CONTRIBUTED PAPER

Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems

This paper proposes a set of neuromorphic engineering solutions to address the challenge of building low-power compact physical artifacts that can behave intelligently in the real world and exhibit cognitive abilities.

By Elisabetta Chicca, *Member IEEE*, Fabio Stefanini, Chiara Bartolozzi, *Member IEEE*, and Giacomo Indiveri, *Senior Member IEEE*

ABSTRACT | Several analog and digital brain-inspired electronic systems have been recently proposed as dedicated solutions for fast simulations of spiking neural networks. While these architectures are useful for exploring the computational properties of large-scale models of the nervous system, the challenge of building low-power compact physical artifacts that can behave intelligently in the real world and exhibit cognitive abilities still remains open. In this paper, we propose a set of neuromorphic engineering solutions to address this challenge. In particular, we review neuromorphic circuits for emulating neural and synaptic dynamics in real time and discuss the role of biophysically realistic temporal dynamics in hardware neural processing architectures; we review the challenges of realizing spike-based plasticity mechanisms in real physical systems and present examples of analog electronic circuits that implement them; we describe the computational properties of recurrent neural networks and show how neuromorphic winner-take-all circuits can implement working-memory and decision-making mechanisms. We validate the neuromorphic approach proposed with experimental results obtained from our own circuits and systems, and argue how the circuits and networks presented in this work represent a useful set of components for efficiently and elegantly implementing neuromorphic cognition.

KEYWORDS | Cognitive systems; learning systems; neuromorphic engineering; real-time neuromorphic systems; spike-timing-dependent plasticity (STDP); spiking neural network architecture; subthreshold analog circuits; very large-scale integration (VLSI); winner-take-all (WTA)

I. INTRODUCTION

Machine simulation of cognitive functions has been a challenging research field since the advent of digital computers. However, despite the large efforts and resources dedicated to this field, humans, mammals, and many other animal species including insects still outperform the most powerful computers in relatively routine functions such as sensory processing, motor control, and pattern recognition. The disparity between conventional computing technologies and biological nervous systems is even more pronounced for tasks involving autonomous real-time interactions with the environment, especially in presence of noisy and uncontrolled sensory input. One important aspect is that the computational and organizing principles followed by the nervous system are fundamentally different from those of present day computers. Rather than using Boolean logic, precise digital representations, and clocked operations, nervous systems carry out robust and reliable computation using hybrid analog/digital unreliable processing elements; they emphasize distributed, event-driven, collective, and massively parallel mechanisms and make extensive use of adaptation, self-organization, and learning.

Manuscript received December 3, 2012; revised October 17, 2013 and March 6, 2014; accepted March 12, 2014. Date of publication May 1, 2014; date of current version August 18, 2014. This work was supported by the European Union (EU) European Research Council (ERC) under Grant "neuroP" (257219); the EU Future & Emerging Technologies (FET) program under Grant "SI-CODE" (284553); and the Excellence Cluster 227 [Cognitive Interaction Technology—Center of Excellence (CITEC), Bielefeld University].

E. Chicca is with the Cognitive Interaction Technology-Center of Excellence, Bielefeld University and Faculty of Technology, Bielefeld 33615, Germany (e-mail: chicca@cit-ec.uni-bielefeld.de).

F. Stefanini and **G. Indiveri** are with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich 8092, Switzerland.

C. Bartolozzi is with the iCub Facility, Istituto Italiano di Tecnologia, Genova 16163, Italy.

Digital Object Identifier: 10.1109/JPROC.2014.2313954

Several approaches have been recently proposed for building custom hardware, brain-like neural processing architectures [1]–[9]. The majority of them are proposed as an alternative electronic substrate to traditional computing architectures for neural simulations [2], [4], [5], [7]. These systems can be very useful tools for neuroscience modeling, e.g., by accelerating the simulation of complex computational neuroscience models by three or more orders of magnitude [4], [7], [10]. However, our work focuses on an alternative approach aimed at the realization of compact, real-time, and energy-efficient computational devices that directly emulate the style of computation of the brain, using the physics of silicon to reproduce the biophysics of the neural tissue. This approach, on the one hand, leads to the implementation of compact and low-power behaving systems ranging from brain–machine interfaces to autonomous robotic agents. On the other hand, it serves as a basic research instrument for exploring the computational properties of the neural system they emulate and hence gain a better understanding of its operational principles. These ideas are not new: they follow the original vision of Mead [11], Mahowald [12], and Douglas *et al.* [13]. Indeed, analog complementary metal–oxide–semiconductor (CMOS) technology has been effectively employed for the construction of simple neuromorphic circuits reproducing basic dynamical properties of their biological counterparts, e.g., neurons and synapses, at some level of precision, reliability, and detail. These circuits have been integrated into very large-scale integration (VLSI) devices for building real-time sensory-motor systems and robotic demonstrators of neural computing architectures [14]–[19]. However, these systems, synthesized using *ad hoc* methods, could only implement very specific sensory-motor mappings or functionalities. The challenge that remains open is to bridge the gap from designing these types of reactive artificial neural modules to building complete neuromorphic behaving systems that are endowed with cognitive abilities. The step from reaction to cognition in neuromorphic systems is not an easy one, because the principles of cognition remain to be unraveled. A formal definition of these principles and their effective implementation in hardware is now an active domain of research [20]–[23]. The construction of brain-like processing systems able to solve cognitive tasks requires sufficient theoretical grounds for understanding the computational properties of such a system (hence its necessary components), and effective methods to combine these components in neuromorphic systems. During the last decade, we pursued this goal by realizing neuromorphic electronic circuits and systems and using them as building blocks for the realization of simple neuromorphic cognitive systems [20]. Here we describe these circuits, analyze their dynamics in comparison with other existing solutions, and present experimental results that demonstrate their functionalities. We describe the limitations and problems of such circuits, and propose effective design strategies for building larger brain-like processing systems. We conclude with a discussion on the advantages and disadvantages of the approach we followed and with a description of the challenges that need to be addressed in order to progress in this domain. Specifically, in Sections III–VI, we show how the building blocks we propose, based on dynamic synapse circuits, hardware models of spiking neurons, and spike-based plasticity circuits, can be integrated to form multichip spiking recurrent and winner-take-all neural networks, which in turn have been proposed as neural models for explaining pattern recognition [24], [25], working memory [9], [26], decision making [27], [28], and state-dependent computation [29], [30] in the brain.

