FPGA-Based Real-Time Multichannel Neural Dataset Generation

László Schäffer*, Zoltán Nagy†, Zoltán Kincses*, and Richárd Fiáth¶†
*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,
Email: nagy.zoltan@itk.ppke.hu
¶ Institute of Cognitive Neuroscience and Psychology, Research Centre for Natural Sciences,
Hungarian Academy of Sciences, Budapest, H-1117, Email: fiath.richard@ttk.mta.hu

***Abstract*—Miniaturized voltage sensors (electrodes) implanted into the brain tissue are capable of recording the brief electrical impulses (spikes) of neurons located close to the electrode sites. To investigate the activity of individual neurons and discriminate spikes generated by different neurons a technique called spike sorting can be applied on the recorded data. However, the performance of current spike sorting methods is challenged by multichannel neural data recorded with high-density, high-channel count silicon probes developed recently. Our group started to develop an FPGA-based solution to accelerate the clustering of spikes detected in high-channel count neural recordings. It is a crucial step of the development to validate the performance of the clustering algorithm. This can be achieved by using ground truth datasets where the exact time of spikes fired by different single units are known. In this paper we present an FPGA-based architecture for real-time generation of multichannel hybrid ground truth datasets, which will be used for the validation of our FPGA-based clustering algorithm.**

I. Introduction

One of the oldest and most widely used experimental technique of neuroscientists to investigate the complex functions of neural networks is the extracellular measurement of brain electrical activity. Miniaturized voltage sensors (electrodes) implanted into the brain tissue are capable of recording the brief electrical impulses (called action potentials or spikes) of dozens of neurons located close to the electrode sites [1]. However, to investigate the activity of individual neurons (single-unit activity) and their relationship with other neurons in the examined neural circuit, we have to discriminate spikes generated by different neurons. This discrimination step is usually performed by applying spike sorting on the recorded data, an analysis method during which individual neuronal identities are assigned to each detected spike [2].

Spike sorting is used in many fields of basic neuroscience research (e.g. to study the dynamics of neural networks [3]) and is usually one of the first analysis methods which is applied on the recorded spiking activity data. The identification of spikes of individual neurons is of great importance in some real-time clinical applications as well (e.g. neuroprosthetic devices, brain-machine interfaces [4]). Usually, the algorithms developed for spike sorting contain multiple steps [2], some of which are computationally intensive, such as spike detection or clustering (assigning the detected spikes to the neuron clusters). Furthermore, the performance of current spike sorting methods is challenged by multichannel neural data recorded with high-density, high-channel count silicon probes developed recently (e.g. [5]). These novel devices consisting of several hundred, closely-spaced electrode sites are able to record the activity of large neural ensembles from up to more than hundred individual brain positions simultaneously. In contrast, most spike sorting algorithms are prepared to process data recorded with only a few (usually four) electrodes. Therefore, new algorithms using novel approaches and/or implementing multiple steps of spike sorting on dedicated hardware (e.g. Field Programmable Gate Array (FPGA), Application-Specific Integrated Circuits (ASIC), Graphics Processing Unit (GPU)) are under intensive development to reduce the computation time required to process the recorded high-dimensional data [6]-[12].

Recently, our group started to develop an FPGA-based solution to accelerate the clustering of spikes detected in high-channel count neural recordings [8]. It is a crucial step of the development to validate the performance of the clustering algorithm. This can be achieved by using ground truth datasets where the exact time of spikes fired by different single units are known. There are several methods to obtain such ground truth datasets [10]. Simultaneous paired recordings collecting the spikes of the same neuron both extracellularly and intracellularly would be the most optimal solution, however, this method is technically challenging, therefore the availability of such datasets are limited [13]. Simulating the activity of biophysically realistic neural networks is also an option [14], but the generation of synthesized multichannel datasets needs high computational power provided only by computer clusters. Finally, hybrid ground truth datasets can be generated by using the mean spike waveforms of a subset of well-separated units isolated from real extracellular recordings as templates or donors and add these spike templates at random times and positions of simulated or real recordings [9]-[12]. We followed the latter method and synthesized our multichannel hybrid ground truth datasets using an FPGA board. These datasets will be used for the real-time validation of our FPGA-based

978-1-5386-3974-0/17/$31.00 ©2017 IEEE

clustering algorithm [15].

