An Efficient Spike Detection VLSI Architecture Based on Normalized Correlator

Wen-Jyi Hwang
Department of Computer Science and
Information Engineering
National Taiwan Normal University
Taipei, 117, Taiwan
Email: whwang@csie.ntnu.edu.tw

Chun-Fu Lin
Instrument Technology Research Center
National Applied Research Laboratories, Taiwan
Email: vincent@itrc.narl.org.tw

Szu-Huai Wang
Department of Computer Science and
Information Engineering
National Taiwan Normal University
Taipei, 117, Taiwan
Email: a0919779123@gmail.com

Abstract—**This paper aims to present an effective circuit for noisy spike detection. The circuit detects spikes by the normalized correlators. The operations of the correlators involve filtering, block energy computation, normalized correlation, and thresholding. All the computations are carried out in a pipelined fashion. The circuit has been implemented by the field programmable gate arrays (FPGAs). The circuit is used as a hardware accelerator in a network-on-chip (NOC) platform for performance evaluation. Experimental results reveal that the proposed circuit provide realtime computation for the noisy spike detection with high true postive and low false alarm rates.**

Keywords*–*Spike Sorting; Spike Detection; FPGA; Network on Chip

I. INTRODUCTION

Spike sorting [1] is often desired for the design of brain machine interface (BMI) [2]. It receives spike trains from extracellular recording systems. Each spike train is a mixture of the trains from neurons near the recording electrodes. Spike sorting is able to segregate the spike trains of individual neurons from this mixture. It usually involves detection, feature extraction, and classification operations. Spike detection is the first step of the spike sorting. The goal of spike detection is to separate spikes from background noise. Extracellularly recorded signals are inevitably corrupted by noise from a number of sources such as the recording hardware and electromagnetic interference. In the presence of large noise, successful spike detection is essential for subsequent accurate feature extraction and classification.

One way to perform the spike detection is based on the energy of spike trains. An example of energy-based spike detection is the nonlinear energy operator (NEO) [3], which computes the energy difference between the signal's current power and the power in adjacent time intervals. The energy of coefficients in wavelet domain may also be useful for spike detection [4]. The energy-based methods are simple and efficient. However, when noise becomes large, proper selection of threshold values for these algorithms may be difficult. Therefore, their performance may deteriorate rapidly as noise energy increases. An alternative to the energy-based methods is to utilize the templates of spikes for detection. A typical technique using templates is based on matched filters [5]. A drawback of the matched filters is the high computational complexities. Realtime spike detection may then be difficult when matched filters are implemented by software. In addition, similar to the energy-based methods, it may be difficult to find an effective threshold levels for matched filters when noise becomes large.

A number of hardware implementations for spike sorting have been proposed for reatime spike sorting. Some hardware implementations [6] are based on NEO because of is simplicity and low area costs so that the circuits may be implantable at the front end for online detection. Nevertheless, the circuits may not be suited for detection with high noise levels. In addition, hardware designs are also beneficial for offline spike sorting [8] because of the requirement for processing large amount of data. With the relaxation of implantation requirement for offline processing, development and implementation of more efficient spike detection algorithms in hardware may be desired.

The objective of this paper is to present a novel VLSI architecture for realtime spike detection for noisy spike trains. The architecture is based on normalized correlator for enhancing detection performance. Segments of spike trains are normalized prior to the correlation computation. The normalization allows the output of the correlators lie inside a range, which is independent of the input spike trains and noise levels. This is beneficial for selecting an effective threshold levels for spike detection as signal-to-noise (SNR) ratios become low.

The proposed architecture can be separated into four units: the filter unit, the block energy computation unit, the correlator unit, and the thresholding unit. All the units are operated in a pipelined fashion to enhance the throughput of the circuit. The filter unit consists of a bandpass Butterworth filter capable of removing DC and high frequency components of spike trains. The filter is helpful for noise removal prior to correlation computation and detection. The block energy computation unit is used for calculating block energy of segments of spike trains. The normalized correlation is then carried out in the correlator unit. The thresholding unit then detects spikes based on the results produced by correlator unit.

