35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Low-Power Hardware for Neural Spike Compression in BMIs

Ângelo C. Lapolli, Bertrand Coppa, and Rodolphe Héliot*

***Abstract*— Within brain-machine interface systems, cortically implanted microelectrode arrays and associated hardware have a low-power budget for data sampling, processing, and transmission. Recent studies have shown the feasibility of data transmission rate reduction using compressed sensing on detected neural spikes. They provide power savings while maintaining clustering and classification abilities. We propose and analyze here a low-power hardware implementation for spike detection and compression. The resulting integrated circuit, designed in CMOS 65nm technology, consumes 2.83 µW and provides 97% of data rate reduction.**

***Index Terms*—Brain-Machine Interface, Compressed Sensing, Integrated circuits, Neural signals processing.**

I. Introduction

BRAIN-MACHINE Interfaces (BMIs) detect and decode neural data aiming the control of an external actuator. Their major applications include the restoration of sensorimotor functions for patients with neurological disorders in which the controlled devices are such as robotic prosthesis [1]. Decoding neural data requires spike sorting, consisting in detecting neuronal spikes and then classifying them by source neuron. Spiking neural data is recorded by microelectrode arrays; state-of-the-art systems aim at implanting the data sampling and transmitting it wirelessly to the external world, to avoid cables passing through the skull. The associated hardware has strong power constraints since it must fit into the microelectrode base area; moreover, its power density is limited to avoid damaging the biological tissue [2]. As stated in [4], the energetic cost for wirelessly transmitting the collected data is much greater than for any other node function (AD conversion, spike detection, etc.). Thus, the reduction of its transmission rate by data compression has been recently addressed seeking the respect of these restrictions.

Under this perspective, the simplicity of the *Compressed Sensing* (CS) compression algorithm has motivated the study of its viability in BMIs. In [3], the use of CS is proposed for generic biosignals sparsely represented in a determined domain; it is suggested the compression of the collected signal in its totality. Specifically for neural data, it has been shown in [4] that CS can be used individually on each detected spike, while maintaining clustering and classification abilities. In that way, the data compression increases allowing a higher reduction on transmission energy consumption. Therefore, we propose and analyze a low-power hardware implementation aiming at spike detection and CS compression. Previously proposed hardware solutions comprise the realization of: spike detection with no further compression [5], or detection followed by feature extraction [6].

This paper is organized as it follows. Section II explains CS compression algorithm and the spike sorting method. Section III specifies the characteristics and constraints of the desired system. Section IV presents the dataset, the design, and the validation of the proposed architectures. Section V describes possible solutions and discusses their implementation results, and is followed by a discussion.

II. Theoretical Backgorund

A. Compressed Sensing

C*ompressed sensing* has been conceived to allow sparse signals to be properly rebuilt from representations with low sampling rates [7]. It consists in representing the original sparse signal $x \in \Re^N$ by a subspace projection as in

$$y = \phi \cdot x \quad \text{with} \quad \phi \in \Re^{m \times N} \text{ and } m < N, \tag{1}$$

where ϕ is called encoding matrix and $y \in \Re^m$ is the compressed signal, thus giving N/m as compression rate. Although (1) does not present high computation complexity, an efficient signal reconstruction involves a complicated minimization problem. In opposition to this drawback, it is shown by [4] that it is possible to retrieve the information required by BMI applications directly from individually compressed spikes. By using a random binary encoding matrix where the probability of each state (± 1) is 0.5, spike sorting is shown feasible with a compression rate of 5.33 considering *N=32* (extracted spike length) and *m=6*. Seeing that this approach is very well adapted to hardware implementation, we study hereinafter its possible architectural implementations.

