

# Design of analog/digital simulator dedicated to real-time neurocomputing

Y. Bornat, J. Tomas, C. Lopez, O. Malot, B. Belhadj, S. Renaud

IMS Labs, CNRS UMR 5218,  
Université Bordeaux 1, ENSEIRB  
Talence, France  
[jean.tomas@ims-bordeaux.fr](mailto:jean.tomas@ims-bordeaux.fr)

**Abstract**—This paper presents a mixed analog/digital hardware simulation system to investigate the dynamics of biomimetic neural networks. The computation core consists in specific integrated circuits (ASIC) that emulate neurons' electrical activity using a biophysical model. The connectivity and plasticity of the network are digitally computed using digital programmable circuits (FPGA). A custom printed board hosts all the components and is connected to a computer by serial link. The network activity respects biological real-time.

**Keywords**—BiCMOS, ASIC, FPGA, analog and mixed design, Hodgkin-Huxley model, spiking neurons, STDP, neural network simulation

## I. Introduction

Neuromorphic engineering is a field of engineering based on the design and fabrication of artificial neural systems, which architecture and design principles are based on those of biological neural systems. Recent research in that field addresses the investigation of spiking neural networks, in a tentative understanding of the temporal coding of information by such networks [1], [2]. We engineered in our research group different systems to process real-time simulations of neural networks, using detailed and biologically-realistic models of neurons [3]. One innovation in those systems is the fact that artificial neurons are integrated on custom analog VLSI circuits (ASICs). These neuromimetic ASICs compute in real-time the electrical activity of neurons, represented by the voltage difference across their membrane ( $V_{MEM}$ ) [4]. Another innovation is that Spike Timing Dependent Plasticity algorithms (STDP) are implemented into digital programmable circuits (FPGA) and optimized to warranty real-time activity of the network.

After presenting the simulation system architecture, we will describe its main components: the analog core, the digital hardware and the custom printed board that hosts the whole system.

## II. Context and specifications

Neuron models exist at different precision levels when considering their biological relevance. The computation mode of artificial neural networks can be classical software, but also digital and/or analog hardware. The implementation choice results from a necessary compromise between the precision and the technical performance, such as computational speed or power consumption. We chose models of spiking neurons that respect the dynamics of the biological neurons activity. The neurons and synapses are implemented on hardware, which improve the computation speed and allow the building of hybrid neural networks, where living cells communicate in real-time with artificial neurons [5].

An active neuron presents action potential - “spikes” - at low frequencies (from 0.1 Hz to 200 Hz) on its membrane  $V_{MEM}$ , that can be spontaneous or induced by synaptic connections from other neurons. The exact timing of spikes is a key information on the neural network activity and the evolution of its connectivity. The synaptic interactions can be subject to short-term and long-term variations, and process cognitive mechanisms that rely on plasticity and learning. In our simulation systems, the neuronal activity is computed in real-time and in analog hardware. Synaptic interactions, that control the information transmission between the neurons, are processed digitally using a hardware medium. The systems are used by neuroscientists to investigate learning or plasticity algorithms that are presumably present in biological neural networks [6].

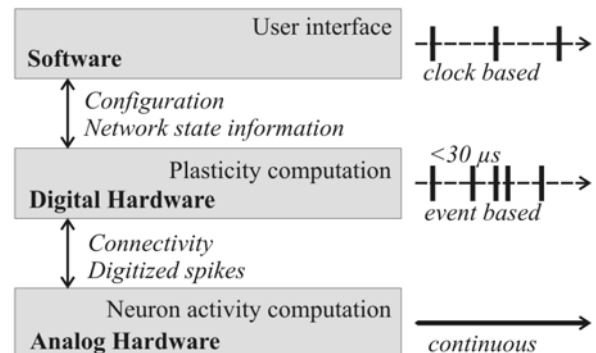


Fig. 1. : Architecture and data flow of the simulation system.