The proposed architecture can be simplified for the design of implantable circuit. By retaining only the block energy computation unit, and thresholding unit, the proposed architecture becomes an incoherent energy detector, which performs generalized likelihood ratio test (GLRT) [7] for spike detection. The incoherent energy detector has the advantages of low area costs and low power consumption, while attaining higher throughput for spike detection.

The proposed architecture has been implemented by the field programmable gate arrays (FPGAs). The circuit is employed as a hardware accelerator in a network-on-chip (NOC) platform for performance evaluation. Experimental results show that the proposed architecture is able to attain high speed detection with high true positive rate and low false alarm rate even when SNR becomes -3 dB. Its simplified version, which performs noncoherent energy detection, has the additional advantages of lower area costs at the expense of slightly inferior detection performance. They are effective alternatives for spike sorting applications requiring real-time computation with superior spike detection performance.

The remaining parts of this paper are organized as follows. Section 2 gives a brief review of the normalized correlation algorithm. Section 3 describes the proposed spike detection architecture. Experimental results are included in Section 4. Finally, the concluding remarks are given in Section 5.

II. The Normalized Correlation Algorithm for Spike Detection

We start with the basic matched filter technique for spike sorting, which can be implemented by convolving the spike trains with the pre-stored templates. For sake of simplicity, we assume the matched filter contains only one template. Let $x[n]$ be the n-th sample of the input spike train. Let $\mathbf{x}_n = [x[n], x[n-1], ..., x[n-N+1]]^T$ be the n-th segment of the spike train, where N is the length of the segment. The template for matched filtering contains also N elements, denoted by $\mathbf{t} = [t[1], ..., t[N-1]]^T$. The matched filter output at n, denoted by, $y[n]$, is computed from the convolution

$$y[n] = \sum_{k=0}^{N-1} x[n-k]t[k] = \mathbf{x}_n^T\mathbf{t}. \quad (1)$$

Note that the convolution is equivalent to the inner product of segment $\mathbf{x}_n$ and template $\mathbf{t}$, which indicates the correlation between these two vectors. The segment $\mathbf{x}_n$ is detected as a spike when $y[n]$ is larger than a pre-specified threshold η.

A drawback of matched filter technique is that the threshold η alone cannot be used to determine the squared distance for template matching. To see this fact, we first observe that the squared distance between $\mathbf{x}_n$ and $\mathbf{t}$, denoted by $d(\mathbf{x}_n, \mathbf{t})$, is given by

$$d(\mathbf{x}_n, \mathbf{t}) = ||\mathbf{x}_n||^2 + ||\mathbf{t}||^2 - 2\mathbf{x}_n^T\mathbf{t}. \quad (2)$$

Therefore, when $\mathbf{x}_n^T\mathbf{t} > \eta$,

$$d(\mathbf{x}_n, \mathbf{t}) \leq ||\mathbf{x}_n||^2 + ||\mathbf{t}||^2 - 2\eta. \quad (3)$$

Therefore, when $\mathbf{x}_n$ is detected as a spike (i.e., $\mathbf{x}_n^T\mathbf{t} > \eta$), we see from (3) that the upper bound of $d(\mathbf{x}_n, \mathbf{t})$ is determined from $||\mathbf{x}_n||^2$, $||\mathbf{t}||^2$ and η, where $||\mathbf{x}_n||^2$ is dependent on the input spike trains. When $||\mathbf{x}_n||^2$ is large, it is possible that $d(\mathbf{x}_n, \mathbf{t})$ is still large even $\mathbf{x}_n^T\mathbf{t} > \eta$. In this case, a false alarm may occur.

One way to overcome this problem is to normalize $\mathbf{x}_n$ and $\mathbf{t}$ before computing the correlation. Define $\bar{\mathbf{x}}_n$ and $\bar{\mathbf{t}}$ as the normalized version of $\mathbf{x}_n$ and $\mathbf{t}$, respectively. That is,

$$\bar{\mathbf{x}}_n = \frac{\mathbf{x}_n}{||\mathbf{x}_n||}, \quad \bar{\mathbf{t}} = \frac{\mathbf{t}}{||\mathbf{t}||}. \quad (4)$$