B. Spike-Sorting

1) Detection and Extraction

The spike detection methodology considered in [4] is the same as for [9] and it is called *Absolute Value* (AV). This technique detects a spike whether the current sample is superior to a threshold defined as:

*A. C. Lapolli, B. Coppa, and R. Héliot are with CEA-LETI, Minatec Campus, Grenoble, France (e-mail: rodolphe.heliot@gmail.com)

$$Thr = 4 \cdot \sigma_N \quad \text{with} \quad \sigma_N = median\left\{\frac{|x|}{0.6745}\right\}, \qquad (2)$$

where x is the band-pass filtered input signal and σ_N is an estimate of the standard deviation of the background noise. According to [8], the multiplication factor 4 can be slightly changed in order to adapt to different noise levels. For each detected spike, 64 samples are saved and then aligned with respect to their maximum.

Other detection approaches have been studied under the perspective of hardware implementation in [9]. It is concluded that the *Nonlinear Energy Operator* (NEO) method [10] is the most appropriate, given its adaptability to a larger range of signal to noise ratios. It consists in calculating the instantaneous energy for each sample of the input signal $\psi[x(n)]$ and comparing it to a threshold as in:

$$\psi[x(n)] = x^2(n) - x(n+1) \cdot x(n-1) \qquad (3)$$

$$Thr = C \frac{1}{N} \sum_{n=1}^{N} \psi[x(n)], \qquad (4)$$

where N is the number of samples in the input signal x and C=8 [9]. Following detection, alignment is necessary for the correct spike extraction.

2) Clustering and Classification

To guarantee the clustering and classification abilities of spikes after hardware CS compression, we use the same classification method as in [4]. Initially, the extracted and compressed spikes $y \in \Re^m$ are represented by means of a *principal component analysis*, and then the resulting points are used to construct a *minimum spanning tree* seeking the initialization of the clusters centers. Finally, the *k-means algorithm* makes the correspondence of each spike to a cluster.

III. System Description

The envisioned system's input signal is filtered and digitally converted neural data collected by one channel of a microelectrode array (see section IV-A), the desired output is detected and compressed spikes. Fig. 1 illustrates the general modular structure chosen for the system. The input signal is set here at a rate of 24 kHz, and is at first stored in a buffer which is accessible by two modules: one responsible for the spike detection and another in charge of the compression. Whenever a spike is detected, the former notifies the latter to start the compression. The output is initially set to zero and it is updated each time the compression of a spike is completed.

We consider each spike to be represented by 32 16-bits samples. The encoding matrix thus has 6 rows and 32 columns and its elements (± 1) are randomly distributed. The runtime of a single compression is limited to the interval until the arrival of the next potential spike. In order to avoid damages to the biological tissue, the power density must be inferior to 800 µW/mm² [3]. The circuit must have an area in the order of 0.01mm² to fit alongside the filter and the analog-to-digital converter into the microelectrode base area.

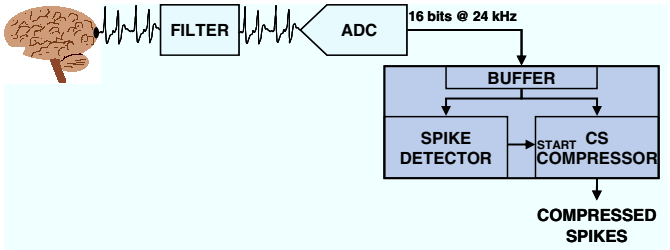

Fig. 1. Global System Modular Schematic

IV. Material and Methods

For this paper, we use simulated neural data made publicly available by the authors of [8]. They are available in the form of a 10-second-long simulated signal containing 507 spikes. Over this data set, we apply a band-pass (300 Hz – 3 kHz) second-order Cauer filter followed by an analog-to-digital 16 bits conversion with uniform quantization. We have previously computed the spike detection and compression using *Matlab®*, and used those results to validate our hardware implementations. In addition, we have performed spike classification on compressed signals in order to evaluate the eventual performance loss after compression. By this mean, the encoding matrix minimizing classification errors has been identified and selected.

