HAL
open science

A Library of Analog Operators Based on the Hodgkin-Huxley Formalism for the Design of Tunable, Real-Time, Silicon Neurons

Sylvain Saïghi, Yannick Bornat, Jean Tomas, Gwendal Le Masson, Sylvie Renaud

To cite this version:

Sylvain Saïghi, Yannick Bornat, Jean Tomas, Gwendal Le Masson, Sylvie Renaud. A Library of Analog Operators Based on the Hodgkin-Huxley Formalism for the Design of Tunable, Real-Time, Silicon Neurons. IEEE Transactions on Biomedical Circuits and Systems, 2011, 5 (1), pp.3-19. 10.1109/TBCAS.2010.2078816 . hal-00562021

HAL Id: hal-00562021

https://hal.science/hal-00562021v1

Submitted on 2 Feb 2011

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.

A Library of Analog Operators Based on the Hodgkin-Huxley Formalism for the Design of Tunable, Real-Time, Silicon Neurons

Sylvain Saïghi, *Member, IEEE*, Yannick Bornat, Jean Tomas, Gwendal Le Masson, and Sylvie Renaud, *Member, IEEE*

***Abstract*—In this paper, we present a library of analog operators used for the analog real-time computation of the Hodgkin–Huxley formalism. These operators make it possible to design a silicon (Si) neuron that is dynamically tunable, and that reproduces different kinds of neurons. We used an original method in neuromorphic engineering to characterize this Si neuron. In electrophysiology, this method is well known as the "voltage-clamp" technique. We also compare the features of an application-specific integrated circuit built with this library with results obtained from software simulations. We then present the complex behavior of neural membrane voltages and the potential applications of this Si neuron.**

***Index Terms*—AVLSI circuits, Hodgkin–Huxley (HH) formalism, neuromorphic engineering, silicon (Si) neuron.**

I. Introduction

SINCE Carver Mead pioneered neuromorphic very-large scale integrated (VLSI) engineering in the early 1990s [1], [2], an ever-increasing number of research groups have adopted his design principles of using analog signals and components for computing primitives of neuron models. Even if the main feature of neuromorphic engineering is its use of the analog rules of electronic components, this discipline also merges knowledge from neurobiology, mathematics, computer science, and integrated-circuit (IC) design. These days, two approaches coexist in the neuromorphic design community: 1) bioinspired methods on the one hand and 2) neuromimetic methods on the other hand.

Bioinspired designers develop new solutions to solve engineering issues [3]. They use biological principles, taking various approximations of nature, with the view of building more efficient systems. Their research topics concern mainly sensory systems, such as vision sensors [4]–[6] or topics relevant to audition, such as auditory scene analysis [7] or sound localization [8], but also signal processing in sensory systems, such as selective attention for vision processes [9] and learning [10].

The second approach in the neuromorphic community concerns neuromimetic systems, which imitate more precisely the activity of biological cells. One of the first designs of this type is the "Silicon Neuron" [11]. Some of these systems are a follow-up of bioinspired designs for visual prosthesis [12], [13]. However, most systems address fundamental neuroscience questions, such as the exploration of activity patterns specific to an identified neural network (for example, the central pattern generator or rhythmic motor control [14]–[17]), or learning principles [18]–[20].

Our group has been designing and exploiting neuromimetic silicon neurons for ten years [20]–[23], in collaboration with Prof. G. Le Masson [24]. We have developed specific ICs from biophysical models following the Hodgkin–Huxley (HH) formalism, in order to address two fields of research: 1) build a hardware simulation system for computational neuroscience to investigate plasticity and learning phenomena in spiking neural networks and 2) develop the hybrid technique, which connects Si and biological neurons in real time. These ICs are designed to provide two main features: 1) enable the construction of biorealistic networks and 2) offer the possibility of dynamically tuning the model parameters. ICs are organized to form a simulation toolbox where a large variety of models can be implemented in real time, as all of the HH formalism parameters can be modified on-chip. References [25]–[27] presented VLSI implementation of HH models where either few parameters were tunable or where the tuning range was limited. Although our choice implies a costly design (Si area, number of pads, power consumption), it is an interesting alternative to digital computation in simulation platforms for computational neuroscience. Moreover, conductance-based models and real-time processing at the sample level will be helpful for the hybrid technique.

In neuromimetic design, the neural network targeted for a given study imposes the model to be implemented. When dealing with large networks, the chip size is limited and, therefore, the implemented model is simplified and not biophysically plausible [18], [19], [28], [29] , even though the biological behavior is preserved. These Si neurons fit a specific model, for the study of a single class of neurons. In the case of the work presented here, our aim was to fit a biophysical model, even if it led to the application of limits on the network size.

The design flow for a specific analog IC consists in circuit simulations, masks design, and fabrication. Due to the considerable time needed for that, reuse is an important issue. We ad-

Manuscript received May 17, 2010; accepted July 18, 2010. Date of publication October 28, 2010; date of current version January 26, 2011. This project was supported in part by the Fifth Research Framework Program of the European Union under Grant no. 34712 (SenseMaker) and in part by the Sixth Research Framework Program of the European Union under Grant no. 15879 (FACETS). This paper was recommended by Associated Editor E. M. Drakakis.

S. Saïghi, Y. Bornat, J. Tomas, and S. Renaud are with the IMS Laboratory at the University of Bordeaux, Talence 33400, France (e-mail: sylvain.saighi@ims-bordeaux.fr).

G. Le Masson is with the Neurocentre Magendie, Bordeaux 33077, France.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2010.2078816

1932-4545/$26.00 © 2010 IEEE

dressed this requirement by designing a library of analog operators, which can participate in the HH formalism, and can be used for the construction of a Si neuron IC. To optimize reuse, these operators have parameters, which correspond to the widest possible range of neuron model parameters. A chip designed from this library computes the neuron's activity and is accessed as a simulation tool, the model parameters of which can be tuned by the user. The user then builds his or her proprietary neurons and network, adapted to his specific application.

In Section II, we present the neuron models implemented on ICs. In Section III, we describe the library of analog operators; in Section V, we present the analog operators, which belong to the library and are based on the basic circuits briefly presented in Section IV. Section VI describes a neuromimetic simulator, with details concerning the integration of its IC onto Si in Section VII. The associated experimental platform is presented in Section VIII. The results and comparisons with software simulations are provided in Sections IX and X, respectively. In Section XI, we show additional simulations to demonstrate the diversity of possible configurations, which can be produced by using the same library. In Section XII, we address upcoming applications.

II. Implemented Model

A. HH Formalism

We used the HH formalism [30] as a design basis for our IC. The main advantage of this formalism is that it relies on biophysically realistic parameters and describes individual ionic and synaptic conductances for each neuron in accordance with the dynamics of ionic channels.

The electrical activity of a neuron is the consequence of the diffusion of different ionic species through its membrane. This activity results in fluctuations of the membrane potential, which is the voltage difference between the outer and inner sides of the cell. Ions flow through the cell membrane through ion-specific channels, modeled as specific ionic currents. A reversal potential is associated with each ionic species, according to the difference between its intracellular and extracellular concentrations. The fraction of open ion-specific channels determines the global membrane conductance of a given ion. This fraction results from the interaction between time and voltage-dependent activation and inactivation processes.

The HH formalism provides a set of equations and an equivalent electrical circuit (Fig. 1), which describes these conductance phenomena.

The current flowing across the membrane is integrated on the membrane capacitance, according to

$$C_M \frac{dV_M}{dt} = -\sum I_I + I_S \tag{1}$$

where V_M is the membrane potential, C_M is the membrane capacitance, and I_S is a stimulation or synaptic current.

I_I is the current for a given channel type, and its associated equation is

$$I_I = g_I \cdot m^p \cdot h^q \cdot (V_M - E_I) \tag{2}$$

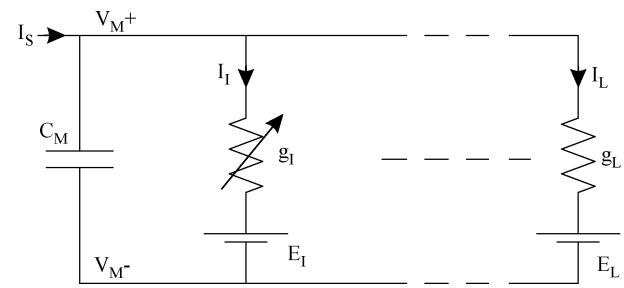


Fig. 1. Equivalent electrical circuit of a neuron following the HH model. Voltage-gated and leak ion channels are represented by nonlinear or linear conductances (g_I). The electrochemical gradients driving the flow of ions are represented by voltage sources (E_I). The membrane is represented by a capacitance (C_M).

in which g_I is the maximum conductance, and m and h represent the activation and inactivation terms, respectively. They are dynamic functions, which describe the permeability of membrane channels to its specific ion. E_I is the ion-specific reverse potential and p and q are integrers.

According to the first-order differential (3), m relaxes back toward its associated steady-state value m_∞, which is a sigmoid function of V_M

$$\tau_m \frac{dm}{dt} = m_\infty - m \tag{3}$$

$$m_\infty = \frac{1}{1 + \exp\left(-\frac{V_M - V_O}{V_S}\right)}. \tag{4}$$

The time constant for convergence is τ_m. In (4), V_O is the activation sigmoid offset, and V_S is the activation sigmoid slope. The inactivation parameter h follows identical equations, except for the sign inside the brackets, which is positive.

The HH primary equations describe sodium, potassium, and leakage channels, with $p = 3$ and $q = 1$; $p = 4$ and $q = 0$; $p = 0$ and $q = 0$, respectively, in (2). These channels are responsible for action potential generation. For more complex activity patterns, such as bursting or the discharge of action potentials with adaptation phenomena, additional channels, such as calcium and calcium-dependent potassium have to be taken into account.

B. Other Ionic Currents

The maximum conductance may also depend on an internal variable, such as an ionic concentration. In this case, the calcium channel still obeys the same equations, but with several possible values for p and q in accordance with (2)[31]. To achieve our initial objective and model various types of neural activity, we chose $p = [1; 2]$ and $q = [0; 1]$.

The potassium channel dynamic also depends on internal variables, such as the calcium concentration. This can be computed by using (5), and then introduced into (6) to evaluate the steady-state activation value. For the calcium-dependent potassium channel, we define $p = 1$ and $q = 0$.

$$\tau_{Ca} \frac{d[Ca^{2+}]}{dt} = I_{Ca^{2+}} - [Ca^{2+}] \tag{5}$$

$$m_\infty = \frac{[Ca^{2+}]}{[Ca^{2+}] + [Ca_0]} \cdot \frac{1}{1 + \exp\left(-\frac{V_M - V_O}{V_S}\right)}. \tag{6}$$

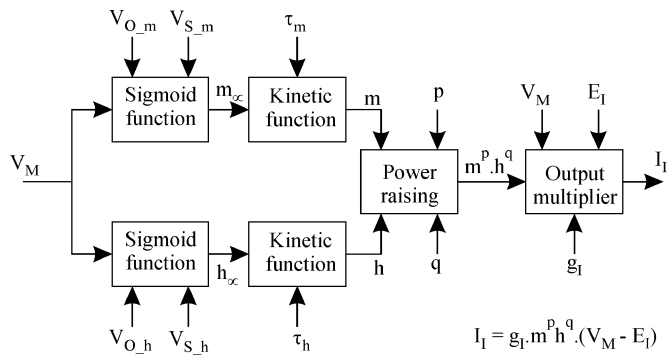


Fig. 2. Block diagram of the ionic current generator. Each block corresponds to a subcircuit performing the indicated operation and parametered by tunable parameters representing the conductance model parameters.

C. Synapses

To address neural computation at the network level, we also integrated synaptic interactions, modeled using conductance-based synapses [32]. The synaptic current is described by

$$I_{Syn} = g_{Syn} \cdot r\,(V_{M_Pre}) \cdot (V_{M_Post} - E_{Syn}) \tag{7}$$

where g_{Syn} is the synaptic weight. If the synapse is not plastic, then g_{Syn} has a fixed value, whereas in the opposite case, it follows a plasticity rule such as that defined by spike timing-dependent plasticity (STDP).

In (7), the overall synaptic current depends on the postsynaptic membrane voltage, and the activation term r depends on the presynaptic membrane voltage. The terms V_{M_Post} and V_{M_Pre} then replace V_M, respectively, in (2) and (4).

III. From Model to Si

A. Library of Mathematical Functions and Specifications

For our design, we retained five channel types: 1) leakage, 2) sodium, 3) potassium, 4) calcium, and 5) calcium-dependent potassium. By combining those channels, we can model a large variety of neurons. Each channel follows the modular principle described before, which results in the block diagram of ionic current generators (Fig. 2).

The repetition of mathematical operators in the model, combined with tunable parameters, is an advantage for systematic development.