Therefore,

$$d(\bar{\mathbf{x}}_n, \bar{\mathbf{t}}) = 2 - 2\bar{\mathbf{x}}_n^T\bar{\mathbf{t}}. \quad (5)$$

Because $d(\bar{\mathbf{x}}_n, \bar{\mathbf{t}}) > 0$, it can be easily shown that

$$\bar{\mathbf{x}}_n^T\bar{\mathbf{t}} \leq 1. \quad (6)$$

Our normalized correlator is based on $\bar{\mathbf{x}}_n$ and $\bar{\mathbf{t}}$. When $\bar{\mathbf{x}}_n^T\bar{\mathbf{t}} > \eta$, then $\mathbf{x}_n$ is detected as a spike. From (6), it follows that

$$\eta \leq 1. \quad (7)$$

In addition, when $\bar{\mathbf{x}}_n^T\bar{\mathbf{t}} > \eta$, from (5) we see that

$$d(\bar{\mathbf{x}}_n, \bar{\mathbf{t}}) \leq 2(1-\eta), \quad (8)$$

which is dependent only on the threshold value η. Therefore, the threshold value for correlation computation uniquely determines the upper bound of squared distance for template matching after a spike is detected. In addition, a larger η implies a smaller squared distance $d(\bar{\mathbf{x}}_n, \bar{\mathbf{t}})$. The upper bound of η is 1, which is independent on the input spike trains.

The normalized correlator has more meaningful interpretation for the threshold value η because $\eta \leq 1$, and the upper bound of squared distance for template matching for a detected spike is $2(1-\eta)$. When $\eta = 1.0$ is selected for detection, only the segments having *full* correlation with the template $\mathbf{t}$ are considered as spikes, and their squared distance with $\mathbf{t}$ is 0. When $\eta = 0.5$, all the segments having *half* correlation (or above) with $\mathbf{t}$ are detected as spikes, and the upper bound of their squared distances is 1. When $\eta = 0$, even the segments having *no* correlation with $\mathbf{t}$ are detected as the spikes, and the upper bound of their squared distances increases to 2. In the presence of noise, it may be impractical to require the detected spikes as the segments having full correlation (i.e., $\eta = 1.0$). In our experiments, the requirement of 70 % correlation (i.e., $\eta = 0.7$) may be sufficient for the normalized correlator to attain high detection hit rate, low miss rate, and low false alarm rate even for high noise levels. Detalied discussions of the normalized correlator can be found in our earlier work in [9].

Although the normalized correlator simply the process for the selection of threshold values, it has higher computation complexities for spike detection as compared with the basic matched filter technique. This is because the block energy of each segment need to be computed prior to the correlation computation. Hardware implementation of the normalized correlator may be beneficial for enhancing the throughput of the normalized correlator for realtime spike sorting.

III. The Proposed Architecture

Figure 1 shows the block diagram of the proposed architecture, which contains the filter unit, and block energy computation unit, the correlator unit, and thresholding unit. The filter unit is the pre-processing unit for the spike detection. It removes both the DC offset and noises before the detection operation. The goal of the block energy computation unit is to compute the block energy $||\mathbf{x}_n||^2$. The correlator unit then calculates $\bar{\mathbf{x}}_n^T\bar{\mathbf{t}}$. The detection results are then produced by the thresholding unit.

A. Filter Unit and Block Energy Computation Unit

In the implementation, the bandpass butterworth filter is used for the preprocessing operations. The filter can be implemented by shift registers, multipliers and adders. For sake of simplicity, the details of the implementation is not included. The direct implementation of the block energy computation involving N multiplications is also straightforward. Although N multipliers can be employed for the multiplications, the area

