



**HAL**  
open science

## Low power and massively parallel simulation of oscillatory biochemical networks on FPGA

Serge Le Thanh, Nicolas Lobato-Dauzier, Farad Khoystatee, Romain Beauvois, Teruo Fujii, Anthony J Genot, Timothée Levi

► **To cite this version:**

Serge Le Thanh, Nicolas Lobato-Dauzier, Farad Khoystatee, Romain Beauvois, Teruo Fujii, et al.. Low power and massively parallel simulation of oscillatory biochemical networks on FPGA. 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Oct 2019, Nara, Japan. pp.1-4, 10.1109/BIO-CAS.2019.8919020 . hal-02406013

**HAL Id: hal-02406013**

**<https://hal.science/hal-02406013v1>**

Submitted on 12 Dec 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Low power and massively parallel simulation of oscillatory biochemical networks on FPGA

Serge Le Thanh, Serge Le Thanh, Nicolas Lobato-Dauzier, Farad Khoystatee, Romain Beaubois, Teruo Fujii, Anthony Genot, Timothée Levi

### ► To cite this version:

Serge Le Thanh, Serge Le Thanh, Nicolas Lobato-Dauzier, Farad Khoystatee, Romain Beaubois, et al.. Low power and massively parallel simulation of oscillatory biochemical networks on FPGA. 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Oct 2019, Nara, Japan. pp.1-4, 10.1109/BIOCAS.2019.8919020 . hal-02406013

**HAL Id: hal-02406013**

**<https://hal.archives-ouvertes.fr/hal-02406013>**

Submitted on 12 Dec 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Low power and massively parallel simulation of oscillatory biochemical networks on FPGA

Serge Le Thanh<sup>1,2</sup>, Nicolas Lobato-Dauzier<sup>1,4</sup>, Farad Khoystatee<sup>3</sup>, Romain Beaubois<sup>1,3</sup>, Teruo Fujii<sup>1,4</sup>, Anthony Genot<sup>1,4</sup>, Timothée Levi<sup>1,3,4</sup>

Email: {lobato, tfujii, genot, levi}@iis.u-tokyo.ac.jp {farad.khoystatee, timothee.levi}@u-bordeaux.fr

<sup>1</sup>Institute of Industrial Science (IIS), the University of Tokyo, Tokyo, Japan

<sup>2</sup>GEII department, IUT de Bordeaux, University of Bordeaux, Talence, France

<sup>3</sup>IMS Laboratory, UMR5218, University of Bordeaux, Bordeaux INP, Talence, France

<sup>4</sup>LIMMS, CNRS-Institute of Industrial Science, UMI 2820, University of Tokyo, Tokyo, Japan

**Abstract**— Biological functions emerge from a multitude of chemical species woven into intricate biochemical networks. It is crucial to compute the dynamics of a biochemical network from its kinetics and topology. In order to reverse engineer networks and map their design space, dynamics needs to be simulated for many different parameters and topologies, leading to a combinatorial explosion that requires heavy computational power. To solve this issue, we show here an application of FPGA platform to simulate biochemical networks. As a toy model, we simulate a structurally simple network with a rich oscillatory dynamics: a predator-prey biochemical oscillators. The network mimics predator-prey dynamics. We show that FPGA can simulate the dynamics of PP faithfully. These results open the door to more energy-efficient simulations.

**Keywords**—Biochemical network, Prey-predator, FPGA

## I. INTRODUCTION

Biological functions emerge from a multitude of chemical species woven into intricate biochemical networks. For instance, signaling networks transmit and process information about the cellular environment through cascades of phosphorylation [1]. In gene regulatory networks, genes mutually influence their production rates, creating nonlinear and cyclical layers of interactions. Gene regulatory networks are the engines of most biological process - ranging from embryogenesis and immune response to tumor formation and wound healing [2-3].

Oscillation is a signature dynamic found in many biochemical networks and is best epitomized by the circadian clock - a light-entrained clock that cadences numerous biological activities. The circadian clock has been found, in various guises, across numerous species. And a ring-oscillator, the repressilator, is one of the first dynamics demonstrated by synthetic biologists [4].

It is crucial to compute the dynamics of a network from its kinetics and topology - both for system biologists who dissect natural networks, and synthetic biologists who build artificial ones. To that end, an extensive battery of mathematical models has been published, ranging from coarse-grained Boolean networks (first published in the 60s [5]) to modern, fine-grained and whole-cell simulations based on ODE and

PDE [6]. In order to reverse engineer networks and map their design space, dynamics needs to be simulated for many different parameters and topologies, leading to a combinatorial explosion that requires heavy computational power [7].