The specifications of the equation parameters, presented in Table I, were set, taking into account models of cortical neurons from the Neuron software database [33]. The resulting constraints of those specifications are strong for IC design. In some cases, we had to split the parameter ranges to preserve interesting dynamics.

B. Design Mode

Taking into account the integration constraints of the electronics, and in order to increase the dynamic range and noise immunity, we applied an x5 gain factor to the voltages. The gain factor for the conductance depends on the membrane capacitance of the hardware, compared with that found in the biological context. Pending the detailed results described in Section X, due to the ratio of capacitor values between hardware and model, we multiplied the conductance by 22.72, leading to a current gain factor of 113.63.

TABLE I
Specification of the Biological Parameters for an Analog Operator Library

Function	Parameters	Range Values
Sigmoid function	V_O (mV)	-100 to +100
	V_S (mV)	2 to 20
Integrator function	τ_{m_Na} (ms)	0.02 to 1
	τ_{h_Na} and τ_{m_K} (ms)	0.2 to 10
	τ_{m_Ca} (ms)	2 to 100
	τ_{h_Ca} (ms)	20 to 1000
	$\tau_{m_K_Ca}$ and $\tau_{m_[Ca]}$ (ms)	0.4 to 20 and 20 to 1000
	τ_r (ms)	0.4 to 20
Output multiplier	g_{Na} and g_K (normalized) [a]	0.02 to 1
	g_{Ca}, g_{K_Ca}, g_{Leak}, and g_{Syn} (normalized) [a]	0.00001 to 0.1
	E_I (mV)	-150 to 150

[a] The overall conductance value depends on cell area. All conductance values are normalized to that of sodium.

We designed the functions in current mode [34], so that all of the internal variables are physically represented by currents. This design mode improves noise immunity and enables all usual operations to be designed with simple circuits. All MOS transistors operate above threshold. Thus, we can address the specifications defined in Table I which have a large dynamic range for input voltages and input/output currents.

IV. Elementary Circuits

Although the elementary circuits are inspired from well-known circuits used by electronics designers, they are described here in detail, to provide the reader with an adequate understanding of the computational methods used for the HH variables.

A. Voltage–Current Converter

The model's parameters are applied to the chip in the form of voltages. Since we chose to use a current mode design, a voltage-current converter (VCC) is needed. The solution with a high linear range uses one operational amplifier (Op-Amp), one resistor, and one MOS transistor (Fig. 3). The op-amp driver stage is not necessary since its output is connected to the MOS gate. To ensure a linear behavior on a large range, we should prevent the threshold voltage from depending on the bulk-substrate voltage. Using the BiCMOS n-well technology available from the common resource center for VLSI prototyping, we can have a bulk-to-source connection only with PMOS transistors. We thus obtained the following equation where V_{CC} is the power-supply voltage:

$$I_O = \frac{V_{CC} - V_I}{R_{Conv}}. \tag{8}$$

B. Current-Mode Multiplier

While additions and subtractions are easily implemented in current mode, multiplications need more elaborated circuits. Here, the input variables are the activation and inactivation terms [see (2)], and these terms are bounded by 0 and 1, so that

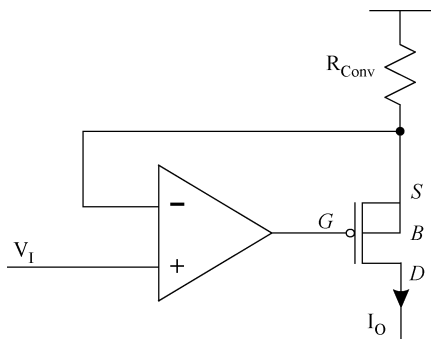


Fig. 3. Voltage-current converter using 1 op-amp, 1 PMOS transistor and 1 resistor (V_I is the input, I_O is the output).

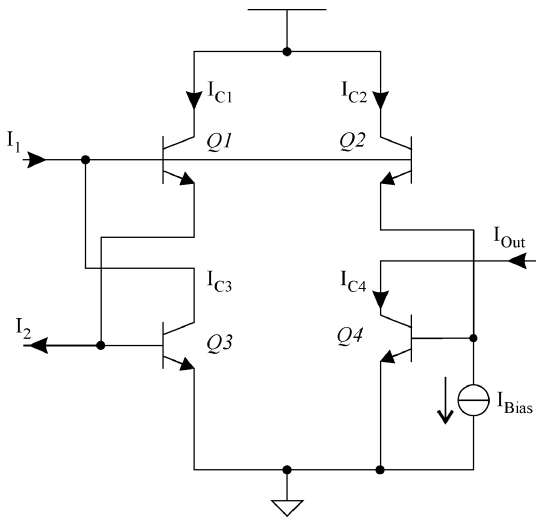


Fig. 4. One-quadrant multiplier in current mode using the NPN transistor with $I_{C1} \cdot I_{C3} = I_{C2} \cdot I_{C4}$, resulting in $I_{Out} = I_1 \cdot I_2 / I_{Bias}$.

only a one-quadrant multiplier is needed, as shown in Fig. 4 [35]. From the translinear loop principle, and with equal emitter areas, we obtain

$$I_{C1} \cdot I_{C3} = I_{C2} \cdot I_{C4}. \tag{9}$$

Using BiCMOS technology, the base current is negligible when compared to the collector current. We then obtain

$$I_{Out} = \frac{I_1 \cdot I_2}{I_{Bias}}. \tag{10}$$

C. MOS Operational Transconductance Amplifier

When we need voltage–current conversion with a differential input voltage, we cannot use the VCC described before. We therefore need to use an operational transconductance amplifier (OTA), as shown in Fig. 5. Reference [36] provides details of the associated equations, which lead to

$$I_{Out} = V_D \cdot \sqrt{\mu \cdot Cox \cdot \frac{W}{L} \cdot I_{Bias} - \left(\frac{\mu \cdot Cox}{2} \cdot \frac{W}{L} \cdot V_D\right)^2} \tag{11}$$

where μ is the electron mobility, C_{OX} is the gate–oxide capacitance, W is the channel width, and L is the channel length.

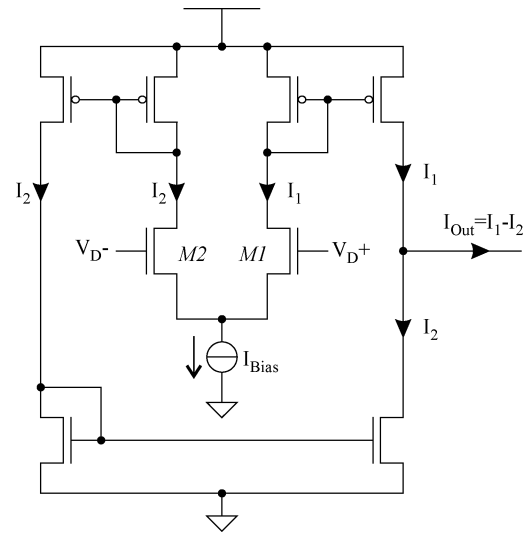


Fig. 5. OTA, with an input V_D and an output I_{Out}.

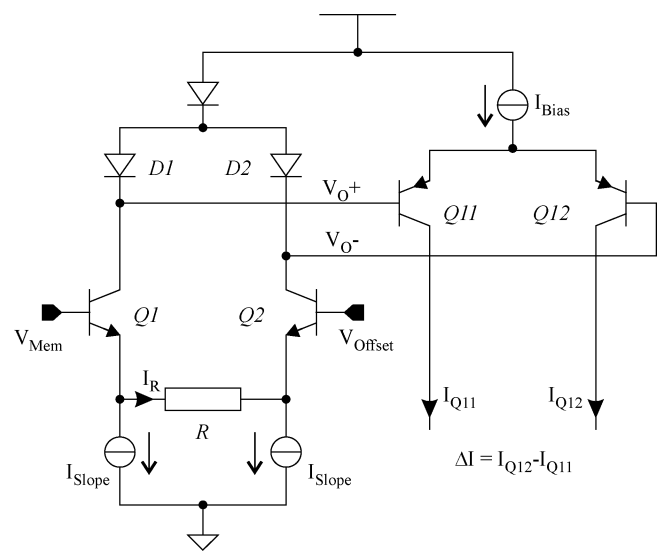


Fig. 6. Bipolar differential pair with predistortion stage, with a differential input $V_{Mem} - V_{Offset}$ and an output ΔI.

The MOS OTA is compact but is linear on a limited range. Equation (12) gives the expression for ε, the relative error, which characterizes the linearity of the circuit. Here, we arbitrarily chose $\varepsilon = 5\%$, by setting appropriate values for W and L

$$\varepsilon = \sqrt{1 - \frac{\mu \cdot Cox}{4 \cdot I_{Bias}} \cdot \frac{W}{L} \cdot {V_D}^2} - 1. \tag{12}$$

D. Bipolar Differential Pair With Predistortion Stage

As we operate MOS above threshold, only bipolar transistors can provide the sigmoid function, which will be detailed in the next section as a collector current from a bipolar differential pair. For the operating range of a standard bipolar differential pair, we can accept linear behavior for an input voltage equal to $4 \times U_T$, where U_T is the thermal voltage. To satisfy the analog operator's specifications where the input voltage dynamics is up to 1 V, this range needs to be increased. This was achieved by using the circuit diagram shown in Fig. 6 (see [37] for the

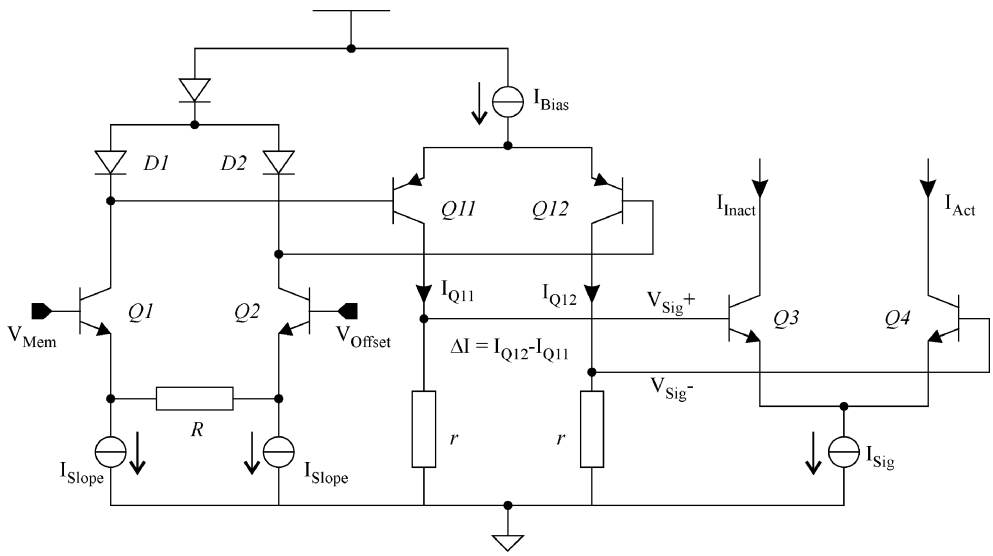


Fig. 7. Tunable sigmoid function, with input voltage $V_{Mem} - V_{Offset}$ and output current I_{Act} or I_{Inact} to compute the activation or the inactivation term, respectively.

detailed equations). Assuming the integrated resistance R to be sufficiently high ($I_R \ll I_{Slope}$), V_O and ΔI are given by (13) and (14), which are the two most important results

$$V_O = -U_T \cdot \ln \frac{I_{Slope} + \frac{(V_{Mem} - V_{Offset})}{R}}{I_{Slope} - \frac{(V_{Mem} - V_{Offset})}{R}} \tag{13}$$

$$\Delta I = \frac{1}{R} \cdot \frac{I_{Bias}}{I_{Slope}} \cdot (V_{Mem} - V_{Offset}). \tag{14}$$

If I_{Slope} is generated with a VCC circuit described in Section IV-A, we can identify the two parameters V_O and V_S in (14) used in (4).

V. Library of Analog Operators

Using the elementary circuits described before, we designed the analog blocks according to their specifications (Table I). For all of the blocks, the input and output dynamics are constrained in the range [2 V – 3 V]. Thanks to this feature, the blocks as mathematical operators can be freely arranged to compute a complete conductance equation. In counterpart, this design mode results in high power consumption and Si area. In order to test the analog operators, we designed a prototype chip called *Violetta*, based on a 0.8-μm BiCMOS technology process from *austria micro systems* (AMS), under the Cadence environment. We ran Monte Carlo simulations on all analog operators. The effect of components mismatch can be compensated by adjusting the model parameters in the range of Table I. We made all measurements with a dedicated printed-circuit board (PCB) and standard signal generators. A 5-V power supply was used with a reference bias voltage of 2.5 V.