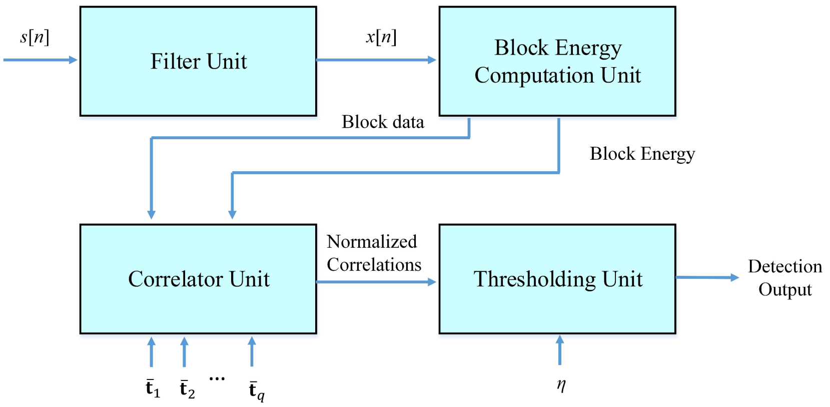

Figure 1. The Block Diagram of the Proposed Architecture for q templates

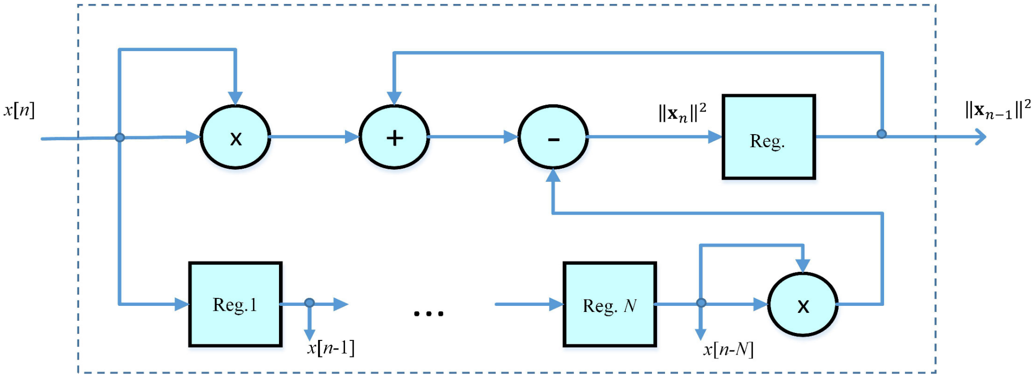

Figure 2. The Architecture of the Block Energy Computation Unit

costs can be high. An alternative is based on the observation that

$$||\mathbf{x}_n||^2 = ||\mathbf{x}_{n-1}||^2 + x^2[n] - x^2[n-N]. \qquad (9)$$

Therefore, when the block energy of the previous block (i.e., $||\mathbf{x}_{n-1}||^2$) is known, the computtaion of the block energy of the current block needs only two multiplication for the computation of the square of the samples $x[n]$ and $x[n-N]$, as shown in Figure 2. There are one N-stage shift register, two multiplier and two adders in the block energy computation unit. The shift register is used to hold the values of the past samples (i.e., $x[k]$, $k = n-1, ..., n-N$) in the first-in-first-out (FIFO) fashion. In addition to providing the value $x[n-N]$ for the computation of $x^2[n-N]$, the shift register is beneficial for the correlation computation in the correlator unit.

B. Correlator Unit

In addition to multiplications, the correlator for the computation of $\bar{y}[n] = \bar{\mathbf{x}}_n^T \bar{\mathbf{t}}$ requires the normalization operations. Although the normalized template $\bar{\mathbf{t}}$ can be obtained offline, the computation of the normalized $\bar{\mathbf{x}}_n$ should be carried out online. A direct implementation of the circuit for the computation of $\bar{\mathbf{x}}_n$ is to divide each sample of $\mathbf{x}_n$ by $||\mathbf{x}_n||$. This would require N dividers, because the dimension of the block $\mathbf{x}_n$ is N. An alternative is based on the post-normalization technique, in which the inner product $\mathbf{x}_n^T \bar{\mathbf{t}}$ is computed first. Because the inner product is a scalar, we can then use only one divider to compute $\bar{\mathbf{x}}_n^T \bar{\mathbf{t}}$ by dividing $\mathbf{x}_n^T \bar{\mathbf{t}}$ by $||\mathbf{x}_n||$.

