A 16-Channel Nonparametric Spike Detection ASIC Based on EC-PC Decomposition

Tong Wu, *Student Member, IEEE*, Jian Xu, *Member, IEEE*, Yong Lian, *Fellow, IEEE*, Azam Khalili, *Member, IEEE*, Amir Rastegarnia, *Member, IEEE*, Cuntai Guan, *Senior Member, IEEE*, and Zhi Yang, *Member, IEEE*

***Abstract*—In extracellular neural recording experiments, detecting neural spikes is an important step for reliable information decoding. A successful implementation in integrated circuits can achieve substantial data volume reduction, potentially enabling a wireless operation and closed-loop system. In this paper, we report a 16-channel neural spike detection chip based on a customized spike detection method named as exponential component-polynomial component (EC-PC) algorithm. This algorithm features a reliable prediction of spikes by applying a probability threshold. The chip takes raw data as input and outputs three data streams simultaneously: field potentials, band-pass filtered neural data, and spiking probability maps. The algorithm parameters are on-chip configured automatically based on input data, which avoids manual parameter tuning. The chip has been tested with both *in vivo* experiments for functional verification and bench-top experiments for quantitative performance assessment. The system has a total power consumption of 1.36 mW and occupies an area of 6.71 mm^2 for 16 channels. When tested on synthesized datasets with spikes and noise segments extracted from *in vivo* preparations and scaled according to required precisions, the chip outperforms other detectors. A credit card sized prototype board is developed to provide power and data management through a USB port.**

***Index Terms*—EC-PC regression, multichannel digital system, neural signal processing, unsupervised spike detection.**

I. Introduction

SPIKE detection refers to differentiating extracellular neural spikes from background noise. Its motivation is twofold: to extract neural spikes for data analysis and closed-loop execution, and to compress neural data and facilitate wireless operations. Many detection algorithms have been proposed in the literature [1]–[17]. Several spike detectors have also been reported [18]–[33]. Among the available spike detection methods, the absolute value thresholding (AT) [1]–[5] and the nonlinear energy operator (NEO) [17] are the simplest ones. An FPGA-based implementation has been proposed using AT [21]. The algorithm is attractive for its computational simplicity, yet its performance is unsatisfactory at moderate and low signal-to-noise ratios (SNRs), making detection threshold a very sensitive parameter. NEO has been realized in several spike-sorting chips [24], [30], [34] due to its efficiency. The algorithm is meant to boost the differentiation between spikes and noise. However, neural noise tends to be non-stationary, thus NEO may unfavorably amplify some noise waveforms and give a raised false alarm.

Other popular spike detection methods include template matching [6]–[9] and wavelet-based detectors [10]–[15]. In template matching, assuming neural spikes follow several templates with white Gaussian noise, matched filters constructed from signal templates can give the best waveform differentiation. These algorithms are effective given decent SNRs and stationary neural signals; however, neural spikes may have both short-term and long-term variations. For example, individual spikes in a burst can have more than 50% amplitude variation according to simultaneous intracellular and extracellular recordings [35]. This challenges the hypothesis of "static waveform template" and makes the performance not as good as expected. Wavelet-based detectors require well-shaped mother wavelets correlated with signals through either discrete-wavelet transform [10], [15] or continuous-wavelet transform [11], [14]. A challenge that is commonly faced is the requirement to fine-tune thresholds, which is difficult given non-stationary signals and noise. In addition, some wavelet-based detectors require excessive hardware resources to implement [36], and an extension to many channels is difficult [37].

Any successful spike detection method should satisfy the following requirements. First, it should be suitable for online implementation without requiring significant computational resources and storage. Second, detectors should be nonparametric and unsupervised to avoid frequent manual parameter tuning. Third, a detector is preferred to consistently perform well with different preparations and experiment protocols. Specifically, it should be robust to practical imperfections such as spike overlapping, waveform variation, low SNRs, unresolved artifacts, and interferences. Considering the mentioned requirements, we report a detection algorithm followed by its ASIC implementation. In comparison with other detectors and unsolved challenges on hardware efficiency, parameter

Manuscript received April 08, 2014; revised August 16, 2014 and December 03, 2014; accepted December 28, 2014. Date of publication March 05, 2015; date of current version February 22, 2016. This work was supported by Singapore grants A*STAR Public Sector Funding R-263-000-699-305, Ministry of Education Tier-1 funding R-263-000-A47-112, and Young Investigator Award R-263-000-A29-133. This paper was recommended by Associate Editor M. Stanacevic.

T. Wu, J. Xu, and Z. Yang are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: eleyangz@nus.edu.sg).

Y. Lian is with the Department of Electrical Engineering and Computer Science, Lassonde School of Engineering, York University, Toronto, ON M3J 1P3 Canada (e-mail: plian@yorku.ca).

A. Khalili and A. Rastegarnia are with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (e-mail: a_rastegar@ieee.org).

C. Guan is with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632 (e-mail: ctguan@i2r.a-star.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/TBCAS.2015.2389266

TABLE I
ALGORITHM FLOW OF EC-PC DETECTOR

EC-PC Spike Detection Algorithm

Input: Digitized neural data $V(m\Delta T)$, m is the sampling index and ΔT is the sampling interval.

Output: Probability map $p_s(m\Delta T)$ to indicate spike presence. For example, $p_s(m\Delta T)$=1 means that the sample has a 100% chance to be true spikes.

- Band-pass filter $V(m\Delta T)$ into $V_{bpf}(m\Delta T)$.
- Transform $V_{bpf}(m\Delta T)$ into Hilbert space as $HV_{bpf}(m\Delta T)$, and form analytic signal $V_{st}(m\Delta T) = V_{bpf}(m\Delta T) + \mathrm{i}HV_{bpf}(m\Delta T)$.
- Estimate the probability density function $f(Z(m\Delta T))$, where $Z(m\Delta T) = |V(m\Delta T)_{st}|^2$.
- Decompose $f(Z)$ into two components, $\widetilde{f}_d(Z(m\Delta T))$ and $\widetilde{f}_n(Zm\Delta T)$, which are represented as $y_{ec} = p \cdot e^{-\lambda_1 \cdot Z} \rightarrow \log(y_{ec}) = -\lambda_1 \cdot Z + a$ and $y_{pc} = \frac{q}{Z^{\lambda_2}+r} \rightarrow \log(y_{pc}) \approx -\lambda_2 \cdot \log(Z) + b$.
- Calculate $p_s(m\Delta T) = \widetilde{f}_d(Z(m\Delta T)) \cdot (\widetilde{f}_d(Z(m\Delta T)) + \widetilde{f}_n(Z(m\Delta T)))^{-1}$.

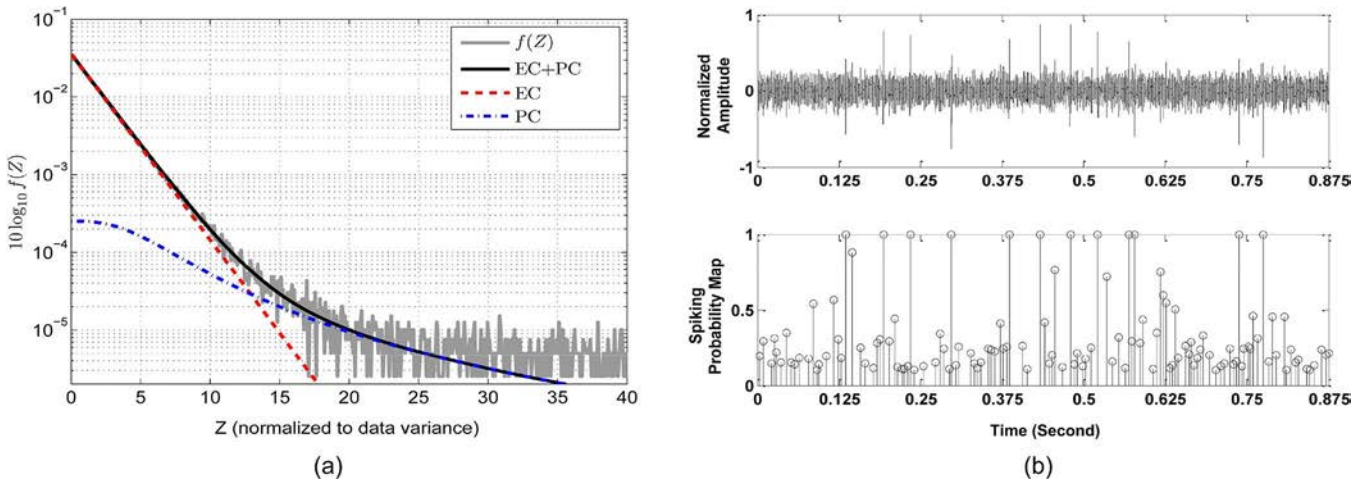


Fig. 1. (a) Decomposition of neural data in Hilbert space into EC and PC. (b) Band-pass filtered neural data with corresponding spiking probability maps.

tuning, and reliability, this work has the following features. First, through online and iterative learning, the required on-chip storage has been reduced, which enables area-efficient hardware design. Second, all parameters except thresholds are estimated from raw data and adaptively updated, requiring no human intervention. Thresholds can be specified independent of data characteristics. Third, the detector has a unique output, spiking probability map, that can reliably predict spike occurrence by assigning probabilities to neural signals. By outputting spiking probability maps, 16-channel raw data are compressed from 10.24 Mbps to 160 kbps, feasible for reliable wireless transmission [38]–[41].

The rest of the paper is organized as follows. Section II outlines the proposed algorithm and presents the chip architecture. Section III discusses the design trade-offs and circuit implementation of individual blocks. Section IV presents the prototype design. Section V evaluates the chip performance with experiment results and comparisons with other detectors. Section VI gives concluding remarks.

II. EC-PC DETECTOR AND SYSTEM ARCHITECTURE

In this section, we start with an overview of the EC-PC spike detection algorithm, followed by the introduction of the system architecture implemented in hardware. Techniques to reduce circuit power and area consumptions on the architecture-level are exploited and discussed.