II. NEURAL DYNAMICS IN ANALOG VLSI

Unlike a von Neumann computing architecture, neuromorphic architectures are composed of massively parallel arrays of simple processing elements in which memory and computation are colocalized. In these architectures, time represents itself and so the synapse and neuron circuits must process input data on demand, as they arrive, and must produce their output responses in real time. Consequently, in order to interact with the environment and process real-world sensory signals efficiently, neuromorphic behaving systems must use circuits that have biologically plausible time constants (i.e., of the order of tens of milliseconds). In this way, they are well matched to the signals they process and are inherently synchronized with the real-world events. This constraint is not easy to satisfy using analog VLSI technology. Standard analog circuit design techniques either lead to bulky and silicon-area expensive solutions [31] or fail to meet this condition, resorting to modeling neural dynamics at "accelerated" unrealistic time scales [32], [33].

One elegant solution to this problem is to use current-mode design techniques [34] and log-domain subthreshold circuits [35]–[39]. When metal–oxide–semiconductor field-effect transistors (MOSFETs) are operated in the subthreshold domain, the main mechanism of carrier transport is that of diffusion, as it is for ions flowing through proteic channels across neuron membranes. As a consequence, MOSFETs have an exponential relationship between gate-to-source voltage and drain current, and produce currents that range from femto- to nano-Ampères. As the time constants of log-domain circuits are inversely proportional to their reference currents, in addition to being directly proportional to the circuit capacitance, the subthreshold domain allows the integration of relatively small capacitors in VLSI to implement temporal filters that are both compact and have biologically realistic time constants, ranging from tens to hundreds of milliseconds.

Neuron conductance dynamics and synaptic transmission can be faithfully modeled by first-order differential

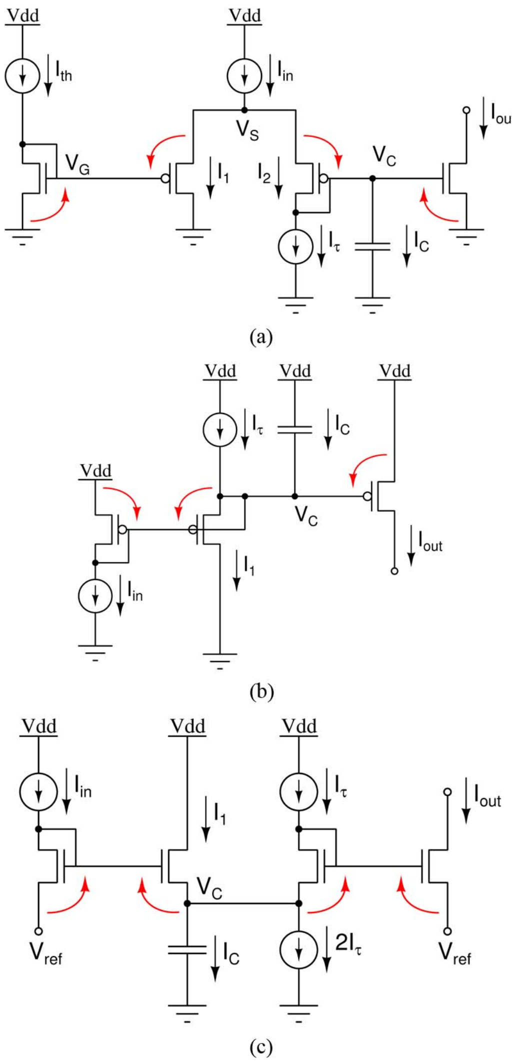


Fig. 1. ***Current-mode LPF circuits. Red arrows show the translinear loop considered for the log-domain analysis. (a) The DPI circuit diagram. (b) The LPF circuit diagram. (c) The "tau-cell" circuit diagram.***

equations [40], therefore subthreshold log-domain circuits that implement first-order low-pass filters (LPFs) can faithfully reproduce biologically plausible temporal dynamics. Several examples of such circuits have been proposed as basic building blocks for the implementation of silicon neurons and synapses. Among them, the differential pair integrator (DPI) [41], [42], the log-domain LPF [43], and the "tau-cell" [44] circuits offer a compact and low-power solution. These circuits, shown in Fig. 1, can be analyzed by applying the translinear principle, whereby the sum of voltages in a chain of transistors that obey an exponential current–voltage characteristic can be expressed as multiplication of the currents flowing across them [45]. For example, if we consider the DPI circuit of Fig. 1(a), and we assume that all transistor have same parameters and operate in the subthreshold regime and in saturation [37], we can derive circuit solution analytically. Specifically, we can write

$$\begin{aligned} I_{\text{out}} &= I_0 e^{\frac{\kappa V_C}{U_T}} \quad I_C = C\frac{d}{dt}V_C \\ I_{\text{in}} &= I_1 + I_2 \quad I_2 = I_\tau + I_C \end{aligned} \tag{1}$$

where I_0 represents the transistor dark current, U_T represents the thermal voltage, and κ represents the subthreshold slope factor [37]. By applying the translinear principle across the loop made by the arrows in the circuit diagram of Fig. 1(a), we can write: $I_{th} \cdot I_1 = I_2 \cdot I_{\text{out}}$. Then, by replacing I_1 and expanding I_2 from (1), we get

$$I_{th} \cdot (I_{\text{in}} - I_\tau - I_C) = (I_\tau + I_C) \cdot I_{\text{out}}. \tag{2}$$

Thanks to the properties of exponential functions, we can express I_C as a function of I_{out}

$$I_C = C\frac{U_T}{\kappa I_{\text{out}}}\frac{d}{dt}I_{\text{out}}. \tag{3}$$

Finally, by replacing I_C from this equation and dividing everything by I_τ in (2), we get

$$\tau\left(1 + \frac{I_{th}}{I_{\text{out}}}\right)\frac{d}{dt}I_{\text{out}} + I_{\text{out}} = \frac{I_{th}I_{\text{in}}}{I_\tau} - I_{th} \tag{4}$$

where $\tau \overset{\Delta}{=} CU_T/\kappa I_\tau$.