However conventional CPUs are ill-equipped to massively and efficiently simulate biochemical networks. Their architecture follows the Von Neuman paradigm of computation, where CPU and memory are physically separated, and data continuously shuttled between them. In modern CPUs, the majority of the transistor budget is spent on caching data to hide memory latencies, on executing complex flows of instruction, or supporting double-precision arithmetic (64 bits). Those features are essential for database management and servers (which are usually bound by memory) or for high performance computing (e.g. chaotic systems like weather are utterly sensitive to initial conditions and must be simulated with very high precision). But these overheads cost energy, CPU cycles, and are not absolutely essential to simulate biochemical networks. First, double precision arithmetic offers  $\sim 16$  significant digits, which exceed, by large, the typical statistical fluctuations of biomolecules. In a cell, most RNAs are typically expressed at less than  $N=$ one million copies, which gives a relative statistical fluctuation ( $1/\sqrt{N}$ ) larger than 0.1%, or 3 digits. Secondly, biochemical networks are massively parallel by nature, while CPU are designed to handle sequential flows of instructions. Ironically, a CPU consumes  $\sim 10$  Joule to simulate a simple oscillatory network with 2 species ( $\sim 10$  W for  $\sim 1$  s of wall-clock time simulation on a laptop), while a cell only consumes  $\sim 100$  nano-joule (1 pW/cell for 24 hours) to run its whole biochemical machinery, which includes the circadian clock. To reduce the power consumption and make parallel computation, the choice of parallel hardware platforms like FPGA or GPU is relevant [8].

The FPGA platform is an attractive alternative to CPU to simulate biomimetic dynamics like neural network [9-13] and biochemical networks [14-16]. By coding directly the resolution of the ODE directly at the level of transistors, it opens the door for massive parallelization and drastically reduces power consumption. Tuning of parameters is also

easy with FPGA and can work in accelerated time performing several simulations in parallel [17]. Compared to GPU – another alternative for massively parallel computing - FPGA boasts a reduced power consumption (~1 W vs 300 W).

Here we investigated FPGA for simulating biochemical reaction networks. As a toy model, we simulated a structurally simple network with a rich oscillatory dynamics: a predator-prey (PP) biochemical oscillators (Fig. 1). This network mimics predator-prey dynamics (Fig. 1C), and was extensively characterized experimentally [18-19]. We first show that FPGA can simulate the dynamics of PP faithfully. In the future, we will use the power of FPGA to compute the bifurcation diagram of the PP. These results will allow more energy-efficient simulations.

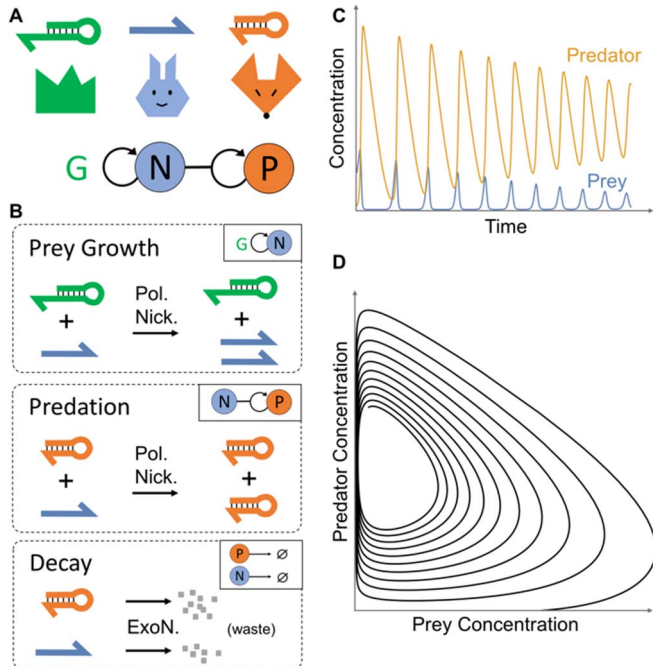


Fig. 1. DNA-based predator-prey system. (A) General design of the predator-prey biochemical network. (B) DNA polymerization–depolymerization reactions constituting the network. (C) Typical dynamical behavior of the system. (D) Typical 2D trajectory in the concentration space.

## II. PREY PREDATOR (PP) NETWORK

The prey consists of a short DNA strand that replicates thanks to another DNA strand, the grass template (Fig. 1A). Briefly, when a prey strand binds to the template strand, it is elongated by a polymerase and then nicked by a nickase, releasing two prey strands and leading to exponential amplification of the prey (Fig. 1B). In the predation reaction, the predator strand binds to the prey strand, and is elongated by the polymerase, turning the prey into a predator. Predator and prey strands are continuously degraded by a specialized enzyme, an exonuclease. This biochemical network has been tried and tested *in vitro*, and exhibited sustained oscillations (Fig. 1C) for the right balance of replication, predation and degradation – which can simply be tuned by adjusting the concentration of template strand and enzymes. The experimental dynamics are well described mathematically by a modified Lotka–Volterra equation that accounts for enzymatic kinetics [18]. Description of the equations is included in [18] supplementary material section 3.

The model with adimensionalized concentrations and time is as follows:

$$\frac{dn}{dt} = \frac{gn}{1+\beta gn} - pn - \lambda\delta \frac{n}{1+p} \quad (1)$$

$$\frac{dp}{dt} = pn - \delta \frac{p}{1+p} \quad (2)$$

The first term of the first line models self-replication of the prey  $n$ , the second term models consumption of the prey  $n$  by the predator  $p$ , and the last term models degradation of the prey  $n$  by the exonuclease (and similarly for the terms of the second line). The role and ranges of the parameters are detailed in Table I. Compared to a classical Lotka–Volterra equation, replication of preys and degradation of predator plateau at high concentrations, which reflects the saturation of the enzymatic process.

TABLE I. NONDIMENSIONAL PARAMETERS OF THE MODEL FOR STANDARD CONDITIONS

| Parameters                                                    | Adimensional |
|---------------------------------------------------------------|--------------|
| $\beta$ (binding affinity of the grass template for the prey) | [0.01 ; 0.2] |
| $\lambda$ (ratio of degradation rates of prey and predator)   | [0 ; 8]      |
| $\delta$ (degradation rate of predators)                      | [0 ; 1]      |
| $g$ (concentration of grass template)                         | [0 ; 4]      |
| $n_0$ (initial concentration of prey)                         | [0 ; 1]      |
| $p_0$ (initial concentration of predator)                     | [0 ; 1]      |

## III. FPGA IMPLEMENTATION

### A. Methodology

To validate our simulations, we first implemented the modeling with Matlab software. Then we compared our FPGA data to Matlab simulations.

For VHDL code, we used fixed-point numbers. We chose 4 bits for integer part and 32 bits for decimal part. These numbers have been chosen to fit the simulation data to keep high resolution and stability.

Basic operations such as multiplication and division required specific VHDL module. The multiplication can be performed by Digital Signal Processing (DSP) into the FPGA. As DSP are on 18 bits, we just need 4 of them per multiplication. On the other hand, the non-restoring algorithm is used as a simple and efficient way to compute the division [20].

The differential equations can be solved using the Euler method which defines derivative function. The value of  $dt$  is chosen to be digitally convenient, to have stable dynamics and to stay in the acceptable range of living variability. By choosing a power of two as  $dt$ , it is possible to replace the multiplier by a logical shift operation. In our application,  $2^{-6}$  for  $dt$  was used.

To reduce data size, we extract the main features of the oscillations for bifurcation diagram analysis. The features of the oscillation peaks (time stamp and amplitude of maximum and surrounding minima) are sent to PC via serial communication. To send all the results of simulation, especially for bifurcation diagram of the PP, we use serial communication from FPGA board to CPU to send extracted

features of the oscillations (maximum, minimum and timing). The whole curve is sent to the scope through a 12 bit DAC.

### B. Implementation

To simplify implementation and avoid division, we transform equation (1) to equation (3).

$$\frac{dn}{d\tau} = \frac{1}{\beta} \frac{n}{\frac{1}{\beta} + n} - p \cdot n - \lambda \cdot \delta \frac{n}{1+p} \quad (3)$$

Figure 4 describes the implementation of the (n,p) dynamic system with pipelined division, DSP and other computation.