The proposed architectures have been described in Register Transfer Level (RTL), then validated with *ModelSim®* and finally synthesized with a 65nm CMOS process from *STMicroelectronics*. The synthesis allowed both surface and power estimation on *Synopsys®* and *Spyglass®* respectively.

V. Architectures and Results

A. Spike Detector

We have designed and evaluated circuits for AV and NEO detection methodologies. For the first one, a few changes have been applied seeking either noise level adaptation or hardware simplification. The NEO detection method has been implemented exactly as described in section II-B.

The AV spike detector proposed circuit calculates the threshold with a multiplication factor of 5 instead of 4 (see eq.2) which results in better response to high noise levels at the expense of increasing the probability of missing spikes. We have chosen by experimentation to use the first 512

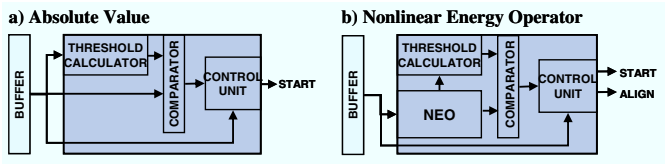

Fig. 2. Spike Detectors Structures – ALIGN is used to shift CS Compressor buffer addressing according to the desired alignment.

samples of the input signal to calculate the threshold. After that, if a sample value is greater than this threshold, this one is considered the twelfth sample of a spike, and no further alignment is performed. This strategy requires that the buffer holds a minimum of 13 samples.

Regarding the NEO approach, for its threshold calculation, we have set N to 1024 which results in a 42.66 ms long setup phase. Then, whenever a spike is detected, the maximum sample value among the threshold-crossing sample and its following 20 samples is set as the twelfth spike data point. With this technique, the buffer must store a minimum of 32 samples.

Fig. 2 demonstrates generically both spike detectors architectures. The estimates of the area and the power for each detector are presented by Table I. In order to compensate the higher complexity of the NEO technique, the detection is done using only the 8 most significant bits of the samples.

TABLE I
SPIKE DETECTOR SYNTHESIS RESULTS

	Area [µm²]		Power [nW]	
Detection Method	AV	NEO	AV	NEO
Buffer	2596.36	6389.76	478	1250
Spike Detector	1608.36	5317.52	193	1000
Total	**4204.72**	**11707.2**	**671**	**2250**

B. CS Compressor

1) Encoding Matrix

There are several possibilities for the implementation of the encoding matrix; in any case, for classification purposes, we must ensure the same matrix multiplication is applied to all spikes. We can either store it in a memory unit or generate it internally during compression.

Memory storage has the advantage of flexibility, i.e., the encoding matrix can be easily changed anytime after the system's conception by reprogramming. In contrast, internal matrix generation can provide area and power optimizations limiting the range possible encoding matrixes once the system is conceived. The authors of [3] propose the use of two *Pseudorandom Bit Sequence* (PRBS) generators; we suggest a *Moore Finite State Machine* (FSM) dedicated to the generation of a specific matrix which has been previously proven well adapted for spike classification (see section IV).

We have synthesized four variations over these described possibilities. For the memory storage, we have considered a SRAM. The PRBS generation method has been implemented with two 6 bits PRBS generators. Additionally, two FSM matrix generators have been analyzed, one producing one matrix column for each clock cycle (32 states) and another doing it element by element (192 states). Table II shows the area and the power consumption estimates for these circuits, including that of their control logic. We have parameterized all implementations to permit the reading of the whole matrix in 1 ms.

2) Compression Computation

The CS compression, i.e., the calculation of (1), is composed by multiplications and sums. To simplify the operations, we consider the binary values 0 and 1 of the encoding matrix as representing the states +1 and -1 respectively, thus the multiplication by these values can be easily done with *XOR* gates and the carry-in input of adders [3].