Figure 3 shows the architecture of the correlator unit for the case of two templates. Correlators for any $q > 0$ templates can be carried out in a similar fashion. As shown in the figure, there are $2N$ multipliers, two accumulators, one squared root circuit, and one divider. The samples of $\mathbf{x}_n$ are obtained from the shift register in the block energy computation unit. The normalized templates $\bar{\mathbf{t}}_1$ and $\bar{\mathbf{t}}_2$ are pre-stored in the registers of the unit. To accelerate the correlation computation, there are N multipliers for the computation of each $\bar{\mathbf{x}}_n^T \bar{\mathbf{t}}_i, i = 1, 2$. In addition, the accumulation of the multication results are carried out in a pipelined fashion. The output of each accumulator is then divided by by $||\mathbf{x}_n||$. Observe from Figure 2 that the output of the block energy computation unit is $||\mathbf{x}_n||^2$. Therefore, the squared root (SQRT) circuit can be used to compute $||\mathbf{x}_n||$, as shown in Figure 3.

C. Thresholding Unit

Although the thresholding operations can be easily accomplished by a simple comparison circuit, the detection accurracy may be further improved by taking the detection results of the neighboring blocks into consideration. Because the neighboring blocks are overlapping, it is then likely that these blocks have similar normalized correlation values. A number of neighboring blocks may then have normalized correlation values larger than a pre-specified threshold. Consequently, it is possible that multiple hits may be declared for the occurrence of a single spike.

One way to solve this problem is not to declare a hit

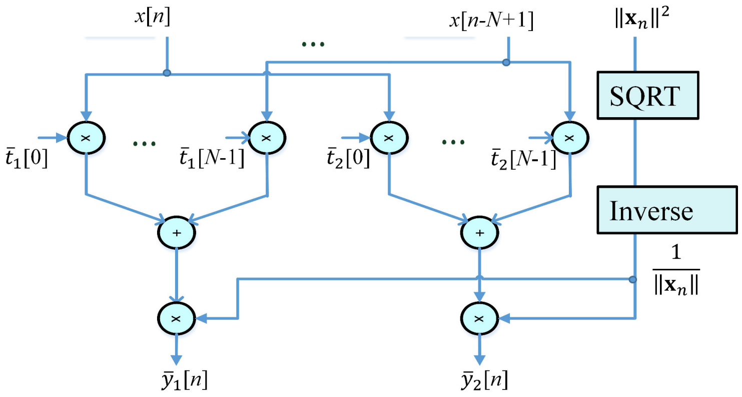

Figure 3. The Architecture of the Correlator Unit for $q = 2$ Templates

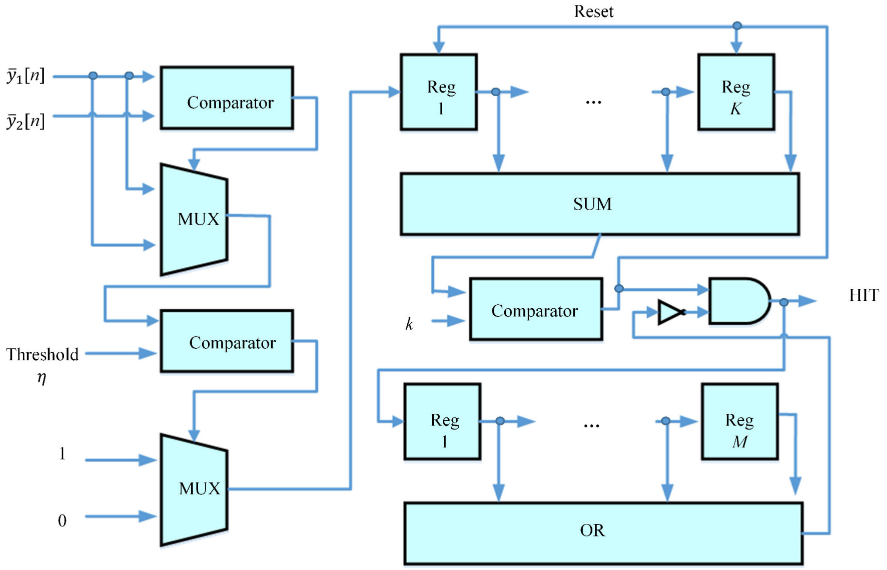

Figure 4. The Architecture of the Thresholding Unit for $q = 2$ templates

when the normalized correlartion value of a block is above the threshold. The normlalized correlation values of the previous blocks are also ckecked. Among its K preceding blocks, if k of them are also above the threshold, a hit is then declared. This may effectively reduce the false alarm rate for the detection. The architecture of the thresholding unit is revealed in Figure 4. It can be observed from the figure that a K-stage shift register is used to store the thresholding results of the K previous blocks. Each stage contains 1-bit information, where 0 and 1 indicate the corresponding block has correlation value below and above the threshold η, respectively. Consequently, when the sum of the output of all the K stages is equal or above k, then k of the K preceding blocks have correlation value above the threshold. A hit is then issued.