The whole simulation system is organized in 3 layers (Fig 1). The analog hardware layer runs the continuous and real-time computation of the neurons activity. The analog integrated circuits (ASICs) are controlled by the digital hardware layer. This hardware is in charge of collecting spike events information from the analog neurons, of computing Spike Timing Dependent Plasticity (STDP) algorithms and of controlling the synaptic connectivity back to the analog hardware. To optimize computational speed, the processing mode is event-based, with a ensured maximum period of  $30\mu\text{s}$ . Predefined stimulation patterns can also be applied to individual neurons. Within the software layer, a computer running a real-time operating system hosts a dedicated software. It provides user interface functions to control off-line and on-line the simulation configuration and to collect the network state information.

### III. Analog hardware

#### A. The neurons model

As mentioned in the introduction, we chose to implement a biologically-realistic model of spiking neurons, able to capture the main intrinsic and response properties of cortical neurons, and compatible with the required level of precision in the timing of spikes. The corresponding category of models is the conductance-based models (Hodgkin-Huxley type [7]). For those models, the electrical activity of a single neuron is computed by summing ionic and synaptic currents on a capacitance  $C_{MEM}$  that represents the neuron membrane (Fig. 2). Each ionic current ( $I_{Na}$ ,  $I_K$ , ...) is described by a set of non-linear and time and voltage-dependent equations, which parameters depend on the biophysical properties of the considered ionic specie. A set of parameters corresponds to a "model card", and describes a specific type of neurons. We chose to design configurable artificial neurons: the model parameters are not fixed, and can vary in a pre-defined range. This range is fixed in a way that the models can represent "prototypical" neurons and synaptic interactions that are frequent in the cerebral cortex.

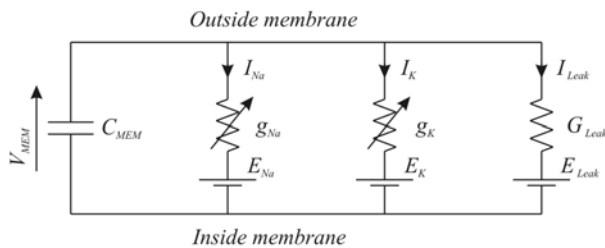


Fig 2 : The electrical circuit representing the Hodgkin-Huxley model

For each neuron (Fig. 3), pre-synaptic events are collected on two synaptic inputs (respectively for inhibitory synapses and for excitatory ones). The events are integrated with their respective synaptic weight to generate the multi-synapse digital control signal. Each of those signals triggers the generation of a synaptic current added to the neuron ionic currents.

A synaptic conductance follows a conductance-based formalism of kinetic synapse [8], with parameters in a range that corresponds to the most classical types of synapses. The digital hardware layer is in charge of generating the synapses control signals. Using this multi-synapses scheme, the neural network can handle all-to-all connections, whatever the number of neurons in the network. A comparator detects a spike and converts the membrane potential  $V_{MEM}$  to a 1-bit digital signal output, directly connected to one FPGA input pad.

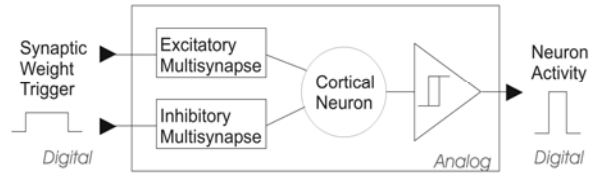


Fig 3 : Digital inputs and output of the analog neuron

#### B. The ASIC architecture

The circuit is organized to provide a large variety of configurations for the simulated neural network. As the neural activity is generated by a sum of ionic and synaptic currents on a membrane capacitance, we decided to integrate a set of generic blocks. Each block is able to compute a conductance-based model of ionic or synaptic current. Parameters of the model are stored on analog memory cells, which values are programmed during the configuration phase of the simulation. During that same phase, the user will also set the topology of the network, i.e. define the blocks connectivity. A set of connected blocks will form an artificial neuron, with their respective currents summed on an external capacitance.