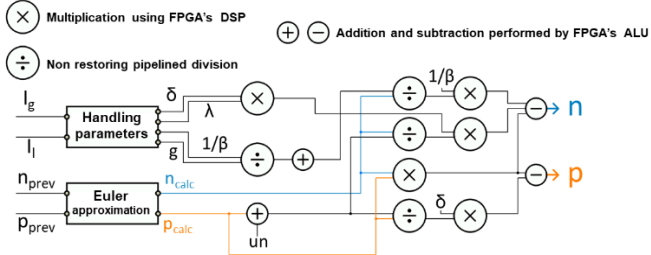


Fig. 4. Description of the computation of the differential equations of the PP system (equation 3).  $I_g$  and  $I_l$  are parameters for tuning  $\lambda$ ,  $\delta$  and  $\beta$ ,  $g$  parameters. After each simulation,  $I_g$  et  $I_l$  are incremented for processing automatic multiple simulations.

The FPGA platform is a Kintex-7 of “xc7k325tffg900” family included in the Genesys 2 evaluation board.

Figure 3 describes the schematic block of the different modules implemented in the FPGA.

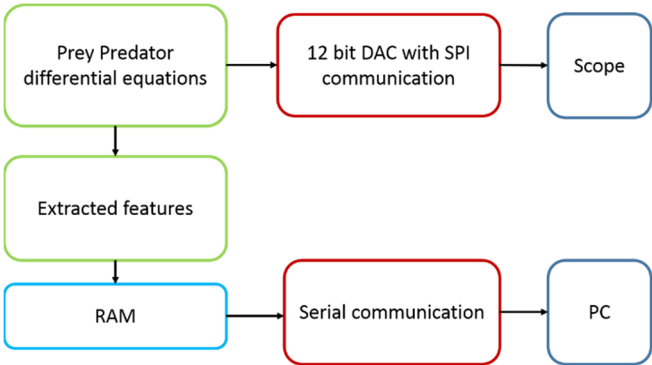


Fig. 3. Description of the different modules implemented in FPGA. Green modules are implementing differential equations and detection of the peaks of the oscillations. Red modules are used for communication with the scope and PC.

This FPGA hardware implementation is low-power, requiring only 0.844W (dynamic and static) - compared to ~300W for CPUs and GPUs. One complete simulation takes 7.5ms - which gives an energy consumption of 6 mJ (3 orders of magnitudes smaller than with CPU).

Table II. FPGA implementation resources

|      | PP implementation | Full resources | Percentage |
|------|-------------------|----------------|------------|
| LUT  | 18799             | 203800         | 9.22       |
| FF   | 8444              | 407600         | 2.07       |
| DSP  | 16                | 840            | 1.90       |
| BRAM | 2                 | 445            | 0.45       |

BRAM are necessary to stock the extracted features of the simulation as serial communication is slow.

As we used less than 10% of the resources, we can implement 10 PP cores in parallel and then decouple the number of simulations.

### C. Results

We first validated the PP dynamics by reproducing oscillations in the concentrations of  $p$  and  $n$ . We found that Matlab and VHDL simulations fit each other satisfyingly (Figure 4), though clearly considerations must be given to the number of bit for fixed-point implementation and the time step  $dt$  for the Euler method. As for figure 5, only VHDL simulations are shown for the sake of readability.

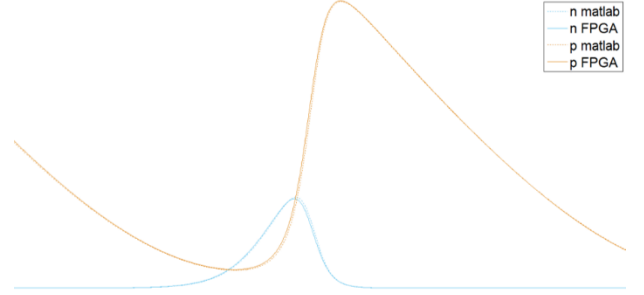


Fig. 4. Comparison between Matlab simulation and VHDL simulation. Set parameter is :  $\beta=0,087$ ;  $\lambda=4,5$ ;  $\delta=0,39$ ;  $g=0,40$ ;  $n_0=0,5$ ;  $p_0=0,5$ ;

Figure 5 describes VHDL simulations using two sets of parameters describing typical dynamics of the PP system and 2D trajectories in the concentration space.

Compared to *in vitro* experiments, FPGA simulations speed up the dynamics by 7 orders of magnitudes (40 hours for the whole experiment in *in vitro*, against 7.5ms in FPGA). The objective of the biochemistry is to vary parameters ( $g$ ,  $\beta$ ,  $\lambda$ ...) to study the behavior of the system and to explore the different dynamics.