Note that we may be able to further reduce the false alarm rate at the expense of a slight increase in true positive rate by imposing the assumption that spikes are at least M samples apart. The enforcement of the assumption can be carried out be an additional M-stage shift register recording the location of the previous hit. Each stage also has values of 0 or 1. If the previous hit is less than M samples apart, one of the stage in the shift register contains value of 1, which disables the hit. A hit is allowed to be issued only when all the stages contain value of 0.

D. Noncoherent Energy Detector

The proposed circuit can be simplified by removing the correlator unit. In this case, the output $||\mathbf{x}_n||^2$ of the block energy computation unit is connected directly to the threshodling unit. The circuit will declare a hit when $||\mathbf{x}_n||^2$ is above the threshold. This is the noncoherent energy detector proposed by [7]. As compared with the proposed circuit, the noncoherent energy detector has the advantages of lower area costs and power consumption at the expense of slightly lower true positive rates and/or higher false alarm rates. The circuit is advantageous for the applications where both the speed and area costs are the important concerns.

IV. Experimental Results

This section presents some experimental results of the proposed architecture. The simulator developed in [10] is

TABLE I. THE TPR AND FAR VALUES OF VARIOUS SPIKE DETECTION ALGORITHMS FOR SPIKE TRAINS WITH VARIOUS SNR LEVELS.

SNR (dB)		Normalized Correlator	Noncoherent Energy Detector	NEO	SWT	Matched Filter
10	TPR	93.64 %	91.37 %	93.10 %	94.82%	89.65 %
	FAR	0.40 %	5.35 %	3.57 %	6.77 %	2.80 %
1	TPR	90.04 %	88.03 %	87.21 %	92.43 %	82.90 %
	FAR	0.92 %	6.36 %	22.49 %	79.36 %	3.02 %
-3	TPR	82.71 %	82.60 %	80.53 %	86.66 %	80.31 %
	FAR	1.06 %	9.52 %	57.87 %	82.43 %	8.92 %

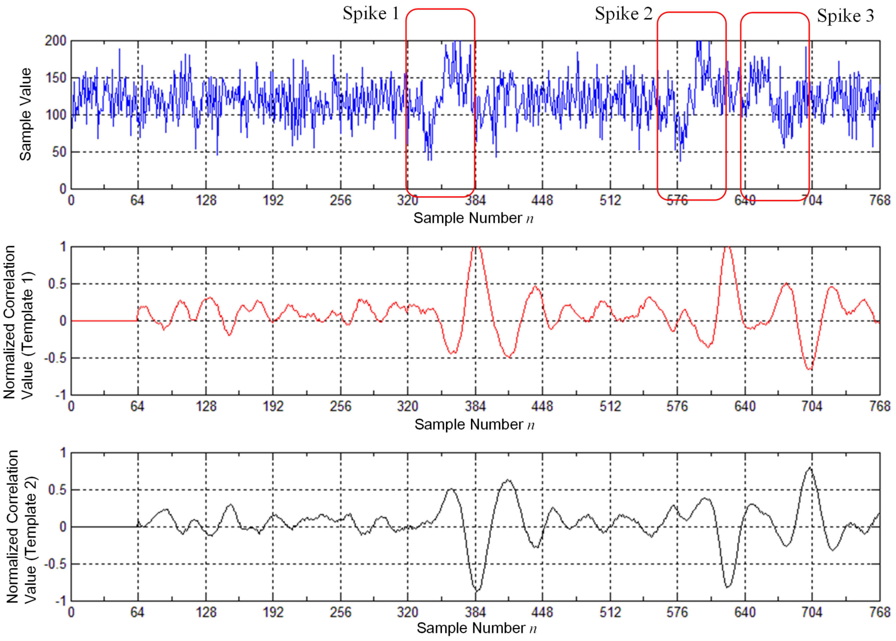

Figure 5. An example of the proposed normalized correlator for noisy spike detection with SNR=-3 dB for $q = 2$ templates.

adopted to generate extracellular recordings. The simulation gives access to ground truth about spiking activity in the recording. This facilitates the quantitative assessment of the proposed architecture, since the features of the spike trains are known a priori. All the spikes are recorded with a sampling rate of 24,000 samples/s. Each spike has 64 samples (i.e., $N = 64$), and the length of each spike is 2.67 ms.

We first consider the true positive rate (TPR) and false alarm rate (FAR) of the proposed architecture. The TPR is defined as the number of detected true spikes divided by the total number of true spikes. The FAR is defined as the number of silent segments, which are detected as spikes, divided by the total number of detected segments. Table I shows the TPR and FAR of the normalized correlator, the noncoherent energy detector, NEO, stationary wavelet transform (SWT), and matched filter for various SNR levels. The number of neurons is 2. The proposed normalized correlator architecture therefore uses 2 templates (i.e., $q = 2$).

It can be observed from Table I that the normalized correlator has higher TPR and lower FAR as compared with those of the other algorithms. This is because the correlation is beneficial for identifying real spikes and ignoring silent segments. This fact can be further observed in Figure 5, where the noisy spike train with SNR= -3 dB, and the normalized correlation values $\bar{y}_i[n], i = 1, 2$, are shown. It can be observed from Figure 5 that it is difficult to locate spikes due to large noise corruption. Nevertheless, the normalized correlation values shown in Figures 5 still provide useful information revealing the location of true spikes. It is also interesting to note that the noncoherent energy detector has TPR and FAR values comparable to those of matched filter. These results show that the energy is also effective for spike detection.

Next we evaluate the area complexities. Because adders, multipliers, dividers, comparators and registers are the basic building blocks of the architecture, the area complexities are separated into four categories: the number of adders, multipliers, dividers, comparators and registers. Table II shows the area complexities of the proposed architecture. It can be observed from Table I that the number of adders, multipliers, and dividers are fixed, and independent of the block dimension N and number of templates q in the filter unit, block energy computation unit and thresholding unit. Although the number of adders and the number of multipliers grows with the N and

TABLE II. THE AREA COMPLEXITIES OF THE PROPOSED ARCHITECTURE

	Filter	Block Energy	Correlator	Thresholding
	Unit	Computation Unit	Unit	Unit
Adders	$O(1)$	$O(1)$	$O(qN)$	$O(1)$
Multipliers	$O(1)$	$O(1)$	$O(qN)$	$O(1)$
Dividers	0	0	$O(1)$	0
Comparators	0	0	0	$O(1)$
Registers	$O(1)$	$O(N)$	$O(qN)$	$O(1)$

TABLE III. HARDWARE UTILIZATION OF THE FPGA IMPLEMENTATION OF THE PROPOSED NORMALIZED CORRELATOR ARCHITECTURE

	Filter	Block Energy	Correlator	Thresholding	Total
	Unit	Computation Unit	Unit	Unit	
ALUTs	750	649	4571	89	6059
Registers	236	866	2788	13	3903
Memory Bits	0	0	0	0	0
DSP Blocks	24	3	528	0	555

TABLE IV. COMPARISONS OF HARDWARE UTILIZATION OF THE NORMALIZED CORRELATOR AND NONCOHERENT ENERGY DETECTOR FPGA IMPLEMENTATIONS

	ALUTs	Registers	Memory Bits	DSP Blocks
Normalized Correlator	6059	3903	0	555
Noncoherent Energy Detector	1488	1115	0	3

q in the block energy computation unit, only a single divider is used in the unit because of the employment of the post-normalization technique. This is beneficial for lowering the area costs of the circuit.

