits and systems, low power, low leakage, variation-tolerant digital circuits, memory, SoC, emerging memory technologies, hardware security primitives, and sensor design.

Dr. Do has served as a reviewer for several IEEE journals and conferences including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS. He was a recipient of the Best Paper Award at ISOCC 2012 and the Second Prize and Best Presentation Award in the Innovation Contest of the International Ph.D. Student Workshop 2007 of the National University of Taiwan.



Seyed Mohammad Ali Zeinolabedin (S'12–M'18) received the B.S. degree in electrical engineering from Azad University, Tehran, Iran, in 2006, the M.S. degree in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, in 2010, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2017.

In 2017, he joined the Highly Parallel VLSI Systems and Neuro-Microelectronics Research Group, Technische Universität Dresden, Dresden, Germany. His current research interests include the hardware implementation of image, video, and biomedical algorithms and ultralow power circuits and systems design with high energy efficiency.

Dr. Zeinolabedin was a recipient of the Best Demo Award at APCCAS2016 and the Low Power Design Contest Award at ISLPED2016.

Dongsuk Jeon (S'10) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2009, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014.

From 2014 to 2015, he was a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently an Assistant Professor with the Graduate School of Convergence Science and Technology, Seoul National University. His current research interests include energy-efficient signal processing, low-power circuit, and SoC for mobile applications.

Dennis Sylvester (S'95–M'00–SM'04–F'11) received the Ph.D. degree in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA.

He is currently a Professor of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA, where he is also the Director of the Michigan Integrated Circuits Laboratory, a group of 10 faculties and 70+ graduate students. He held research staff positions with the Advanced Technology Group, Synopsys, Mountain View, CA, USA, and the Hewlett-Packard Laboratories, Palo Alto, CA, USA, and Visiting Professorships with the National University of Singapore, Singapore, and Nanyang Technological University, Singapore. He has authored or coauthored over 450 articles along with one book and several book chapters. He holds 34 U.S. patents. His current research interests include the design of millimeter-scale computing systems and energy-efficient near-threshold computing.

Dr. Sylvester serves as a Consultant and Technical Advisory Board Member for electronic design automation and semiconductor firms in his research areas. He was a recipient of the NSF CAREER Award, the Beatrice Winner Award at ISSCC, the IBM Faculty Award, the SRC Inventor Recognition Award, the 10 Best Paper Awards and nominations, and the University of Michigan Henry Russel Award for Distinguished Scholarship. He was named one of the Top Contributing Authors at ISSCC. His dissertation was recognized with the David J. Sakrison Memorial Prize as the most outstanding research with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. He served on the Executive Committee for the ACM/IEEE Design Automation Conference. He serves on the Technical Program Committee for the IEEE International Solid-State Circuits Conference. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, and the Guest Editor of the IEEE JOURNAL OF SOLID-STATE CIRCUITS and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II.

Tony Tae-Hyoung Kim (M'06–SM'14) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2009.

From 2001 to 2005, he was with Samsung Electronics, Hwasung, South Korea, where he was involved in the design of high-speed SRAM memories, clock generators, and IO interface circuits. From 2007 to 2009, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, and Broadcom Corporation, Edina, MN, USA, where he was involved in circuit reliability, low-power SRAM, and battery backed memory design. In 2009, he joined Nanyang Technological University, Singapore, where he is currently an Associate Professor. He has authored or coauthored +140 journal and conference papers. He holds 17 U.S. and Korean patents registered. His current research interests include low-power and high-performance digital, mixed-mode, and memory circuit design, ultralow voltage circuits and systems design, variation and aging tolerant circuits and systems, and circuit techniques for 3-D ICs.

Dr. Kim was a recipient of the Best Demo Award at APCCAS2016, the Low Power Design Contest Award at ISLPED2016, the Best Paper Awards at 2014 and 2011 ISOCC, the AMD/CICC Student Scholarship Award at IEEE CICC2008, the Departmental Research Fellowship from the University of Minnesota in 2008, the DAC/ISSCC Student Design Contest Award in 2008, the Samsung Humantec Thesis Award in 2008, 2001, and 1999, and the ETRI Journal Paper of the Year Award in 2005. He is the Chair of the IEEE Solid-State Circuits Society Singapore Chapter. He has served numerous conferences as a Committee Member. He serves as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS.