A sub- $1\mu\text{W}$ Neural Spike-Peak Detection and Spike-Count Rate Encoding Circuit

Sivylla E. Paraskevopoulou^{*†} and Timothy G. Constandinou^{*†}

^{*}Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK

[†]Centre for Bio-Inspired Technology, Institute of Biomedical Engineering, Imperial College London, SW7 2AZ, UK

Email: {s.paraskevopoulou09,t.constandinou}@imperial.ac.uk

Abstract—In this paper we present a circuit for determining neural spike features such as peak occurrence, peak amplitude and spike count rate in continuous-time. The system achieves these functions concurrently and in real-time achieving an accuracy higher than a typical digital solution (constrained by the sampling time and/or resolution). For an average spike rate of 50spikes/s the system consumes 815nW designed in a commercially-available $0.18\mu\text{m}$ CMOS technology. The complete circuit core (excluding bondpads) occupies a total area of approximately 0.022mm^2

I. INTRODUCTION

Neurotechnology, the application of technology to neuroscience, is a topic that is currently enjoying much interest in the research community. With ever progressing advances in semiconductor technology, never before have there been so many opportunities to develop devices that effectively interface with neurobiology. The ultimate aim of such research is to improve the quality of life of individuals with dysfunction or damage to the nervous system. The potential benefits have already been demonstrated in over 200,000 people with cochlear implants for impaired hearing and 80,000 with DBS devices for Parkinson's, Dystonia and Essential Tremor.

To develop next generation devices that monitor, decipher and act on neural activity, it is crucial to examine common neural coding strategies studied by neuroscientists. Of these, three popular schemes include: rate coding (instantaneous spike rate), spike-count rate (average spikes taken over trial) and population coding (joint activity of several neurons) [1]. A common requirement in such schemes is that the information is coded in the firing pattern (in time and space) of either single (or clusters of) neurons. Typically, to access such signals, implantable microelectrode arrays are used but each channel receives the spike activity of several (up to 5) neurons. It is therefore essential to separate the individual neuron activity through a signal processing chain which consists of spike detection, alignment, feature extraction, and spike sorting [2]. With the number of channels in such systems continuing to increase [3], it is becoming more beneficial to integrate as much of this processing as possible in the front-end.

A crucial aspect (often neglected), in this processing chain is the alignment. During the feature extraction, the spike waveform is projected to a space with reduced dimensions compared to the sampled signal. This projection can be severely compromised without a stable reference point. An

Fig. 1: System architecture of the spike peak detection and count-rate encoding circuit.

obvious spike feature suitable as the alignment marker is its maximum. However, the maximum can be dislocated because of insufficient sampling, hence more elaborate alignment algorithms have been proposed [4]. By implementing the peak detection in the analogue domain, this issue can be eliminated.

In addition to accurately detecting and aligning the spike features, it is also useful for neural interfaces to encode the timing information [5]. In previous work [6], we explored the different possible configurations by varying the level (high vs. low) and quality (lossless vs. lossy) of data to be transmitted. Amongst these topologies, the spike rate transmission configuration is an extremely bandwidth efficient option for high channel count devices (although lossy). However, such an implementation typically requires a ripple counter (of at least 6-bits), to encode either the spike-count rate or inter-spike interval. This can itself be a relatively power hungry method of encoding timing information.

In this paper, we present a novel system that locates the spike peaks and also monitors the spike rate using a continuous-time implementation. This paper is organised as follows: Section II outlines the system architecture, Section III describes the circuit implementation, Section IV presents the simulated results and Section V discusses the system performance.

II. SYSTEM ARCHITECTURE

The system presented herein takes an amplified neural signal as its input, locates the spike peaks and computes the spike occurrence rate. The block diagram of the system is shown in Fig. 1. This operates as follows:

- A window is sampled around the peak of the neural signal by using a single threshold spike detector with threshold derived from input (i.e. $k \times \sigma$) [6], [7].

- The peak detector follows the increasing monotony of the signal. A change of monotonicity signals the peak, triggering an event.
- The phase generator then produces two non-overlapping pulses ϕ_1 & ϕ_2 , with ϕ_1 being aligned to the spike peak.
- The pulse ϕ_1 disconnects the input to a sample and hold switch, capturing the peak magnitude.
- The spike rate encoder counts the spike events, thus converting the timing information into a voltage.
- A 2Hz clock samples the current count of the spike rate encoder before resetting it.

The front-end neural amplifier will not be discussed in this design as this can be implemented using well-established topologies [8].

III. CIRCUIT IMPLEMENTATION

The circuit has been implemented in a commercially-available $0.18\mu\text{m}$ 1P4M CMOS technology provided by IBM.

A. Spike Detector

The method selected for spike detection is based on a single threshold comparison. The input neural signal (pre-amplified) is compared to a preset threshold value (derived from the signal itself- typically a multiple of its standard deviation [6], [7]). The comparator circuit uses a double differential stage (for offset compensation) and regenerative load (to boost its response time) adopted from [9]. The output of the comparator is used to create a window around the peak (i.e. when the neural signal is larger than the threshold) by controlling a feedthrough switch. This is crucial for robustly locating the spike peaks, as this essentially rejects noise that may otherwise cause false positives.

B. Peak Detector

The peak detector circuit is shown in Fig. 2, adapted from the designs reported in [10]–[12]. The basic idea behind this circuit is to follow the input for increasing signals but not for decreasing signals. This is achieved by comparing the level of $V_{in}(t)$ to its previous value (stored in C1), thus monitoring the monotony of the signal (if increasing). The output of the OTA is $V_s(t) = A \cdot V_{in}(t) \cdot V_{peak}(t)$, where A is the OTA gain. Device Q9 has two roles: to charge the capacitor C1 (for increasing signals) but disconnect the input (for decreasing signals). For increasing signals, $V_{in} > V_{peak}$ and therefore device Q9 charges C1 until $V_{in} = V_{peak}$. This charging inherently causes some hysteresis, depending on the value of the capacitor (the smaller the capacitor, the closer V_{peak} tracks V_{in}). At the peak, the monotony of the signal changes from positive to negative and $V_{in} < V_{peak}$, causing Q9 to switch off. C1 is then slowly discharged through the *leak* network (R1 and Q10) which can be tuned using V_{leak} . The target is to set this such that V_{peak} discharges to below the threshold voltage of the comparator in 4ms , which is considered to be the minimum inter-spike interval. The polysilicon resistor (R1) has been inserted in series with Q10 to improve the noise performance.

Fig. 2: Peak detector circuit (with bias current, $I_{bias} = 40\text{nA}$)

C. Pulse Generator

The pulse generator circuit is shown in Fig. 3. This converts the output from the peak detector to a discrete event, i.e. an asynchronous pulse. The peak detector (maximum) output range is between GND (i.e. mid-rail), to VDD but signal amplitudes (and offsets) can vary from spike to spike. The pulse generator therefore has to amplify this range such that the signal spans the entire supply without distorting the peak timing information. This is achieved using an OTA (X1) with the reference (non-inverting input) tied to VSS and capacitive feedback (to the inverting input) such that the gain is $C_1/C_2 \approx 6$. This feeds a current-starved inverter chain (X2) to shape a *steep* digital pulse to minimize the dynamic power consumption of proceeding digital logic (X3). However, the pulse width varies from pulse to pulse, depending on the amplitude, width and profile of the input signal.

D. Phase Generator

The phase generator produces two non-overlapping pulses of preset duration 1ms given a digital input (from the pulse

Fig. 3: Pulse generator circuit. The OTA is based on a balanced transconductor topology with a 5nA bias. The current-starved inverter chain is biased at 1nA with a 1:4:12:24 ratio.

Fig. 4: 2Hz clock generator circuit. Current starved inverter chains are as in Fig. 3 but use only a current-starved source (in X1) and sink (in X4).

generator). The input is edge-triggered to ensure the first phase is aligned (see Section IV for alignment accuracy) with the spike peak, and generates the two pulse phases in sequence (ϕ_1 followed by ϕ_2). These two phases are required to increment the SC integrator. The pulses are generated in a similar fashion to the 2Hz clock generator (described below).

E. 2Hz Clock Generator

The 2Hz clock generator circuit is shown in Fig. 4. This is based on charging and discharging a capacitor using a small bias current ($\approx 25pA$) generating a dual slope. The capacitor ($C1$) is charged by the current source mirror ($Q1-2$) and discharged through the current sink mirror ($Q5-6$). Devices $Q3-4$ are used as switches to control the charge and discharge phases. By selecting these to be complementary, it is ensured that the source and sink cannot be active simultaneously. The capacitor voltage (V_{cap}) is monitored by the two current-starved inverter chains ($X1-2$ and $X4-5$). Inverter chain $X1-2$ is designed to have a limited current source, whereas chain $X4-5$ is designed to have a limited current sink. By utilising these two chains in parallel, both low ($\approx V_{SS} + 200mV$) and high ($\approx V_{DD} - 200mV$) threshold crossings can be detected. These, in turn, are used to control an edge-triggered RS latch ($X3$) which toggles between the two states (defining the charge and discharge phases). Finally, the output is fed through a D-flip-flop to half the frequency and ensure a 50% duty cycle.

F. Spike Rate Encoder

The spike rate encoder circuit is shown in Fig. 5. This is based on a switched-capacitor inverting (lossless) integrator with clocks ϕ_1 and ϕ_2 being non-overlapping (phase generator described previously). The advantage of this topology is its insensitivity to stray (or parasitic) capacitances. Its transfer function is given by:

$$V_{out}(n) = V_{out}(n-1) - a_1 \cdot V_{ref}(n-1) \quad (1)$$

Where a_1 is the ratio between input and feedback capacitors (i.e. C_1/C_2). If V_{ref} is set to a present DC bias level, the output will drop by a fixed amount every time a spike occurs. This decrement will be equal to $a_1 \cdot (V_{ref} - GND)$ and is

Fig. 5: Switched-capacitor integrator (with reset) for determining the spike count rate. All MOS switches are $W/L=0.5/0.5$.

designed such that for each integration period ($500ms$), a maximum of 100 steps can be represented (corresponding to a maximum spike rate of $200spikes/s$). To achieve this resolution, $V_{ref} = V_{DD}$ and $C_2/C_1 = 100$. The output is sampled and integrator is reset (by shorting C_2 with Q_5) at $500ms$ intervals.

IV. SIMULATED RESULTS

The circuit has been simulated using the Cadence Spectre (5.1.41sr1) simulator with foundry supplied BSIM3v3 models. The test signals used are neural data sets sampled at $24KHz$ ¹.

A. Peak Detection

The peak detector operation is illustrated in Fig. 6. This illustrates the noise immunity and tuneability of the peak detector circuit. By tuning the bias voltage on transistor Q_{10} , i.e. V_{leak} , the spike recovery time can be adjusted. Here it is essential to select an intermediate value, as if this is too long, the detector may miss spikes (at high rates) and if too short, the noise immunity will be compromised.

B. Spike Timing

The spike timing generation is illustrated in Fig. 7. This shows the operation of the circuit for a $100ms$ duration of a neural signal containing 6 spikes. The simulation shows the generation of the clock phases ϕ_1 and ϕ_2 using the peak detector circuit.

¹<http://www2.le.ac.uk/departments/engineering/research/bioengineering/neuroengineering-lab/spike-sorting>

Fig. 6: Peak detector output for different values of V_{leak} . The nominal value used is: $V_{leak} = 275mV$.

Fig. 7: Peak detection and timing generation showing: (a) input signal, (b) peak detector output, (c) ϕ_1 pulse, (d) ϕ_2 pulse.

Fig. 8: Temporal variation in the offset of detected spike peaks.