A. Sigmoid Function

The sigmoid block computes the steady-state value corresponding to activation or inactivation. This mathematical function appears in the transfer function of a bipolar junction transistor (BJT) differential pair. To facilitate management of the large input ranges needed for V_M and V_O (see Table I), we used a predistortion stage, described in Section IV-D. Fig. 7 shows the corresponding circuit.

The current difference ΔI is applied to the resistors r, and the resulting voltage is applied to the differential input pair (Q3-Q4), with V_{Sig} defined by (15). Expression (16) gives the collector currents in the BJT differential pair Q3-Q4, where U_T is the thermal voltage and I_{Sig} is the bias current of the differential pair, equal to 20 μA. We indicate I_{Act} or I_{Inact} as the activation or inactivation terms of the neuron model in (4)

$$V_{Sig} = -\frac{r}{R} \cdot \frac{I_{Bias}}{I_{Slope}} \cdot (V_{Mem} - V_{Offset}) \tag{15}$$

$$\begin{cases} I_{Inact} = \frac{I_{Sig}}{1+\exp\left(+\frac{V_{Sig}}{U_T}\right)} \\ I_{Act} = \frac{I_{Sig}}{1+\exp\left(-\frac{V_{Sig}}{U_T}\right)} \end{cases}. \tag{16}$$

Power consumption for this block is 2.3 mW. Fig. 8 presents the measured values of the current I_{Act} applied to an external 100 kΩ resistor (for measurements only), with various parameter combinations. The parameter V_{Offset} corresponds to the half-activation or half-inactivation voltage, and V_{Slope} sets the slope of the linear part of the sigmoid. It can be seen that the circuit follows the originally specified trend, over the full range of model parameters.

B. Integrator Function

In the design of the integrator function, for reasons of noise immunity, we chose to use a closed-loop integrator rather than a differentiator. The sigmoid generator output signals to the integration module are shown in Fig. 9. This generator contains two current-mode multipliers (CMM), as shown in Section IV-B. The sigmoid generator output and the integrator function are always positive, since they obey (3). For reasons of simplicity, it was thus preferred to design two one-quadrant multipliers, rather than a single two-quadrant multiplier placed after a subtractor. One of the multiplier inputs is the current provided by

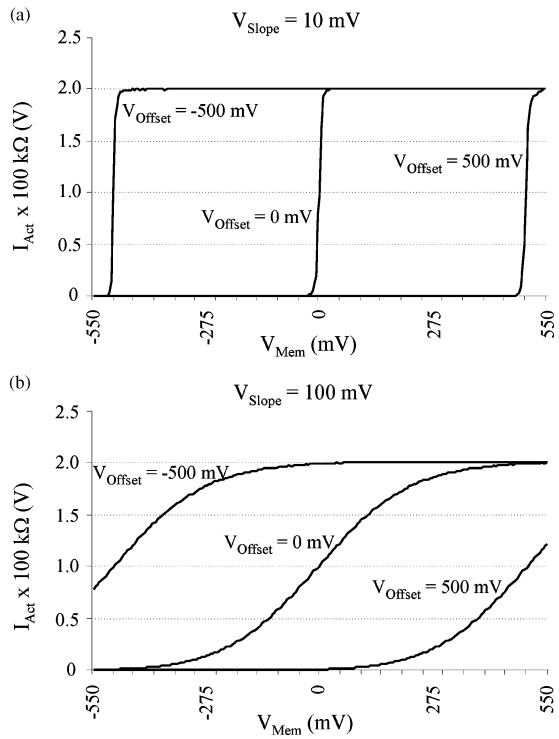


Fig. 8. Measurements of I_{Act} (activation term). The bias current and the output resistor are 20 μA and 100 kΩ, respectively. The x-axis is the clamped voltage V_{Mem} for the measurements ($V_{Mem} = V_M \times 5$, V_M from the biological model). (a) Response for V_{Slope} equal to 10 mV ($V_{Slope} = V_S \times 5$, V_S from the biological model). (b) Response for the V_{Slope} equal to 100 mV. For both conditions on V_{Slope}, V_{Offset} is tuned to its extreme values (from -500 mV to $+500$ mV).

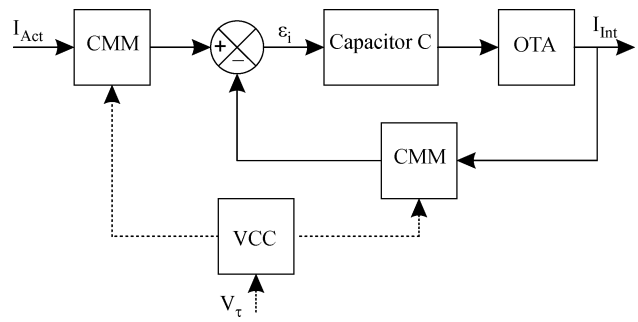


Fig. 9. Circuit diagram of the integrator function. The input and output currents, respectively I_{Act} and I_{Int}, are amplified by the current-mode multiplier (CMM). The second input to the CMM is current delivered by a voltage-controlled current (VCC) source, and the gain of the CMM is $A(V_\tau)$. The capacitor transfer function is $1/Cs$, and the gain of the OTA is B.

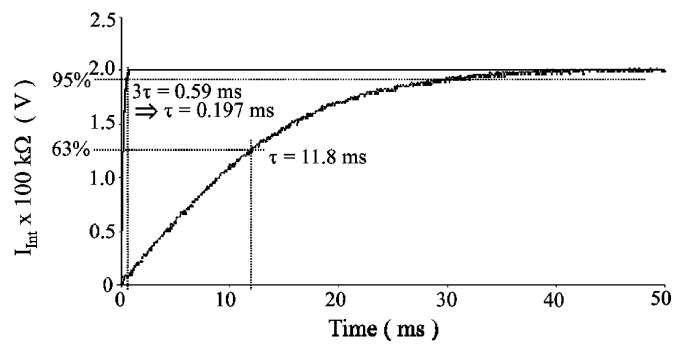


Fig. 10. Integrator current response to an input current step, for two different values of the control voltage V_τ corresponding, respectively, to $\tau = 0.197$ ms and $\tau = 11.8$ ms.

the sigmoid function (I_{Act} or I_{Inact}). The second input is a controlled current supplied by a VCC. We modify the VCC circuit to obtain two identical output currents thanks to a current mirror. One can thus think of the CMM as a current amplifier, with controlled gain $A(V_\tau)$.

The OTA in Fig. 9 has a fixed gain B and the capacitor C transfer function is $1/Cs$, where s is the Laplace variable. This leads to (17), and by comparing this with (3), we identify the time constant as $C/(A(V_\tau) \cdot B)$

$$\frac{C \cdot s}{A(V_\tau) \cdot B} \cdot I_{Int} = I_{Act} - I_{Int}. \tag{17}$$

Since it is not feasible to perform on-chip tuning of the time constant over 6 decades (i.e., from 0.02 ms to 1 s (Table I), we use an external capacitor C, which allows a reduced range to be selected on the chip. The control voltage V_τ (from 1.5 V to 4.95 V) allows fine tuning within each range. Fig. 10 illustrates the output current from this block, with $C = 10$ nF and for two values of $A(V_\tau)$. The output current is applied to an external 100-kΩ resistor (for measurements only), and a good match is observed with respect to the specifications. Power consumption for this block is 1.4 mW.

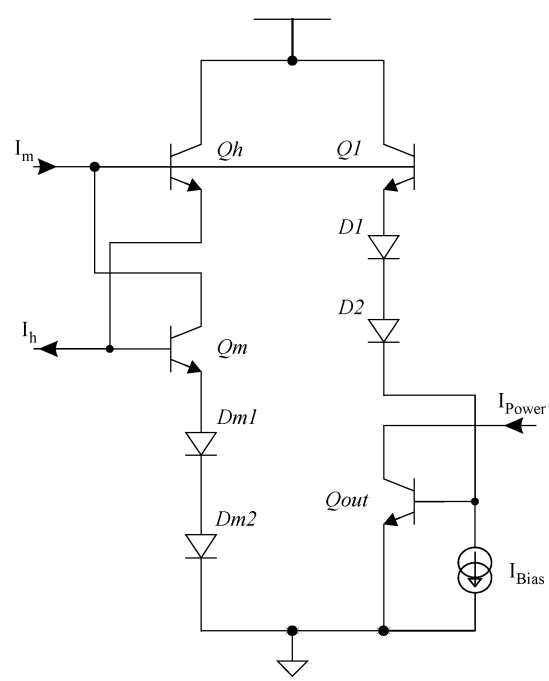


Fig. 11. Circuit used for power raising, using the translinear loop principle. In that case, $I_{Power} = I_m^3 \cdot I_h / I_{Bias}^3$ is used for the sodium channel m^3h function.

C. Power Raising

This stage is a variant of the CMM presented before. We implemented all possible combinations of p and q (respectively, m for the activation and h for the inactivation power), according to the specifications (e.g., mh, m^2h, m^3h, and m^4). The power is not tunable in this stage. Fig. 11 shows the circuit diagram for the example of m^3h (corresponding to the Na current). This

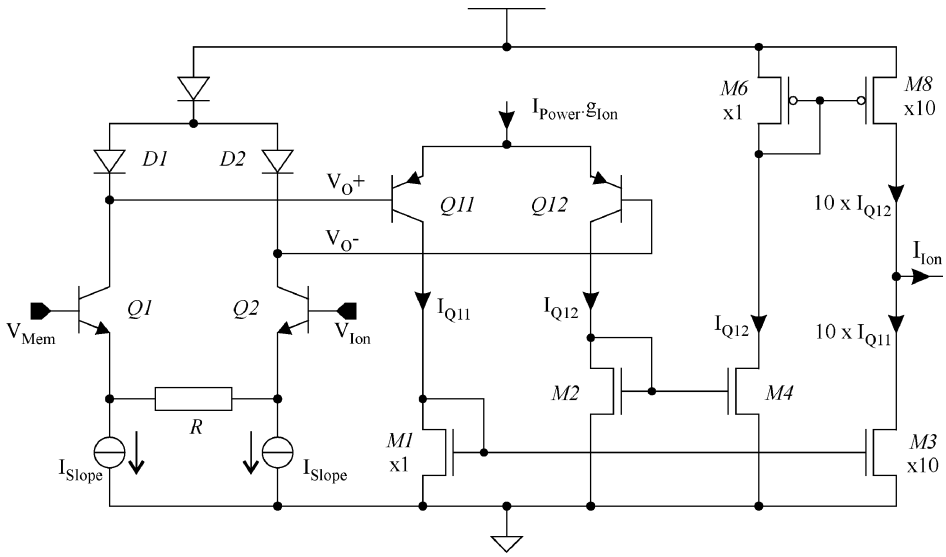


Fig. 12. Multiplier circuit generating the ionic current I_{Ion}, with an input bipolar differential pair $Q_1 - Q_2$ controlled by V_{Mem} and V_{Ion} and a predistorsion stage $Q_{11} - Q_{12}$ biased by $I_{Power} \cdot g_{Ion}$.

circuit computes this product using the translinear loop principle. Since $I_B \ll I_C$ and $I_B \ll I_E$ for all bipolar transistors, we can assume the following current relationships: $I_{Qh} = I_h$; $I_{Qm} = I_{Dm1} = I_{Dm2} = I_m$; $I_{Q1} = I_{D1} = I_{D2} = I_{pol}$. Applying the translinear principle to the V_{BE} loop, we thus obtain the following expression for the output current:

$$I_{out} = \frac{{I_m}^3 \cdot I_h}{{I_{Bias}}^3}. \tag{18}$$

It is important to note that for all combinations of p and q according to the model, the dedicated circuits are always raised to a power in the numerator of (18) which is 1 higher than that in the denominator. We know that I_{Int}, which becomes I_m or I_h here, lies between 0 and 20 μA, so that with $I_{Bias} = 20\ \mu A$, I_{out} always lies in the range between 0 and 20 μA, for all combinations of p and q. The average power consumption is 0.25 mW for $(p + q) = 4$.

D. Output Multiplier

The last element of our library is the output multiplier. The specifications presented here cover a very large range for the value of g_I. As already stated, we split this range into two parts, depending on the type of ion under consideration. The multiplier shown in Fig. 12 is based on a bipolar differential pair with a predistorsion stage. The bias current I_{Bias} of the bipolar pair Q11-Q12 is replaced by $I_{Power} \cdot g_{Ion}$, where I_{Power} comes from the power raising stage and g_{Ion} is the current supplied by a VCC (the parameter g_{Ion} represents the maximum conductance value). The collector current difference, between Q11 and Q12, can thus be determined from (14). For the first version of the multiplier used for the Na and K channels, the width of M3 and M8 is ten times higher than that of M1 and M6. The output current is also ten times higher, and is given by the following expression:

$$\Delta I = \frac{10}{R \cdot I_{Slope}} \cdot I_{Power} \cdot g_{Ion} \cdot (V_{Mem} - V_{Ion}). \tag{19}$$

Power consumption for this first version is 3.2 mW. For the second version of the multiplier used for the other channels (i.e., for the weakest g_I values), we maintain the same width for M3 and M8 as for M1 and M6. As specified in Table I, the maximum output currents for calcium, calcium-dependent potassium, and leakage channels are one decade smaller than those of sodium and potassium. Power consumption for this second version is 2.5 mW.

VI. Toward Real-Time IC Simulations

Our library of tunable mathematical operators for custom analog simulator designs was validated with a preliminary prototype – the *Violetta* IC. We then implemented these functions on a second prototype IC, referred to as *Pamina*, which allowed us to build a real-time simulator for various sets of computational neuroscience applications. For this, we first defined the specifications of this new chip generation, and defined the details of the required new functions. The analog computational core, built with various analog operators, represents a set of ionic current generators. Digital functions are added to manage the core topology, and analog memory cells are included in order to store the model parameters.

A. IC Pamina Specifications

As shown in Fig. 13, this chip includes: 1) analog computational cores inspired from previous developments (*Analog Core* block); 2) a set of SRAM digital memory cells defining the connections needed between the current generators (*Topology* block); 3) a set of DRAM analog memory cells, used to store the user-defined set of model parameters (*Analog Memory* block). An artificial neuron consists of synaptic currents and a set of ionic currents, summed on a membrane capacitance. In this version, two artificial neurons are implemented, each of them being built with five ionic conductances according to the model described in Section II (Na, K, Ca, K(Ca) and Leakage). For each

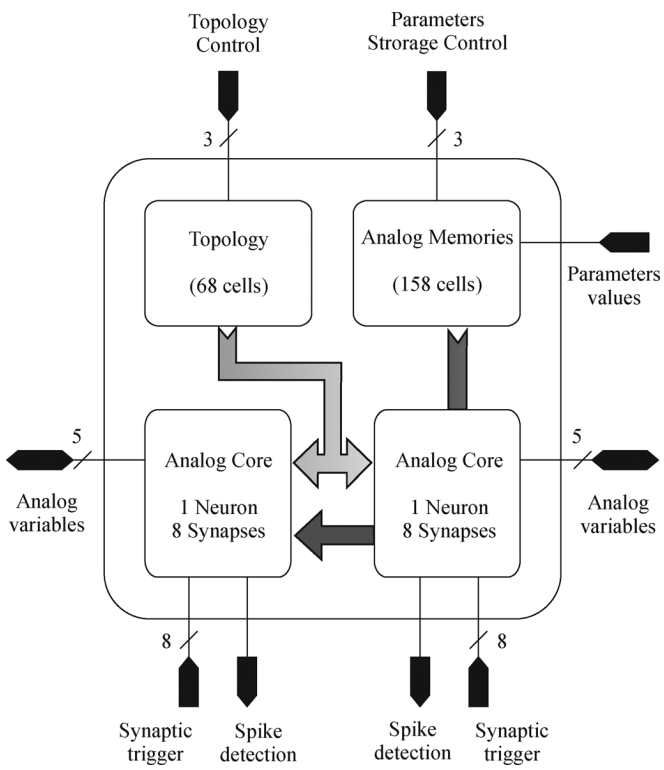


Fig. 13. I/O mapping, block diagram, and inter-block connections in the Pamina IC. The analog cores are in charge of computing the ionic and synaptic currents. The function parameters are stored in the analog memory cells. The conductance's connection is controlled by the topology digital memory cells.

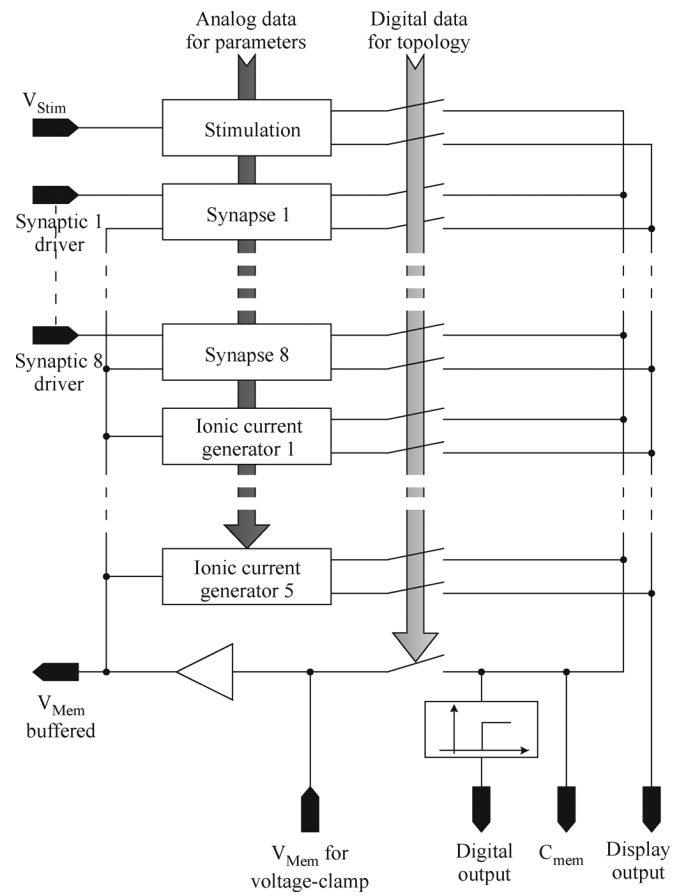


Fig. 14. Architecture of an analog core. It includes 5 ionic current generators and 8 synaptic current generators with their own pre-synaptic inputs. Currents selected by the digital topology control are applied to the C_{mem} pin to be summed on an external capacitance. Analog signals (V_{Mem} and one I_{Ion}) are output for display or storage. V_{Mem} is digitized on a digital output, using a threshold comparator. V_{Stim} and V_{Clamp} are used for stimulation and V_{Mem} clamping, respectively.

neuron, we added eight synaptic conductances for network applications, and one stimulation input.

The analog parameters are sent serially and periodically to the memory cells. The digital parameters control the topology of the analog simulation core. Some of the currents and voltages of the Si neurons (such as membrane voltages and ionic currents) are available in the form of analog outputs for an oscilloscope display, or inputs for voltage-clamp experiments. Neural activity (spikes), as well as presynaptic information is available in digital form. This also provides access for the real-time management of simulations (e.g., software learning computations that modulate the synaptic weights in the neural network).

B. Analog Cores: Ionic Channels and Synapses

We now give further details of the architecture and I/O of the analog cores (see Fig. 14). All current generators have two outputs: the first can send its output to the external capacitor C_{Mem} (representing the neuron membrane capacitor); the second one is used as a display output. A current buffer authorizes the monitoring of electrical membrane activity with an oscilloscope probe, through a third output (V_{Mem} buffered). The switch between the current buffer and the external capacitor is closed for neuronal activity simulations and opened for voltage-clamp experiments (this technique involves setting V_{Mem} to predetermined levels to characterize the ionic conductances). Thanks to the external source V_{Stim}, we can use arbitrary signals to stimulate the neuron.

We also implemented eight synaptic current generators, triggered by digital pre-synaptic inputs, together with a tunable detection threshold, to digitize the action potential.

C. Topology Management

The connectivity of these different blocks is externally programmable, via the on-chip SRAM memory. The data sent to the chip is decoded and drives static switches. We define three types of data. The first type configures each analog core, e.g., selects, from the ionic channels, synapses, a stimulation current, and those current generators to be connected to C_{Mem}. The second type describes the analog core configuration according to Section II, e.g., the time constant range for $\tau_{m_K_Ca}$ and $\tau_{m_[Ca]}$, the power raising for the activation term in the calcium channel, or the computational mode (closed-loop neural membrane or voltage-clamp experiment). The last data set defines the monitored current output(s).

D. Analog Memories

We designed a specific analog memory bank for our application. The simplest dynamic memory cell is a capacitive device, which is periodically refreshed. A switch, opened when the cell

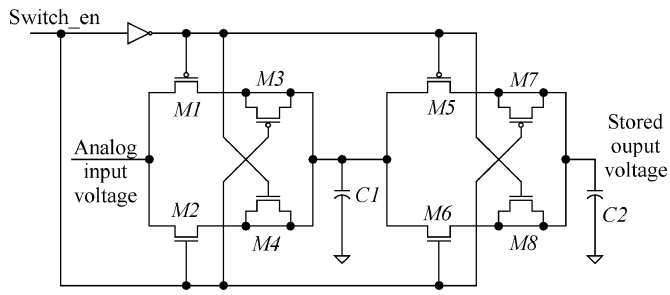


Fig. 15. The 2 stages analog memory cell, addressed by the signal Switch_en. C2 is the final storage capacitor.

is addressed for writing, controls its access. We apply the 158 model parameters in serial fashion, coded by analog voltages, onto one input of the ASIC. The access frequency on this pin is 100 kHz, which limits the refresh frequency of any one cell to 633 Hz (a hold period of 1.6 ms). This provides a good compromise, limiting the clocking noise and the discharge of the storage capacitor.

The memory cell itself is a 2-stage circuit, as shown in Fig. 15. A digital state-machine (not represented in the figure) addresses the cell by controlling the *Switch_en* signal. The design was optimized [38] in order to minimize leakage currents, in particular during $\mathrm{open} \rightarrow \mathrm{close}$ and $\mathrm{close} \rightarrow \mathrm{open}$ transitions: M3-M4 and M7-M8 respectively compensate the leakage transition currents of the switches M1-M2 and M5-M6. The analog data is stored on C2. The $\mathrm{R_{off}}$ value of the M5-M6 switch is maximized during the time when, due to its first stage storage capacitance C1, the voltage difference across M5-M6 is being reduced. These combined features limit the leakage current on the M5-M6 switch. The first stage switch M1-M2 has a small resistance $\mathrm{R_{on}}$, to limit the loading period. Simulations show that in a period of 2 ms, the memory value is degraded by 1 mV, from an initial total range of 5 V (12 bit encoding); we observe glitches with a maximum duration of 200 ns and an amplitude of 10 mV which is not sufficient to trigger a spike, whatever the parameter. This performance matches our requirements perfectly, since the parameters lie in the range [1 V – 5 V], with a refresh period of 1.6 ms.

VII. Silicon Integration

The second prototype ASIC was designed in full-custom mode, using a BiCMOS SiGe 0.35 $\mu\mathrm{m}$ technology process from *austria**micro**systems* (AMS), under the Cadence environment. Fig. 16 is a microphotograph of this ASIC, called *Pamina*. The analog cores, topology, and analog memory cells are indicated on the figure.

Pamina contains around 19000 MOS transistors, 2000 bipolar ones, and 1200 passive elements; its surface area is $4170 \times 3480\ \mu\mathrm{m}^2$. All analog cells are designed in full-custom mode. For this, we drew the analog operator layouts, and merged them in such a way as to build the target current generators. The average power consumption for a five conductances neuron is 38 mW. Whereas the digital cells are taken from the *austria**micro**systems's* library, 71% of the 22 000 components are produced by a fully customized design procedure. We used optimized analog layout procedures, such as common-centroid and dummy devices, in order to implement critical structures and to harden the design against technological process mismatches and variations [39].

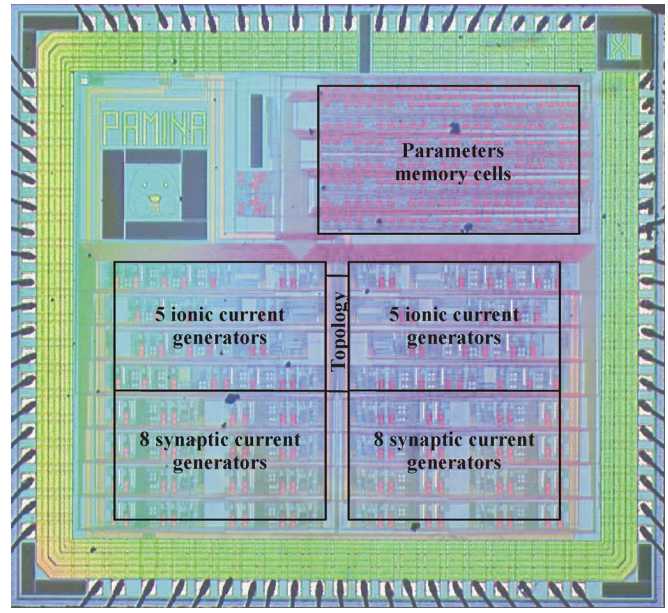


Fig. 16. Microphotograph of the *Pamina* chip ($4170 \times 3480\ \mu\mathrm{m}^2$) after fabrication. The areas for the current generators and the analog memory cells are identified.