II. Spike Generation Requirements

In order to generate realistic neural data (spikes) many requirements have to be fulfilled. The absolute refractory period of a neuron is 1-2 milliseconds long, during this period the neuron is not able to generate spikes, and the interspike intervals of a neuron have a log-normal distribution [16]. Therefore, a log-normally distributed random number generator should be applied. Furthermore, it was found that the spike-amplitude decays as $\frac{1}{r}^n$ with $1 <= n <= 2$ close to soma and $n >= 2$ far away, where r corresponds to the distance of the electrode from the recorded neuron. Based on this decay and on our 128-channel model probe, where the electrodes are arranged in a 32x4 matrix shape with an electrode pitch of 20 μm, the action potential of an average neuron can be detected at most six electrodes (120 μm) away from the center, where the spike can be recorded with the highest amplitude [17]. Finally, in our first approach, noise with uniform distribution was added to the generated data.

In addition to these requirements, real-time multichannel spike generation is required to test our multichannel Online Sorting [15] architecture where the main parameters like the number of neurons, the number of channels and the size of the inter-channel action potential spreading should be configurable.

The sampling rate of the real measurement system where the OSort architecture will be used is 20 kHz, therefore in our case real-time means that the spike generator should produce a sample on all channels with at least 20 kHz frequency.

III. Random Number Generation

To generate the neural dataset, uniform and log-normal distributed random number generation is required. In the literature various uniform random number generation methods can be found. We have chosen the pseudo-random Mersenne Twister (MT) algorithm, which can provide an astronomical period of $2^{19937} - 1$ and 623-dimensional equidistribution up to 32-bit accuracy. Furthermore, it passes several statistical tests, including diehard. The algorithm is based on polynomial calculations over the two-element field and uses a twisted generalised feedback shift register with the help of a tempering matrix. The pseudo code and the detailed description of the algorithm can be found in [18].

A continuous probability distribution of a randomly generated variable, which logarithm is normally distributed, is called log-normal distribution. The log-normally distributed random number generation is based on the output of the MT algorithm normalized into the $(0, 1]$ interval. The inverse transform sampling method can be used to generate random numbers from any probability distribution, if the cumulative distribution function (CDF) is known and invertible. The CDF of the log-normal distribution can be written as follows:

$$\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\Big[\frac{\ln x - \mu}{\sqrt{2}\sigma}\Big], \tag{1}$$

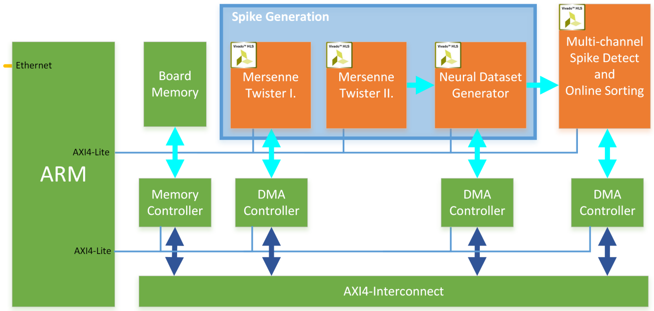

Fig. 1. The schematic diagram of the overall system

where x is the probability variable, μ is the mean, σ is the variance, and erf is the error function, which can be written as follows:

$$\frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt, \tag{2}$$

and $erf(x)$ describes the probability of a random variable X falling in the range $[-x, x]$. Inverting the log-normal CDF, the quantile function can be given by:

$$e^{\mu + \sigma \Phi^{-1}(p)}, \tag{3}$$

where p is a uniformly distributed random number in the range of $(0, 1]$, and Φ^{-1} is the inverse error function, which can be approximated using a Maclaurin series as follows:

$$\sqrt{\pi}(\frac{1}{2}z + \frac{\pi}{24}z^3 + \frac{7\pi^2}{960}z^5 + \frac{127\pi^3}{80640}z^7 ...), \tag{4}$$

where z is an uniformly distributed random number in the range of $(-1, 1)$.