A. Algorithm Overview

We have reported an EC-PC spike detection algorithm in [42] and outlined in Table I. Our research shows that in Hilbert space *in vivo* noise follows an exponential component (EC) and extracellular spikes follow a polynomial component (PC). By online estimating both EC and PC, the algorithm can quantitatively predict the occurrence of neural spikes at any time based on a spiking probability map, which plots the probabilities of each data point being part of a spike over time. Specifically, EC-PC tries to train two straight lines in the linear-log and log-log scales of neural data distributions, which are represented by two pairs of coefficients for EC and PC, respectively. Spikes can be detected by thresholding on the probability map. In addition, the probability threshold has a unique feature that predicts the *Precision* of detection, defined as the percentage of correctly detected spikes in all detected spikes [43]. For example, around 50% detected spikes are true spikes with a 50% threshold. This prediction is valid within a wide range of different SNRs, firing rates, and background noise. It can significantly simplify the threshold setting in real-time multi-channel neural recording experiments. Fig. 1 illustrates the decomposition of neural data

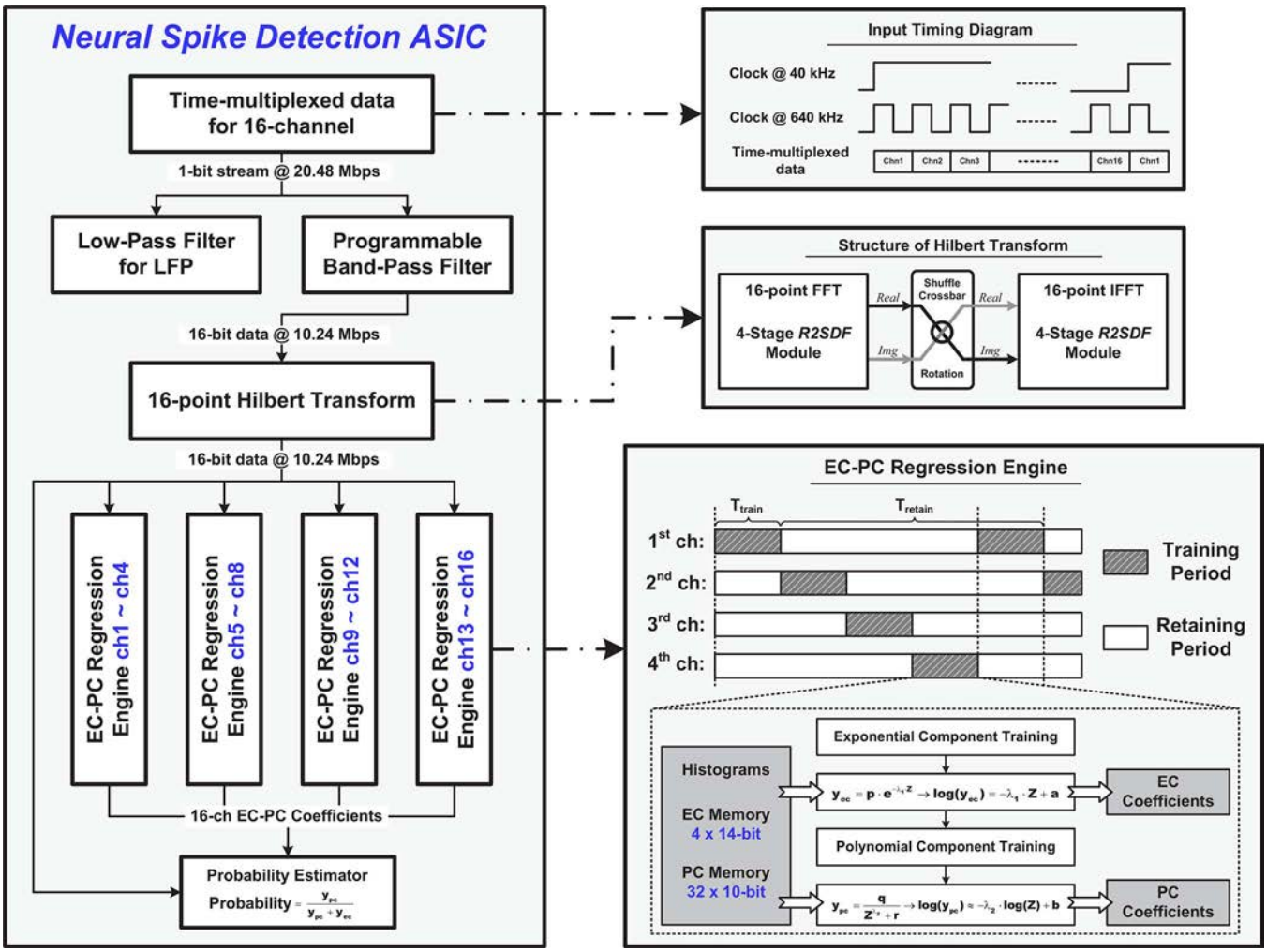


Fig. 2. Block diagram of the proposed EC-PC spike detection ASIC.

into EC and PC, as well as the spiking probability map with corresponding band-pass filtered neural data.

B. System Architecture

Fig. 2 shows the proposed 16-channel neural spike detection ASIC. The inputs to the chip are 16-channel time-division multiplexed neural data, serialized and encoded for processing. Each channel neural data are sampled at 40 kHz and packaged in 32 bits, where 16 bits are used for data representation and the rest are for protocols. The use of a 40 kHz sampling frequency is mainly to be consistent with the on-chip integrated analog frontends and ADC circuits which operate on 40 kHz as a reference sampling rate, as well as offer better waveform alignment precision. This word-length is chosen to enable the simultaneous representation of spikes and local field potentials (LFP), which are usually contaminated by artifacts and interferences that can be large in magnitude. The chip can record both LFPs and spikes simultaneously: LFPs are separated from the input by low-pass filtering the neural data at a corner frequency of 250 Hz. To obtain spikes, a programmable band-pass filter with default corner frequencies at 300 Hz and 8 kHz is used. The corner frequency of the band-pass filter can be configured anywhere between 5 kHz and 9 kHz with over 64 dB stop-band attenuations and less than 0.08 dB pass-band ripples. This is to provide maximum flexible programmability given the variability of testing subjects and application requirements [24].

The band-limited data are then fed into the Hilbert transform module, which is implemented as a combination of fast Fourier transform (FFT) and inverse-FFT (IFFT) with an intermediate rotation in the frequency domain. We chose to implement the Hilbert transform module as pipelined FFT-IFFT instead of time-domain convolution to facilitate multichannel hardware sharing. The outputs of Hilbert transform module enter the EC-PC regression engines, where Hilbert transformed data are first normalized to their estimated variances and accumulated to build histograms. A fully autonomous training mechanism is realized in the regression engines to extract the EC-PC parameters from histograms for each channel within 2.5 sec. At last, a probability estimator is deployed to calculate the spiking probability maps based on the trained EC-PC parameters. A winner-take-all strategy is implemented in the probability estimator, where the data sample with the highest probability score in a 64-point sliding window is identified. By outputting the probability scores associated with the identified data points, a 64× data rate reduction from 10.24 Mbps to 160 kbps is achieved, facilitating wireless data transmission.

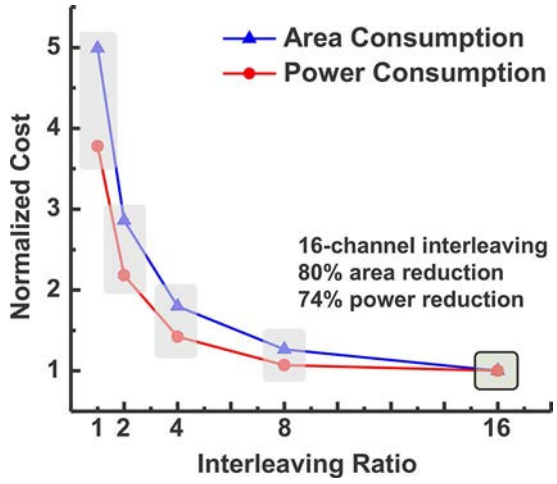


Fig. 3. Comparison of power and area reductions for different interleaving ratios. All values are normalized to the 16-channel interleaving.

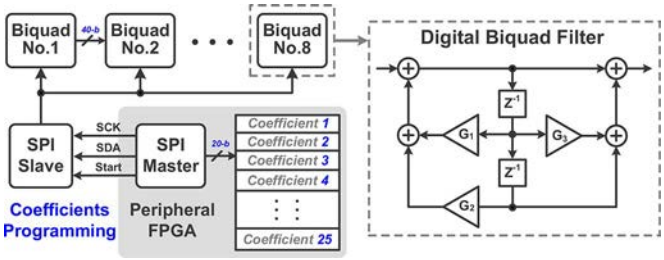


Fig. 4. Structure of the band-pass filter. The IIR filter is configured in a cascade form, with each stage being one biquad filter. Filter coefficients are programmed through the SPI, communicating with the peripheral FPGA where coefficients are stored.

The behavior of the autonomous training mechanism is described as follows. Histograms are used to estimate probability distributions of neural data. Ideally, each channel should have its own histogram to keep track of neural dynamics. However, 16-channel histograms would require a large amount of memory and consume excessive power and area. To avoid the storage overhead, we split the operation of EC-PC regression into two sessions, as shown in Fig. 2. The regression starts with an accumulation of neural data into histograms, which lasts for T_{train} and is executed sequentially from channel to channel to approximate data distributions. The length of T_{train} is evaluated in Section III.C. At the end of T_{train}, EC-PC parameters of the current channel are derived from the accumulated histogram within 0.75 ms. After that, the channel enters a retaining phase (T_{retain}) where parameters remain unchanged and are used for spiking probability estimation. Meanwhile, regression engines are allocated to other channels for histogram training. To balance hardware cost and EC-PC parameter updating latency, we have deployed four EC-PC regression engines to process 16-channel neural data, which results in a $4\times$ reduction in storage cost compared with a fully parallel EC-PC regression configuration at a negligible performance loss.

C. Interleaved Structure

Interleaving is an effective technique to save circuit power and area by sharing combinational logic across channels with increased clock frequencies, which leads to a trade-off analysis between the reduced leakage power and increased switching power to determine the optimal interleaving ratio [30]. To maximize the power and area saving, all modules in our design except EC-PC regression engines are eligible for interleaving. For a 16-channel design, candidates of interleaving ratios include 1, 2, 4, 8, and 16. We have implemented each option in Verilog RTL description and synthesized it using a 0.13 μm process standard cell library. Neural data recorded from *in vivo* preparations are used to simulate the synthesized netlists to obtain realistic switching activities for accurate power estimation. Based on the post-synthesis simulation results, the area and power consumptions of all five candidates are normalized to the smallest and plotted in Fig. 3: the 16-channel interleaving ratio gives the best power and area reduction by up to 74% and 80%, respectively, compared with the non-interleaved version, thus is chosen in our design.

III. Circuit Blocks

In this section, we discuss the design and implementation of individual circuit blocks in terms of the trade-offs and optimizations of functionality, performance, and hardware complexity. The programmable band-pass filter will be discussed first, followed by the Hilbert transform module, the EC-PC regression engine, and the spiking probability estimator. We also compare the hardware complexity of our EC-PC detector with other popular detectors. Finally, the processing latency based on our implementation is discussed.

A. Band-Pass Filter Design

In general, infinite impulse response (IIR) filters are more efficient than finite impulse response (FIR) filters to achieve almost identical specifications. Therefore we used an elliptic IIR filter to realize the band-pass filter. The main advantage of elliptic filter is that it has the sharpest transition bands for a given order than any other type of IIR filters. In addition, the amount of ripples and attenuations of elliptic filters are independently adjustable, which is desirable for a reconfigurable filter design. In our implementation, the elliptic filter is arranged in the cascade structure, which can have better performance on hardware complexity and coefficient sensitivity against arithmetic roundoff and coefficient quantization compared with other filter structures, including direct form II, parallel, and lattice, etc [44]. As shown in Fig. 4, the 16-order band-pass filter consists of 8 digital Biquad filters, and each Biquad filter has a set of 3 coefficients. All the 25 coefficients, including the overall gain factor, are programmed through a serial peripheral interface (SPI) from an off-chip FPGA, and can be online updated. The word-lengths of coefficients and intermediate data are set as 20-bit and 40-bit, respectively, which are determined by fixed-point modeling of the filter structure in Simulink using synthetic data with varied amplitudes, firing rates, and SNRs to ensure low ripples and reserve original spike waveforms without data overflow. Another motivation of the coefficient word-length is to provide more accurate tuning of pass-band, which is helpful to improve the signal SNR by better isolating spikes [45].

The corner frequencies of the band-pass filter are at 300 Hz and 8 kHz, by default, and can be programmed in wide ranges (250–600 Hz and 5–9 kHz) with over 64 dB stop-band attenuations and less than 0.08 dB pass-band ripples. Simulations results confirmed that the achieved filtering introduces tolerable distortions to neural data such that the incurred deviations of estimated EC-PC parameters are less than 0.1%.