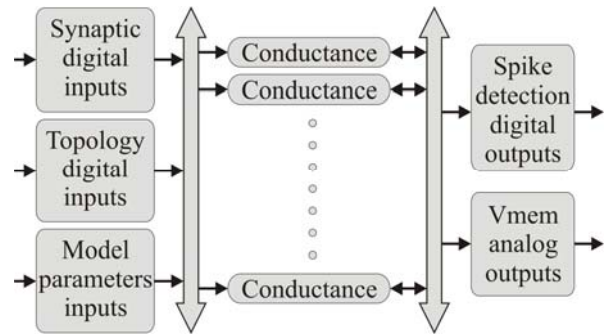


Fig. 4 : The Galway chip structure and data I/O

The Galway chip we present here comprises (Fig. 4):

- a set of conductance modules, each able to generate an ionic or synaptic current following the conductance-based model
- spike-detection modules, to code on 1-bit the neuron membrane voltage
- a set of synaptic input modules, that activate synaptic conductance modules with a digitally-controlled weight

- an analog memory cells array, to store the model parameters
- a matrix of switches, to control the neurons topology (i.e. the arrangement of the conductance and synaptic modules that form the artificial neuron)
- digital functions to control data transfer from and to external devices.

### C. ASIC design

Successive generations of ASICs were designed, that integrate the conductance-based neuron models [9], [10]. These circuits were exploited to build a library of electronics function. Each mathematical function appearing in the neuron model corresponds to a generic analog module in the library. Conductance blocks that compute the ionic currents models are built using these generic modules.

Except for the state machine, synthesized from a VHDL description, all the blocks have been validated using the analog simulator *Spectre* under *Cadence* environment. Due to the important number of components in the chip, it is not possible to perform analog simulation of the final chip on long durations: the neural activity is a low frequency activity, and the neural activity has to be simulated for at least tens of ms. To solve that problem, we simulated the circuit using a mixed description: models are given at the transistor level and using a behavioral description. These simulations validate the blocks' connectivity and ensure the functionality of the whole design.

Each *Galway* chip includes 21 ionic conductance blocks and 10 synaptic conductance blocks. 205 analog parameters are stored in the memory cells. A typical arrangement of the modules is the building of 5 artificial neurons comprising from 3 to 5 ionic conductances; each neuron receives 2 multi-synaptic inputs. Such a structure allows us to address standard configuration of small neural networks.

The chip has been integrated using *austriamicrosystems* BiCMOS 0.35 $\mu$ m technology (Fig. 5); its area is 10.5 mm<sup>2</sup>, it has 105 pads and the number of transistors of the core is 47000, with a ratio of 93% for the analog part.

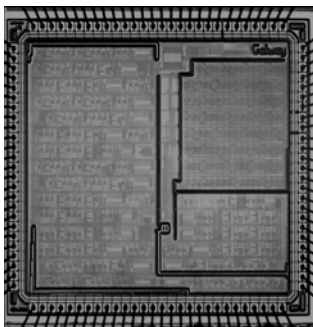


Fig. 5: microphotography of *Galway* chip..

### D. ASIC measurements

On the ASICs, the electrical potentials are 5 times greater than the biological potential they model, and the biological

reference potential (0 V) corresponds to 2.5 V on the chip. The power supply of the analog part of *Galway* is 5 V.

We configured an artificial neuron on *Galway* using the model parameters of a Fast Spiking (FS) neuron. We measured various analog electrical activities (see Fig. 6) on three different  $V_{MEM}$  outputs on a *Galway* chip [11], that fit the desired model.