For the peak detection to be useful for spike alignment (as a pre-process to spike sorting), a crucial requirement is for it to be able to encode accurate timing with low variation over a wide variety of spike signals. Fig. 8 shows a histogram illustrating the distribution of the time offset between the actual spike peak (neural input signal) and the rising edge of the generated pulse (ϕ_1) for 100 spikes. Over this dataset the average offset is $0.215\mu s$ with a standard deviation of $47.5\mu s$. It can be observed that the histogram in fact exhibits two peaks. This can however be attributed to the quantisation error in the sampled dataset, since $f_{sample}=24KHz$. This equates to a $41.6\mu s$ temporally-referred quantisation offset, therefore accounting for the majority of this offset.

C. Spike Rate Encoding

A transient simulation of the complete system over a $1.5s$ duration is shown in Fig. 9. This illustrates the operation of the peak detector for an input signal composed of multiple neurons and of varying spike rate over several integration periods. Also, the operation of the spike rate encoder can be observed to integrate the spikes and hold the final value (representing the current spike count rate) after each $500ms$ integration period.

V. CONCLUSION

We have presented a circuit that extracts three specific features from a neural signal. These are: (1) peak (timing) detection, (2) peak amplitude extraction, and (3) spike count encoding. Such features are crucial to facilitate subsequent processing such as spike alignment for spike sorting or envelope tracking for automatic gain control. This is particularly

Fig. 9: Spike rate encoding showing: (a) input signal, (b) ϕ_1 pulse, (c) SC integrator output, (d) hold, (e) spike rate output.

important if the quantisation error in a sampled signal would degrade the quality of the subsequent processing. The circuit is able to represent a spike count rate of $0-200spikes/s$ updated in $500ms$ intervals and can provide asynchronous timing for spike peak occurrence. The static power consumption of the circuit is $720nW$ with a dynamic energy requirement of $1.9nJ/spike$. The circuit occupies a silicon area of $0.022mm^2$ in a $0.18\mu m$ CMOS technology.

ACKNOWLEDGMENT

This work was supported by the UK Engineering and Physical Sciences Research Council (Grant ref: EP/I000569/1).

REFERENCES

- [1] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*. Cambridge University Press, 2002.
- [2] M. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Comput. Neural Syst.*, vol. 9, no. 1998, pp. 53–78, 1998.
- [3] J. N. Y. Aziz, et al., "256-channel neural recording and delta compression microsystem with 3d electrodes," *IEEE JSSC*, vol. 44, no. 3, pp. 995–1005, 2009.
- [4] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Low Power Architectures for Spike Sorting," *Proc. IEEE EMBC*, pp. 162–165, 2005.
- [5] S. G. Mason and G. E. Birch, "A general framework for brain-computer interface design," *IEEE Neural Syst. & Rehab. Eng.*, vol. 11, no. 1, pp. 70–85, 2003.
- [6] A. Eftekhari, S. E. Paraskevopoulou, and T. G. Constandinou, "Towards a next generation neural interface: Optimizing power, bandwidth and data quality," *Proc. IEEE BioCAS*, pp. 122–125, 2010.
- [7] R. Harrison, "The design of integrated circuits to observe brain activity," *Proc. IEEE*, vol. 96, no. 7, pp. 1203–1216, 2008.
- [8] R. R. Harrison and C. Charles, "A low-power low-noise cmos amplifier for neural recording applications," *IEEE JSSC*, vol. 38, no. 6, pp. 958–965, 2003.
- [9] R. Baker, *CMOS: Circuit design, layout, and simulation*, pp. 909–920. Wiley-IEEE Press, 2010.
- [10] G. De Geronimo, et al., "Analog cmos peak detect and hold circuits," *Nuclear Instr. Meth. Phys. Res. A*, vol. 484, no. 1-3, pp. 533–543, 2002.
- [11] H. Abdalla and T. K. Horiuchi, "An analog vlsi low-power envelope periodicity detector," *IEEE TCAS-I*, vol. 52, no. 9, pp. 1709–1720, 2005.
- [12] J. Holleman, et al., "A micro-power neural spike detector and feature extractor in $13\mu m$ cmos," *Proc. IEEE CICC*, pp. 333–336, 2008.