MULTIPLE PROCESSOR CORE SYSTEMS ON FPGA CIRCUITS IMPLEMENTING BIO-INSPIRED NEURAL NETWORKS FOR CLASSIFICATION TASKS

László BAKÓ[1,2], Sándor-Tihamér BRASSAI[1,2], Lajos LOSONCZI[2,3] and László-Ferenc MÁRTON[1,2]

[1] Sapientia – Hungarian University of Transylvania, Electrical Engineering Department, Tirgu-Mures, Romania
[2] NSRG – Neural Systems Research Group of the Sapientia – Hungarian University of Transylvania, Romania
[3] Lambda Communications Ltd., Tirgu-Mures, Romania

lbako@ms.sapientia.ro

HiPEAC 2013
8th International Conference on High-Performance and Embedded Architectures and Compilers
January 21-23, 2013
Berlin, Germany

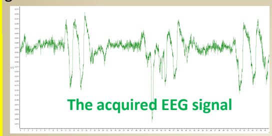
Neural Systems Research Group

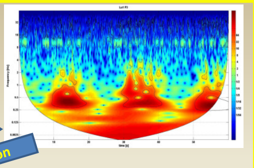
ABSTRACT

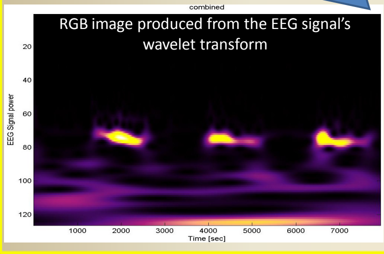
The presented implementations of a SNNs, partly sacrifice the fully parallel nature of the design, by embedding either soft-core processor modules (PicoBlaze or MicroBlaze) or hard-core modules (PowerPC). The input values of the implemented SNNs have been encoded by overlapped and graded sensitivity profiles, algorithm executed by one of the embedded cores. The execution of the neuron body algorithm for all cells in the network is implemented on individual cores. These run in parallel but still execute instructions sequentially. On the other hand, the synapse modules, where the spikes are weighted and the learning rules reside, are all built of dedicated hardware elements, yielding a parallel execution by a heterogeneous many-core architecture.

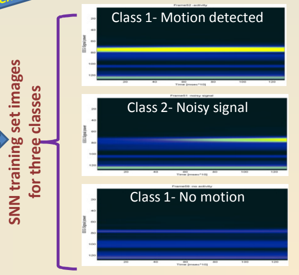
CLASSIFICATION OF EEG SIGNALS – DATASET PREPARATION

The proposed goal was to detect the EEG measurement subject's movements by recognizing patterns in the images generated from the wavelet transform of the acquired signal, into a training image set for the hardware embedded SNN.


The acquired EEG signal



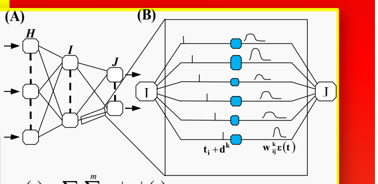

RGB image produced from the EEG signal's wavelet transform


SNN training set images for three classes

EMBEDDED SPIKING NEURON MODEL

- **Synapse:** Modulates the incoming pre-synaptic pulses, Implements the learning algorithm
- **Soma:** accumulates the post-synaptic values => membrane potential value (MP), Generates an axonal spike, if the MP exceeds a pre-defined threshold value

✓ SNN processing is divided into time-steps
✓ Each time-step features several phases:
- α phase: encoding the input values
- β phase: activate synapses, run learning algorithm
- γ phase: executing the soma algorithm



$$x_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^{m} w_{ij}^k y_i^k(t)$$

Figure 1 - Theoretical concept of the implemented neural network

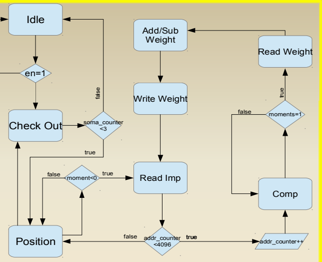
This poster aims to report on the identification of a number of challenges facing the area in terms of creating large-scale implementations of spiking neural networks on reconfigurable hardware, particularly that operate in real time, and yet demonstrate biological plausibility in terms of the adaptability of the architecture.

THE FPGA IMPLEMENTED LEARNING ALGORITHM

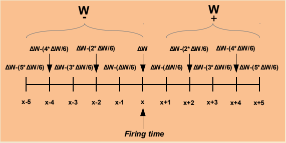
❖ A modified, supervised version of the Hebb algorithm has been devised for this application. Only those weights are adapted that have been active in a timestep close to the moment of the soma activation (at most +/- 5 steps) during the 37-timestep frame.

❖ The weights of the synapses that have activated in the same timestep as the soma or in the previous five will be decreased, while those that activated later than the soma will have their weight increased. The maximum value of the weight change, ΔW, is applied only to one synapse, the one that has activated simultaneously with the soma. The rest of the synapses will change weight by a decreasing value in ΔW/6 increments. Those synapses that have activated in more distant timesteps will retain their current weight values.

❖ The learning process is implemented in the LEARN unit, consisting of an elaborate finite-state machine, as the Figure shows.