This is a first-order nonlinear differential equation that cannot be solved explicitly. However, in the case of sufficiently large input currents (i.e., $I_{\text{in}} \gg I_\tau$), term $-I_{th}$ on the right-hand side of (4) can be neglected. Furthermore, under this assumption and starting from an initial condition $I_{\text{out}} = 0$, I_{out} will increase monotonically and eventually condition $I_{\text{out}} \gg I_{th}$ will be met. In this case, also term I_{th}/I_{out} on the left-hand side of (4) can be neglected. So the response of the DPI reduces to a first-order linear differential equation

$$\boxed{\tau\frac{d}{dt}I_{\text{out}} + I_{\text{out}} = \frac{I_{th}}{I_\tau}I_{\text{in}}} \tag{5}$$

The general solution of the other two log-domain circuits shown in Fig. 1(b) and (c) can be derived analytically following a similar procedure. Table 1 shows the equations used for the derivation of all three circuits, and their general solution.

The LPF circuit of Fig. 1 is the one that has the least number of components. However, it is not the most compact, because, to apply the translinear principle correctly,

Table 1 Characteristic Equations of the DPI, LPF, and Tau-Cell Log-Domain Filters

DPI	LPF	Tau-Cell
Circuit equations		
$I_{out} = I_0 e^{\frac{\kappa V_C}{U_T}}$	$I_{out} = I_0 e^{\frac{\kappa(V_{dd}-V_C)}{U_T}}$	$I_{out} = I_0 e^{\frac{\kappa V_2 - V_{ref}}{U_T}}$
$I_C = C\frac{dV_C}{dt}$	$I_C = -C\frac{dV_C}{dt}$	$I_C = C\frac{dV_C}{dt}$
$I_{in} = I_1 + I_\tau + I_C$	$I_1 = I_\tau + I_C$	$I_1 = I_\tau + I_C$
$I_C = C\frac{U_T}{\kappa I_{out}}\frac{dI_{out}}{dt}$	$I_C = C\frac{U_T}{\kappa I_{out}}\frac{dI_{out}}{dt}$	$I_C = C\frac{U_T}{I_{out}}\frac{dI_{out}}{dt}$
Translinear Loop		
$I_{th} \cdot I_1 = (I_\tau + I_C) \cdot I_{out}$	$I_{in} \cdot I_0 = I_1 \cdot I_{out}$	$I_{in} \cdot I_\tau = I_1 \cdot I_{out}$
Solution		
$\tau\frac{dI_{out}}{dt} + I_{out} = \frac{I_{th}}{I_\tau}I_{in}$	$\tau\frac{dI_{out}}{dt} + I_{out} = \frac{I_0}{I_\tau}I_{in}$	$\tau\frac{dI_{out}}{dt} + I_{out} = I_{in}$
$\tau = \frac{CU_T}{\kappa I_\tau}$	$\tau = \frac{CU_T}{\kappa I_\tau}$	$\tau = \frac{CU_T}{I_\tau}$

it is necessary to use a *p*-type field-effect transistor (FET) with its bulk connected to its source node [see *p*-FET with I_1 current flowing through it in Fig. 1(b)]. This requires an isolated well in the circuit layout, which leads to larger area usage, and makes the overall size of the circuit comparable to the size of the other two solutions. Furthermore, the requirement of an isolated well for the *p*-FET does not allow the design of the complementary version of the circuit in standard CMOS processes (e.g., to have negative currents). The tau-cell circuit does not have this problem, but it requires precise matching of the two current sources producing I_τ and $-2I_\tau$, which can also lead to large area usage at the layout level. The DPI can implement in a compact way both positive and negative currents [e.g., by using the complementary version of the schematic of Fig. 1(a)]. Another advantage of the DPI, with respect to the other two solutions, is the availability of the additional control parameter I_{th} that can be used to change the gain of the filter.

The LPF circuit has been used to model both synaptic excitation and shunting inhibition [46]. The tau-cell has been used to implement log-domain implementations [47], [48] of Mihalas-Niebur and Izhikevich neuron models, and the DPI has been used to implement both synapse and neuron models [41], [49]. In Sections III and IV, we will show examples of neurons and synapses that exploit the properties of the DPI to implement the relevant dynamics.

III. SILICON NEURONS

Several VLSI implementations of conductance-based models of neurons have been proposed in the past [50]–[54]. Given their complexity, these circuits require significant silicon real estate and a large number of bias voltages or currents to configure the circuit properties. Simplified integrate-and-fire (I&F) models typically require far less transistors and parameters but often fail at reproducing the rich repertoire of behaviors of more complex ones [55], [56].

A recently proposed class of generalized I&F models, however, has been shown to capture many of the properties of biological neurons, while requiring fewer and simpler differential equations compared to more elaborate conductance-based models, such as the Hodgkin & Huxley (H&H) one [56], [57]. Their computational simplicity and compactness make them valuable options for VLSI implementations [32], [47], [48], [58], [59].

We describe here a generalized I&F neuron circuit originally presented in [59], which makes use of the DPI circuit described in Section II and which represents an excellent compromise between circuit complexity and computational power: the circuit is compact, both in terms of transistor count and layout size; it is low power; it has biologically realistic time constants; and it implements refractory period and spike-frequency adaptation, which are key ingredients for producing resonances and oscillatory behaviors often emphasized in more complex models [55], [57].