B. Hilbert Transform

The use of Hilbert transform is 1) to estimate the probability density function (pdf) of neural data; 2) to simplify EC-PC decomposition, since neural data tend to have more compact representations in Hilbert space. In general, Hilbert transform can be implemented as time-domain convolution or FFT plus IFFT in the frequency-domain. We chose to implement it in the frequency domain thus can leverage existing efficient FFT design techniques. In the frequency domain, Hilbert transform corresponds to a $\pm\pi/2$ rotation [46] and is described as

$$H(k) = \begin{cases} -i & \text{for } k = 1, 2, \ldots, N/2 - 1 \\ 0 & \text{for } k = 0 \text{ and } N/2 \\ i & \text{for } k = N/2 + 1, \ldots, N - 2, N - 1 \end{cases} \quad (1)$$

where $H(k)$ is the rotated sequence, k is the index and N is the length of the Hilbert transform. The rotation can be simply realized by changing the sign and swapping the real and imaginary parts of FFT outputs.

The length of Hilbert transform is an important design parameter. A longer series give more accurate estimation of data distribution at the cost of proportionally increased circuit area, power consumption, and processing latency. To find an optimal length, we have compared several candidates from 4-point to 128-point in terms of the accuracy of histograms. Candidate lengths are restricted to be power of 2 to allow efficient FFT design. 100 sequences neural data extracted from *in vivo* recordings are used in simulation. In each trial, a Hilbert transform with the same length as the testing sequence is used to obtain the ground truth. Simulation results using candidate lengths of Hilbert transform are obtained by averaging over the 100 sequences and compared with the ground truth. The result is evaluated in R^2 score and plotted in Fig. 5. It shows that the histogram accuracy saturates from the 16-point with an over 97% similarity. The area consumption grows linearly with the length of Hilbert transform. Compared with the 16-point, the 32-point design achieved a less than 1% accuracy improvement at the cost of doubled area consumption and processing latency. Therefore, the 16-point Hilbert transform is chosen in our implementation to trade minor improvement in accuracy in favor of increased area saving.

Commonly used architectures to implement FFT and IFFT include structures of memory-based [47], pipelined [48], array [49], and cached-memory [50]. Among these structures, the pipelined provides a balanced trade-off between area and processing latency [51]. Specifically, the choice of pipeline structure is necessitated by the need to interleave multiple channels, which enables significant hardware sharing so as to save computational resources. The pipelined approach has two

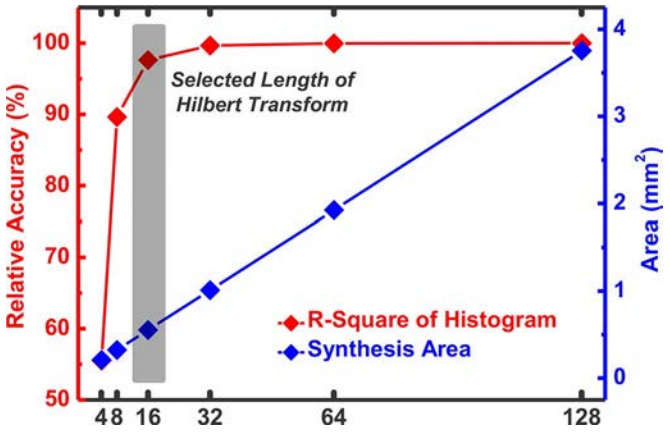


Fig. 5. Comparison of different lengths of Hilbert transform using R^2. A Hilbert transform with the same length of testing data is used to obtain the ground truth histograms. The x-axis: different Hilbert transform lengths from 4-point to 128-point. The y-axis: coefficients of determination R^2 of the histograms derived from candidate lengths relative to the ground truth.

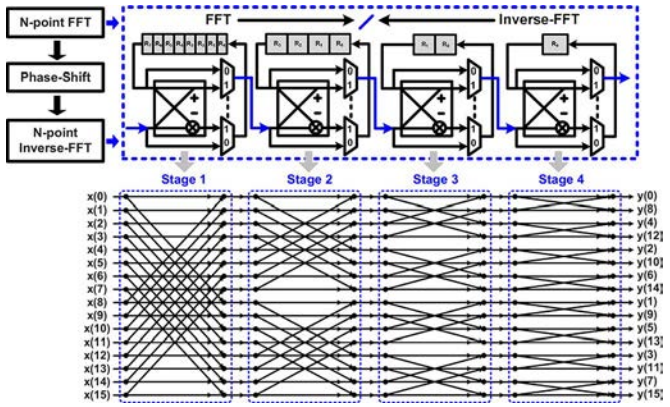


Fig. 6. Illustration of a 16-point Hilbert transform module implemented as FFT and IFFT in the radix-2 single-path delay feedback structure.

major types which are multi-path delay commutator (MDC) and single-path delay feedback (SDF) [52]. We prefer SDF because of its lower hardware complexity. The detailed implementation of the 16-point FFT with the radix-2 SDF (R2SDF) structure is given in Fig. 6. The 16-point FFT is a cascade of four R2SDF butterfly units, each of which is responsible for the computation in one stage of the decimation-in-frequency FFT. Since the summation in each butterfly unit may increase the word-length of intermediate data, the word-length of the buffers are extended with a step of 1-bit per stage to avoid overflow. The 16-point IFFT can be easily designed as a transpose of its FFT counterpart. In summary, the Hilbert transform consumes 370-bit storage and has a total processing delay of 0.8 ms.

C. EC-PC Regression Engine Design

Towards an efficient ASIC implementation, it is important to reduce on-chip storage and enable hardware sharing. In the rest of this section, related design trade-offs are discussed.

1) Evaluation of Training Period: In general, PC converges slower than EC thus the bottleneck, and the convergence is slower on a low firing rate sequence than a high firing rate one. Therefore, the training period T_{train} needs to be larger than

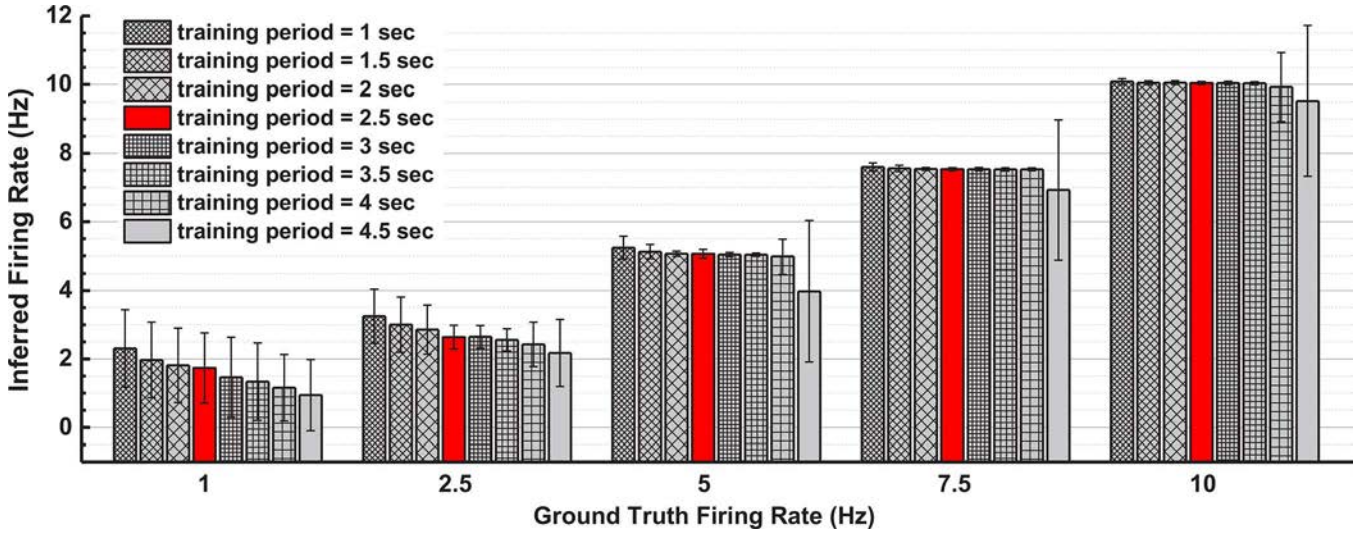


Fig. 7. Evaluation of the training periods in terms of the accuracy of inferred firing rates. Ground truth firing rates are 1 Hz, 2.5 Hz, 5 Hz, 7.5 Hz and 10 Hz. For each firing rate, we run 100 trials with 9 different training periods. In each trial, the testing sequence is divided into segments with each consisting of $T_{\text{train}}+T_{\text{retain}}$ data, where the first T_{train} data is used for parameter training. The inferred firing rate for each training period is averaged from 100 trials. The length of all trials for one training period is $100 \times (T_{\text{train}} + T_{\text{retain}})$, where in each $T_{\text{train}} + T_{\text{retain}}$, the first T_{train} is used for parameter training.

a minimum to ensure PC convergence. On the other hand, as shown in Fig. 2, EC-PC parameters of one channel remain unchanged during T_{retain}, which is 3 times of T_{train}. This makes a long T_{train} unfavorable because the unchanged EC-PC parameters during T_{retain} may be out-dated for the evolving neuronal properties and experimental conditions. Additionally, a long T_{train} would result in a long initial waiting period to train all channels once. To evaluate the trade-off between required convergence time and frequent parameter updating, we measure the means and standard deviations of inferred firing rates obtained with different T_{train} from 1 to 4.5 sec with an increment of 0.5. The inferred firing rate can serve as an intuition on the convergence of histograms. The accuracy of detection based on the selected T_{train} and histograms is verified in Section V-C. Since typical firing rate of one inactive neuron falls in the range of 1 to 10 Hz [20], we run 100 trials for each of five firing rates (1, 2.5, 5, 7.5, and 10 Hz) over all different T_{train}. The length of each trial for a particular training period is $100 \times (T_{\text{train}} + T_{\text{retain}})$. We are interested in low firing rate situations to determine a lower boundary of T_{train} to accumulate sufficient PC information.

Fig. 7 summarized the simulation results. For 1 Hz firing rate, all T_{train} except the 4.5 sec have large deviations, and the degree of deviation is inversely proportional to the length of T_{train}. This is because synthesized spike events follow a Poisson distribution, which makes the spike occurrence irregular and thus difficult to be covered and counted by shorter T_{train}. The problem with the 4.5 sec option is that it will incur an excessively long initial waiting time of 18 sec. In the rest of the simulations, for small T_{train} ($\leqslant$2 sec), the deviations and the fluctuations of estimated firing rates are caused by incomplete PC convergence; For large T_{train} ($\geqslant$4 sec), the imperfections are due to the out-dated EC-PC parameters in the long retaining session. Among all the candidates, the 2.5 sec training period achieves a balanced performance in terms of the estimation accuracy, fast parameter updating, and initial waiting period.

2) Evaluation of Histograms: The storage required for histograms would consume major circuit area and power without optimization due to two parameters: the bin width and the number of bins. In general, bins with large width can smooth the envelope curve of histograms and facilitate linear regression. We have used uniform bin width to simplify histogram design and empirically set the bin width to be 0.25. Based on the that, the word-length of bins are determined by simulations with varied firing rates (1–150 Hz) and SNRs ($-$5–15 dB), confirming that 14-bit and 10-bit are sufficient to represent histogram bins in the EC and PC dominant regions.