TABLE II
ENCODING MATRIX SYNTHESIS RESULTS

Implementation	Area	Power
SRAM	720.5 µm²	58.37 nW
PRBS	319. 8 µm²	53.5 nW
FSM 32 States	282.36 µm²	24.5 nW
FSM 192 States	403.52 µm²	50.5 nW

The simplest strategy is to implement all the computation in a completely combinatory logic including the encoding matrix, however it is probably the most area and power consuming approach. Alternatively, the compression can be calculated on several cycles, so we consider an adaptation of the strategy in [3] with the use of XOR ports and accumulators (Fig. 3); in addition, we suggest the reduction to only one adder structure with multiplexed inputs as illustrated by Fig. 4. These implementations compress one spike throughout 32 and 192 clock cycles respectively.

In order to correctly compare all these solutions, we have standardized the output word length to 18 aiming to reduce the incidences of overflows. Besides, we have set the frequencies at 32 kHz and 192 kHz to have the compressed spikes available at the output 1 ms after their detection. The previously described FSMs have been used to generate the matrix for the sequential calculation approaches. Table III shows the synthesis results, it comprises the matrix generation, and the compression calculation modules alongside their control units.

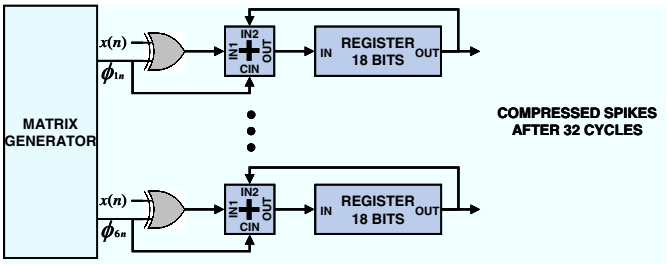

Fig. 3. Adapted Implementation – n represents the current calculation cycle, it iterates through the spike samples and the matrix columns.

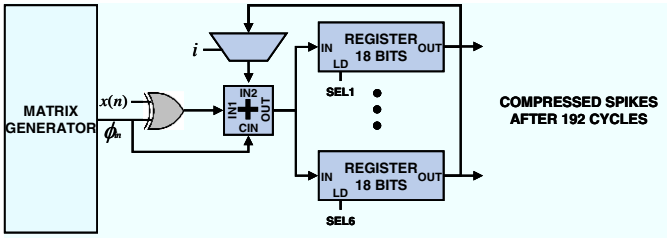

Fig. 4. Multiplexed Accumulators – the combinations of n and i represent the current calculation cycle, n iterates through the spike samples and the matrix columns whereas i do it through its rows.

TABLE III
COMPRESSION COMPUTATION SYNTHESIS RESULTS

Implementation	Area	Power
Combinatory	40899.04 µm²	4.58 µW
Seq. 32 Cycles	4306.9 µm²	561.5 nW
Seq. 192 Cycles	3482.44 µm²	670.5 nW

VI. DISCUSSION

As already stated, the main goal of this work is to optimize power on implanted BMI sensor nodes. We recommend the use of the NEO detection methodology that shows better performance for larger noise levels than the AV approach. Considering the implementation of the encoding matrix, we observe that memory storage, though flexible, presents the worst area and power results among the proposed solutions. On the other hand, the FSM matrix generators are the best optimized circuits in terms of energy consumption, still the choice between these two approaches depends on the compression computation strategy. Though the 192 cycles implementation occupies a smaller area, its higher frequency rate in order to achieve the same runtime as the one with 32 cycles makes it more power consuming; hence the latter will be considered for our final solution. Table IV presents the final results for each chosen module as well for the whole circuit considering both the AV and the NEO spike detection methods.

The resulting power density is 136.9 µW/mm² and 170.14 µW/mm² with the AV and the NEO spike detectors respectively, they are both inferior to 800 µW/mm², thus these circuits respect the power density constraint. The input rate is 384 kbps (24000 samples/s x 16 bits/sample), and, assuming an average spike firing rate of 100 spikes/s, we have 10.8 kbps (100 spikes/s x 6 words/spike x 18 bits/word) at the output which results in a 97.19% data rate reduction.