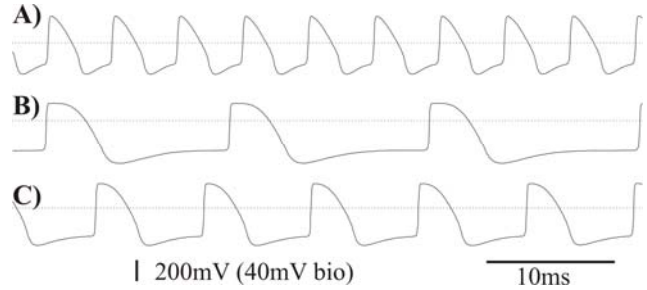


Fig. 6: electrical activity of 3 Fast Spiking (FS) neurons with different model parameters

## IV. Digital hardware

The digital hardware has to perform several tasks as shown in Fig 7 :

- receive the neurons configuration parameters from the user via the software layer and send them to the corresponding ASICs.
- map the topology of the neural network (connectivity and plasticity).
- send back to the user the neural state information (time stamped spike events, synaptic weights evolution).
- receive spike events, send the synaptic input signal to post-synaptic neurons
- compute in real-time STDP algorithms and update synaptic weight.

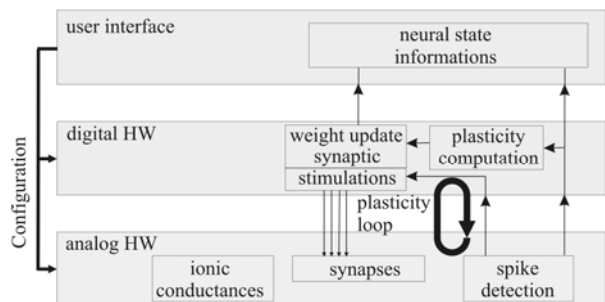


Fig 7 : Distribution of computational tasks in the system

The timing of these tasks must respect the biological real-time; using FPGAs allows to run parallel processing of different tasks. Each function is described in VHDL and synthesized into a *Xilinx Spartan3*<sup>TM</sup> device using *ISE*<sup>TM</sup> software from Xilinx Inc. .

## A. Neurons configuration

Each *Galway* chip includes up to five neurons. A matrix of switches has to be configured to control the topology of each neuron. We send 3 words of 14 bits each to the dedicated serial input of *Galway* as shown by the chronographs of Fig. 8. Note that the clock signal is only activated when needed, to minimize parasitic coupling.

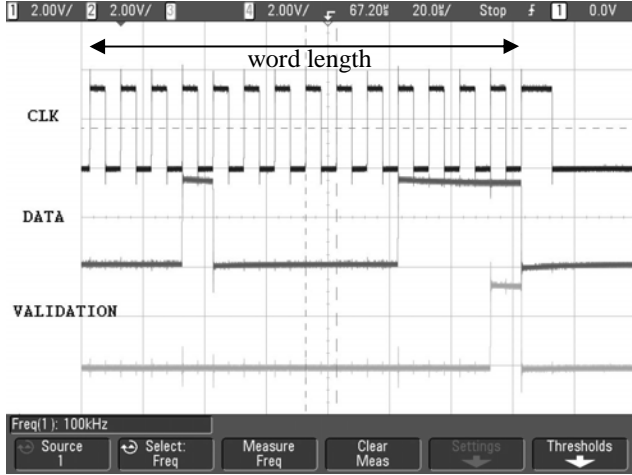


Fig. 8: Timing of the inputs for topology configuration

To emulate a neuron activity, each neuron needs a set of parameters, encoded as voltage values, that are stored in analog memory cells inside *Galway*. The values of the original parameters are coded on 14 bits, stored in a RAM synthesized into the FPGA and sent via a serial DAC to the ASIC. For 5 neurons, i.e. for one chip, we have to refresh 205 analog parameters every 2 ms. Fig. 9 displays the chronograms of the 4 signals input to the ASIC to control the memory cells.