Comparative experiments based on 100 *in vivo* data sequences have been performed as in Fig. 8 to determine the bin numbers. First, ideal EC-PC regression is performed without imposing any constraints on storage to hold histograms, and the fitted parameters are averaged over the 100 trials and used as ground truth. Next, EC-PC regression using different numbers of bins are performed and the fitted parameters are averaged and compared with the ground truth. All candidate bin numbers are assumed to be power of 2 to facilitate time-multiplexing thus enable hardware sharing. Since PC is more unstable than EC and more sensitive to training parameters, we measured the influence of 1% parameter error of λ_2 in Fig. 9 as the similarity score in R^2 between the ideal spiking probability map and the one obtained with biased λ_2. Fig. 9 show that 1% PC error on λ_2 would result in a R^2 of 90.23%–99.93% with varied SNRs and firing rates. Based on the simulation results on accuracy and storage cost, we have chosen 4 bins for EC and 32 bins for PC regression. The choice of 4 bins for EC instead of 8 bins is to trade-off the minor parameter training accuracy for halved storage cost and computational resources. A trend can be observed that the PC error of 16 bins is slightly higher and more diverse than others, while the rests are almost the same. Therefore, we chose 32 bins for PC estimation.

3) EC-PC Regression Engine: Fig. 10 shows the architecture of the EC-PC regression engines. The word-length of bins

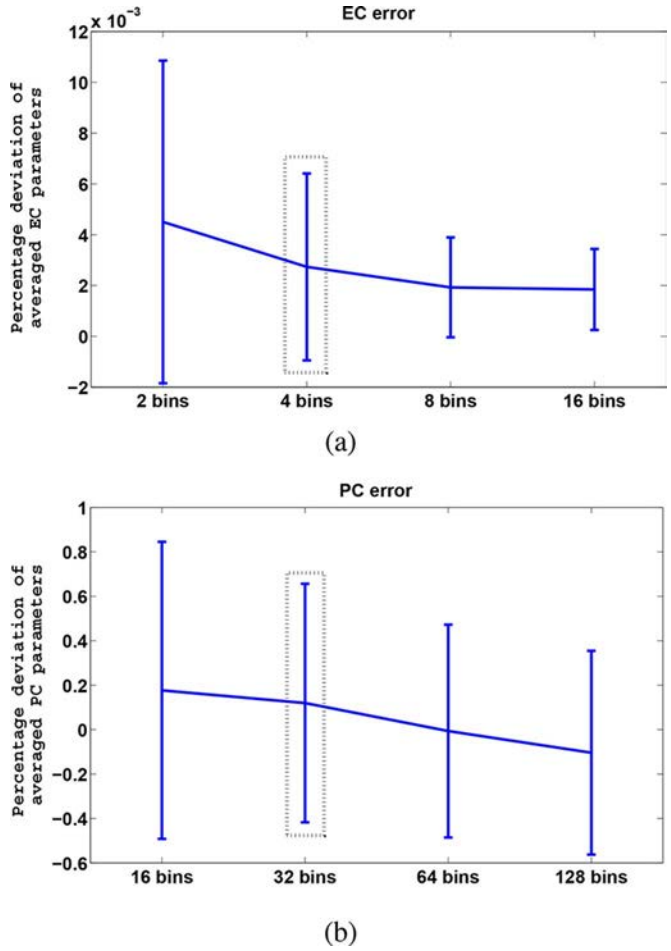


Fig. 8. Evaluations of EC and PC bin numbers for regression. More than 200 consecutive bins from $Z = 0$ to $Z = 50$ are used. Several bin numbers are compared with the ideal case in terms of the fitted parameters. Each candidate has been run over 100 20-sec data sequences and the averaged parameters are shown with error bar. (a) Evaluation of EC bins using EC parameters. (b) Evaluation of PC bins using PC parameters. The x-axis: histogram bin numbers; the y-axis: the averaged EC-PC parameter relative error in percentage.

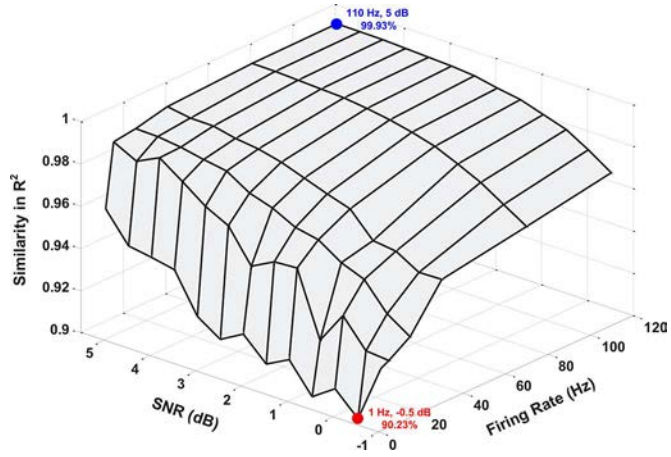


Fig. 9. Quantitative measurement of 1% PC error in R^2 between biased spiking probabilities and ideal ones. Each dot on the surface is averaged from the training results of 100 datasets, each of which is synthesized under one particular combination of SNR and firing rate. Tested SNRs are from -0.5 to 5 dB with an increment of 0.5. Tested firing rates are 1, 5, 15, 30, 70, and 110 Hz.

in histograms for EC and PC estimation are 14-bit and 10-bit, respectively. Histograms are implemented in register arrays instead of SRAMs to simplify the frequent updating operations. As discussed in Section II.B, the structure of one histogram consisting of $4\times$ EC bins and $32\times$ PC bins is shared by 4 channels sequentially, and 4 histograms are deployed to compensate for the training latency caused by hardware sharing. The consumed register array is 1.504 kb, achieving a $4\times$ reduction in storage cost compared with a fully parallel configuration. At the end of training sessions, the values of EC and PC bins are time-multiplexed and processed by the curve fitting units, which performs two regressions in the linear-log and log-log scales, respectively. The regression takes less than 0.75 ms to finish. The estimated EC-PC parameters of all 16 channels are used to update the parameters of the spiking probability estimator.

D. Hardware Complexity

The complexity of our design is evaluated in comparison with AT based on root-mean-square (AT-RMS), AT based on median (AT-median), NEO, and MTEO. These detectors are simple in structure and suitable for real-time application. To give a fair comparison, all detectors are translated into single-channel RTL descriptions and synthesized using the same 0.13 μm CMOS process used for this design. Necessary components include band-pass filters for de-noising signals and sliding windows for threshold estimation or parameter updating. The storage requirements for AT-RMS, NEO, and MTEO are trivial: the parameters for thresholding can be updated by in-place computation. The estimation of medians in AT-median requires a large amount of memories for sorting operations, resulting in relatively high power and area consumptions [37]. Details of the detector implementations are given as follows.

- **AT-RMS**: The threshold is set as

$$\text{Thr} = k \cdot \sigma_n \tag{2}$$

  where σ_n is the data standard deviation and is estimated from a sliding window of several seconds.
- **AT-median**: The threshold is determined as

$$\text{Thr} = k \cdot \sigma_n, \quad \sigma_n = \text{median}\left\{\frac{|x(n)|}{0.6745}\right\}. \tag{3}$$

  The median is estimated from a sequence of on-the-fly stored and sorted neural data. The sliding window for median estimation is 128-point, corresponding to 3.2 ms neural data. Shorter sliding windows may hardly cover a typical spike, tending to compromise estimation accuracy.
- **NEO**: The NEO of neural data is defined as

$$\psi[x(n)] = x^2(n) - x(n+1)x(n-1) \tag{4}$$

  and the threshold is set as a scaled mean of NEO [17]

$$\text{Thr} = C\frac{1}{N}\sum_{n=1}^{N}\psi[x(n)] \tag{5}$$

  where N is set with the same length as the sliding window in AT-RMS.

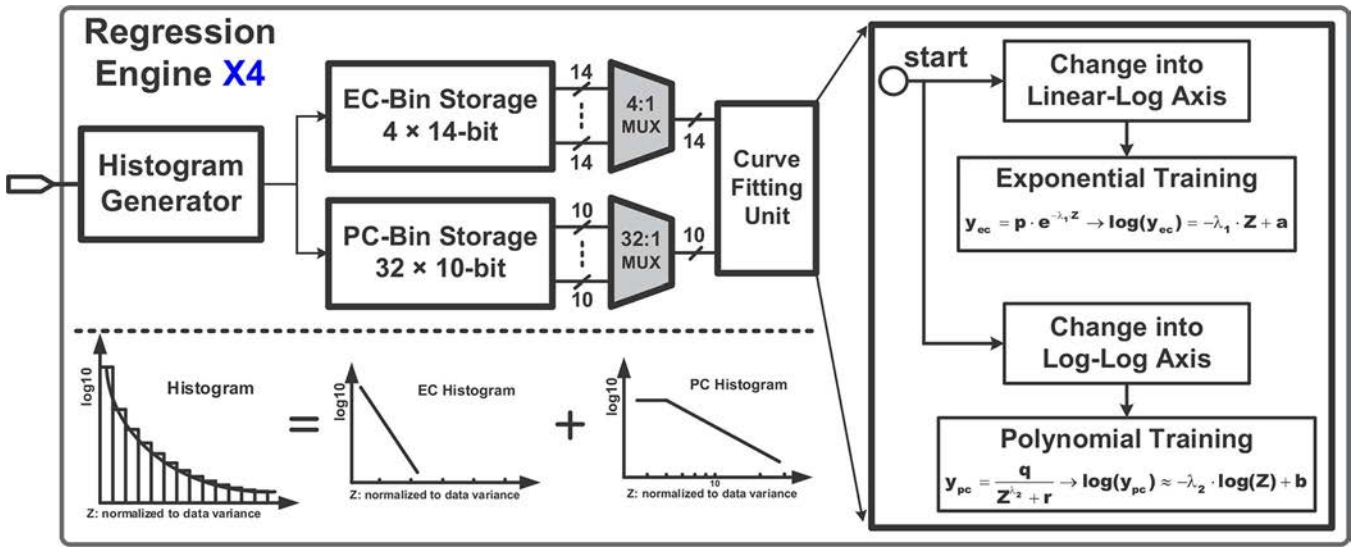


Fig. 10. Architecture of multichannel EC-PC linear regression engine.

TABLE II
SUMMARY OF NORMALIZED POWER AND AREA CONSUMPTIONS OF SEVERAL DETECTORS

	Power[a]	Combinational[b]	Sequential[b]
EC-PC	1	5.871	1
AT-RMS	0.263	2.001	0.187
AT-median	1.570	11.675	1.317
NEO	0.259	1.920	0.205
MTEO	0.512	3.543	0.785

[a] Normalized to the power of EC-PC detector.
[b] Normalized to the sequential circuit area of EC-PC detector.

- **MTEO**: MTEO is composed of several k-TEO, which is defined as

$$\psi_k[x(n)] = x^2(n) - x(n+k)x(n-k) \tag{6}$$

where k is the resolution parameter to enhance the performance of TEO. Normally MTEO requires about 6 or 7 k-TEOs to cover the sampling frequencies from 10 kHz to 40 kHz. Since the sampling frequency in our design is 40 kHz, the values of k can be chosen as 1, 3, and 5 to reduce complexity [53]. The smoothing window of each k-TEO is implemented as a hamming window of length $4k+1$. The thresholding operation is the same as NEO.

Power and area consumptions of these detectors are tabulated in Table II, where values are normalized to the power and sequential circuit area of EC-PC detector. The hardware cost of EC-PC is less than AT-median. The structure of NEO and AT-RMS are more suitable for multichannel implementation, yet their performance are suboptimal. Our EC-PC detector has a moderate power consumption and provides useful features for reliable recording, achieving balanced trade-off among hardware complexity, functionality, and detection performance.