The circuit schematic is shown in Fig. 2. It comprises an input DPI circuit used as an LPF (M_{L1-3}), a spike-event generating amplifier with current-based positive feedback (M_{A1-6}), a spike reset circuit with refractory period functionality (M_{R1-6}), and a spike-frequency adaptation mechanism implemented by an additional DPI LPF (M_{G1-6}). The DPI block M_{L1-3} models the neuron's leak conductance; it produces exponential subthreshold dynamics in response to constant input currents. The neuron's membrane capacitance is represented by the capacitor $C_{\rm mem}$, while sodium channel activation and inactivation dynamics are modeled by the positive-feedback circuits in the spike-generation amplifier M_{A1-6}. The reset M_{R1-6} block models the potassium conductance and refractory period functionality. The spike-frequency adaptation block M_{G1-6} models the neuron's calcium conductance that produces the after-hyperpolarizing current I_{ahp}, which is proportional to the neuron's mean firing rate.

By applying the current-mode analysis of Section II to both input and spike-frequency adaptation DPI circuits, we derive the complete equation that describes the neuron's subthreshold behavior

$$\left(1 + \frac{I_{th}}{I_{\rm mem}}\right)\tau\frac{d}{dt}I_{\rm mem} + I_{\rm mem}\left(1 + \frac{I_{ahp}}{I_\tau}\right) = I_{{\rm mem}_\infty} + f(I_{\rm mem})$$

$$\tau_{ahp}\frac{d}{dt}I_{ahp} + I_{ahp} = I_{ahp_\infty}u(t) \qquad (6)$$

where $I_{\rm mem}$ is the subthreshold current that represents the real neuron's membrane potential variable, I_{ahp} is the slow

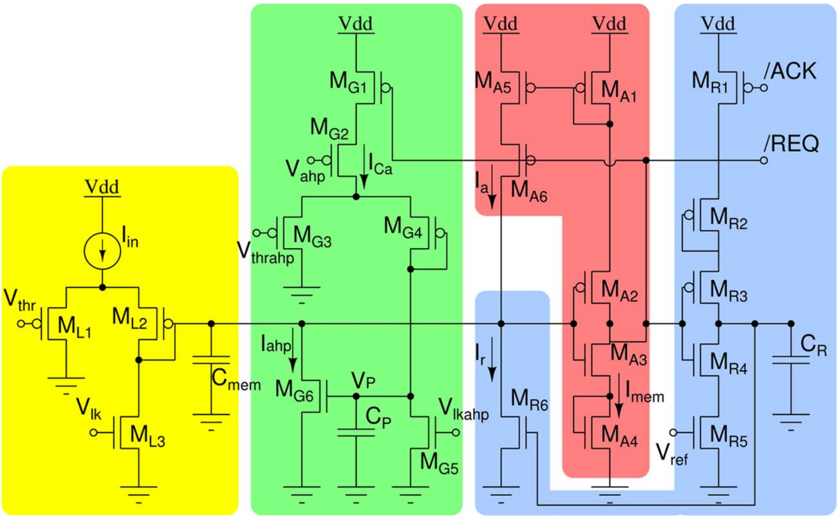


Fig. 2. ***Adaptive exponential I&F neuron circuit schematic. The input DPI circuit*** (M_{L1-3}) ***models the neuron's leak conductance. A spike event generation amplifier*** (M_{A1-6}) ***implements current-based positive feedback (modeling both sodium activation and inactivation conductances) and produces address–events at extremely low-power operation. The reset block*** (M_{R1-6}) ***resets the neuron and keeps it in a resting state for a refractory period, set by the*** V_{ref} ***bias voltage. An additional LPF*** (M_{G1-6}) ***integrates the spikes and produces a slow after-hyperpolarizing current*** I_{ahp} ***responsible for spike-frequency adaptation.***

variable responsible for the spike-frequency adaptation mechanisms, and $u(t)$ is a step function that is unity for the period in which the neuron spikes and null in other periods. Term $f(I_{\text{mem}})$ is a function that depends on both membrane potential variable I_{mem} and positive-feedback current I_a of Fig. 2

$$f(I_{\text{mem}}) = \frac{I_a}{I_\tau}(I_{\text{mem}} + I_{th}). \qquad (7)$$

In [49], Indiveri *et al.* measured I_{mem} experimentally and showed how $f(I_{\text{mem}})$ could be fitted with an exponential function of I_{mem}. The other parameters of (6) are defined as

$$\tau \triangleq \frac{C_{\text{mem}} U_T}{\kappa I_\tau} \qquad \tau_{ahp} \triangleq \frac{C_p U_T}{\kappa I_{\tau_{ahp}}}$$

$$I_\tau \triangleq I_0 e^{\frac{\kappa}{U_T} V_{\text{lk}}} \qquad I_{\tau_{ahp}} \triangleq I_0 e^{\frac{\kappa}{U_T} V_{\text{lkahp}}}$$

$$I_{\text{mem}_\infty} \triangleq \frac{I_{th}}{I_\tau}(I_{\text{in}} - I_{ahp} - I_\tau) \qquad I_{ahp_\infty} \triangleq \frac{I_{th_{ahp}}}{I_{\tau_{ahp}}} I_{\text{Ca}}$$

where I_{th} and $I_{\tau_{ahp}}$ represent currents through n-type MOSFETs not present in Fig. 2, and defined as $I_{th} \triangleq I_0 e^{(\kappa/U_T)V_{thr}}$, and $I_{th_{ahp}} \triangleq I_0 e^{(\kappa/U_T)V_{thrahp}}$, respectively.

In addition to emulating calcium-dependent after-hyperpolarization Potassium currents observed in real neurons [60], the spike-frequency adaptation block M_{G1-6} reduces power consumption and bandwidth usage in networks of these neurons. For values of $I_{\text{in}} \gg I_\tau$, we can make the same simplifying assumptions made in Section II. Under these assumptions, and ignoring the adaptation current I_{ahp}, (6) reduces to

$$\tau \frac{d}{dt} I_{\text{mem}} + I_{\text{mem}} = \frac{I_{th}}{I_\tau} I_{\text{in}} + f(I_{\text{mem}}) \qquad (8)$$

where $f(I_{\text{mem}}) \approx (I_a/I_\tau) I_{\text{mem}}$.

So under these conditions, the circuit of Fig. 2 implements a generalized I&F neuron model [61], which has been shown to be extremely versatile and capable of faithfully reproducing the action potentials measured from real cortical neurons [62], [63]. Indeed, by changing the biases that control the neuron's time constants, refractory period, and spike-frequency adaptation dynamics, this circuit can produce a wide range of spiking behaviors ranging from regular spiking to bursting (see Section VII).

While this circuit can express dynamics with time constants of hundreds of milliseconds, it is also compatible with fast asynchronous digital circuits (e.g., < 100-ns pulse widths), which are required to build large spiking neural network architectures (see the /REQ and /ACK signals of Fig. 2 and Section VI). This allows us to integrate multiple neuron circuits in event-based VLSI devices and construct large distributed reconfigurable neural networks.

IV. SILICON SYNAPSES

Synapses are fundamental elements for computation and information transfer in both real and artificial neural systems, and play a crucial role in neural coding and learning.

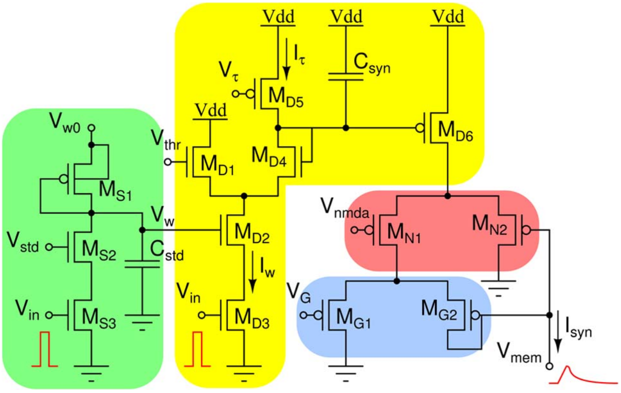


Fig. 3. ***Complete DPI synapse circuit, including short-term plasticity, NMDA voltage gating, and conductance-based functional blocks. The short-term depression block is implemented by MOSFETs*** M_{S1-3}***; the basic DPI dynamics are implemented by the block*** M_{D1-6}***; the NMDA voltage-gated channels are implemented by*** M_{N1-2}***, and conductance-based voltage dependence is achieved with*** M_{G1-2}.

While modeling the nonlinear properties and the dynamics of large ensembles of synapses can be extremely onerous for software (SW) simulations (e.g., in terms of computational power and simulation time), dedicated neuromorphic hardware (HW) can faithfully reproduce synaptic dynamics in real time using massively parallel arrays of pulse (spike) integrators. In this case, the bottleneck is not in the complexity of the spiking processes being modeled, but in the number of spikes being received and transmitted (see Section VI for more details).

An example of a full excitatory synapse circuit is shown in Fig. 3. This circuit, based on the DPI circuit described in Section II, produces biologically realistic excitatory postsynaptic currents (EPSCs), and can express short term plasticity, N-Methyl-D-Aspartate (NMDA) voltage gating, and conductance-based behaviors. The input spike (the voltage pulse $V_{\rm in}$) is applied to both M_{D3} and M_{S3}. The output current $I_{\rm syn}$, sourced from M_{D6} and through M_{G2}, rises and decays exponentially with time. The temporal dynamics are implemented by the DPI block M_{D1-6}. The circuit time constant is set by V_τ while the synaptic efficacy, which determines the EPSC amplitude, depends on both V_{w0} and V_{thr} [41].

A. Short-Term Depression and Short-Term Facilitation

Short-term plasticity mechanisms can be extremely effective tools for processing temporal signals and decoding temporal information [64], [65]. Several circuit solutions have been proposed to implement these types of dynamics, using different types of devices and following a wide range of design techniques [66]–[71]. These short-term dynamic mechanisms are subdivided into short-term depression and short-term facilitation. The circuit block M_{S1-3} is responsible for implementing short-term depression: with every voltage pulse $V_{\rm in}$ the synaptic weight voltage V_w decreases, at a rate set by V_{std}. When no spikes are being received, V_w "recovers" toward the resting state set by V_{w0}. In [67], Boegerhausen *et al.* demonstrate that this subcircuit is functionally equivalent to the one described in theoretical models, and often used in computational neuroscience simulations [72], [73]. In addition to short-term depression, this DPI synapse is capable also of short-term facilitation: if the bias V_{thr} of M_{D1} is set so that $I_{th} \gg I_{\rm syn}$ at the onset of the stimulation (i.e., during the first spikes), the circuit equation, derived from (4) in the analysis of Section II, reduces to

$$\tau \frac{d}{dt} I_{\rm syn} + \frac{I_{\rm syn}^2}{I_{th}} - I_{\rm syn}\left(\frac{I_w}{I_\tau} + 1\right) = 0 \qquad (9)$$

which can be further simplified to

$$\tau \frac{d}{dt} I_{\rm syn} = I_{\rm syn}\left(\frac{I_w}{I_\tau} + 1\right). \qquad (10)$$

In other words, the change in circuit response increases with every spike, by an amount greater than one, for as long as condition $I_{\rm syn} \ll I_{th}$ is satisfied. As $I_{\rm syn}$ increases, this condition starts to fail, and eventually the opposite condition $(I_{\rm syn} \gg I_{th})$ is reached. This is the condition for linearity, under which the circuit starts to behave as a first-order LPF, as described in Section II.

B. NMDA Voltage Gating and Conductance Behavior

The output differential pairs of Fig. 3 (M_{N1-2} and M_{G1-2}) are responsible for implementing NMDA voltage-gated channels and conductance-based behavior, respectively. The response properties of these circuits have been thoroughly characterized in [41].

C. Homeostatic Plasticity: Synaptic Scaling

Synaptic scaling is a stabilizing homeostatic mechanism used by biological neural systems to keep the network's activity within proper operating bounds. It operates by globally scaling the synaptic weights of all the synapses afferent to a neuron, for maintaining the neuron's firing rate within a functional range, in face of chronic changes of their activity level, while preserving the relative differences between individual synapses [74]. In VLSI, synaptic scaling is an appealing mechanism that can be used to compensate for undesired behaviors that can arise, for example, because of temperature drifts or sudden changes in the system input activity levels. Thanks to its independent controls on synaptic efficacy set by V_w and V_{thr}, the DPI synapse of Fig. 3 is compatible with both conventional spike-based learning rules, and homeostatic synaptic scaling mechanisms. Specifically, while learning circuits can be designed to locally change the synaptic weight by acting on the V_w of each individual synapse (e.g., see Section V), it is possible to implement adaptive circuits that act on V_{thr} of all the synapses connected to a given neuron to keep its firing rate within desired control boundaries. This strategy has been recently demonstrated in [75].

V. SYNAPTIC PLASTICITY: SPIKE-BASED LEARNING CIRCUITS

One of the key properties of biological synapses is their ability to exhibit different forms of plasticity. Plasticity mechanisms produce long-term changes in the synaptic strength of individual synapses in order to form memories and learn about the statistics of the input stimuli. Plasticity mechanisms that induce changes that increase the synaptic weights are denoted as long-term potentiation (LTP) mechanisms, and those that induce changes that decrease synaptic weights are denoted as long-term depression (LTD) mechanisms [76].

In neuromorphic VLSI chips, implementations of long-term plasticity mechanisms allow us to implement learning algorithms and set synaptic weights automatically, without requiring dedicated external read and write access to each individual synapse.

As opposed to the case of theory, or software simulation, the realization of synapses in hardware imposes a set of important physical constraints. For example, synaptic weights can only have bounded values, and with a limited (and typically small) precision. These constraints have dramatic effects on the memory capacity of the neural network that uses such synapses [77], [78]. So when developing computational models of biological synapses that will be mapped onto neuromorphic hardware, it is important to develop plasticity mechanisms that work with limited resolution and bounded synaptic weights [24]. Another important constraint that should be taken into account when developing hardware learning systems that are expected to operate continuously (as is the case for real-time behaving systems) is related to the blackout effect [79]. Classical Hopfield networks are affected by this effect: in Hopfield networks the memory capacity is limited, and is related to the number of synapses available. Learning new patterns uses memory resources and if the number of stored patterns reaches a critical value the storage of even one single new pattern destroys the whole memory because none of the old patterns can be recalled. Unfortunately, this catastrophic condition is unavoidable in most practical scenarios, since continuous, uninterrupted learning will always lead to the blackout effect. However, it is possible to avoid this effect, by building networks that can progressively forget old memories to make room for new ones, thus exhibiting the palimpsest property [80]. It has been demonstrated that the optimal strategy for implementing this palimpsest property, while maintaining a high storage capacity, is to use synapses that have a discrete number of stable states and that exhibit stochastic transitions between states [81]. Specifically, it was demonstrated that by modifying only a random subset of the network synapses with a small probability, memory lifetimes increase by a factor inversely proportional to the probability of synaptic modification [82]. In addition, the probability of synaptic transitions can be used as a free parameter to set the tradeoff between the speed of learning against the memory capacity.

These types of plastic synapse circuits can be implemented in a very compact way by reducing to the minimum the resolution of the synaptic weight (i.e., just two stable states) and using variability in the input spike trains as the source of stochasticity for the transition of the synaptic weights (e.g., from an LTD to an LTP stable state). The low resolution in the synaptic weights can be compensated by redundancy (i.e., using large numbers of synapses), and the variability in the input spike trains can be obtained by encoding signals with the mean rates of Poisson distributed spike trains [83]–[85]. An important advantage of delegating the onus of generating the stochasticity to the input spiking activity is that no additional circuitry is needed for the stochastic state transitions [86]. Furthermore, since the spiking activity controls the speed of learning, the network can easily switch between a slow-learning regime (i.e., to learn pattern of mean firing rates with uncorrelated stimuli) to a fast learning one (i.e., to learn highly correlated patterns) without changing its internal parameters [84], [87].

In addition to allowing compact circuit designs, these types of plastic synapse circuits do not require precisely matched analog devices. As the dominant source of variability lies in the (typically Poisson distributed) input spikes driving the learning, additional sources of variability, for example, induced by device mismatch, do not affect the main outcome of the learning process. As a consequence, analog VLSI designers do not have to allocate precious silicon real-estate resources to minimize device mismatch effects in these circuits.

An example of a circuit that implements a weight update mechanism compatible with this stochastic learning rule is shown in Fig. 4(a). The circuit comprises three main blocks: an input stage M_{I1-2}, a spike-triggered weight update block M_{L1-4}, and a bistability weight storage/refresh block [see transconductance amplifier in Fig. 4(a)]. The input stage receives spikes from presynaptic neurons and triggers increases or decreases in weights, depending on the two signals V_{UP} and V_{DN} generated downstream by the postsynaptic neuron. The bistability weight refresh circuit is a positive-feedback amplifier with very small "slew rate" that compares the weight voltage V_w to a set threshold V_{thw} and slowly drives it toward one of the two rails V_{whi} or V_{wlo}, depending on whether $V_w > V_{thw}$ or $V_w < V_{thw}$, respectively. This bistable drive is continuous and its effect is superimposed to the one from the spike-triggered weight update circuit. The analog, bistable,