VIII. Hardware Simulation System

In this section, we describe the hardware simulation system, based on the Pamina IC, which was designed to explore the neuron conductance based model.

A. System Specifications

To facilitate the use of the simulation system (by neuroscientists), we interfaced *Pamina* with a computer: the experimenter can define the neuron model parameters using a software interface. These characteristics include the ionic channel type, selected from the options of sodium, potassium, leakage, calcium, and calcium-dependent potassium, and the parameter values for each channel. The control software sends this data to the IC, through the analog and digital data paths described in Section VI. The analog computation core simulates, in real-time, the membrane potential which is digitized by an analog to digital converter and sent to the computer for display, storage or further processing (see Fig. 17). We do not insert synapse parameters or synaptic plasticity at this level, because a layout mistake prevented the synapses from functioning correctly.

B. PCI and Interface Boards

The analog ICs are interfaced with the computer using a specifically designed PCI board and a circuit-dedicated interface board. The interface board supports the *Pamina* chip, one DAC to directly stimulate each neuron (see Section VI.B.), and two analog ADCs to sample the membrane voltages.

The PCI board is controlled by an on-board FPGA, which is configured to control the main interfacing operations. The

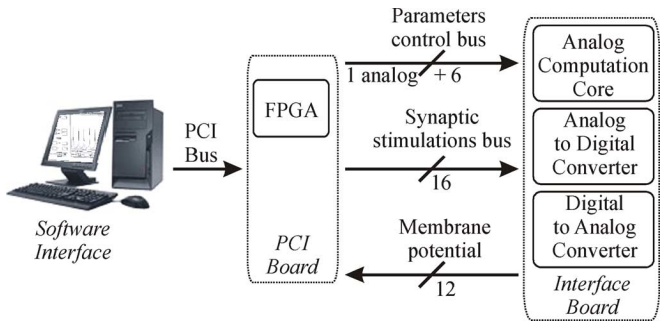


Fig. 17. Synopsis of the simulation system based on the *Pamina* IC. The IC is controlled by a software and a dedicated PCI board.

FPGA accesses the *Pamina* IC through the daughter board and a PCI bridge, which handles data transfer from and to the computer. The FPGA decodes the information sent by the computer, and dispatches it to the different components on the daughter board. Among these components, an external 1 MB RAM buffers the sampled membrane voltage in real-time and sends it to the PC storage unit through the PCI bus. Lastly, the FPGA provides the signals required by the chip topology, and for analog memory cell refreshment.

C. Software Layers

The host computer runs on a GNU/Linux operating system, based on the RedHat distribution. An open source system facilitates the development of dedicated driver software to control the PCI board. Among the required functions to manage the PCI board, two were optimized for speed and latency. The first one of these, *poke*(address,data), writes to the FPGA, and the second one, *peek*(address), reads from it. Associated with the address table of the FPGA, these two functions provide easy access for system development and debugging.

All simulations are preceded by a configuration phase involving the definition and transmission of a set of parameters. During execution of the effective simulation, the users can, at any time, send a set of configuration data to change any simulation parameter.

IX. Ionic Channel Identification

We performed a set of benchmark tests on *Pamina*, prior to running any explorative experiments. The classical method used to check a modeled neuron relies on measurements of the membrane voltage, which should exhibit an oscillatory activity. At the behavioral level, however, this type of verification remains very imprecise.

Neurophysiologists have developed specific techniques to extract current-based models from the biological recordings of neuronal electrical activity. The most popular method is the voltage-clamp technique [40]. For such an experiment, the neuroscientists record the *in vitro* neural activity using intra-cellular electrodes. It is possible to measure the conductance of an individual ionic channel by inhibiting the other channels, through the injection of specific drugs, and by clamping the neuron membrane voltage. After measuring the neuron's surface area and membrane capacitance, the experimentalist extracts the current-voltage relationships of the ionic conductances for each type of ionic channel. In complex neuron models, such as the Hodgkin-Huxley enhanced models, 15 parameters are needed to describe a 3-conductance neuron.

In analog design, the influence of dispersion and mismatch in the manufacturing process, in addition to the influence of the IC electronic environment, lead to a certain degree of uncertainty concerning the exact equations computed in the circuit. The use of a voltage-clamp method, such as in electrophysiology, enables the exact model implemented on the IC to be extracted after manufacture. If the parameters are tunable, the model parameters can be adjusted, in order to increase the precision of the model. These extracted characteristics also provide useful information, which can be used for further modifications of the analog blocks. The relevance of these extracted parameters when compared to the original biological model is discussed in Section X.

Using the voltage-clamp technique, we can identify, one by one, the parameters of each ionic channel. We open the membrane voltage loop with the switch between the $\mathrm{C_{Mem}}$ output and the Voltage-clamp input (see Fig. 14). We then study the responses of the ionic current generators to successive steps in the value of $\Delta \mathrm{V_{Mem}}$, applied to the voltage-clamp input connected to the ionic channel. Fig. 18 illustrates the experimental process: Fig. 18(a) shows the membrane voltage steps successively applied to $\mathrm{V_{Mem}}$, and Fig. 18(bB) shows the associated response of the isolated potassium channel. The process used to extract the potassium channel parameters is described in Section IX(A). In this paper, we studied inhibitory and excitatory neuron models to check the behavior of the *Pamina* IC. In the literature, these two types of model are also referred to as, respectively, Fast Spiking (FS) and Regular Spiking (RS) neurons. These require respectively three and four conductance values: sodium, potassium, and leakage channels for the FS neuron, and an additional modulator channel for the RS neuron. The modulator channel is a slow potassium current commonly used for spike-frequency adaptation [41]. It has the same equation as the calcium one, but different parameters. We use the calcium current generator to implement this modulator current. In the following paragraphs, we apply this identification technique to the potassium, sodium, leakage, and modulator channels. In Section X, we compare the experimental and theoretical results.

A. Potassium Channel

The potassium (K) channel is easy to identify because it has only one activation term, n. When (2) is adapted to the electronic variables of this channel, it gives:

$$\mathrm{I_K} = \mathrm{g_K}\ \mathrm{n}^4(\mathrm{V_{Mem}} - \mathrm{V_K}). \tag{20}$$

Fig. 18 presents the results of the voltage-clamp experiment on the K channel of *Pamina*. For $\mathrm{t} > 30\ \mathrm{ms}$ in Fig. 18(b), the current $\mathrm{I_K}$ reaches its steady state value and $\mathrm{n} = \mathrm{n}_\infty$, where n_∞ is the fraction of the activated channel. For $\mathrm{V_{Mem}} > 300\ \mathrm{mV}$, (4) adapted also to electronic values gives: $\mathrm{n}_\infty = 1$. Expression (20) thus becomes:

$$\mathrm{I_K} = \mathrm{g_K}(\mathrm{V_{Mem}} - \mathrm{V_K}). \tag{21}$$

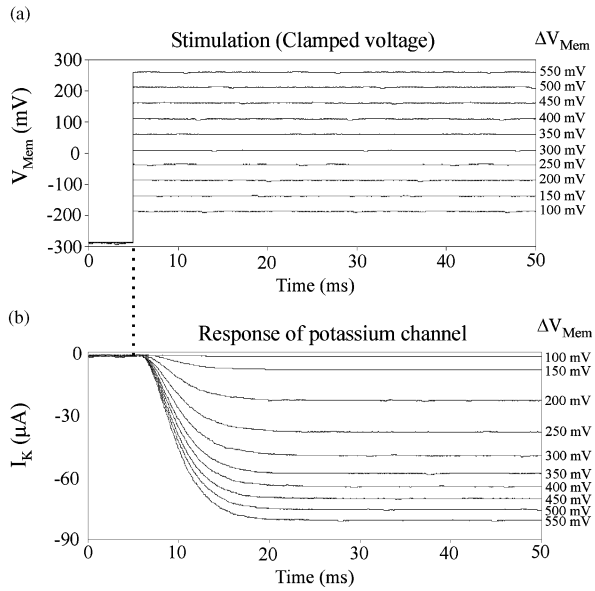


Fig. 18. Illustration of the voltage-clamp method on the ASIC, applied to the potassium channel. A) ΔV_{Mem} input steps. During the experiment, voltage steps are applied successively, with a rest period of 50 ms. Y axis scale: $V_{Mem_ASIC} = 5 \times V_{M_Bio}$. B) Response current of the Potassium channels to the voltage step ΔV_{Mem} of A. The current absolute value for $t \propto$ increases with V_{Mem}. Y axis scale: $I_{K_ASIC} = 113.6 \times I_{K_bio}$. X axis: simulation time is equal to biological real time.

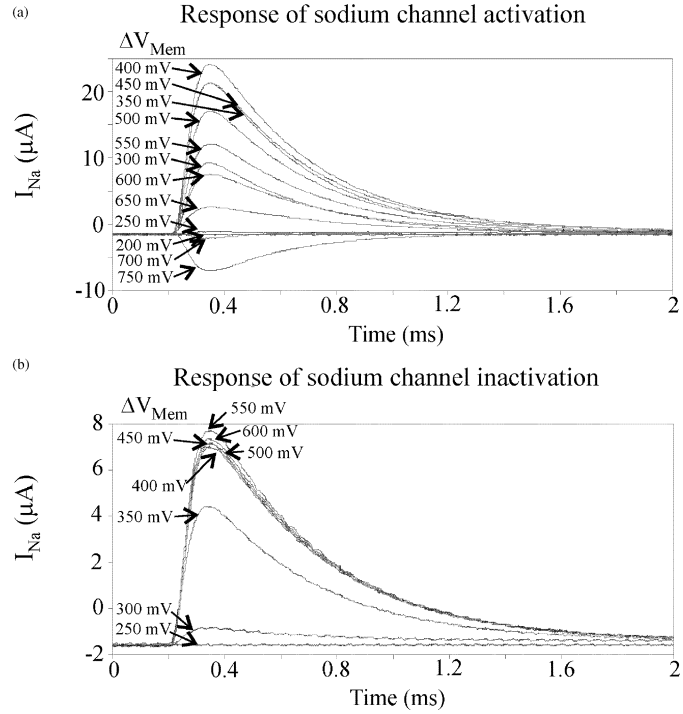


Fig. 19. Measurements on the sodium channel. A) represents the response to Sodium channel activation. B) represents the response to Sodium channel inactivation. On the chip, both currents are a factor of 113.6 greater than the biological values.

By applying a simple linear regression method, we obtain $V_K = -493$ mV and $g_K = 107.2\ \mu$S.

Then, for $t > 30$ ms, we can use (20) to plot the activated channel fraction n_∞ as a function of V_{Mem}:

$$n_\infty = \sqrt[4]{\frac{I_K}{g_K \cdot (V_{Mem} - V_K)}} \tag{22}$$

for each parameter pair (I_K, V_{Mem}).

The resulting curve, $n\infty(V_{Mem})$, is fitted by a sigmoid function, with $V_{Offset_n} = -186.7$ mV and $V_{Slope_n} = 56.4$ mV.

For the activation time constant, we use a classical approximation method (81.5% of the full range at $t = 3 \cdot \tau_n$), leading to $\tau_n = 2.5$ ms.

B. Sodium Channel

The Sodium channel has an activation term (m) and an inactivation term (h). Then, when (2) is adapted to the electronic variables, it becomes:

$$I_{Na} = g_{Na}\, m^3 h (V_{Mem} - V_{Na}). \tag{23}$$

If we consider $\tau_m \ll \tau_h$ (a biologically realistic assumption) and for Δt equivalent to τ_m, $h(t)$ can be approximated by a constant, whereas $m(t)$ can vary. We can thus identify m and h separately. With the ΔV_{Mem} stimulation steps used in Fig. 18(a), the measured I_{Na} responses are as shown in Fig. 19(a). These curves attain their maximum values, when the fraction of the activated channel m reaches its steady state value m_∞. As mentioned above, h remains equal to 1, which is its initial value when V_{Mem} starts at -300 mV. From these curves, we can identify the following parameters: $V_{Na} = 193$ mV, $g_{Na} = 83.7\ \mu$S, $V_{Offset_Na_m} = -211.8$ mV, $V_{Slope_Na_m} = 29.3$ mV, and $\tau_m = 0.037$ ms.

Identification of the inactivation term h requires a second set of measurements. The stimulation involves steps, ΔV_{Mem}, which start from different initial values (-500 mV to -150 mV) and finish at an identical final value ($+100$ mV). We apply the initial voltage for 10 ms, after which m and h reach their respective steady states m_∞ and h_∞. Using the same hypothesis, i.e., that $\tau_m \ll \tau_h$, when the final value of V_{Mem} is applied the current I_{Na} should reach its maximum value, for $m = m_\infty = 1$. The inactivation term h remains at its initial value, which depends on the initial value of V_{Mem} (see Fig. 19(b)). We then obtain $V_{Offset_Na_h} = -231.7$ mV, $V_{Slope_Na_h} = 19.0$ mV, and $\tau_h = 0.42$ ms. The hypothesis $\tau_m \ll \tau_h$ is verified.