E. System Latency

In applications of neural prostheses or brain-computer interfaces (BCI), low latency is desirable to enable fast responses and instantaneous feed-back control. Our system has an initial waiting period of 10 sec caused by the sequential and cyclic training of each EC-PC regression engine over 4 channels, which is affordable. Afterwards, chip outputs are reported almost instantly with trivial delays of less than 2.5 ms contributed by the band-pass filter, Hilbert transform, etc. It can be concluded that our system is eligible for neural prosthetic devices or BCI experiments in terms of the requirement on latency.

IV. PROTOTYPING

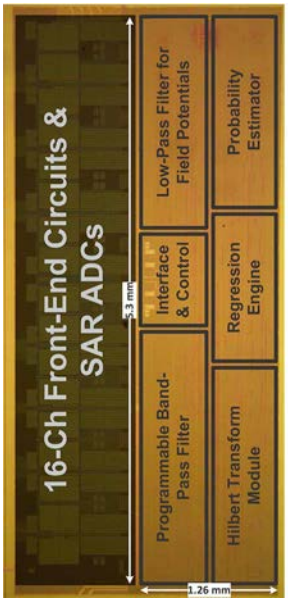
A. Chip Summary

The multichannel spike detection ASIC has been implemented and tested. Fig. 11 shows the die micrograph and summarizes the chip performance. The chip occupies a core area of 6.71 mm^2 in a 0.13 μm CMOS process and consumes a total power of 1.36 mW for 16 channels from a 1.2 V supply voltage, corresponding to 85 μW and 0.42 mm^2 per channel. The use of a 1.2 V supply is to facilitate the integration of the digital ASIC with the analog frontends and ADCs, which are on the same die as the digital ASIC and work exclusively with a 1.2 V supply voltage. Level shifters would be on-chip integrated in future developments to allow separate power supply of the digital ASIC and further power reduction through voltage scaling.

B. Experiment Setup

The chip is bonded onto a small printed circuit board (PCB) with a size of 1.9 cm × 1.5 cm connected to a NeuroNexus microelectrode array. A credit card size board (5.4 cm × 7.5 cm) including FPGA, SRAMs, level shifters, power management and communication is used as an evaluation platform to provide a complete testing bench-top that requires only one USB cable for power supply and data storage, as shown in Fig. 12. The chip bonding board communicates with the evaluation board through wired connections.

The experiment setup features single-board solution of both power and data links. The power link originates from a 5 V



Whole System	
Technology	CMOS .13 µm Process
Core Area	6.71 or 0.42 mm^2/ch
Reference Clock	20.48 MHz
Supply Voltage	1.2 V
Num. of Channels	16
Power	1.36 mW or 85 µW/ch
Memory Usage	1.536 kb
Training Period	2.5 sec
Interface & Clock & Control Logic	243.4 µW 23.1%
Low-Pass Filter for LFP	115.6 µW 8.5%
Programmable Band-Pass Filter	225.8 µW 16.6%
Hilbert Transform	364.4 µW 26.8%
EC-PC Regression Engine	163.2 µW 12.1%
Probability Esimator	175.4 µW 12.9%

Fig. 11. Chip die micrograph and measured circuit specifications. Power consumptions of individual blocks are also given.

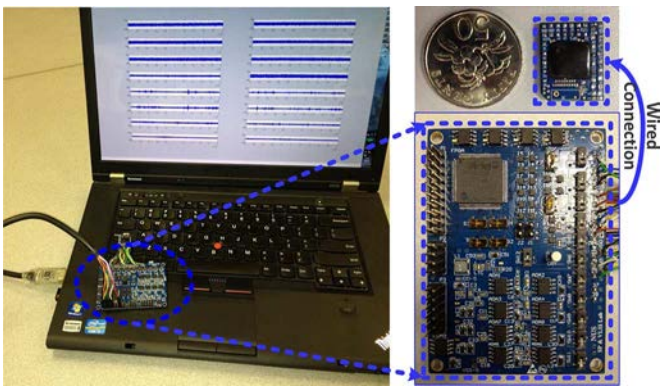


Fig. 12. The evaluation board used for both bench-top and *in vivo* experiments. Chip bonding board is also shown, which communicates with the evaluation board through wired connections.

supply in the USB port, integrating one low dropout regulator and several DC-DC converters. The data link, as shown in Fig. 13, is used to store the chip's three output bit-streams. A programmable USB cable is configured in SPI mode to transfer data between the host PC and the evaluation board. All interfacing are coordinated by a on-board FPGA. The operations of the chip and the FPGA are synchronized to a on-board 20.48 MHz reference clock, which is asynchronous to the system clock of the host PC. To enable continuous data communication between the host PC and the evaluation board, a pair of SRAMs are configured in ping-pong mode. During data acquisition, the FPGA directs the chip outputs to be stored in one of the two SRAMs. Once the SRAM is full, the other SRAM is scheduled to accept the chip outputs unintermittently. Meanwhile, the host PC will read out the content of the unused and full SRAM. As shown in Fig. 13, a flag signal DD is asserted each time when one SRAM is full to notify the host PC for data collection. The data feeding from the host PC to the

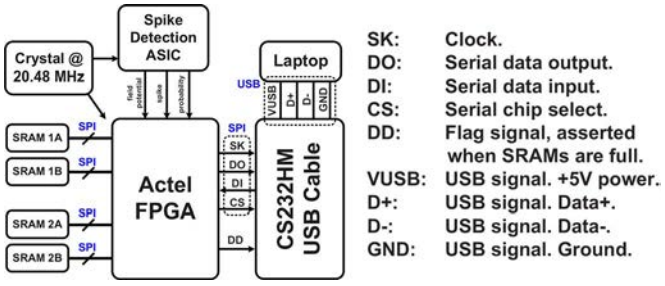


Fig. 13. Block diagram of the designed data link for storing chip outputs.

chip works in a similar manner in the reverse direction, which is supported by another pair of SRAMs.

To further reduce the size of testing setup and make it suitable for freely moving animal recording experiments, on-chip integration of the reference generation circuits and the communication protocols with external hosts for data transmission is necessary.

C. Output Packaging Scheme

To ensure correct data transmission, we have packaged each output data sample into a frame consisting of a header and the data sequence. As shown in Fig. 14, to distinguish different channels, header "10101011" is concatenated in front of each first-channel data sample, while header "10111101" is used for other channels. To avoid mis-recognition of headers inside the data samples, a pair of redundancy bits "00" is inserted into each data sample at every six digits. Mis-recognition is therefore avoided due to the absence of "00" within headers. By doing so, one 16-bit data sample is extended to 32-bit with a sampling frequency of 20.48 MHz. Fig. 15 illustrates the format of chip outputs observed on the oscilloscope.

V. Chip Measurements

A. Data Preparation

Protocols of data synthesis for simulation is briefly introduced as follows. A library of 70 spike templates with clear waveforms is used for synthesis and the spike amplitudes are normalized by their peak values. The firing pattern of individual neuron is assumed to follow an inhomogeneous Poisson process with a refractory period of 3 ms. To simulate background activities, recorded *in vivo* data that contain a small amount of visually detectable spikes are scaled and used as background noise, which can emulate the noise introduced by the recording electronics. Specifically, the sequence used for simulating noise is randomly picked up from a more than 20-minute *in vivo* data and scaling it according to required SNR in each trial. The SNR is defined as the average peak value of spikes divided by 3σ

$$\mathrm{SNR} = \frac{\frac{1}{n}\sum_{i=1}^{n}|V_i|}{3\sigma} \tag{7}$$

where σ represents the standard deviation of background noise and V_i is the spike peak. To be consistent with our hardware implementation, the sampling frequency is 40 kHz and each spike is quantized with a precision of 16-bit.

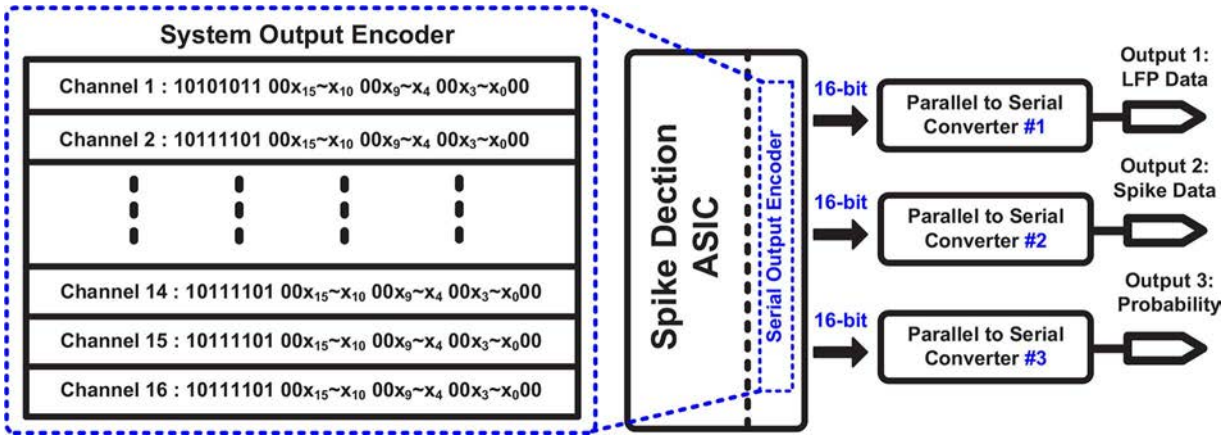


Fig. 14. Chip outputs encoding scheme.

B. Adaptability to Neural Dynamics

Neurons and neural networks exhibit nonlinearity and non-stationarity, resulting in the spatial-temporal variations of recorded waveforms. Therefore, it is desired that our chip can adapt to this variation in a real-time fashion. To test the adaptability, firing rates are derived from the output spiking probability maps and compared with ground truth. Two spike templates with distinct waveforms are used, and the firing of spikes follows a uniform distribution to minimize the bias caused by the estimation method of firing rate. The ground truth firing rate is increased abruptly from 5 Hz to 45 Hz, and the SNR is varied from 0 dB to 10 dB for different sequences. Firing rate is estimated by counting the spikes in a 1 sec sliding window along the spiking probability map, where only spikes scoring 100% are identified.

Simulation results indicate that our detector works robustly to adapt to firing rate variation in a wide range of SNRs. As shown in Fig. 16, our chip is able to report the 40 Hz firing rate increase within 0.5 sec. The adaptation speed is approximately invariant to different SNRs (0 dB, 2.5 dB, 5 dB, and 10 dB). The estimated firing rates exhibit more fluctuations before and after the transition region for low-SNR sequences than high-SNR sequences. It should be noted that the reported delays in adaptation are negatively influenced by the method of firing rate estimation. Shorter sliding windows can lead to faster response, but with more fluctuations in the estimated firing rates.

C. Performance Comparison With Other Detectors

We have run extensive experiments to evaluate the performance of our hardware detector in comparison with the implementations of AT-RMS, AT-median, NEO, and MTEO. The performance of these detectors are measured using the true positive rate (TPR) and the false positive rate (FPR), defined as

$$\mathrm{TPR} = \frac{\text{Number of correctly detected spikes}}{\text{Number of total true spikes}} \tag{8}$$

$$\mathrm{FPR} = \frac{\text{Number of false alarms}}{\text{Number of true negatives spikes}} \tag{9}$$

where the number of true negatives is estimated by subtracting true spikes from the testing sequence and dividing the length of remaining data by the typical length of a spike.