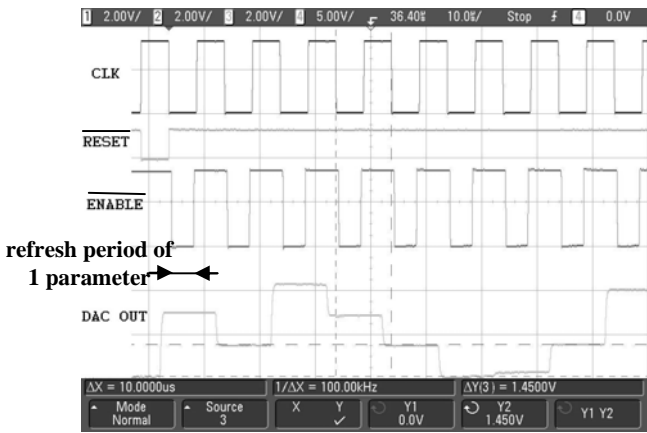


Fig. 9: Chronograms of the ASIC inputs for the memory cells update

## B. Network configuration & STDP computation

As we will explain in section V, our system in its current version supports a maximum of 25 neurons that can be all-to-all connected. To configure the network, we have to decide which pre-synaptic neuron is connected to which post-synaptic one and if this connection is adaptive (“plastic”) or not. To do this, we have defined two matrices. Both have 25 rows corresponding to the number of pre-synaptic neurons and 25 columns corresponding to the number of post-synaptic neurons. The first matrix called Plasticity [P] indicates if an individual synapse from pre-synaptic neuron  $N_j$  to post-synaptic neuron  $N_i$  follows a plasticity rule (“1”) or not (“0”). The second matrix, called Weight [W], gives the value (row  $j$ , column  $i$ ) of the weight  $W_{ji}$  of the synaptic connection from pre-synaptic neuron  $N_j$  to post-synaptic neuron  $N_i$ . It is coded on 10 bits (Fig. 10). If the values (row  $j$ , column  $i$ ) of [P] and [W] are 0, it means that the connection is not adaptive and the weight is null, so  $N_j$  is not connected to  $N_i$ . In the case of a non-adaptive connection between  $N_j$  and  $N_i$ , the initial weight  $W_{ji}$  will remain constant. Otherwise, the weight will be updated over time according to the STDP rule that we detail below [12].

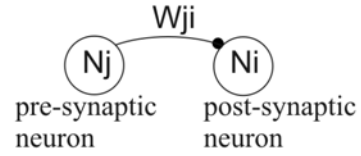


Fig. 10: Synaptic connection between pre synaptic neuron  $N_j$  and post synaptic neuron  $N_i$

Spike timing dependent plasticity (STDP) is believed to be the most important phenomenon in the remodeling of neural circuits in human brain. This plasticity depends on the relative timing of pre-synaptic and post-synaptic spikes.

Using a single connection from  $N_j$  to  $N_i$ , the simplest STDP algorithm describes 2 phenomena: a long term potentiation (LTP) noted  $P(t)$  and a long term depression (LTD) noted  $Q(t)$  according to eq.1 where  $t_j^{last}(t)$  and  $t_i^{last}(t)$  give the last spikes occurrences at time  $t$  in neurons  $N_j$  and  $N_i$ .

$$\frac{dw_{ji}}{dt} = \sum_{t_i} P[(t - t_j^{last}(t))\delta(t - t_i)] - \sum_{t_j} Q[(t - t_i^{last}(t))\delta(t - t_j)] \quad (1)$$

Potentiation means that the weight  $W_{ji}$  increases when the post-synaptic neuron  $N_i$  spikes at time  $t_i$  after a spike of the pre-synaptic neuron  $N_j$  at time  $t_j$ . Depression means that the weight  $W_{ji}$  decreases when the pre-synaptic neuron  $N_j$  spikes at time  $t_j$  after a spike of the post-synaptic neuron  $N_i$  at time  $t_i$ . LTP and LTD are illustrated in Fig.11

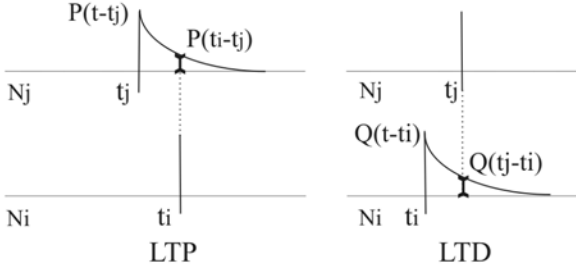


Fig. 11: Long-term potentiation and long-term depression evaluation, depending on the occurrence of events on Ni and Nj