C. Leak Channel

The last channel for the FS neuron corresponds to the leakage current. Its model, adapted to the electronics variables, is:

$$I_{Leak} = g_{Leak}\,(V_{Mem} - V_{Leak}). \tag{24}$$

We can verify this equation by a simple series of voltage-clamp measurements, since I_{Leak} has a linear dependence on V_{Mem}. The extracted parameters are $V_{Leak} = -626$ mV, $g_{Leak} = 538$ nS.

D. Modulator Channel

The regular spiking neuron contains a modulator channel. This is more straightforward to identify than the sodium

TABLE II
PARAMETER MODEL FOR FS AND RS NEURONS.

Parameters	Programmed biological values	Measured biological values
$V_{Offset_Na_m}$	-37 mV	-42.4 mV
$V_{Slope_Na_m}$	5.0 mV	5.9 mV
τ_{m_Na}	0.014 ms	0.037 ms
$V_{Offset_Na_h}$	-41.1 mV	-46.3 mV
$V_{Slope_Na_h}$	2.5 mV	3.8 mV
τ_{h_Na}	0.157 ms	0.42 ms
g_{Na}	24.7 mS / cm²	16.74 mS / cm²
V_{Na}	50.2 mV	38.6 mV
$V_{Offset_K_n}$	-36.8 mV	-37.3 mV
$V_{Slope_K_n}$	7.7 mV	11.3 mV
τ_{n_K}	1.3 ms	2.5 ms
g_K	25 mS / cm²	21.44 mS / cm²
V_K	-90.0 mV	-98.6 mV
g_{Leak}	73.9 µS / cm²	107.4 µS / cm²
V_{Leak}	-70.0 mV	-125.2 mV
$V_{Offset_Mod_n}$	-36.8 mV	-53.92 mV
$V_{Slope_Mod_n}$	8.4 mV	17.8 mV
τ_{n_Mod}	52.2 ms	19 ms
g_{Mod}	137 µS / cm²	82.4 µS / cm²
V_{Mod}	-100.0 mV	-168.8 mV

channel, because, like the potassium channel, it has only one activation term n. We can thus write the following equation, adapted to the electronics variables:

$$I_{Mod} = g_{Mod}\, n(V_{Mem} - V_{Mod}). \tag{25}$$

The results of the voltage-clamp experiment on the Modulator channel are not shown here, because the process is similar to that used for the K channel. We extracted the following values: $V_{Mod} = -844\,\mathrm{mV}$, $g_{Mod} = 414\,\mathrm{nS}$, $V_{Offset_n} = -269.6\,\mathrm{mV}$, $V_{Slope_n} = 89.0\,\mathrm{mV}$, and $\tau_n = 19\,\mathrm{ms}$.

X. RESULTS AND COMPARISON WITH SIMULATIONS

We simulated the FS and RS neurons using the neural simulation software *Neuron* [42], and translated the model parameters, extracted from the measurements, to their equivalent values for the biological model. The ICs are designed to compute voltages and currents with the following gains: $V_{Mem_ASIC} = 5 \times V_{M_Bio}$ and $t_{ASIC} = 1 \times t_{Bio}$. If we choose $C_{Mem_ASIC} = 5\,\mathrm{nF}$, and since $C_{M_Bio} = 1\ \mu\mathrm{F/cm^2}$, with a membrane area of $22.10^{-5}\ \mathrm{cm^2}$, we obtain a ratio of 22.72 between the hardware and biological conductances. We can thus write $I_{ASIC} = 113.6 \times I_{Bio}$. The resulting parameters for the FS and RS biological neuron model are summarized in Table II.

Before looking into the details of this table, we can observe the effect of a process mismatch on all of the parameters and errors coming from the DAC and analog memories. The time constants for the predefined and measured values are very different, because we use external capacitors with a precision of $\pm 20\%$. The leakage and modulator currents are of the order of hundreds of nA, in a range equivalent to CMOS leakage currents: we cannot evaluate the precision of the parameters extracted using this technique. To solve this problem, we plan to integrate current amplifiers into each current generator, in our next generation of ICs.

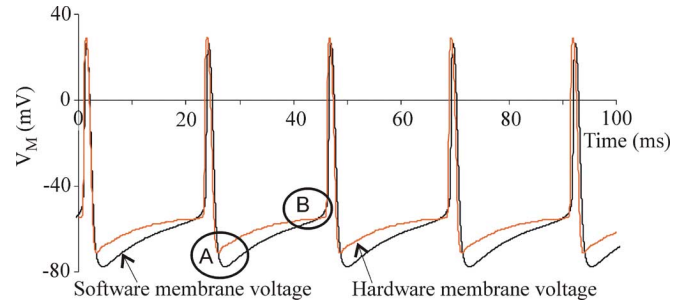


Fig. 20. Software and hardware membrane voltages when the neurons are spiking at 44.3 Hz. Area A: hyperpolarization of the software membrane voltage. Area B: sharp rise of hardware membrane voltage.

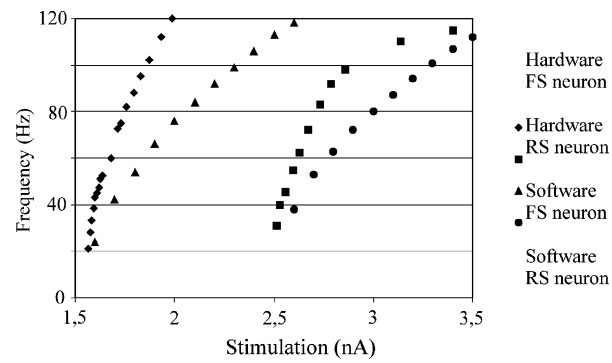


Fig. 21. Frequency vs. stimulation current curves, for FS and RS neurons, from hardware and software simulations.

To compare the shape of the action potentials between software and hardware simulations, we used the extracted parameters and tuned the current stimulation of both neurons so as to achieve the same spiking frequency. Fig. 20 plots the membrane voltage for both simulations, when the FS neuron spikes at 44.3 Hz. The shapes of the hardware and software spikes are similar, in particular the spike widths. Nevertheless, the hardware simulation membrane voltage is characterized by hyperpolarization (shown in area A). This phenomenon arises from the very negative value of E_L: although a value of $-70\ \mathrm{mV}$ could be expected, we extracted $-125.2\ \mathrm{mV}$ and used this value in our software simulations. The area B shows, around the threshold voltage, a difference in the shape of the membrane voltage; the result is smoother in the simulation. We were able to visualize this defect in post layout simulations.

We ran another benchmark test to validate the library of analog operators. In Fig. 21, the frequencies of the action potentials are plotted for different values of a constant stimulation current (I_S). This plot is characteristic of the FS and RS neurons. We observe that theses curves have the same shape, but with a different gain. Once again, the mismatch comes from the estimation errors in the leakage and modulator channels.

These errors do not however compromise the module's functionalities, and we conclude that the library of analog operators is operational.

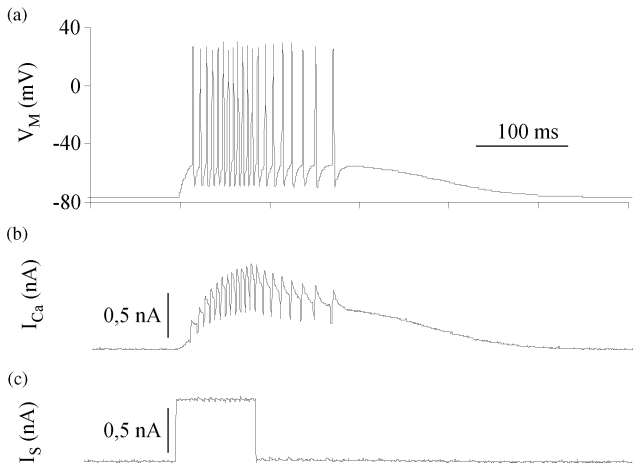


Fig. 22. A 600 ms simulation of a 4 conductance neuron receiving a stimulation pulse. A: Membrane voltage $\mathrm{V_M(t)}$. B: Calcium current $\mathrm{I_{Ca}(t)}$. C: Stimulation current $\mathrm{I_S(t)}$.

XI. Additional Results

A. Calcium Plateau

Here, the ICs simulate the activity of a 4-conductance neuron (sodium, potassium, leak, and calcium). Fig. 22 shows plots of the stimulation current, the calcium current, and the membrane voltage. Before stimulation, the artificial neuron is silent. When the stimulation current is applied, the neuron starts oscillating and the calcium current increases, which in turn raises the oscillation frequency. When the stimulation pulse stops, the oscillations continue until the calcium current is sufficiently weak. Finally, the neuronal activity ceases. With this simulation, we show that the same ionic current generator can be tuned over a large range of parameters, thereby simulating different ionic channels. In the present case, $\mathrm{E_I}$ is tuned to $+100\,\mathrm{mV}$, whereas it was set to $-120\,\mathrm{mV}$ for the modulator channel.

B. Real-Time Interactions

We used the same set of parameters to illustrate two features of the system: real-time simulation and dynamic reconfiguration. Whereas the circuit needs initial values to start the simulation, it is possible to dynamically modify one or more values during the course of the simulation. Fig. 23 shows the electrical activity of a 4-conductance neuron measured at the IC output. From A) to C), the maximum conductance of the calcium channel $\mathrm{g_{Ca}}$ is increased, resulting in continued oscillations, even after stimulation reset.

We performed this change in parameter value without stopping or resetting the simulation; the influence of an individual parameter is therefore easier to visualize. By combining dynamic tuning and membrane electrical potential acquisition, we can use this real-time system to explore with greater precision the different interactions between the model parameters and neuronal activity.

C. Spiking Activity

We added a model of the calcium-dependent potassium channel to the neuron described above. The stimulation current and the neuron's electrical activity can be seen in Fig. 24. When the stimulation current starts, the neuron begins to spike and activates the calcium channel. The calcium channel activates, in turn, the calcium-dependent potassium channel. The calcium conductance tends to increase the spiking frequency, whereas the calcium-dependent potassium conductance tends to decrease it. The calcium-dependent potassium effect is finally predominant, and the spiking frequency decreases, whereas the stimulation current is still present. When the stimulation current stops, the calcium current is not strong enough to maintain the activity; we then observe hyperpolarization of the membrane, while the calcium-dependent potassium channel still provides current. The calcium channel finally becomes inactive, leading to inactivation of the calcium-dependent potassium channel, and the membrane potential returns to its rest state.

If the maximum conductance of the calcium channel is slightly decreased, the membrane voltage does not decrease to hyperpolarization. In this case, the two ionic channels find an equilibrium, which allows the spiking activity of the membrane voltage to be maintained (see Fig. 25).

XII. Applications

In this section, we present two upcoming applications, using the same analog library.

A. Optimization Technique

With the *Pamina* neuromimetic IC, and communications with the host computer via the PCI-bus, we have all the components needed to build a system for determining the full set of model parameters needed to represent a biological neuron.

Parameter extraction by means of optimization algorithms requires the minimization of an error function between 2 signals. This error function compares two membrane potentials, one measured from a biological cell (the reference), and the other from the hardware simulation (the model whose parameters are examined). Optimization algorithms are responsible for error function minimization. An error function in the static temporal domain would have to deal with the arbitrary phase difference, which arises between the two signals. Synchronization of the reference and simulated activities, by adjusting (for example) the stimulations, is not straightforward. We chose to define an error function that compares the membrane potential dynamics rather than their static values. Considering the definition of the mathematical expressions in (1) and (2) for the membrane potentials, we calculated the phase diagram as specified in (26), over a single oscillatory period, regardless of the absolute phase. This approach is possible only if we consider periodic activities, which is the case in the following example.

$$\frac{\mathrm{dV_M}}{\mathrm{dt}} = \mathrm{f(V_M)}. \tag{26}$$

Fig. 26 presents the phase diagram for the case of a three conductance neuron model (sodium, potassium, and leakage; the same model as for FS in Section IX). We plot the phase diagram of the reference and IC activities, which simulate the model with a set of arbitrarily chosen parameters. In the case of this demonstration, the reference activity is taken from an already known model card, obtained from voltage-clamp experiments, and simulated by software. This plot highlights the difference between

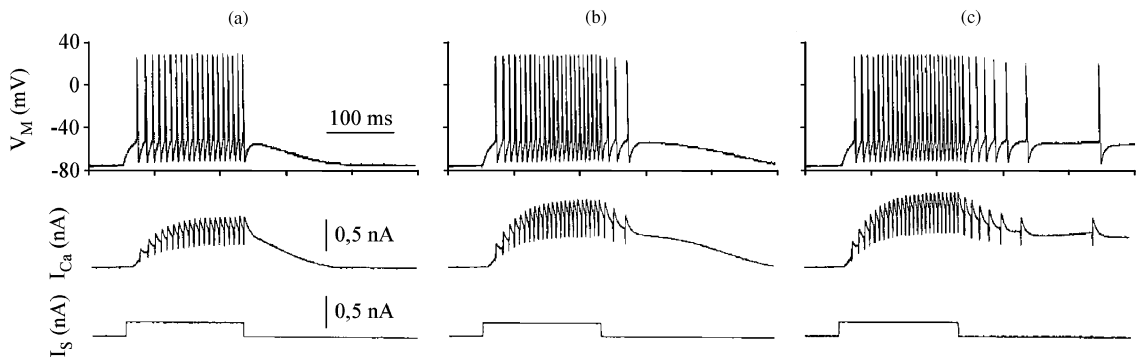


Fig. 23. Effect of the dynamic modification of a parameter during a real-time simulation. Measurements show the effect of parameter change on the membrane voltage. For A), B), and C) and from bottom to top: Stimulation current, calcium current, and membrane voltage. In A) g_{Ca} is the smallest; the neuron electrical activity stops with the stimulation. In B), C), the oscillations are maintained for a duration that increases with g_{Ca}.