Three spike templates with distinct waveforms are used for data synthesis. In each trial, the testing data is 10 sec long, and the threshold for each detector is determined based on a 2.5 sec initial training session. Specifically, a range of thresholds are swept to find the one that gives the largest difference between TPR and FPR, and is subsequently used as the threshold. Detection results of all detectors are summarized in Fig. 17 in terms of TPR and FPR over different firing rates and SNRs. The six panels correspond to firing rates of 1 Hz, 5 Hz, 15 Hz, 30 Hz, 70 Hz, and 110 Hz. In each panel, TPR and FPR of all five detectors are measured over a wide range of SNRs from -1 dB to 5 dB with an increment of 0.5. Each dot in the figure is averaged over 100 trials with different synthetic data, and the error bar associated with each dot represents the standard deviation of the TPRs or FPRs in the 100 trials.

The results shows that EC-PC outperforms others by achieving a 10%–30% higher TPR than all other detectors over a wide range of SNRs and firing rates, especially for datasets with low firing rates and SNRs, which potentially enables detecting more spikes in noisy recording environment to establish causal connectivity. We also observed that the FPR of EC-PC is higher than some of other detectors by no more than 10%, and decreases quickly when SNR is higher. Since false alarms tend to have lower spiking probabilities, this higher FPR with low-SNR datasets can be compensated for by incorporating spiking probability scores of false alarms into subsequent processing.

D. Performance Comparisons in ROC Curves

We have run a comparative experiment of the five detectors using a public database available from [5], where the most noisy sequence is used. The result is shown in Fig. 18, where the performance is measured by the receiver operating characteristics (ROCs). A ROC curve near the top and left boundary of the plot is considered as better performance. For each detector, threshold is varied on the pre-emphasized signal from the minimum to the maximum [36]. Since both AT-RMS and AT-median pre-emphasize neural data as absolute values, their ROC curves are the

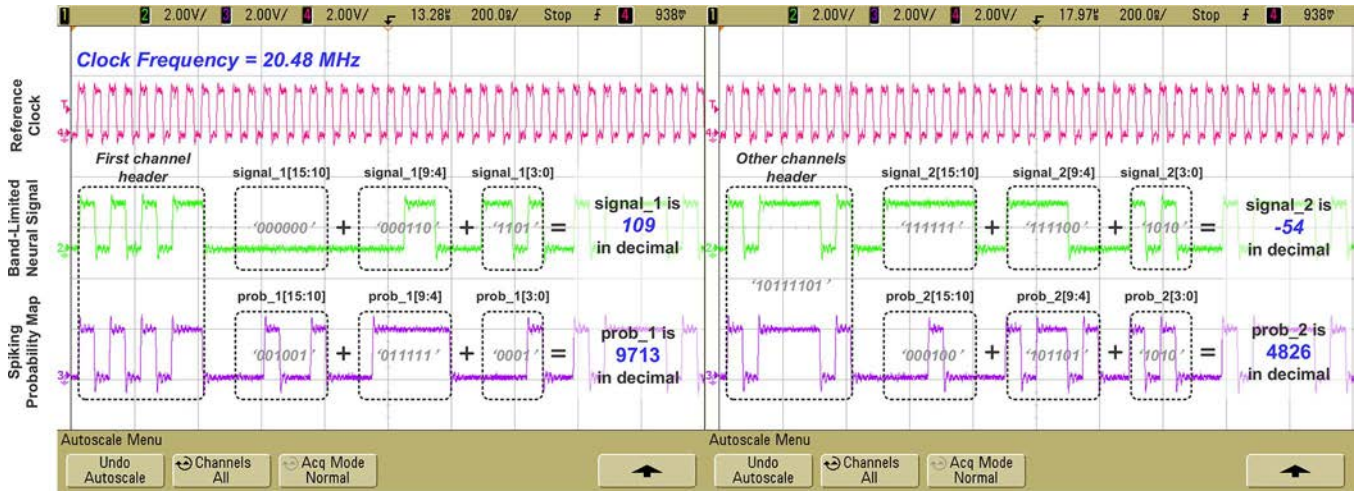


Fig. 15. Formats of chip outputs including the serialized and encoded band-pass filtered neural signals and the spiking probability maps.

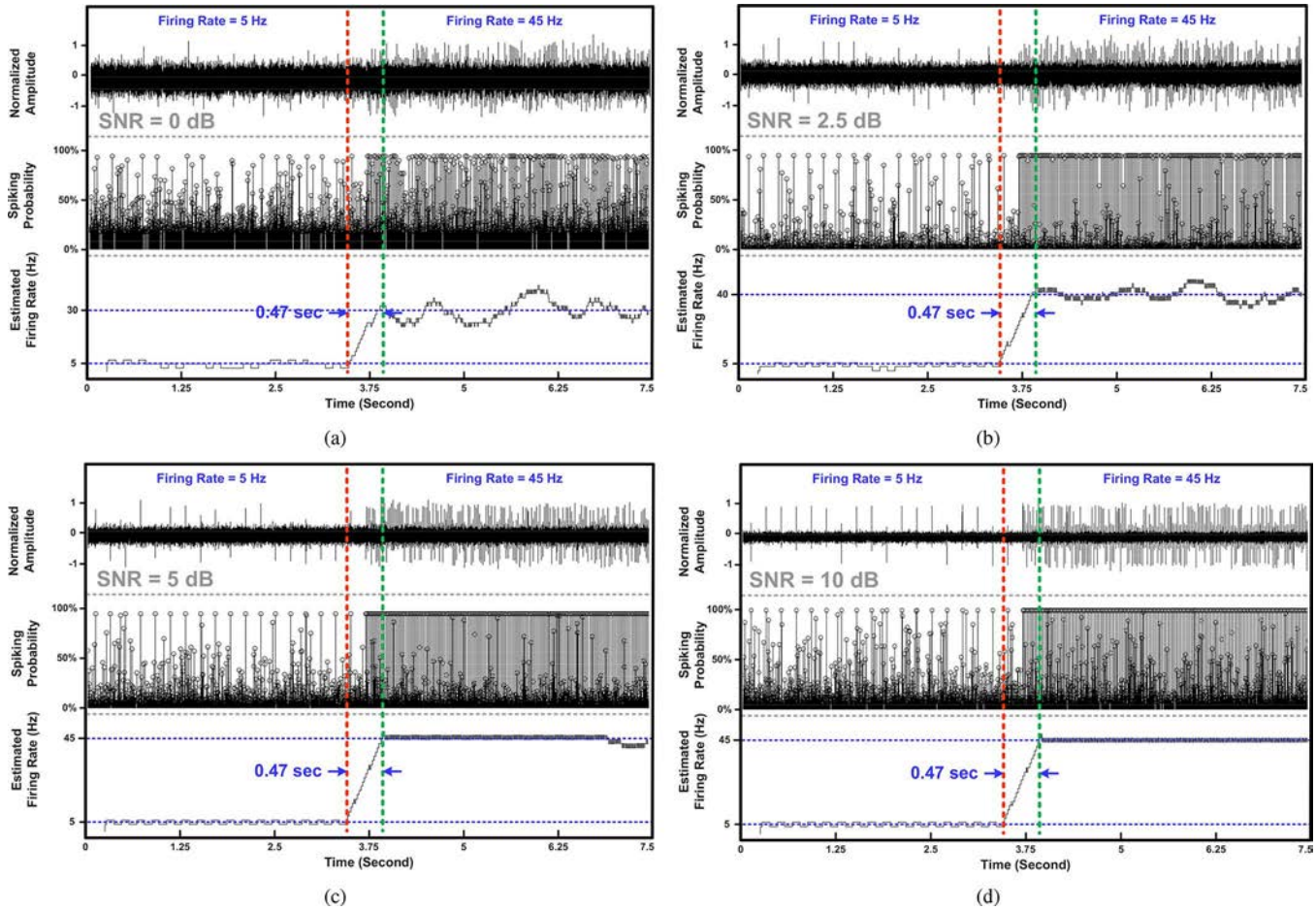


Fig. 16. EC-PC detection chip outputs of sequences with varying firing rate from 5 Hz to 45 Hz. All sequences are 7.5 sec long. The SNRs are 0, 2.5, 5, and 10 dB, respectively. In each sub-figure, the three stacked panels are neural data, spiking probability maps, and estimated firing rates. Our chip takes approximately 0.47 sec to adapt to the firing rate increase for all sequences.

same. EC-PC has achieved comparable performance as NEO and better than the rest.

In addition, we compared the EC-PC spike detection ASIC with its software version (in Matlab) using ROC curves. As shown in Fig. 19, the ROC curve of the hardware EC-PC is slightly lower than that of the software EC-PC, indicating an acceptable performance loss caused by real-time implementation. The area under ROC curves (AUC) are provided to quantify the difference, which can be mainly attributed to the limited hardware resources that the EC-PC ASIC can leverage: trade-off signal processing specifications and simplified parameter training based on approximated linear models.