IV. Overall System

The schematic diagram of the overall multichannel spike sorting system extended with the real-time neural dataset generator can be seen in Fig. 1. The system consists of the *ARM Processing System* (PS), the *DDR3 Board Memory*, the *Memory Controller* core, the *DMA Controller* cores, the *Mersenne Twister* cores, the *Neural Dataset Generator* core and the *Multi-channel Spike Detect and Online Sorting* core.

The *DDR3 Board Memory*, the *Memory Controller* core, some of the *DMA Controller* cores and the *Multi-channel Spike Detect and Online Sorting* core are responsible for the spike detection and classification [15].

In this paper, the Spike Generation will be discussed, which contains two main parts: the *Mersenne Twister* cores, and the *Neural Dataset Generator* core. The *Mersenne Twister* cores are responsible for the generation of random numbers used in the inverse transform sampling to compute the log-normal distribution of interspike intervals (*Mersenne Twister I.* core) and the noise generation (*Mersenne Twister II.* core).

However, the FPGA implementation of the log-normally distributed random number generator is done, but the *ARM PS* has enough computation performance to complete this task. In this way, the log-normal random number calculation does not occupy any FPGA resources.

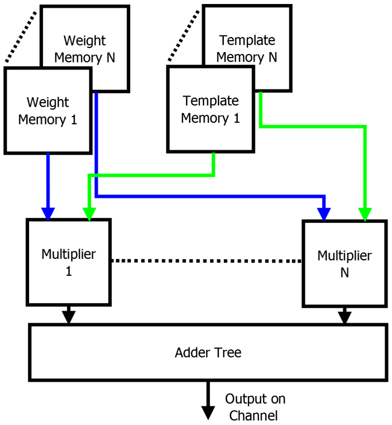

Fig. 2. Schematic diagram of the output generation for one channel

The *ARM PS* and the cores are connected to each other through *AXI4-Lite* and *AXI4-Interconnect* buses, while the *ARM PS* communicates with a host PC through an Ethernet port.

A. *Mersenne Twister I-II cores*

The Mersenne Twister I core calculates the random numbers required in inverse transform sampling to generate log-normally distributed firing times. The Mersenne Twister II core is responsible for the generation of the noise.

These cores are based on the pseudo-code and parameters published in [18], but some modifications were required to achieve optimal performance during the FPGA implementation. The initialization phase of the algorithm is not changed, but the twist and the output generation is separated, since only the output generation can be pipelined. Using this modified algorithm the twist step can be performed in 1872 clock cycles, when N is 624. After this initialization period is completed a new random number can be generated in each clock cycle. The twist step is called after generating 624 random numbers.

B. *Neural Dataset Generator core*

The *Neural Dataset Generator* core generates the multi-channel neural dataset. The first part of this process is the simulation of the action potential spreading through the neighbouring 6 channels. This is based on a pre-defined normally distributed Gaussian-kernel. The shape of the electrode array is stored in a Look Up Table (LUT), which is used for proper indexing of the weight matrix. For each channel and each neuron, the weight matrix contains the corresponding spreading weight value from the Gaussian-kernel. The electrode array LUT and the weight matrix is pre-calculated on the *ARM PS*. Therefore, the generation of one output sample is a simple multiplication between the weight coefficient of the actual channel and the value of the actual spike template. Finally, an adder tree summarizes the weighted template values and the result will be the output of one channel at one time. The architecture of the *Neural Dataset Generator* core can be seen in Fig. 2.

TABLE I
AREA REQUIREMENTS AND DEVICE UTILIZATION OF THE SYNTHESIZED MODULES

	Available resources			
	BRAM	DSP	FF	LUT
	102	120	12,150	14,034
	Device Utilization			
	BRAM	DSP	FF	LUT
Mersenne Twister x2	1%	7%	1%	4%
Log-normal (ARM)	4%	20%	3%	11%
Neural Signal Generator	7%	34%	6%	25%
Summarized	8%	41%	7%	29%

Another important part of the dataset generation is to set up the log-normally distributed intra-channel spike firing distances for each individual neuron. On the *ARM PS* the generated log-normally distributed random numbers are sorted in ascending order with the corresponding neuron values and stored in a structure. This firing structure contains the firing time and the identification number of the firing neuron.

