FPGA-based clustering of multi-channel neural spike trains

László Schäffer*, Zoltán Nagy†‡, Zoltán Kincses*, Zsolt Vörösházi§, Richárd Fiáth¶, István Ulbert¶† and Péter Szolgay†‡
*Dept. of Technical Informatics, Faculty of Science and Informatics, University of Szeged, Szeged, H-6725
Email: {schaffer,kincsesz}@inf.u-szeged.hu
†Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, H-1083
‡Cellular Sensory and Optical Wave Computing Laboratory, Hungarian Academy of Sciences, Budapest, H-1111
Email:{nagy.zoltan,szolgay}@sztaki.mta.hu
§Dept. of Electrical Engineering, University of Pannonia,Veszprém, H-8200
Email: voroshazi@vision.uni-pannon.hu
¶Institute of Cognitive Neuroscience and Psychology, Hungarian Academy of Sciences, Budapest, H-1117,
Email: {fiath.richard,ulbert.istvan}@ttk.mta.hu

***Abstract*—Electro-physiological recording of neural bio-electrical activity contains local field potentials and unit activities. Unit activity is a mixture of action potentials generated by the neurons. Spike sorting is a method to determine which individual neurons produce the recorded unit activity. High-channel-count neural probes can measure more than a hundred different positions of the brain in parallel, so large amount of high-dimensional data is generated. To increase the computational speed and decrease the processing time Field-Programmable Gate Array (FPGA) architectures can be applied as hardware accelerators. In this paper an FPGA-based implementation of the Expectation-Maximization (EM) algorithm for neural spike clustering is presented.**

I. Introduction

Electro-physiological recording methods are one of the dominant experimental techniques in the field of neuroscience used to investigate fundamental neuronal mechanisms and higher-order brain functions, such as perception, learning and memory. The bioelectrical activity recorded with neural probes from the extracellular space of the brain tissue can be separated into two main frequency bands: local field potentials containing low-frequency components (below 500 Hz) and unit activity comprising high frequencies (500-5000 Hz). Unit activity is the mixture of spike trains, which are sequences of brief, electrical impulses generated by neurons surrounding the neural probe.

Spike sorting is a method used to separate the spike trains of individual neurons from the recorded unit activity [1]. A typical spike sorting algorithm contains computationally intensive steps, such as feature extraction and clustering [1]. High-channel-count neural probes are capable of recording from up to more than hundred individual brain positions simultaneously pose an even greater challenge for spike sorting applied on general-purpose hardware. However, implementing the spike sorting algorithm on dedicated hardware (e.g. FPGA, GPU, or ASIC (Application-Specific Integrated Circuits)) can significantly reduce the computation time required to process large amounts of high-dimensional data [2]. Therefore, such hardware-accelerated data processing could greatly increase the sorting speed both for real-time clinical applications (e.g. brain-machine interfaces [3]) and for offline analysis of experimental data (e.g. studies of neural network dynamics [4]).

The most computationally intensive task for spike sorting is the offline clustering of neural data recorded with high-channel-count probes. During clustering spikes are classified into different groups based on their extracted features, where the groups correspond to different neurons. In the recent years, many clustering methods were used for spike sorting with different properties and classification performance [5], [6]. In this paper an FPGA-based implementation of an unsupervised clustering algorithm - the modified Expectation-Maximization (EM) algorithm [7] - is proposed to achieve performance increase during clustering the neural spike trains.

II. Expectation-Maximization algorithm

The EM algorithm is an iterative method for maximizing the expectation of the log likelihood function over a distribution. The main equation of the algorithm is the following (1):

$$L(w_k,\mu_k,\Sigma_k) := \sum_{n=1}^{N} E_{\tilde{x}}[\log(\sum_{k=1}^{K} w_k \frac{\exp(-\frac{1}{2}(\tilde{x}_n-\mu_k)^T\Sigma_k^{-1}(\tilde{x}_n-\mu_k))}{(2\pi)^{\frac{d}{2}}||\Sigma_k||^{\frac{1}{2}}})] \quad (1)$$

Where L is the log likelihood function, w_k is the weight of cluster k, while μ_k is the mean of cluster k, Σ_k is the covariance matrix of cluster k, d is the number of features, $E_{\tilde{x}}$ is the expected value based on $\tilde{x}_n$ which is the virtual distribution.

The unmasked EM algorithm consists of two main steps: M-step and E-step. Before the M-step the weight w_k has to be calculated, then (in the M-step) the mean μ_k and the covariance Σ_k of each cluster is computed. In the E-step the computation of the log of responsibilities is performed.