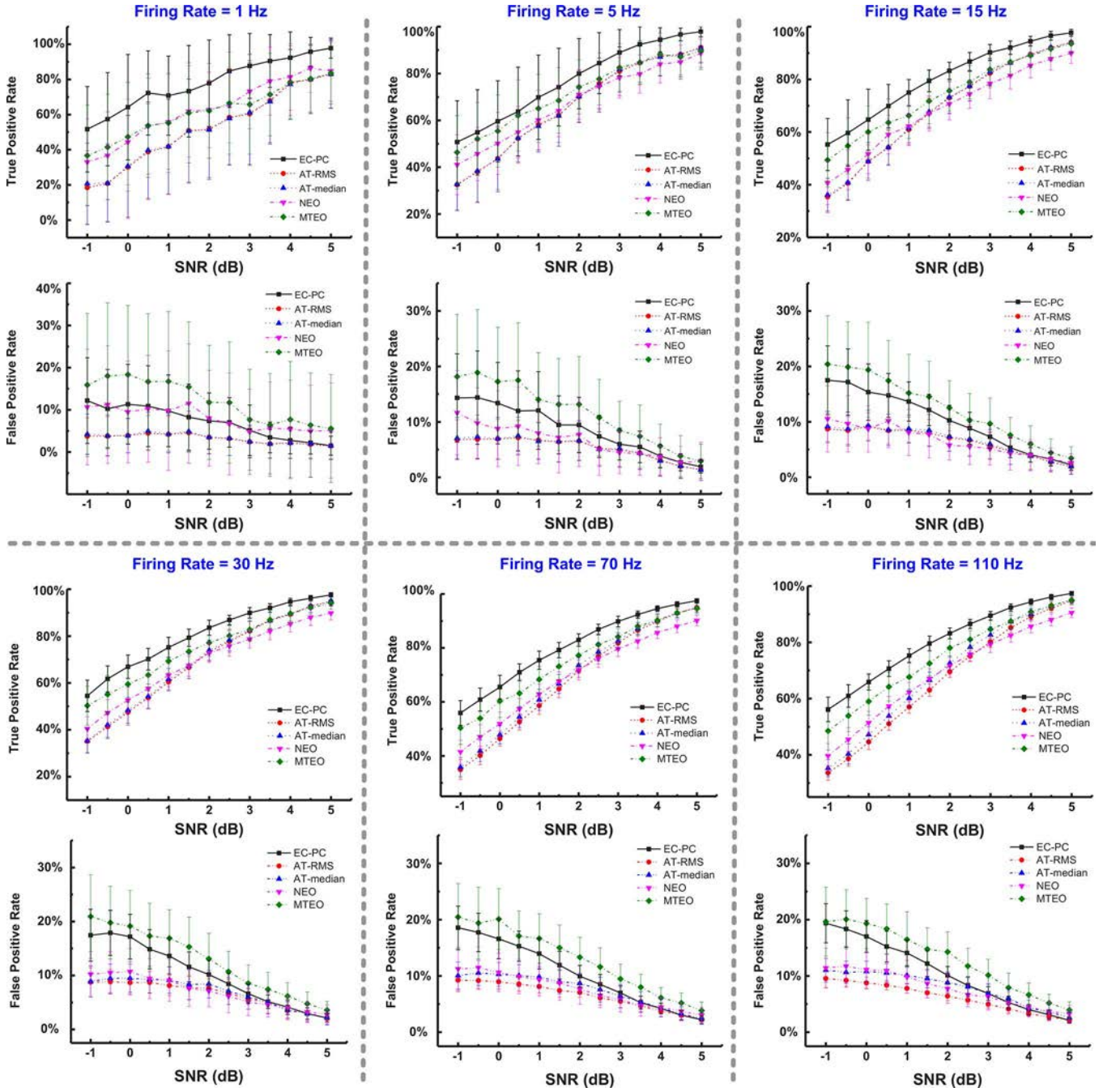


Fig. 17. Performance comparison of hardware EC-PC, AT-RMS, AT-median, NEO, and MTEO in terms of TPR and FPR in different firing rates and SNRs. SNR is adjusted from -1 dB to 5 dB with an increment of 0.5 dB, and firing rate is ranged over 1 Hz, 5 Hz, 15 Hz, 30 Hz, 70 Hz, and 110 Hz. In each sub-column, TPR and FPR measurements of all detectors for a fixed firing rate are shown, where the x-axis is SNR, and the y-axis is TPR or FPR. Each dot in the figure is averaged from 100 trials labeled with standard variations. The ranges of thresholds for each detector are: AT-RMS, 3–8$\times$ data rms; AT-median, 3–8$\times$ estimated median; NEO, 6–12$\times$ averaged $\psi[x(n)]$; MTEO, 6–12$\times$ averaged $\psi_k[x(n)]$; EC-PC, 20%–100%.

E. Comparison With Previous Detection Hardware

Measured circuits performances in comparison with previously reported neural signal acquisition and processing systems are presented in Table III. Our 16-channel EC-PC detection ASIC has a moderate power consumption and provides additional features for reliable recording, achieving balanced trade-offs among implementation cost, performances, and functionalities. In terms of the performance of detection, our chip achieves an over 82% probability of detection and a below 8% probability of false alarm for SNR above 2.5 dB, and outperforms [54] which achieves 80% for probability of detection and 15% for false alarms, respectively, with the same data SNR.

VI. Conclusion

In this paper, a 16-channel spike detection ASIC is presented. Our primary goal is to demonstrate the feasibility of real-time execution of the EC-PC spike detection algorithm, which is

TABLE III
FEATURES SUMMARY OF NEURAL RECORDING HARDWARE INCLUDING SPIKE DETECTION

Ref.	Process	Detector	Output signals	Bit/sample	Power	Power cons. / (channel · bit/sample)	Channel counts
[21]	FPGA	AT	Compressed/raw spikes; Bin counts	8 ∼ 12-bit	104 μW/ch	$< 13\mu$W	96-chn
[24]	0.35 μm	NEO	Complete spikes; Neural events	16-bit	256 μW/ch	N / A	16-chn
[25]	0.18 μm	AT	Partial/complete spikes	8-bit	138 μW/ch	17.25 μW	16-chn
[54]	0.13 μm	AT	Compressed/raw spikes	8-bit	0.2 μW/ch	0.025 μW	1-chn
[55]	0.18 μm	MTEO	Complete spikes	analog	0.78 μW/ch	0.78 μW[a]	1-chn
Ours	0.13 μm	EC-PC	Field potentials Complete spikes Probability maps Others	16-bit 16-bit 16-bit N / A	7.2 μW/ch 14.1 μW/ch 43.8 μW/ch 20 μW/ch	0.45 μW 0.88 μW 2.74 μW N / A	16-chn

[a] The power consumption is for the complete analog spike detector.

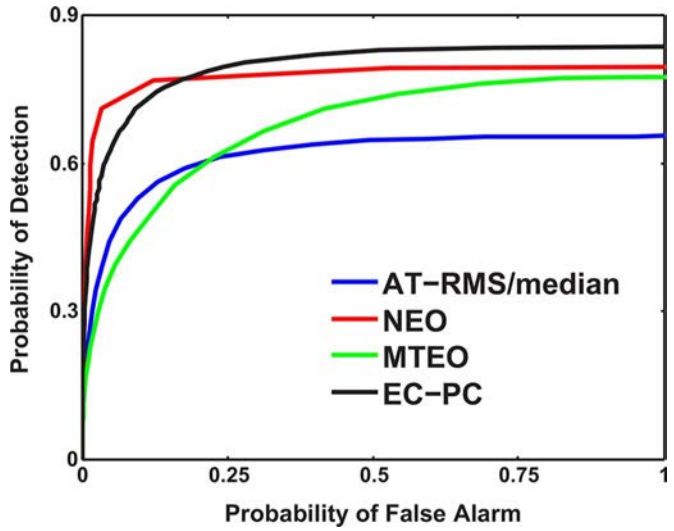


Fig. 18. ROC curves of five detectors using a standard data set available at http://www.vis.caltech.edu/~rodri/Wave_clus/Simulator.zip from [5], which is named as "C_Easy1_noise04.mat".

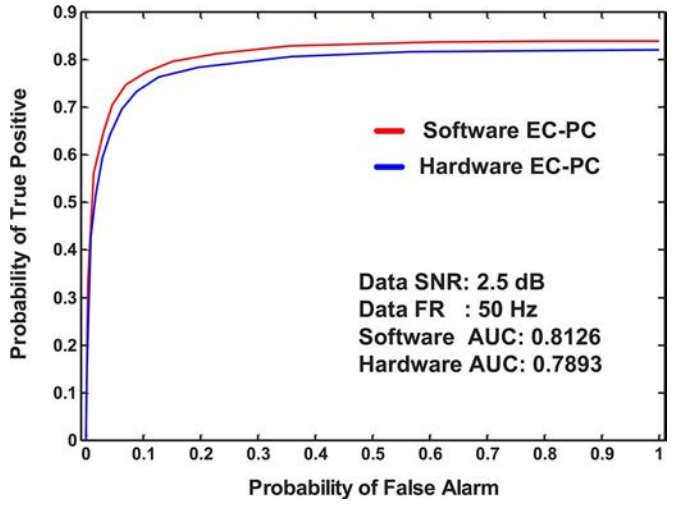


Fig. 19. ROC curves of software and hardware EC-PC. The SNR of simulated data is 2.5 dB, and the firing rate is 50 Hz. The AUCs are 0.8126 and 0.7893 for software and hardware versions, respectively.

made further challenging by simultaneous multichannel processing. In this transformation from batch-mode computation to on-line processing, we have focused on the algorithmic and architectural optimizations of the proposed method without compromising its unique features to reliably predict spiking activities. The chip performs band-pass filtering, Hilbert transform, and EC-PC regression for threshold estimation, and outputs 16-channel field potentials, spike signals, and spiking probability maps simultaneously. The achieved data-rate reduction is over 98% from 10.24 Mbps to 160 kbps when outputting spiking probability maps. To realize a cost-effective implementation, we have interleaved the design over 16 channels to enable optimum hardware sharing. The system occupies an area of 6.71 mm^2 for 16 channels and has a total power consumption of 1.36 mW, which corresponds to an averaged power of 85 μW per channel. Testing prototypes have also been developed to facilitate the use of the proposed ASIC in neural recording experiments without the requirement of human calibrations. Regarding the area overhead of this work, detailed analysis shows that a significant amount of area is consumed by the direct-mapping of many floating-point computations employed in the parameter training, suggesting much room for area improvement with fixed point optimizations. Towards a more power-efficient implementation, potential improvements based on this work include 1) reducing the data sampling frequency and resolution after band-pass filtering, 2) replacing the direct-mapping by floating-point realizations of many arithmetic computations with efficient fixed-point structures, 3) using deep sub-threshold circuit design techniques, and 4) developing customized low-power on-chip memory.

REFERENCES

[1] M. S. Lewicki, "A review of methods for spike sorting: The detection and classification of neural action potentials," *Network: Comput. Neural Syst.*, vol. 9, no. 4, pp. R53–R78, 1998.
[2] R. Chandra and L. Optican, "Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network," *IEEE Trans. Biomed. Eng.*, vol. 44, pp. 403–412, 1997.
[3] S. N. Gozani and J. P. Miller, "Optimal discrimination and classification of neuronal action potential waveforms from multiunit, multichannel recordings using software-based linear filters," *IEEE Trans. Biomed. Eng.*, vol. 41, pp. 358–372, 1994.
[4] K. S. Guillory and R. A. Normann, "A 100-channel system for real time detection and storage of extracellular spike waveforms," *J. Neurosci. Methods*, vol. 91, pp. 21–29, 1999.
[5] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, pp. 1661–1687, 2004.
[6] G. E and H. K, "Separation of action potentials in multiunit intrafascicular recordings," *IEEE Trans. Biomed. Eng.*, vol. 39, pp. 289–295, 1992.

[7] N. Mtetwa and L. S. Smith, "Smoothing and thresholding in neuronal spike detection," *Neurocomput.*, vol. 69, pp. 1366–1370, 2006.

[8] H. Kaneko, S. S. Suzuki, J. Okada, and M. Akamatsu, "Multineuronal spike classification based on multisite electrode recording, whole-waveform analysis, and hierarchical clustering," *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 280–290, 1999.

[9] S. Kim and J. McNames, "Automatic spike detection based on adaptive template matching for extracellular neural recordings," *J. Neurosci. Methods*, vol. 165, pp. 165–174, 2007.

[10] K. Kim and S. Kim, "A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio," *IEEE Trans. Biomed. Eng.*, vol. 50, pp. 999–1011, 2003.

[11] Z. Nenadic and J. W. Burdick, "Spike detection using continuous wavelet transform," *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 74–87, 2005.

[12] M. Frisch and H. Masser, "The use of the wavelet transform in the detection of an unknown transient signal," *IEEE Trans. Inf. Theory*, vol. 38, pp. 892–897, 1992.

[13] T. T. Liu and A. C. Fraser-Smith, "Detection of transients in 1/f noise with the undecimated discrete wavelet transform," *IEEE Trans. Signal Process.*, vol. 48, pp. 1458–1462, 2000.

[14] G. Zouridakis and D. C. Tam, "Multi-unit spike discrimination using wavelet transforms," *Comput. Biol. Med.*, vol. 27, pp. 9–18, 1997.

[15] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *J. Neurosci. Methods*, vol. 101, pp. 93–106, 2000.

[16] J. H. Choi, H. K. Jung, and T. Kim, "A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios," *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 738–746, 2006.

[17] K. Kim and S. Kim, "Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 1406–1411, 2000.

[18] R. R. Harrison, "A low-power integrated circuit for adaptive detection of action potentials in noisy signals," in *Proc. 25th IEEE Annu. Int. Conf. Engineering in Medicine and Biology Soc.*, 2003, pp. 3325–3328.

[19] A. M. Sodagar, K. D. Wise, and K. Najafi, "A fully integrated mixed-signal neural processor for implantable multichannel cortical recording," *IEEE Trans. Biomed. Eng.*, vol. 54, pp. 1075–1088, 2007.

[20] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE J. Solid-State Circuits*, vol. 42, pp. 123–133, 2007.