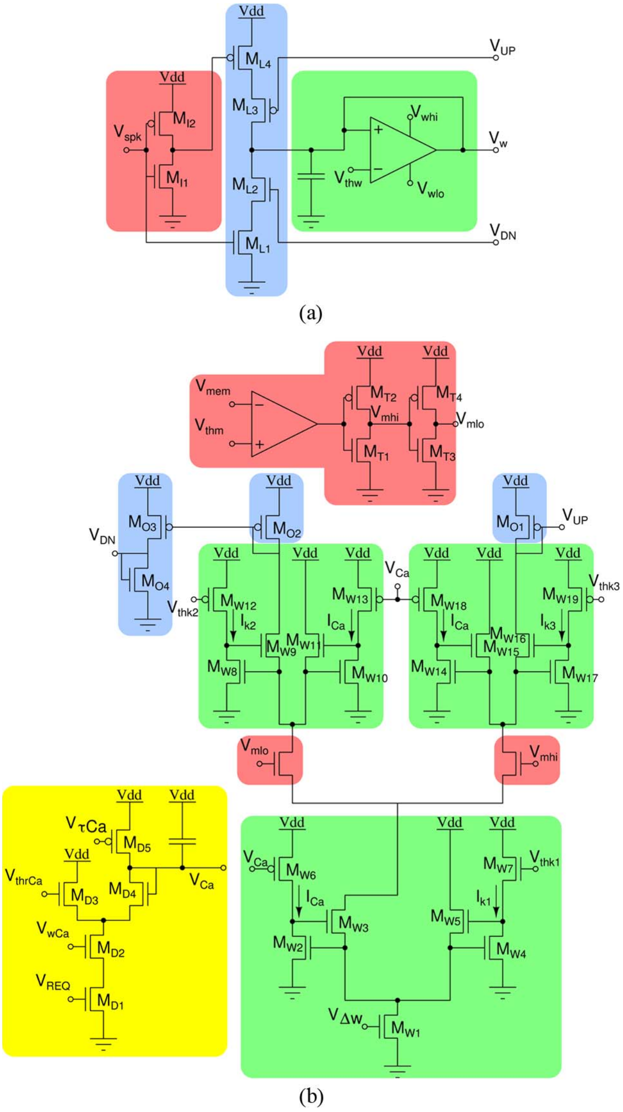


Fig. 4. ***Spike-based learning circuits. (a) Presynaptic weight-update module (present at each synapse). (b) Postsynaptic learning control circuits (present at the soma).***

synaptic weight voltage V_w is then used to set the amplitude of the EPSC generated by the synapse integrator circuit (e.g., the circuit shown in Fig. 3). Note that while the weight voltage V_w is linearly driven by the bistability circuit, its effect on the EPSC produced by the connected DPI synapse is exponential. This nonlinearity can, in principle, affect adversely the dynamics of learning and is more relevant at small scales (tens of synapses) since the contribution of each synapse is important. However, the nonlinearity has a negligible effect in practice because in the slow-learning regime only a small subset of a much larger number of synapses is involved in the learning process, each one participating with a small contribution. The circuit presented here can be easily modified to better reproduce the linear dynamics of the theoretical model by decoupling the synaptic weight from the internal variable, as in [88].

The two signals V_{UP} and V_{DN} of Fig. 4(a) that determine whether to increase or decrease the synaptic weight are shared globally among all synapses afferent to a neuron. The circuits that control these signals can be triggered by the neuron's postsynaptic spike, to implement standard spike-timing-dependent plasticity (STDP) learning rules [76]. In general, STDP mechanisms that update the synaptic weight values based on the relative timing of presynaptic and postsynaptic spikes can be implemented very effectively in analog [83], [89]–[92] or mixed analog–digital VLSI technology [93]. However, while standard STDP mechanisms can be effective in learning to classify spatio–temporal spike patterns [93], [94], these algorithms and circuits are not suitable for both encoding information represented in a spike correlation code and a means rate code without spike correlations [95], [96]. For this reason, we focus on more elaborate plasticity mechanisms that not only depend on the timing of the presynaptic spikes but also on other state variables present at the postsynaptic terminal, such as the neuron membrane potential or its calcium concentration. An example of such type of learning rule is the one proposed in [25], which has been shown to be able to classify patterns of mean firing rates, to capture the rich phenomenology observed in neurophysiological experiments on synaptic plasticity, and to reproduce the classical STDP phenomenology both in HW [9], [85], [88] and in SW simulations [25], [97]. This rule can be used to implement unsupervised and supervised learning protocols, and to train neurons to act as perceptrons or binary classifiers [24]. Typically, input patterns are encoded as sets of spike trains that stimulate the neuron's input synapses with different mean frequencies, while the neuron's output firing rate represents the binary classifier output.

Examples of circuits that implement such a learning rule are shown in Fig. 4(b). The spikes produced by the postsynaptic neuron are integrated by the DPI circuit M_{D1-5} to produce a voltage V_{Ca} which represents a postsynaptic calcium concentration and is a measure of the recent spiking activity of the neuron. The three current-mode winner-take-all circuits [98] M_{W1-19} compare V_{Ca} to three different thresholds V_{thk1}, V_{thk2}, and V_{thk3}. In parallel, the neuron's membrane potential V_{mem} is compared to a fixed threshold V_{thm} by a voltage comparator. The outcomes of these comparisons set V_{UP} and V_{DN} such that, whenever a presynaptic spike V_{spk} reaches the synapse weigh-update block of Fig. 4(a)

$$\begin{cases} V_w = V_w + \Delta w, & \text{if } V_{\mathrm{mem}} > V_{mth} \\ & \quad \text{and } V_{thk1} < V_{\mathrm{Ca}} < V_{thk3} \\ V_w = V_w - \Delta w, & \text{if } V_{\mathrm{mem}} < V_{mth} \\ & \quad \text{and } V_{thk1} < V_{\mathrm{Ca}} < V_{thk2} \end{cases}$$