We further consider the hardware utilization of the proposed normalized correlation architecture implemented by FPGA. In the experiments, we set the dimension of the spikes to be $N = 64$. There are $q = 2$ templates. The target FPGA in the experiments is Altera Stratix III EP3SE80F780C2, which contains 64,000 adaptive lookup tables (ALUTs), 64,000 registers, 6,331,392 memory bits, and 672 DSP blocks. The FPGA design platform is Altera Quartus II 13.0. Table III shows the number of ALUTs, the number of registers, the number of memory bits, and the number of DSP blocks consumed by each unit of the proposed circuit. It can be observed from Table III that many of the ALUTs, registers and DSP blocks provided by the target FPGA are consumed by the correlator unit because the inner product operations are required in the unit.

When only the noncoherent energy detection is necessary, the correlator can be removed. Therefore, the area costs can be effectively lowered. Table IV shows the hardware utilization of the proposed normalized correlation architecture and the proposed noncoherent energy detection architectures. It can be observed from Table IV that the noncoherent energy detection architecture has lower hardware utilization. In particular, the utilization of DSP blocks is 3, which is only 0.54 % (i.e., 3/555) of that utilized by the normalized correlator architecture.

The proposed architecture is used as a hardware accelerator in a NOC platform for the speed evaluation. The NOC is designed by Altera Qsys 13.1. The NOC consists of a NIOS II softcore processor, an embedded RAM, and the proposed circuit. The noisy spike sequences are stored in the embedded RAM. The NIOS II processor activates the delivery of the spike sequence from the RAM to the proposed circuit for spike detection. Upon the completion of spike detection operations, it also collects the results of the spike detection for subsequent spike sorting operations. When operating at the clock rate 50 MHz, the proposed architecture is able to complete the detection operation in 52 ms for a spike sequence with length of 100 seconds. By contrast, the computation time of its software counterpart running in the 1.7 GHz Intel I-7 processor for the same spike sequence is 1.58 second. The speedup of the hardware acceleration therefore in 30.38 (i.e., 1.58 second vs. 52 ms). All these facts demonstrate the effectiveness of the proposed architecture.

V. CONCLUSION

The proposed normalized correlator architecture has been implemented by FPGA for performance evaluation. Experimental results show that the architecture is effective for spike detection. It has the advantages of high TPR, low FAR, and fast computation. For spike trains with SNR = -3 dB, the proposed normalized correlator is able to achieve TPR 82.71 % and FAR 1.06 %. In addition, the speedup of the proposed architecture in the NOC operating at 50 MHz over its counterpart is 30.38. The proposed architecture can also be simplified to a noncoherent energy detector when lower hardware costs are desired at the expense of a slight degradation in detection performance.

REFERENCES

[1] S. Gibson, J. W. Judy, and D. Markovic, “Spike sorting: the first step in decoding the brain,” IEEE Signal Processing Magazine, 2012, pp. 124-143.

[2] M. A. Lebedev and M. A. L. Nicolelis, “Brainmachine interfaces: past, present and future,” Trends in Neurosciences, Vol. 29, 2006, pp. 536-546.

[3] 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, 1998, pp. 180-187.

[4] 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, 2003, pp. 999-l011.

[5] N. Mtetwa and L. S. Smith, “Smoothing and thresholding in neuronal spike detection,” Neurocomputing, Vol. 69, 2006, pp. 1366-1370.

[6] J. Drolet, H. Semmaoui, and M. Sawan, “Low-power energy-Based CMOS digital detector for neural recording arrays,” IEEE Biomedical circuits and systems conference, 2011, pp.13-16.

[7] K. Oweiss and M. Aghagolzadeh, Detection and classification of extracellular action potential recordings, Chapter 2 of Statistical Signal Processing for Neuroscience, 2010, pp. 15-74.

[8] S. Gibson, J. W. Judy, and D. Markovic, “An FPGA-based platform for accelerated offline spike sorting,” Journal of Neuroscience Methods, Vol. 215, 2013, pp. 1-11.

[9] W. J. Hwang, S. H. Wang, and Y. T. Hsu, “Spike Detection Based on Normalized Correlation with Automatic Template Generation,” Sensors, 2014, pp. 11049-11069.

[10] L. S. Smith and N. Mtetwa, “A tool for synthesizing spike trains with realistic interference,” Journal of Neuroscience Methods, Vol. 159, 2007, pp. 170-180.