The functions are defined and parameterized to fit experimental results on real neural networks [6].  $P(t)$  and  $Q(t)$  are exponential functions like in eq.2:

$$P(t) = \exp(-t/\tau_p) \text{ and } Q(t) = \exp(-t/\tau_q) \quad (2)$$

Using this model, the relative change of synaptic weight is shown in Fig. 12, where  $(t_i - t_j)$  represent the time difference between an event on the post-synaptic neuron Ni and the pre-synaptic neuron Nj.

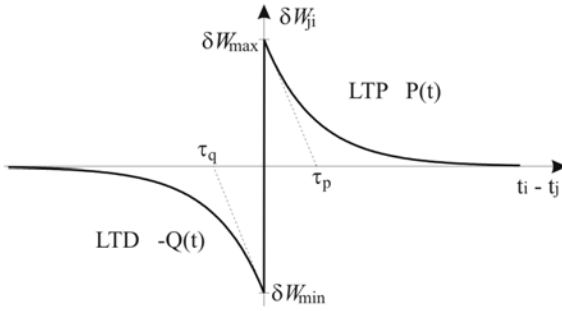


Fig. 12: Relative change in synaptic strength

We can take into account two more effects on STDP: the memory effect [13] and the saturation effect. The resulting expression is (eq 3).

$$\frac{dw_{ji}}{dt} = \epsilon_i \epsilon_j \left\{ (w_{LTP} - w_{ji}) \sum_{t_i} P[(t - t_j^{last}(t))] \delta(t - t_i) - (w_{ji} - w_{LTD}) \sum_{t_j} Q[(t - t_i^{last}(t))] \delta(t - t_j) \right\} \quad (3)$$

The  $\epsilon$  term, called spike efficacy, indicates that the change of synaptic weight  $W_{ji}$  is less important for a second occurrence of pre-synaptic spike at time  $t_j^{last}$  than for the first occurrence at time  $t_j^{last-1}$ . This memory effect can be modelled for Nj and Ni by eq. 4 and eq. 5.

$$\epsilon_j = 1 - \exp(-(t_j^{last} - t_j^{last-1})/\tau_{pre}) \quad (4)$$

$$\epsilon_i = 1 - \exp(-(t_i^{last} - t_i^{last-1})/\tau_{post}) \quad (5)$$

In the previous model, there is no saturation effect on  $W_{ij}$ , which is not really biologically-realistic. To address this

problem, we can add in the STDP algorithms a maximum and a minimum saturating limit, called “soft-bound” values, respectively  $W_{LTP}$  and  $W_{LTD}$ .  $W_{ji}$  will be limited by  $W_{LTD} < W_{ji} < W_{LTP}$ .

The key point of the STDP computation is the calculation of exponential functions with time constant around tens of ms. According to biological experiments, typical models use the following values for time constants:  $\tau_p = 14.8$  ms,  $\tau_q = 33.8$  ms,  $\tau_{pre} = 28$  ms and  $\tau_{post} = 88$  ms. To optimize the FPGA computation of these functions, we propose to design a function that gives in real time the value of the exponential function every  $\Delta t$  using a first order approximation. For  $\Delta t \ll \tau$ , we have:

$$\exp\left(-\frac{t + \Delta t}{\tau}\right) = \exp\left(-\frac{t}{\tau}\right) \exp\left(-\frac{\Delta t}{\tau}\right) \approx \exp\left(-\frac{t}{\tau}\right) \left(1 - \frac{\Delta t}{\tau}\right)$$

Implementing division in FPGA is rather simple when the operand is equal to  $2^n$  with n integer; it is equivalent to a shift of n bits from MSB to LSB. So, we chose to calculate the evolution of the exponential function using 20 bits and to take  $n=10$ . In this case,  $\Delta t = 2^{-n} \cdot \tau = \tau/1024$  and the approximation is right. It leads to the schematic of Fig. 13, where  $f(0)$  is the initial value validated by *init* signal.