Table V compares these solutions with previously proposed alternatives. For multi-channels solutions, only one channel is taken into account. Among all circuits, our implementations present the lowest area occupation. As well, they offer an excellent data compression rate. As an example, considering the data transmission consumption at 3 nJ/bit [11] and the same input rate the transmission power dissipation, using our circuit for one-channel is reduced to 32.4 µW whereas it achieves a minimum of 92.16 µW with the other referenced solutions.

VII. CONCLUSION

Starting from the results of [4] about the feasibility of the use of CS on BMIs for the compression of neural spikes, we have designed and analyzed a hardware detection and CS compression system. The results indicate a potential data rate reduction of approximately 97%, which greatly decreases the power consumption of wireless transmission in implanted BMI systems. This implementation represents a great improvement in terms of compression rate over the recently proposed alternatives. Future studies may include the final stages of circuit conception and tests in a real BMI environment.

TABLE IV
FINAL SOLUTION SYNTHESIS RESULTS

	Area [µm²]		Power [nW]	
Detection Method	AV	NEO	AV	NEO
Buffer	2596.36	6389.76	478	1250
Spike Detector	1608.36	5317.52	193	1001
CS Compressor	4925.44		579	
Total	**9130.16**	**16632.72**	**1250**	**2830**

TABLE V
COMPARATIVE TABLE

References / Solutions	Process	Area	Power	Data Reduct. Rate
[4]	90 nm	0.09 mm²	1.9 µW	90%
[6]	500 nm	0.11mm²	75 µW	92%
[7]	90 nm	0.06 mm²	2.03 µW	91.25%
Solution AV	**65 nm**	**0.009 mm²**	**1.25 µW**	**97.19%**
Solution NEO		**0.017 mm²**	**2.83 µW**	

REFERENCES

[1] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A.H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," in *Nature*, 1998, vol. 442, no. July, pp. R53-R78.

[2] T. M. Sees et al., "Characterization of tissue morphology, angiogenesis, and temperature in adaptive response of muscle tissue to chronic heating," in *Lab. Investigation*, 1998, vol. 78, no. 12.

[3] F. Chen, A. P. Chandrakasan, and V. M. Stojanović, "Design and Analysis of Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors," in *IEEE J. Solid-State Circuits*, 2012, vol. 47, no.3, pp. 744-756.

[4] B. Coppa, R. Héliot, O. Michel, and D. David, "Low-cost intracortical spiking recordings compression with classification abilities for implanted BMI devices," in *34th IEEE EMBS Conference*, 2012.

[5] R. Olsson III, and K. Wise, "A three dimensional neural recording microsystem with implantable data compression circuitry," in *IEEE Journal Solid-State Circuits*, 2005, vol. 40, no. 12, pp. 2796-2804.

[6] V. Karkare, S. Gibson, and D. Marcović, "A 130-µW, 64-Channel Neural Spike-Sorting DSP Chip," in *IEEE J. Solid-State Circuits*, 2011, vol. 46, no. 5, pp. 1214-1222.

[7] D. L. Donoho, "Compressed Sensing," in *IEEE Trans. Inf. Theory*, 2006, vol. 52, pp. 1289-1306.

[8] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering," in *Neural* Computation, 2004, no. 16.8, pp. 1661-1667.

[9] S. Gibson, J. W. Judy, and D. Marković, "Comparison of Spike-Sorting Algorithms for Future Hardware Implementation," in *30th Annual International IEEE EMBS Conference*, 2008, pp. 5015-5020.

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

[11] J. L. Bohorquez, J. L. Dawson, and A. P. Chandrakasan, "A 350 µW CMOS MSK transmitter and 400 µW OOK super-regenerative receiver for medical implant communications", in *Symp. VLSI Circuits Dig. Tech.*, 2008, pp. 32-33.