To generate the final multichannel dataset, the *Neural Dataset Generator* core reads the neuron identification value from the firing time structure and enables that neuron to fire at the corresponding time stamp. The next fire time will be read in the next clock cycle. When all neurons are enabled to fire, then the *Neural Dataset Generator core* pauses the reading.

The *Multi-channel Spike Detection and Online Sorting* core processes the data from the electrode by channel, so the core gets one channel sample per clock cycle. Therefore the *Neural Dataset Generator* core should work in the same fashion.

V. IMPLEMENTATION RESULTS

The proposed *Mersenne Twister* and *Neural Spike Generator* cores (Signal Generation subsystem) are developed and synthesized using the Vivado High Level Synthesis (HLS) version 2016.4. The overall architecture is synthesized to a ZedBoard equipped with a Zynq-7020 FPGA. The available resources on the Zynq-7020 and the resource requirements of the cores can be seen in Table I.

During the development, the FPGA and the ARM-based implementation of the *Log-normal* core were tested. Since the *ARM PS* is capable to generate log-normally distributed random numbers in 150 FPGA clock cycles, the ARM PS was chosen for this calculation. In this case this core does not require any FPGA Programmable Logic resources.

If the resource requirement of the Multichannel Spike Detect and Online Sorting [15] is added to the resource requirements of the Spike Generation subsystem, the DSP resource utilization is greater than 115%. Therefore, the Log-normal core can not be implemented on the Zynq-7020 FPGA and this is the second reason why it is important to run this computation on the *ARM PS*.

After the Vivado HLS synthesis, the latencies of the cores can be seen in Table II. In the case of the *Mersenne Twister* core, the twist step requires 1872 clock cycles, while the output

TABLE II
DETAILED LATENCIES OF THE SYNTHESIZED CORES

		Latency	Iteration Latency	Trip Count
Mersenne Twister	init	5607	9	623
	twist	1872	3	624
	output_gen	625	3	624
Log-normal (ARM)	init	57	27	32
	maclaurin	140	14	128
	output_gen	65	35	32
Neural Signal Generator	dataset_gen	2560026	28	2560000

generation requires 624 of it. So, $1872 + 624 = 2496$ clock cycles are needed to generate 624 random numbers. Therefore, an average of 4 clock cycles are required to generate one random number. The *Log-normal* core running on the *ARM PS* requires 150 clock cycle latency to generate one log-normal value. The spike generation is performed in one second windows using the *Neural Signal Generator* core. This core requires only one clock cycle to generate one sample on all 128 channels. Before the first window, a proper initialization is required. This initialization consists of the electrode array LUT, the weight matrix, the random number and the log-normal number generation. These calculations are performed by the *ARM PS*, except the random number generation, which is done by the *Mersenne Twister II.* core. During the generation of the channel samples of the next windows, only the log-normal number generation is required, which can be performed parallel to the previous window. The clock frequency of the Zynq-7020 on the ZedBoard is 100 MHz, so the generation of the channel samples can reach and even exceed the necessary 20 kHz sampling rates.

In Fig. 3 the results of the dataset generation can be seen on 5 adjacent channels with 3 firing neurons in a 0.1 second window, where the inter-channel signal spreading can be observed. The spikes are grouped by the firing neurons, each neuron has a unique color, while multi-unit spikes have black color.

VI. CONCLUSION

In this paper a real-time multichannel hybrid ground truth neural dataset generator is presented using a Zynq-7020 FPGA. The results show that the presented architecture can be implemented on the Zynq-7020 FPGA with 100 MHz clock frequency and it is capable of generating multichannel neural samples at every clock cycle. The neural dataset generator will be used for the validation of our Multichannel OSort spike sorting architecture. Another purpose of the proposed neural spike generator could be the pre-calibration or pre-teaching of the Multichannel OSort architecture.