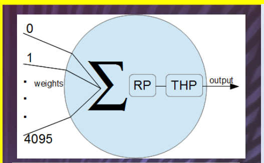


State diagram of the LEARN unit



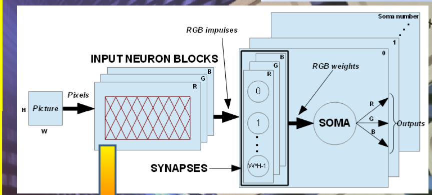
The applied learning rules

FPGA IMPLEMENTATION DETAILS

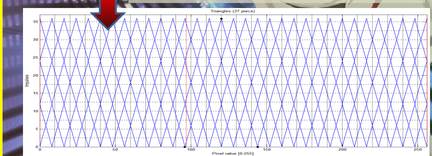


Conceptual structure of the implemented SNs

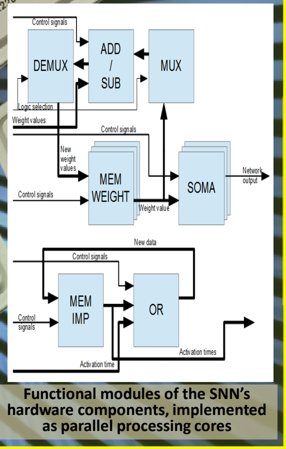
- The three cell body or ***SOMA*** modules are responsible for the spiking neuron's output generation.
- Each soma has a number of synapses per color component (RGB) equal to the number of pixels in the input image. Therefore the number of synapses in an output neuron is three times the number of pixels in the input images.
- This yields a number of about 150000 synapses in the implemented SNN.



Block diagram of the SNN with the input encoding units, synapse modules and somas, all parallelized as processor cores or dedicated logic circuitry



The method of encoding the input variables into delayed spikes, implemented in the PowerPC core

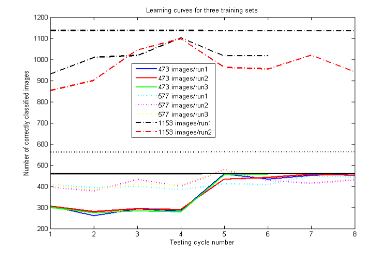


Functional modules of the SNN's hardware components, implemented as parallel processing cores

RESULTS OF THE EEG DATASET CLASSIFICATION

❖ The method used in the experimental setup has varied the number of cycles in which a single image was presented to the SNN as well as the number of cycles that repeated the whole set of images. Testing cycles have been inserted between the training loops in order to track the progress and the convergence of the training.

❖ The experiments have been carried out using EEG signals acquired with the Smart Active Electrode system designed and produced by our team. Two of these measured signals were used to create three training image sets. As Table 1 presents, the SNN managed to produce promising classification results for all datasets.



Learning curves for three experiments using differents training sets of different size

Training set description [Electrode, EEG signal length, nr. of images]	Nr. of cycles [per image*overall]	Correctly classified images [nr.]	Correctly classified images [%]
F3, 60s, 473	20 * 20	455	96.2
F3, 60s, 473	40 * 10	457	96.6
F3, 60s, 473	30 * 15	460	97.2
F4, 73s, 577	30 * 15	450	78
F4, 73s, 577	30 * 20	478	82.3
F4, 73s, 577	20 * 20	475	82.8
F4, 73s, 1153	20 * 20	1102	95.5
F4, 73s, 1153	30 * 15	1098	95.2

Device Utilization Summary of a Xilinx Virtex™-5 FX30T FPGA

Logic Utilization	Used	Available	Utilization
Slice Registers	11060	20480	54%
Slice LUTs	18610	20480	91%
Fully used LUT-FF pairs	552	608	90%
Bonded IOBs	77	360	21%
Block RAMs / FIFOs	27	68	40%
BUFG/BUFGCTRLs	4	32	12%

CONCLUSIONS

❑ Previously built, similar systems validated this method using benchmark tests [7] (IRIS dataset error rate was 18% after 300 training cycles, while accuracy of the WBCD dataset classification was 92%, after 150 training cycles. Execution times were: IRIS 1ms, WBCD 5-10ms.).

❑ The input neurons of the SNN application implemented on FPGA for EEG signal classification were implemented on the 400 MHz PowerPC core embedded into this circuit.

❑ The output neurons with their synapses and somas are all implemented as units working in parallel. The control unit coordinates the various memory modules and finite-state machines that contribute to the successful implementation of the supervised version of the Hebb learning rule.

REFERENCES

[1] G. Wulfram, W. Werner, ''Spiking Neuron Models,'' *Cambridge University Press, Cambridge*, 2002.

[2] M. Schaefer, T. Schoenauer, C. Wolff, G. Hartmann, H. Klar, U. Rueckert, Simulation of spiking neural networks—architectures and implementations, *Neurocomputing* 48 (1) (2002) 647–679(33).

[3] Bohte, S.M., Kok, J.N. and La Poutré, H. Spike-prop: error-backpropagation in multi-layer networks of spiking neurons, *Neurocomputing*, 48(1–4), 2002, pp. 17–37.

[4] Richard S. Zemel, Quentin J. M. Huys, Rama Natarajan, and Peter Dayan. Probabilistic computation in spiking populations. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17, MIT Press, Cambridge, MA*, 2005, pp. 1609–1616

[5] Marchesi, M., Orlandi, G., Piazza, F. and Uncini, A. "Fast neural networks without multipliers," *IEEE Transactions on Neural Networks*, vol. 4, no. 1, Jan. 1993, pp. 53–62.

[6] Maass, W., Natschlager, T. and Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14 (11), 2002, pp. 2531–2560.

[7] László Bakó, Real-time classification of datasets with hardware embedded neuromorphic neural networks, Briefings in Bioinformatics, Special Issue: Parallel and Ubiquitous Methods and Tools in Systems Biology: May 2010; Vol. 11, No. 3, p348-363, doi: 10.1093/bib/bbp066, Oxford University Press

ACKNOWLEDGMENT

This work is part of the project funded by the Romanian National Authority for Scientific Research, grant No. 347/23.08.2011.