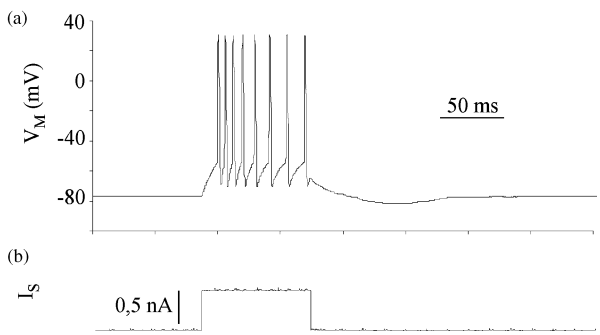


Fig. 24. A 400 ms simulation of a 5 conductance neuron receiving a stimulation pulse. A) Membrane potential $V_M(t)$. B) Stimulation current $I_S(t)$. The membrane potential exhibits hyperpolarization when the stimulation current stops.

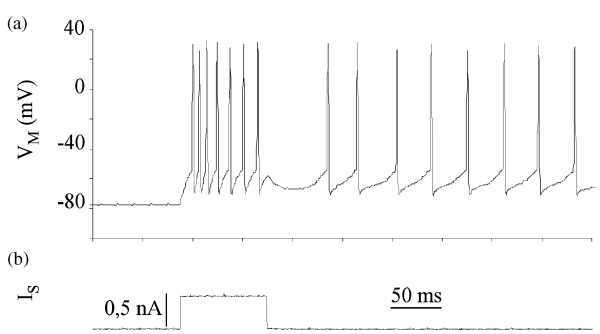


Fig. 25. A 500 ms simulation of a 5 conductance neuron receiving a stimulation pulse. A) Membrane potential $V_M(t)$. B) Stimulation current $I_S(t)$. The membrane potential exhibits tonic activity when the stimulation current stops.

the two electrical activities, showing that the IC parameters are not well tuned to the reference model. The optimization technique minimizes an error function, to obtain closer trajectories in the phase diagram. [43] discusses the definition of an error function, and the choice of optimization algorithms.

The main advantages of this technique are that it simplifies the process of extracting a neuron model, when compared to the complex and time-consuming voltage-clamp experiments. It also provides the opportunity of systematically exploring the parameter space of the studied model. With this technique, the neuroscientist only needs to measure the membrane voltage. The parameter exploration set will provide neuroscientists with (possibly more than one) mathematically possible solutions, from which the most realistic model card can be retained. Although

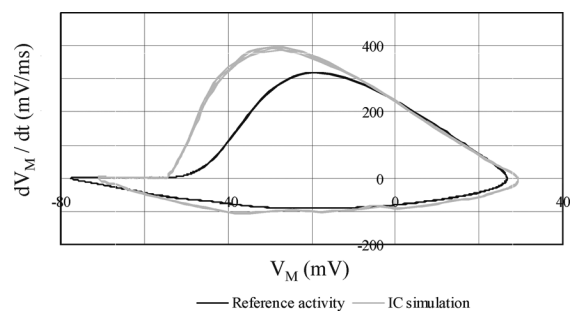


Fig. 26. Phase diagrams ($dV_M/dt = f(V_M)$) of the software and the IC simulations, for a 3-conductance neuron exhibiting a spiking activity (1 cycle = 1 action potential).

this method is currently limited to periodic activities, we are studying its adaptation to non-periodic applications.

B. Increase of Computational Speed

Analog neuromimetic ICs have a considerable advantage over digital implementations: their computational speed can easily be increased. This is possible, provided we remain within the bandwidth of the circuit. As neural activities occur at low frequencies (< 1 kHz), circuits designed with BiCMOS technology have operating frequencies higher than real-time neural activities. In expression (1), if we arbitrarily divide the value of the membrane capacitor by a term α, and divide all the time constants of the activation and inactivation variables by the same term α, the simulation time scale is immediately divided by α (i.e., t in expression (1) is replaced by t/α). This feature is not shared with digital implementation because the software changes also the computational step to solve the equations, and it lasts the same computational time. In Fig. 27, we show test measurements made with the *Pamina* ASIC, for the following cases: $\alpha = 1$, 10, and 100. Each simulation is represented with its own time scale t/α. When $\alpha = 10$, the action potential shape is almost identical to the reference shape ($\alpha = 1$), and the activity frequency increases slightly. When $\alpha = 100$, noise can be observed during the depolarizing phase of the spike, before the action potential. These results show that, although the *Pamina* IC was not designed to enable computational speeds to be increased, an optimal factor can be found which accelerates the simulations, while maintaining an activity identical to the reference one. Application of this property to the software/hardware optimization technique will reduce simulation times; this

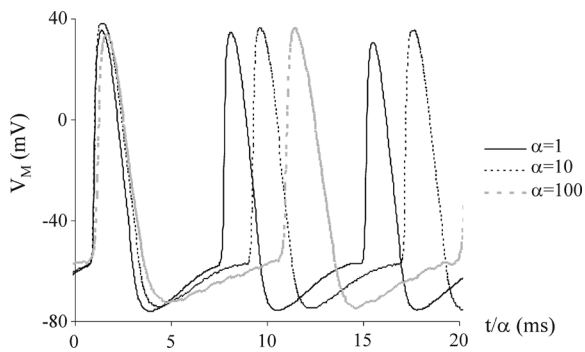


Fig. 27. Membrane voltage for a 3-conductance neuron, when increasing the IC simulation speed, as a function of the term α.

can be a key issue when exploring a large set of parameters in the Hodgkin-Huxley models. In this case, the sub-circuits in the analog library will also be characterized by their respective maximum values of α. The minimum of all the sub-circuit α values implemented in a neuromimetic IC will give an indication of the maximum accelerated time scale, which can be achieved.

XIII. Summary and Outlook

In this paper we have presented a prototype analog neuromimetic IC, which validates the functionality of the library of analog circuit operating functions of the Hodgkin-Huxley formalism, to describe biologically realistic neurons. This library will be enriched with current amplifiers (see Section X), and additional circuits to simulate neuronal ionic currents that have a complex dependence on the ionic species. The potential increase in speed, compared to biological real-time, can also be specified for each module (see Section XII.B.).

We propose that such a library is a good starting point to design neuromimetic ICs acting the computational cores in a simulation platform for conductance-based neural networks. Thanks to its real-time, such a system is a good candidate for supporting experiments on hybrid neural networks combining biological and silicon neural networks. The prototype IC presented in this paper demonstrates the use of the library. In its current state, such an analog and custom IC is less performing than a digital solution (FPGA or stand-alone processor) in terms of design cost and power consumption.

The main advantage of the analog implementation of neural network models, compared to their numerical simulation, arises from the locally analog and parallel nature of the computations. In addition to the facilitation of analog connections to biological systems, this leads to neuromorphic network models being typically highly scalable and being able to emulate neural networks in real time or much faster, independent of the underlying network size [44]. It is difficult to quantify the effective gain of this hardware system in terms of computation time. Running times are specified for some supercomputer installations such as *Lenngren* from Dell or *BlueGene/L* from IBM, which declared goal is to simulate substantial parts of a mammal brain [45]. We did not found in the literature experiments in a context equivalent to ours (conductance-based models running on standard computers or ASICs) giving complete information about the simulation experiments: simulation time depends on the simulation software (or hardware), the operating system, and the computer architecture, processors and memory. [46] is for example a recent digital implementation of conductance-based models. The authors mention a "standard personal computer" under Windows XP to compute in real-time 1000 conductances with no more details. However, we did software simulations that gave us some comparison points. The first experiment is the simulation of a two-neuron oscillator, as part of the leech heartbeat system [47], where we simplified the neuron models down to 7 conductances by neuron (synapses are equivalent to 1 additional conductance). The simulation was done using Scilab 5.0.2 with a CPU Intel Core 6600, 2.4 GHz, 2 GB RAM, and Windows Vista 32 bits. The simulation of 1 s in biological time ran during 42.26 s with a time step of 1 ms. A second experiment was done with a network of 6 excitatory neurons (4-conductance neuron) with STDP [48] using Neuron software with a Core 2 CPU, 2.13 GHz, 3.5 GB RAM, and GNU/Linux Ubuntu 6.10. The simulation of 6 minutes in biological time took about 3 hours. In this case, the STDP computation is time consuming, but this comparison remains interesting, considering that hardware systems can implement STDP using a digital or mixed computation, as proposed in another version of our system [49]. We feel then that analog implementation can be part of the solution when building large networks with adaptation functions at the synaptic level: it can then preserve the digital computational power for highly configurable functions such as plasticity. For networks with several hundreds of cells, the parallel computation mode available with analog blocks is clearly an advantage and will lead to embedded solutions easily connected to analog sensors and actuators. An ongoing study in our group proposes a method to optimize re-use and synthesis when designing analog ICs [50]. It uses a library of functions based to the one presented here and completed by the VHDL-AMS description of each block. A reuse-based design mode will also support the successive technological migrations of the library.

We have already used this library to design another IC dedicated to the real-time hardware simulation of medium-sized neural networks (up to 128 neurons). This IC includes 5 analog cores, synapses, and digital functions to manage the analog components [51]. We are also working on the design of an embedded simulation platform merging analog conductance-based silicon neurons and digital adaptive synapses [52]. Conductance-based models provide enough complexity to allow the exploration of complex activity patterns or adaptation sequences in a small scale neural network.

Acknowledgment

The first author thanks Professor H. Chen (National Tsing-Hwa University – Taiwan) for his helpful questions, and his encouragement to write this paper.

References

[1] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.
[2] R. Douglas, M. Mahowald, and C. Mead, *Neuromorphic Analogue VLSI*. New-York: Oxford Univ. Press, 1995, p. 701.
[3] T. S. Lande, "Editorial," *Analog Integr. Circuits Signal Process.*, vol. 13, no. 1-2, pp. 5–7, 1997.
[4] K. A. Boahen, "A retinomorphic vision system," *IEEE Micro*, vol. 16, no. 5, pp. 30–39, Oct. 1996.

[5] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, "A biomorphic digital image sensor," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.

[6] P. Lichtsteiner, C. Posch, and T. Delbrück, "A 128 × 128 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 566–576, Feb. 2008.

[7] J. Lazzaro and J. Wawrzynek, "Silicon models for auditory scene analysis," in *Advances in Neural Information Processing Systems5 (NIPS95)*. Cambridge, MA: MIT Press, 1995.

[8] A. Van Schaik and S. Shamma, "A neuromorphic sound localizer for a smart MEMS system," *Analog Integr.Circuit Signal Process.*, vol. 39, no. 3, pp. 267–273, 2004.

[9] G. Indiveri, "Neuromorphic selective attention systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, Bangkok, Thailand, 2003, pp. 770–773.

[10] C. Dioro, D. Hsu, and M. Figueroa, "Adaptive CMOS: From biological inspiration to system-on-a-chip," *Proc. IEEE*, vol. 90, no. 3, pp. 345–357, Mar. 2002.

[11] M. Mahowald and R. Douglas, "A silicon neuron," *Nature*, vol. 354, pp. 515–518, 1991.

[12] T. Delbrück and S. C. Liu, "A silicon early visual system as a model animal," *Vis. Res.*, vol. 44, no. 17, pp. 2083–2089, 2004.

[13] K. A. Zaghoul and K. Boahen, "Optic nerve signals in a neuromorphic chip II: Testing and results," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 667–675, Apr. 2004.