ACKNOWLEDGMENT

The authors would like to thank Istvan Ulbert for the neural templates and the theoretical background. The research leading to these results has received funding from the Hungarian Brain Research Program Grants (Grant Nos. KTIA_13_NAP-A-I/1 and KTIA-13-NAP-A-IV/1-4,6).

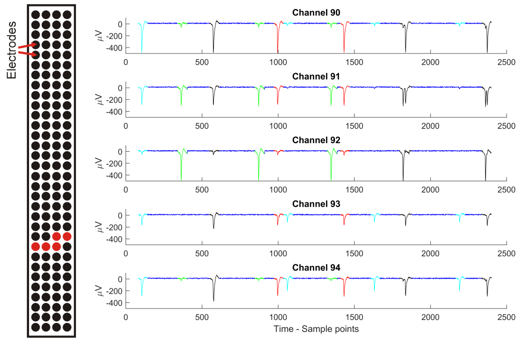

Fig. 3. A 0.1 second long segment of the generated neural data on 5 adjacent channels (electrodes corresponding to the channels are indicated on the left side with red circles on the schematic image of the model probe). The spikes of the three different neurons are highlighted in different colors.

REFERENCES

[1] G. Buzsaki, "Large-scale recording of neuronal ensembles," Nat Neurosci, vol. 7, pp. 446-51, May 2004.

[2] 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.

[3] 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.

[4] L. R. Hochberg et al., Neuronal ensemble control of prosthetic devices by a human with tetraplegia, Nature, vol. 442, pp. 164-71, Jul 13 2006.

[5] C. M. Lopez et al., "An Implantable 455-Active-Electrode 52-Channel CMOS Neural Probe," IEEE Journal of Solid-State Circuits, vol. 49, pp. 248-261, Jan 2014.

[6] 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.

[7] W. J. Hwang, W. H. Lee, S. J. Lin, and S. Y. Lai, "Efficient architecture for spike sorting in reconfigurable hardware," Sensors (Basel), vol. 13, pp. 14860-87, 2013.

[8] L. Schaffer, Z. Nagy, Z. Kincses, R. Fiath, I. Ulbert, FPGA-based clustering of multi-channel neural spike trains, CNNA 2016, Dresden, Germany, August 23-25. 2016.

[9] M. Pachitariu, N. A. Steinmetz, S. Kadir, M. Carandini and K. D. Harris, Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels, bioRxiv, dx.doi.org/10.1101/061481, 2016

[10] J. J. Jun e al., Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction, bioRxiv, dx.doi.org/10.1101/101030, 2017

[11] C. Rossant et al., "Spike sorting for large, dense electrode arrays," Nat Neurosci, vol. 19, no. 4, pp. 634-41, Mar 14 2016.

[12] P. Yger et al., Fast and accurate spike sorting in vitro and in vivo for up to thousands of electrodes, bioRxiv, dx.doi.org/10.1101/067843, 2016

[13] J. P. Neto et al., "Validating silicon polytrodes with paired juxtacellular recordings: method and dataset," Journal of Neurophysiology, vol. 116, no. 2, pp. 892-903, Aug 1 2016.

[14] E. Hagen et al., "ViSAPy: A Python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms," Journal of Neuroscience Methods, vol. 245, pp. 182-204, Apr 30 2015.

[15] L. Schaffer, Z. Nagy, Z. Kincses, R. Fiath, "FPGA-based neural probe positioning to improve spike sorting with OSort algorithm", ISCAS, Baltimore, United States, May 28-31. 2017

[16] G. Buzsaki and K. Mizuseki, "The log-dynamic brain: how skewed distributions affect network operations," Nature Reviews Neuroscience, vol. 15, no. 4, pp. 264-278, 2014.

[17] K. H. Pettersen and G. T. Einevoll, "Amplitude variability and extracellular low-pass filtering of neuronal spikes," Biophys J, vol. 94, no. 3, pp. 784-802, Feb 1 2008.

[18] M. Matsumoto, T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Transactions on Modeling and Computer Simulation (TOMACS), Vol. 8, Issue 1, pp. 3-30, Jan 1 1998