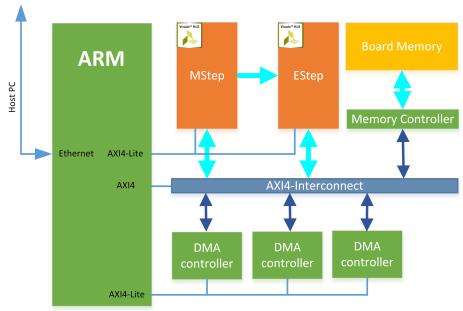

Fig. 1. The architecture of the system

III. The Proposed FPGA-based Architecture

A. The experimental setup

The experimental setup is an Avnet/Digilent Zedboard [8] based on a Xilinx Zynq-7020 APSoC architecture, which is built-up from a dual-core ARM Cortex-A9 PS (Processing System) and a Xilinx Series-7 PL (Programmable Logic). The PS has wide variety of different I/O interfaces to connect the system to the outside world such as Gigabit Ethernet and DDR3 memory controller to name a few. The features of the neural spike trains are fed into the external DDR3 memory of the ZedBoard via Gigabit Ethernet. During the computation, partial and final results are also stored in this memory, which can be transferred back to the PC for further processing (e.g. visualization).

B. The architecture

The architecture of the proposed system is built-up from six main parts as can be seen on Fig. 1. These are the *ARM Processor*, the *DMA Controllers*, the *AXI-4 Interconnect*, the *Memory Controller*, the *Board Memory* and the *EM Core*.

The *ARM Processor* communicates with the host computer via Ethernet and controls the data-flow on the AXI4-Lite and AXI4 buses. Furthermore it pre-calculates the required matrices for the *EM Core*. The *Memory Controller* and the *Board Memory* are responsible for storing the recorded features extracted from the neural spike trains and the results of the clustering process. The data transfer between the *EM Core* and the *Board Memory* is handled by the *DMA Controllers*. The data is stored sequentially in the memory, but the *EM Core* requires it in a mixed manner. Therefore scatter-gather DMA instructions are used. The *EM Core* computes the algorithmic steps of the EM algorithm. In the first step it calculates the covariance matrices. In the second step the upper triangular part of the covariance matrices are computed using Cholesky decomposition, then a matrix inverse operation is performed to solve the resulting equation system. In the *EM Core* only the data for the actual cluster is stored, to minimize the BRAM memory usage. The *AXI-4 Interconnect* provides the connection between the parts of the system.

C. The test results

In this implementation the unmasked version of the EM algorithm was used, so the number of possible clusters and the number of initial clusters can be configured. The size of the covariance matrix (*nDims*nDims*) is defined by the number of channels and features (*nDims=channelNum*featureNum*). The proposed architecture was tested using different covariance matrix sizes from 24-192. The corresponding channel numbers are 8-64. The test results can be seen on Table I.

TABLE I
Resource Requirement and Computational Speed

nDims	BRAM(18K)	DSP48E	FF	LUT	SpeedUp (PC/FPGA)
24	8	36	8865	12217	0.18
48	20	36	8911	12342	0.11
96	68	36	8957	12425	0.08
192	260	36	9003	12509	0.12

The test results show, that the BRAM memory requirement of the *EM Core* increasing quadratically with the channel number, and the computational performance of the proposed architecture is lower than the PC CPU-based solution. But the energy consumption of the Zedboard is only 2 Watts, while the Core i7 CPU of the PC requires more than 60 Watts. Therefore the FPGA-based implementation is more energy-efficient, than the PC CPU-based solution, even the FPGA uses exactly the same algorithm, with the same parameters, and data.

IV. Conclusion

In this paper an FPGA-based implementation of the Expectation-Maximization algorithm for high-channel-count neural spike clustering is proposed.

The test results show that the computational performance of our architecture is lower, than the PC CPU-based solution, but it is more energy-efficient. The main reason of lower performance is that the FPGA-based solution is fully sequential.

Our future work is to make this architecture to parallel, which will further increase the performance of this FPGA-based system.

References

[1] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," Network, vol. 9, pp. R53-78, Nov 1998.
[2] S. Gibson, J. W. Judy, and D. Markovic, "An FPGA-based platform for accelerated offline spike sorting," J Neurosci Methods, vol. 215, pp. 1-11, Apr 30 2013.
[3] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, et al., "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," Nature, vol. 442, pp. 164-71, Jul 13 2006.
[4] S. Fujisawa, A. Amarasingham, M. T. Harrison, and G. Buzsaki, "Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex," Nat Neurosci, vol. 11, pp. 823-33, Jul 2008.
[5] A. Bar-Hillel, A. Spiro, and E. Stark, "Spike sorting: Bayesian clustering of non-stationary data," J Neurosci Methods, vol. 157, pp. 303-16, Oct 30 2006.
[6] A. Oliynyk, C. Bonifazzi, F. Montani, and L. Fadiga, "Automatic online spike sorting with singular value decomposition and fuzzy C-mean clustering," BMC Neurosci, vol. 13, p. 96, 2012.
[7] S. N. Kadir, D. F. Goodman, and K. D. Harris, "High-dimensional cluster analysis with the masked EM algorithm," Neural Comput, vol. 26, pp. 2379-94, Nov 2014.
[8] Digilent webpage (2016): www.digilentinc.com