The implementation of eq 2 occupies 126 slices of the FPGA.

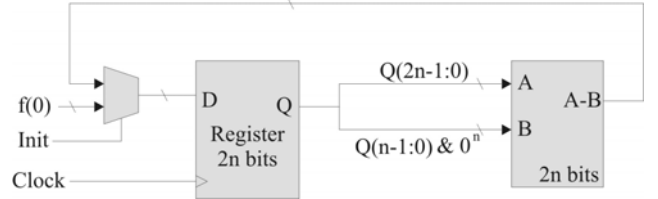


Fig 13 : Structural implementation of exponential calculation into FPGA (& is used as concatenation symbol)

## V. Simulation system

As shown in Fig 14, the simulation system is composed by the *Gaillimh* board connected to a PC via a serial RS232 interface. On the PC, we have developed a friendly Human Machine Interface (HMI) for the user to define the configuration of both neurons and network. The configuration is initially sent to the *Gaillimh* board and can be modified at any time using on-line commands via the HMI or a hyper terminal.

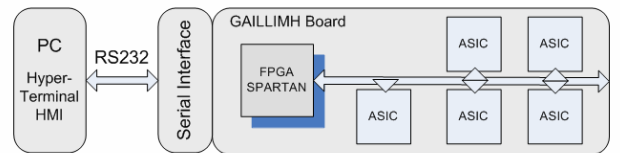


Fig. 14: Structure of the neural network simulation system

The *Gaillimh* board (Fig. 15) is a 6 layers full-custom boards. It hosts 5 *Galway* ASICs for a total of 25 neurons which digital synaptic inputs and digital spike detection

outputs are individually connected to the FPGA (Xilinx Spartan3 XC3S1500FG456). The supply voltage is 5 V and all the other power supplies are regulated on board. The FPGA has an external clock at 100 Mhz; it is associated with two SDRAM (256 Mbits each) clocked at 50 MHz and can be programmed by JTAG.

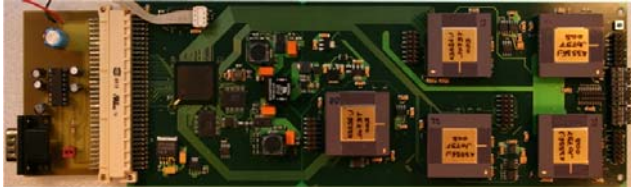


Fig. 15: Photograph of the *Gaillimh* Board

All the functions performed by the FPGA and written in VHDL have been individually implemented, tested and validated on the *Gaillimh* board.

We have to manage the plasticity computation when the 25 neurons network is fully active and all to all connections are activated. When a spike occurs at neuron  $N$ , we have to launch a series of tasks considering this neuron  $N$  both as a pre-synaptic neuron for up to 25 post-synaptic ones and as a post-synaptic neuron for up to 25 pre-synaptic ones. In the first case, we need to activate up to 25 synapse inputs, start the real-time calculation of  $P(t)$  and calculate the values of up to 25  $\epsilon_j$  and  $Q(t)$  terms for updating synaptic weights. In the second case, we need to start the real-time computation of  $Q(t)$  and calculate the values of up to 25  $\epsilon_i$  and  $P(t)$  terms for updating synaptic weights.

Entirely parallel computation as a task solution for a fast computation of these calculations is not realistic, as it would necessitate to duplicate the computational blocks for  $25 \times 25 = 625$  connections. A solution consists in combining replication and time division multiple access (TDMA) to the computational blocks. For instance, the exponential block is clocked at a frequency dependent of the time-constant. If we consider a value  $\tau = 20$  ms, associated with the FPGA clock period (20 ns), the calculation of the exponential value has to be done every  $\Delta t = \tau \cdot 2^{-n} = \tau / 1024 = 19,5 \mu\text{s}$  which is equivalent to 976 FPGA clock periods. So, this same computational block, coupled with SDRAM downloading and uploading operations, will be also available for other exponential calculation tasks. We are now working on optimizing the ratio of duplication and TDMA.