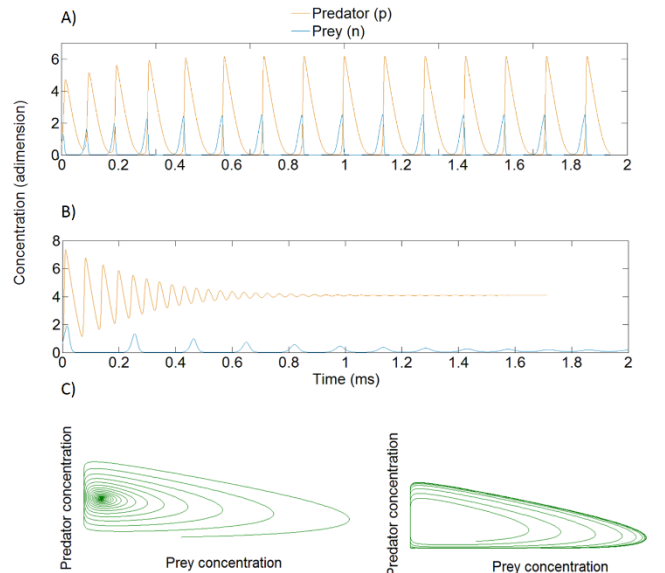


Fig. 5. FPGA simulations of the PP dynamics. A) typical dynamic of oscillations with this set of parameters:  $\beta=0,087$ ;  $\lambda=4,5$ ;  $\delta=0,39$ ;  $g=0,40$ ;  $n_0=0,5$ ;  $p_0=0,5$ ; B) another typical dynamic of oscillations with this set of parameters:  $\beta=0,087$ ;  $\lambda=6$ ;  $\delta=0,39$ ;  $g=3,80$ ;  $n_0=0,5$ ;  $p_0=0,5$ ; C) Left part: 2D trajectory in the concentration space of A) set of parameters, Right part: 2D trajectory in the concentration space of B) set of parameters. FPGA frequency is set to 100 MHz.



In order to output the signals on scope, a Digital to Analog Converter (DAC) with a precision of 12 bits with SPI communication and a 2.5V reference voltage converts the digital data into analog curves. The frequency of this implemented system is 25MHz. Figure 6 describes the scope output of the two typical dynamics.

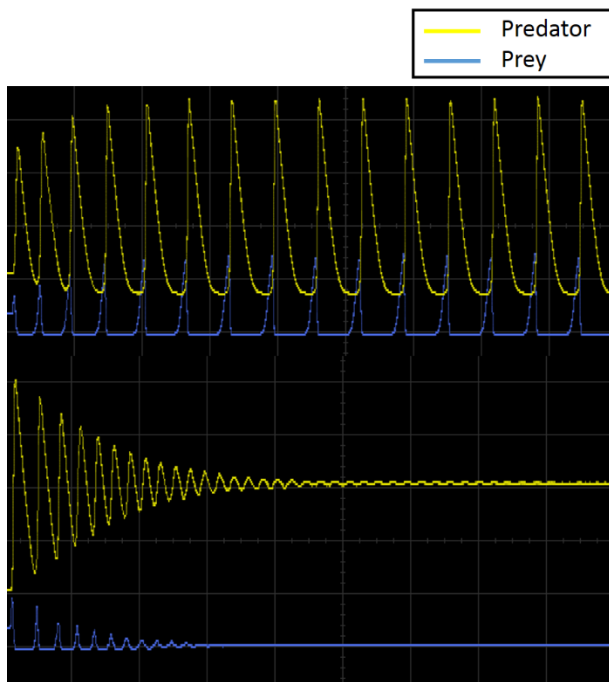


Fig. 6. Output of scope after 12 bit DAC. On the top, PP dynamic using Fig.5 A) parameters. On the bottom, PP dynamic using Fig.5 B) parameters. X axis is 5ms by division. Y axis is 800mV by division. FPGA frequency is set to 25MHz.

After validating individual simulation, we implemented parallel simulation (3200) varying parameter lambda and g by a step of 0.1 in their interval in standard condition (Table I). To reduce amount of data, the extracted features of the oscillation dynamic were sent to the computer. As one simulation took 7.5ms at 100 MHz and this implementation allows 10 parallel simulations, it needs 2.4s to execute the 3200 simulations.

#### IV. CONCLUSION AND PERSPECTIVES

We simulated ODEs of oscillatory biochemical network on FPGA. Results were not distinguishable from CPU-based Matlab resolution of the same ODE. The FPGA efficiently simulated biochemical networks in terms of speed and low-power consumption. Scope outputs proved that the implementation is correctly running. We expect that this hardware implementation will assist the forward and reverse engineering of biochemical networks in cells.