[21] M. Rizk, I. Obeid, S. Callender, and P. Wolf, "A single-chip signal processing and telemetry engine for an implantable 96-channel neural data acquisition system," *J. Neural. Eng.*, vol. 4, pp. 09–21, 2007.

[22] M. Chae, W. Liu, Z. Yang, T. Chen, J. Kim, M. Sivaprakasam, and M. Yuce, "A 128-channel 6 mW wireless neural recording IC with on-the-fly spike sorting and UWB tansmitter," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, 2008, pp. 146–148.

[23] R. Sarpeshkar, W. Wattanapanitch, S. K. Arfin, B. I. Rapoport, S. Mandal, M. W. Baker, M. S. Fee, S. Musallam, and R. A. Andersen, "Low-power circuits for brain-machine interfaces," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 3, pp. 173–183, 2008.

[24] T.-C. Chen, K. Chen, Z. Yang, K. Cockerham, and W. Liu, "A biomedical multiprocessor SoC for closed-loop neuroprosthetic applications," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, 2009, pp. 434–435.

[25] B. Gosselin, A. E. Ayoub, J. Roy, M. Sawan, F. Lepore, A. Chaudhuri, and D. Guitton, "A mixed-signal multichip neural recording interface with bandwidth reduction," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, pp. 129–141, 2009.

[26] D. J. Yeager, J. Holleman, R. Prasad, J. R. Smith, and B. P. Otis, "Neuralwisp: A wirelessly powered neural interface with 1-m range," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 6, pp. 379–387, 2009.

[27] H. Miranda, V. Gilja, C. A. Chestek, K. V. Shenoy, and T. H. Meng, "HermesD: A high-rate long-range wireless transmission system for simultaneous multichannel neural recording applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 181–191, 2010.

[28] S. Yuwen, H. Shimeng, J. J. Oresko, and A. C. Cheng, "Programmable neural processing on a smartdust for brain-computer interfaces," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 5, pp. 265–273, 2010.

[29] S. Narasimhan, H. J. Chiel, and S. Bhunia, "Ultra-low-power and robust digital-signal-processing hardware for implantable neural interface microsystems," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 2, pp. 169–178, 2011.

[30] V. Karkare, S. Gibson, and D. Marković, "A 130-μW, 64-channel neural spike-sorting DSP chip," *IEEE J. Solid-State Circuits*, vol. 46, pp. 1214–1222, 2011.

[31] C. M. Lopez, D. Prodanov, D. Braeken, I. Gligorijevic, W. Eberle, C. Bartic, R. Puers, and G. Gielen, "A multichannel integrated circuit for electrical recording of neural activity, with independent channel programmability," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 2, pp. 101–110, 2012.

[32] H. A. Alzaher, N. Tasadduq, and Y. Mahnashi, "A highly linear fully integrated powerline filter for biopotential acquisition systems," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 5, pp. 703–712, 2013.

[33] M. Yin, D. A. Borton, J. Aceros, W. R. Patterson, and A. V. Nurmikko, "A 100-channel hermetically sealed implantable device for chronic wireless neurosensing applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 2, pp. 115–128, 2013.

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

[35] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsáki, "Intracellular features predicted by extracellular recordings in the hippocampus in vivo," *J. Neurophys.*, vol. 84, no. 1, pp. 390–400, 2000.

[36] 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, 2010.

[37] 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, 2013.

[38] S. B. Lee, H.-M. Lee, M. Kiani, U.-M. Jow, and M. Ghovanloo, "An inductively powered scalable 32-channel wireless neural recording system-on-a-chip for neuroscience applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 6, pp. 360–371, 2010.

[39] Y. Gao, Y. Zheng, S. Diao, W. Toh, C. Ang, M. Je, and C. Heng, "Low-power ultrawideband wireless telemetry transceiver for medical sensor applications," *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 768–772, 2011.

[40] R. R. Harrison, H. Fotowat, R. Chan, R. J. Kier, R. Olberg, A. Leonardo, and F. Gabbiani, "Wireless neural/emg telemetry systems for small freely moving animals," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 2, pp. 103–111, 2011.

[41] J. Tan, W.-S. Liew, C.-H. Heng, and Y. Lian, "A 2.4 GHz ULP reconfigurable asymmetric transceiver for single-chip wireless neural recording IC," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 4, pp. 497–509, 2014.

[42] Z. Yang, W. Liu, M. R. Keshtkaran, Y. Zhou, J. Xu, V. Pikov, C. Guan, and Y. Lian, "A new EC-PC threshold estimation method for in-*vivo* neural spike detection," *J. Neural. Eng.*, vol. 9, no. 4, 2012.

[43] Y. Zhou, T. Wu, A. Rastegarnia, C. Guan, E. Keefer, and Z. Yang, "On the robustness of EC-PC spike detection method for online neural recording," *J. Neurosci. Methods*, vol. 235, pp. 316–330, Sept. 2014.

[44] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, no. 4, pp. 581–595, 1995.

[45] Z. Yang, Q. Zhao, and W. Liu, "Spike feature extraction using informative samples," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2008, pp. 1865–1872.

[46] F. W. King, *Hilbert Transforms*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[47] S. Magar, S. Shen, G. Luikuo, M. Fleming, and R. Aguilar, "An application specific DSP chip set for 100 MHz WTA rates," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 1988, vol. 4, pp. 1989–1992.

[48] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1998, pp. 131–134.

[49] J. O'Brien, J. Mather, and B. Holland, "A 200 MIPS single-chip 1 K FFT processor," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, Feb. 1989, pp. 166–167.

[50] B. M. Baas, "A low-power, high-performance, 1024-point FFT processor," *IEEE J. Solid-State Circuits*, vol. 34, no. 3, pp. 380–387, Mar. 1999.

[51] C.-H. Yang, T.-H. Yu, and D. Marković, "Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 757–768, Mar. 2012.

[52] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. URSI Int. Symp. Signals, Syst. Electron*, 1998, pp. 257–262.

[53] J. H. Choi, H. K. Jung, and T. Kim, "A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios," *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 738–746, 2006.
[54] A. Rodriguez-Perez, J. Ruiz-Amaya, M. Delgado-Restituto, and A. Rodriguez-Vazquez, "A low-power programmable neural spike detection channel with embedded calibration and data compression," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, pp. 87–100, 2012.
[55] B. Gosselin and M. Sawan, "An ultra low-power CMOS automatic action potential detector," *IEEE Trans. Neural. Syst. Rehabil. Eng.*, vol. 17, pp. 346–353, 2009.

Tong Wu (M'13–S'15) received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, and the M.S. degree from Zhejiang University, Hangzhou, China, in 2009 and 2012, respectively.

In October 2011, he joined the Translational System and Signal Processing Group in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he works as a Research Engineer. His research interests include the development of neural signal processing algorithms and their efficient ASIC/FPGA implementations. He is also interested in designing bio-inspired ASIC that can mimic the way humans interpret the visual input and allocate attentions.

Jian Xu (S'10–M'13) was born in Zhejiang Province, China. He received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2007 and 2012, respectively.

From March 2011 to July 2012, he was a Visiting Scholar in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he currently works as a Research Fellow. His research interests include high-precision low power data converters, low-power low-noise mixed-signal circuits, wireless power management, and data transmission for biomedical applications. His current work focuses on the design and implementation of an ultra high impedance and wide dynamic range biosensor interface.

Dr. Xu's Delta-Sigma ADCs got the best thermal figure-of-merit in the global state-of-the-art ranked by ADMS Design AB, Sweden, in 2012.

Yong Lian (M'90–SM'99–F'09) received the B.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from the National University of Singapore (NUS), Singapore, in 1984 and 1994, respectively.

He spent nine years in industry and joined NUS in 1996. He was appointed as the first Provost's Chair Professor in the Department of Electrical and Computer Engineering of NUS in 2011. He is also the Founder of ClearBridge VitalSigns Pte. Ltd. His research interests include biomedical circuits and systems and signal processing.

Dr. Lian has received many awards including the 1996 IEEE CAS Society's Guillemin–Cauer Award, the 2008 Multimedia Communications Best Paper Award, the 2011 IES Prestigious Engineering Achievement Award, and the 2012 Faculty Research Award. He was a coauthor of the Best Student Paper Award in ICME 2007, winner of the 47th DAC/ISSCC Student Design Contest in 2010, and Best Design Award in A-SSCC 2013 Student Design Contest. He serves as the Vice President for Publications of the IEEE Circuits and Systems Society, Steering Committee Member of IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and IEEE TRANSACTIONS ON MULTIMEDIA, and the Past Chair of the DSP Technical Committee of the IEEE Circuits and Systems Society. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II from 2010 to 2013. He also served as the Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, *Journal of Circuits*, and *Systems Signal Processing* in the past 10 years, and was the Guest Editor for eight special issues in IEEE journals. He is the Founder of the International Conference on Green Circuits and Systems, the Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, and the IEEE Biomedical Circuits and Systems Conference. He is a Fellow of the Academy of Engineering Singapore.

Azam Khalili (M'13) received the Ph.D. degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2011.

In 2011, she joined the Department of Electrical Engineering, Malayer University, Hamedan, Iran, as an Assistant Professor. Her current research interests are theory and methods for adaptive filtering, distributed adaptive estimation, and signal processing for communications.

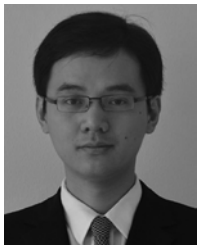
Amir Rastegarnia (M'13) received the Ph.D. degree in electrical engineering at the University of Tabriz, Tabriz, Iran, in 2011.

In 2011, he joined the Department of Electrical Engineering, Malayer University, Hamedan, Iran, as an Assistant Professor. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation, and signal processing for communications.

Cuntai Guan (SM'03) received the Ph.D. degree in electrical and electronic engineering from Southeast University, Nanjing, Jiangsu Province, China, in 1993.

Currently, he is a Principal Scientist and Department Head at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include neural and biomedical signal processing; neural and cognitive process and its clinical application; and brain-computer interface algorithms, systems and their applications. He has authored more than 160 refereed journal and conference papers and holds 13 granted patents and applications. He has delivered over 30 keynote and invited talks.

Dr. Guan was the recipient of several awards including the g.tec Annual BCI Research Award 2010 and IES Prestigious Engineering Achievement Award 2009. He serves on the editorial board of IEEE ACCESS, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, and *Frontiers in Neuroprosthetics*.



Zhi Yang (S'04–M'10) received the M.S. and Ph.D. degrees from the University of California, Santa Cruz, Santa Cruz, CA, USA, in 2007 and 2010, respectively.

Currently, he is an Assistant Professor at the National University of Singapore (NUS), Singapore. He joined the Electrical and Computer Engineering Department in 2010. Since then, he has built up the translational system and signal processing group http://www.ece.nus.edu.sg/stfpage/eleyangz), working on implantable and wearable bioelectronics, algorithms, and systems. He is particularly interested in neural/nerve recorder, stimulator, and signal processors. He is also an Assistant Professor in the Ophthalmology Department and a Principal Investigator at the Interactive and Digital Media Institute at NUS. He has coauthored more than 60 publications on bioelectronics and signal processing, and edited one book titled *Neural Computation, Neural Device, and Neural Prosthesis* (New York, NY, USA: Springer).

Dr. Yang was a recipient of the Best Paper Honorable Mention at ACCV 2009 and the Singapore Young Investigator Award 2012. He was also an invited contributor for a keynote paper at ESSCC 2010, and a winner of the regional MIT TR35 2014.