## VI. Conclusion

In this paper, we have presented a second-generation [14] analog/digital neural network simulator. The equations describing the Hodgkin-Huxley model of the neurons and synapses are integrated into a BiCMOS ASIC (*Galway*) and are computed in continuous time within the so-called analog hardware layer. The neural network is built at the level of the digital hardware layer where an FPGA manages in real-time

the all to all connectivity and the plasticity algorithms according classical STDP rules. This quasi-standalone network is currently limited to 25 neurons. Nevertheless, the *Gaillimh* board is the prototype of a more extended system dimensioned for 512 neurons. This system will host 21 daughter boards *Gaillimh* connected to a common bus and a mother board *Thalamos* for the management of the communication of all the daughter boards.

## REFERENCES

- [1] R. Hahnloser, R. Sarpeshkar, M. Mahowald, R. Douglas, S. Seung, "Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex", *Nature*, vol.405, pp. 947-951, 2000.
- [2] J. Schemmel, S. Hoffman, K. Meier, F. Schurman, "A mixed-mode analog neural network using current-steering synapse", *Analog Integrated Circuits and Signal Processing*, vol.38, pp.233-244, 2004.
- [3] S. Renaud-Le Masson, A. Laflaquière, D. Dupeyron, T. Bal, G. Le Masson, "Analog circuits for modeling biological neural networks: design and applications", *IEEE Transactions on Biomedical Engineering*, vol. 46-6, pp. 638-645, 1999.
- [4] V. Douence, A. Laflaquiere, S. Le Masson, T. Bal, G. Le Masson, "Analog electronic system for simulating biological neurons", *IWANN'99*, Alicante, Spain, June 1999.
- [5] G. Le Masson, S. Renaud, D. Debay and T. Bal, "Feedback inhibition controls spike transfer in hybrid thalamic circuits", *Nature*, vol. 417, pp. 854-858, 2002.
- [6] A. Destexhe, E. Marder, "Plasticity in single neuron and circuit computations", *Nature Review Neuroscience*, vol. 431, pp.789-795, 2004.
- [7] C. Koch, I. Segev, *Methods in neuronal modeling: from synapses to networks*, MIT Press, Cambridge, 1989.
- [8] A. Destexhe, Z.F. Mainen, T.J. Sejnowski, "An efficient method for computing synaptic conductances based on a kinetic model of receptors binding", *Neural Computation*, vol. 6, pp. 10-14, 1994.
- [9] L. Alvado, J. Tomas, S. Saighi, S. Renaud, T. Bal, A. Destexhe, G. Le Masson, "Hardware computation of conductance-based neuron models", *Neurocomputing*, Vol. 58-60, pp. 109-115, 2004
- [10] Y. Bornat, J. Tomas, S. Saighi, S. Renaud, "BiCMOS Analog Integrated Circuits for Embedded Spiking Neural Networks", *XX Conference on Design of Circuits and Integrated Systems (DCIS)*, Lisbon, Portugal, 2005.
- [11] B. Connors, M. Gutnick, "Intrinsic firing patterns of diverse neocortical neurons", *Trends Neuroscience*, vol.13, pp. 99-104, 1990.
- [12] W. Gerstner and W. Kistler, "Spiking Neuron Models: Single Neurons, Populations, Plasticity." *United Kingdom, Cambridge University Press*, 2002.
- [13] R.C. Froemke., Y. Dan "Spike timing-dependent synaptic modification induced by natural spike trains". *Nature*, vol. 416, pp. 433-438, 2002.
- [14] Q. Zou, Y. Bornat, J. Tomas, S. Renaud, A. Destexhe, « Real-time simulations of networks of Hodgkin-Huxley neurons using analog circuits », *Neurocomputing*, Vol. 69, pp. 1137-1140, 2006..