Firstly this hardware system can assist system biologists who, given an observed phenotype, search for the topology and parameters that fit their observation (reverse engineering).

Secondly, this system will contribute in finding genetic networks that perform a given function (e.g. regulating the level of glucose in the blood). It is notoriously difficult, because unlike artificial neural networks, the output of a genetic networks does not have a closed analytical form in general. In that case the classical training methods like backpropagation break down, and brute-force methods like derivative-free training are required. In both case, FPGA-

based simulations could help them to explore the design space in a massively parallel way [21].

#### ACKNOWLEDGMENT

We would like to thank the MITI of CNRS (project DNA2 in Biomimetisme 2019).

#### REFERENCES

- [1] L. Chang and M. Karin. "Mammalian MAP kinase signalling cascades." *Nature*, 410:6824, 37-40, 2001
- [2] H. Busch et al. "Gene network dynamics controlling keratinocyte migration," *Molecular systems biology* 4:199, 2008
- [3] S. Brenner et al. "Genes and development: molecular and logical themes." *Genetics* 126:3, 1990
- [4] M. Elowitz and S. Leibler. "A synthetic oscillatory network of transcriptional regulators." *Nature*, 403, 335-338, 2000
- [5] S. Kauffman, "Homeostasis and differentiation in random genetic control networks." *Nature*, 224, 177-178, 1969
- [6] M. Tomita, "Whole-cell simulation: a grand challenge of the 21st century." *Trends in Biotechnology*, 19:6, 205-210, 2001
- [7] J. Hastay et al. "Computational studies of gene regulatory networks: in numero molecular biology.", *Nature Reviews Genetics*, 2:4, 268-279, 2001
- [8] S. Van Albada et al., "Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model", *Front. Neurosci.*, 2018
- [9] R. Wang, G. Cohen, K. Stiefel, T. Hamilton, J. Tapson, and A. Van Schaik, A., "An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation", *Frontiers in Neuroscience*, 7:1-14, 2013
- [10] P. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface", *Science*, 345:6197, 668-673, 2014
- [11] M. Ambroise, T. Levi, S. Joucla, B. Yvert and S. Saïghi, "Real-time biomimetic Central Pattern Generators in an FPGA for hybrid experiments", *Frontiers in Neurosciences*, 7:215, 2013
- [12] T. Nanami and T. Kohno, "Simple Cortical and Thalamic Neuron Models for Digital Arithmetic Circuit Implementation", *Frontiers on Neuroscience*, 10:181, 2016
- [13] F. Grassia, T. Kohno, T. Levi, "Digital hardware implementation of a stochastic two-dimensional neuron model", *Journal of Physiology-Paris*, 2017
- [14] I. Karon et al., "FPGA implementation of the Predator-Prey algorithm with adrenalin boost based on a Spiking Neural Network", *Proceedings of the 18th IEEE International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2011, Gliwice, Poland*, 2011
- [15] H. Amano et al., "Performance Evaluation of an Fpga-Based Biochemical Simulator ReCSip," *Proceedings of IEEE International Conference on Field Programmable Logic and Applications, Madrid, Spain*, 2006
- [16] M. Yoshimi et al., "Practical implementation of a network-based stochastic biochemical simulation system on an FPGA," *Proceedings of IEEE International Conference on Field Programmable Logic and Applications, Heidelberg, Germany*, 2008
- [17] S. Schmitt et al., "Neuromorphic Hardware In The Loop: Training a Deep Spiking Network on the BrainScaleS Wafer-Scale System," *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2227-2234, Anchorage, USA, 2017
- [18] T. Fujii and Y. Rondelez. "Predator-prey molecular ecosystems." *ACS nano*, 7:1, 27-34, 2012
- [19] A. J. Genot et al. "High-resolution mapping of bifurcations in nonlinear biochemical circuits." *Nature chemistry*, 8:8, 760-767, 2016
- [20] F. Khoiratee, F. Grassia, S. Saighi, T. Levi, "Optimized Real-Time Biomimetic Neural Network on FPGA for Bio-hybridization", *Frontiers on Neuroscience*, 13:377, 2019
- [21] T. Levi, T. Nanami, A. Tange, K. Aihara, T. Kohno, "Development and applications of biomimetic neuronal networks toward brainmorphic artificial intelligence", *IEEE Transactions on Circuits and Systems II*, 65-5, 577-581, 2019