[14] M. F. Simoni, G. S. Cymbalyuk, M. E. Sorensen, R. L. Calabrese, and S. P. DeWerth, "A multiconductance silicon neuron with biologically matched dynamics," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 2, pp. 342–354, Feb. 2004.

[15] M. F. Simoni and S. P. DeWerth, "Two-dimensional variation of bursting properties in a silicon-neuron half-center oscillator," *IEEE Trans. Neural Sys. Rehab. Eng.*, vol. 14, no. 3, pp. 281–289, Sep. 2006.

[16] F. Tenore, R. Etienne-Cummings, and M. A. Lewis, "Entrainment of silicon central pattern generators for legged locomotory control," in *Advances in Neural Information Processing systems (NIPS04)*. Cambridge, MA: MIT Press, 2004.

[17] R. L. Calabrese, "Half-center oscillators underlying rhythmic movements," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 444–447.

[18] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in Neural Information Processing Systems (NIPS05)*. Cambridge, MA: MIT Press, 2005.

[19] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array to low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, Jan. 2006.

[20] Q. Zou, Y. Bornat, J. Tomas, S. Renaud, and A. Destexhe, "Real-time simulations of networks of Hodgking-Huxley neurons using analog circuits," *Neurocomputing*, vol. 69, pp. 1137–1140, Jan. 2006.

[21] L. Avaldo, J. Tomas, S. Saïghi, S. Renaud-Le Masson, T. Bal, A. Destexhe, and G. Le Masson, "Hardware computation of conductanced-based models," *Neurocomputing*, vol. 58–60, pp. 109–115, 2004.

[22] S. Saïghi, J. Tomas, Y. Bornat, and S. Renaud, "A conductance-based silicon neuron with dynamically tunable model parameters," in *Proc. 2nd Int. IEEE EMBS Conf. Neural Engineering*, Arlington, VA, Mar. 2005, pp. 285–288.

[23] Q. Zou, Y. Bornat, S. Saïghi, J. Tomas, S. Renaud, and A. Destexhe, "Analog-digital simulations of full conductance-based networks of spiking neurons," *Netw.: Comput. Neural Syst.*, vol. 17, no. 3, pp. 211–233, Sep. 2006.

[24] G. Le Masson, S. Renaud-Le Masson, D. Debay, and T. Bal, "Feedback inhibition controls spike transfer in hybrid thalamic circuits," *Nature*, vol. 417, pp. 854–858, Jun. 2002.

[25] K. Hynna and K. Boahen, "Neuron ion-channel dynamics in silicon," in *Proc. IEEE Int. Symp. Circuits and Systems*, Island of Kos, Greece, 2006, pp. 3614–3617.

[26] E. Farquhar and P. Hasler, "A bio-physically inspired silicon neuron," *IEEE Trans. Circuits and Systems I, Reg. Papers*, vol. 52, no. 3, pp. 477–488, Mar. 2005.

[27] J. Georgiou, E. M. Drakakis, C. Toumazou, and P. Premanoj, "An analogue micropower log-domain silicon circuit for the Hodgkin and Huxley nerve axon," in *Proc. Int. Symp. Circuits and Systems*, Orlando, FL, 1999, pp. 286–289.

[28] R. J. Vogelstein, U. Mallik, J. T. Volgenstein, and G. Cauwenberghs, "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 253–265, Jan. 2007.

[29] J. Schemmel, D. Bruederle, K. Meier, and B. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *Proc. IEEE Int. Symp. Circuits and Systems*, New Orleans, LA, May 2007, pp. 3367–3370.

[30] A. L. Hodgkin and F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, pp. 500–544, 1952.

[31] C. Koch, "Beyond Hodgkin and Huxley: Calcium and calcium-dependent potassium currents," in *Biophysics of Computation*. New York: Oxford Univ. Press, 1999, pp. 212–231.

[32] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski, "An efficient method for computing synaptic conductances based on a kinetic model of receptor binding," *Neural Comput.*, vol. 6, pp. 14–18, 1994.

[33] [Online]. Available: http://senselab.med.yale.edu/modeldb/

[34] C. Toumazou, F. G. Lidgey, and D. G. Haigh, "Introduction," in *Analogue IC Design: The Current-Mode Approach*. London, U.K.: Peregrinus, 1990, pp. 1–10.

[35] B. Gilbert, "Current-mode circuits from translinear viewpoint: A tutorial," in *Analogue IC Design: The Current-Mode Approach*. London, U.K.: Peregrinus, 1990, pp. 11–91.

[36] P. E. Allen and D. G. Holberg, "CMOS amplifiers," in *CMOS Analog Circuit Design*. New York: Oxford Univ. Press, 1987, pp. 257–321.

[37] Y. Tietze and C. Schenck, "Operational amplifiers applications," in *Electronics Circuits: Handbook for Design and Applications*, 2nd ed. New York: Springer, 2008, pp. 725–765.

[38] Y. Tietze and C. Schenck, "Analog switches and sample-and-hold circuits," in *Electronics Circuits: Handbook for Design and Applications*, 2nd ed. New York: Springer, 2008, pp. 929–944.

[39] A. Hastings, *The Art of Analog Layout*. Upper Saddle River, NJ: Prentice-Hall, 2000, p. 576.

[40] K. S. Cole, *Membranes, Ions, and Impulses: A Chapter of Classical Biophysics*. Berkeley, CA: Univ. California Press, 1968, p. 569.

[41] M. Pospischil, M. Toledo-Rodriguez, C. Monier, Z. Piwkowska, T. Bal, Y. Frégnac, H. Markram, and A. Destexhe, "Minimal Hodgkin-Huxley type models for different classes of cortical and thalamic neurons," *Biol. Cybern.*, vol. 99, no. 4-5, pp. 427–441, 2008.

[42] N. T. Carnevale and M. L. Hines, *The NEURON Book*. Cambridge, U.K.: Cambridge Univ. Press, 2006, p. 479.

[43] G. Le Masson and R. Maex, "Introduction to equation solving parameter fitting," in *Computational Neuroscience: Realistic Modeling for Experimentalists*. Boca Raton, FL: CRC, 2001, pp. 1–23.

[44] D. Brüderle, E. Müller, A. Davison, E. Muller, J. Schemmel, and K. Meier, "Etablishing a novel modeling tool: A python-based interface for a neuromorphic system," *Frontiers in Neuroinformatics*, vol. 3, no. Art. 17, 2009.

[45] M. Djurfeldt, C. Johanson, Ö. Ekeberg, M. Rehn, M. Lundqvist, and A. Lansner, "Massively parallel simulation of brain-scale neuronal network models," 2005. [Online]. Available: www-03.ibm.com/systems/resources/systems_ deepcomputing_pdf_Blue_Gene_Applications_Paper_KTH_0306.pdf

[46] S. W. Hughes, M. Lorincz, D. W. Cope, and V. Crunelli, "NeuReal: An interactive simulation system for implementing artificial dendrites and large hybrid networks," *J. Neurosci. Meth.*, vol. 169, pp. 290–301, 2008.

[47] A. A. V. Hill, J. Lu, M. A. Masino, O. H. Olsen, and R. L. Calabrese, "A model of a segmental oscillator in the leech heartbeat neuronal network," *J. Comput. Neurosci.*, vol. 10, pp. 281–302, 2001.

[48] A. Daouzli, "Systèmes neuromorphiques: Etude et implantation de fonctions d'apprentissage et de plasticité," Ph.D. dissertation, Ecole doctorate des Sciences Physiques et de l'Ingénieur, Bordeaux 1 Univ., Bordeaux, France, 2009.

[49] S. Renaud, J. Tomas, N. Lewis, Y. Bornat, A. Daouzli, M. Rudolph, A. Destexhe, and S. Saïghi, "PAX: A mixed hardware/software simulation platform for spiking neural networks," *Neural Netw.*, to be published.

[50] T. Lévi, N. Lewis, J. Tomas, and P. Fouillat, "IP-based methodology for analog design flow: Application on neuromorphic engineering," in *Proc. IEEE NEWCAS-TAISA Conf.*, Montréal, QC, Canada, 2008, pp. 343–346.

[51] J. Tomas, Y. Bornat, S. Saïghi, T. Lévi, and S. Renaud, "Design of a modular and mixed neuromimetic ASIC," in *Proc. IEEE Int. Conf. Electronics, Circuits and Systems*, Nice, France, 2006, pp. 946–949.

[52] Y. B. J. Tomas, C. Lopez, O. Malot, B. Belhadj, and S. Renaud, "Design of analog/digital simulator dedicated to real-time neurocomputing," in *Proc. XXII Conf. Design of Circuits and Integrated Systems*, Sevilla, Spain, 2007, pp. 391–396.

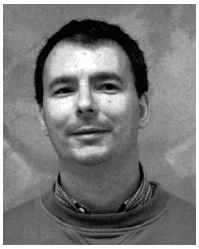
Sylvain Saïghi (M'01) received the M.Sc. degree in electronics, electrotechnical engineering and automatics from University Bordeaux 1, Bordeaux, France, in 1998, the M.Sc. degree in cognitive sciences from University Victor Segalen Bordeaux 2, Bordeaux, France, in 2004, and the Ph.D. degree in electronics from University Bordeaux 1, in 2005.

Currently, he is Associate Professor in the IMS Laboratory (UMR 5218 CNRS/University of Bordeaux) where he has been the head of the "Architecture of Silicon Neural Networks" group since 2010. The group's research focuses on analog circuits and systems as artificial neural networks. He has also authored and co-authored more than 30 peer-review publications. His current research is focused on spike timing–dependent plasticity in learning and on optimization algorithms to identify single neuron biophysic parameters. During training at the "Ecole Normale Supérieure de Cachan," France, he obtained the "Agrégation de Génie Electrique" in 1999.

Dr. Saïghi received the Fulbright Scholar grant in 2010.

Yannick Bornat received the M.Sc. degree in electrical engineering and the Ph.D. degree in electronics from the University of Bordeaux, Bordeaux, France, in 2003 and 2006, respectively.

After a postdoctoral position at the Institute of Microtechnology (IMT), University of Neuchâtel, Neuchâtel, Switzerland, in 2007, he joined the IMS Laboratory at the University of Bordeaux, Talence, France (integration from materials to systems), where he is currently an Associate Professor. His current research interests include real-time hardware computing for experiments mixing electronics and biology.



Jean Tomas received the Ph.D. degree in electrical engineering from University Bordeaux 1, Bordeaux, France, in 1988.

Currently, he is Associate Professor with the IMS Laboratory at the University of Bordeaux, Talence, France (UMR 5218 CNRS/Université de Bordeaux). He has also authored and coauthored more than 50 peer-review publications. His research interests include the design of analog and mixed–signal circuits and systems dedicated to neuromorphic applications.

Gwendal Le Masson received the M.D. and Ph.D. degrees from the University of Bordeaux, Bordeaux, France, in 1994 and 1998, respectively.

He was then recruited as a Researcher in neuroscience at the Institut National de la Santé et de la Recherche Médicale (INSERM). Since 2002, he has been a Tenured Professor in Clinical Neuroscience at Bordeaux University Hospital (CHU Pellegrin-Bordeaux). In 2008–2009, he was an Invited Professor at Columbia University, New York, for the academic year. He conducts his research activities in the Neurocentre Magendie, in the team Pathophysiology of motoneurons. His research interests are computational neurosciences, synaptic plasticity, and amyotrophic lateral sclerosis. He has authored and coauthored many international articles, including publications in *Science*, *Nature*, and in *Nature Neuroscience*.

Sylvie Renaud (M'09) received the M.Sc. degree in electronic engineering from Supelec, Paris, France, in 1986 and the Ph.D. degree in physics from the University of Bordeaux, Bordeaux, France, in 1990, and the Research Habilitation (HDR) in 2001.

After a postdoctoral stay at Brandeis University, MA, in 1991–1992, she was appointed Assistant Professor and then to Professor at ENSEIRB Bordeaux (National Engineering School), where she was recently appointed Research Board Director in the "Institut Polytechnique de Bordeaux." In 1994, she created the Engineering of Neuromorphic Systems Group at IMS-Labs University Bordeaux (CNRS, ENSEIRB), and now heads the Bioelectronics Group with 12 permanent researchers. Her research interests are analog and mixed neuromorphic very-large scale integration (VLSI) systems; real-time hardware simulation platforms of spiking neural networks; hybrid systems interfacing living and artificial neurons; analog application-specific integrated circuits for biological signal conditioning and events detection; active VLSI implants for neurodegenerative diseases and diabetes; and closed-loop living-artificial systems. She has authored and coauthored more than 50 reviewed international articles and communications.

Prof Renaud has been a reviewer for IEEE journals and conferences since 1997. She organized one Special Session in ISCAS2007 and ran a Tutorial in ISCAS2010.