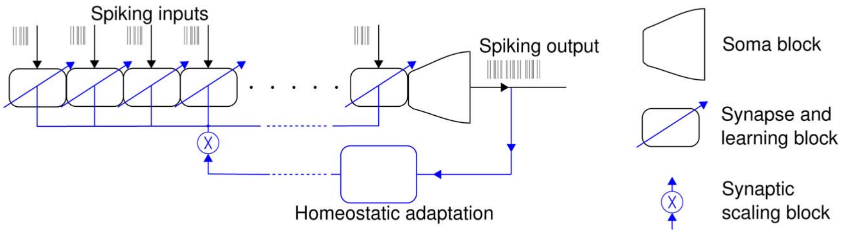


Fig. 5. ***Silicon neuron diagram. This is a schematic representation of a typical circuital block comprising multiple synapse blocks, an I&F soma block, and a homeostatic plasticity control block. The synapses receive input spikes, integrate them, and convey the resulting currents to the soma. The soma integrates these currents and produces output spikes with a mean rate that is proportional to the total net input current. Synapse circuits can implement both local plasticity mechanisms to change their efficacy, and global scaling mechanisms via additional homeostatic control block.***

where Δw is a factor that depends on $V_{\Delta w}$ of Fig. 4(b), and is gated by the eligibility traces $V_{\rm UP}$ or $V_{\rm DN}$. If none of the conditions above are met, Δw is set to zero by setting $V_{\rm UP} = V_{dd}$, and $V_{\rm DN} = 0$.

The conditions on $V_{\rm Ca}$ implement a "stop-learning" mechanism that greatly improves the generalization performance of the system by preventing overfitting when the input pattern has already been learned [24], [25]. For example, when the pattern stored in the synaptic weights and the pattern provided in input are highly correlated, the postsynaptic neuron will fire with a high rate and $V_{\rm Ca}$ will rise such that $V_{\rm Ca} > V_{thk3}$, and no more synapses will be modified.

Mitra *et al.* [85] and Giulioni *et al.* [88] show how such types of circuits can be used to carry out classification tasks with a supervised learning protocol, and characterize the performance of these types of VLSI learning systems. Additional experimental results from the circuits shown in Fig. 4 are presented in Section VII.

VI. FROM CIRCUITS TO NETWORKS

The silicon neuron, synapse, and plasticity circuits presented in Sections III–V can be combined together to form full networks of spiking neurons. Typical spiking neural network chips have the elements described in Fig. 5. Multiple instances of these elements can be integrated onto single chips and connected among each other either with on-chip hard-wired connections [e.g., see Fig. 6(a)], or via off-chip reconfigurable connectivity infrastructures [99]–[103].

A. Recurrent Neural Networks

In the most general recurrent neural network (RNN), each neuron is connected to every other neuron (fully recurrent network). Unlike feedforward networks, the response of RNNs to the input does not only depend on the external input but also on their internal dynamics, which in turn is determined by the connectivity profile. Thus, specific changes in connectivity, for example through

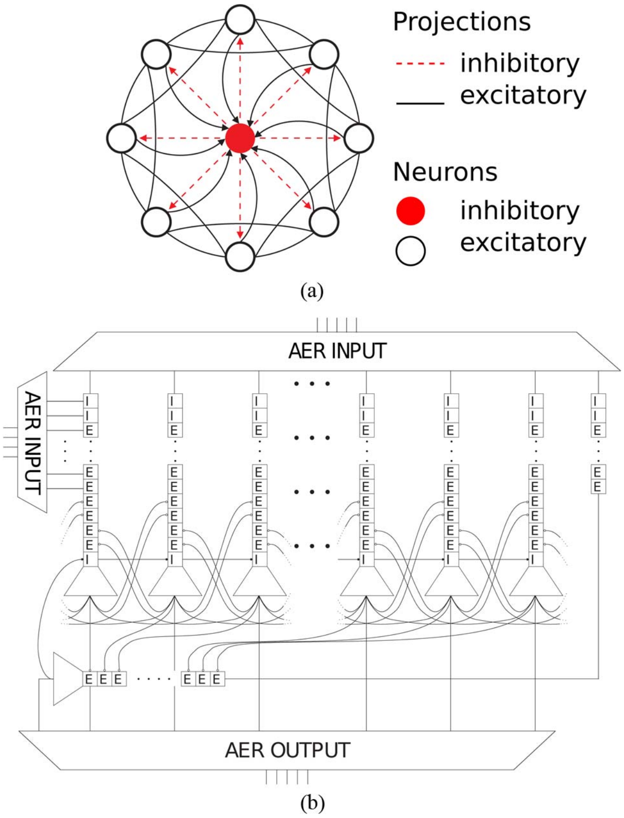


Fig. 6. ***sWTA network topology. (a) Schematic representation of the connectivity pattern of the sWTA network. These connections are implemented by synapses with hard-wired connections to presynaptic and postsynaptic neurons. Empty circles represent excitatory neurons and the filled circle represents the global inhibitory neuron. Solid/dashed lines represent excitatory/inhibitory connections. Connections with arrowheads are mono-directional; all the others are bidirectional. Only eight excitatory neurons are shown for simplicity. (b) Chip architecture. Squares represent excitatory (E) and inhibitory (I) synapses, and small unlabeled trapezoids represent I&F neurons. The I&F neurons transmit their spikes off-chip and/or to locally connected synapses implementing the network topology depicted in (a). Adapted from [117].***

learning, can tune the RNN behavior, which corresponds to the storage of internal representations of different external stimuli. This property makes RNNs suitable for implementing, among other properties, associative memories [81], working memory [104], and context-dependent decision making [30].

There is reason to believe that, despite significant variation across cortical areas, the pattern of connectivity between cortical neurons is similar throughout neocortex. This fact would imply that the remarkably wide range of capabilities of the cortex are the results of a specialization of different areas with similar structures to the various tasks [105], [106]. An intriguing hypothesis about how computation is carried out by the brain is the existence of a finite set of computational primitives used throughout the cerebral cortex. If we could identify these computational primitives and understand how they are implemented in hardware, then we would make a significant step toward understanding how to build brain-like processors. There is an accumulating body of evidence that suggests that one potential computational primitive consists of an RNN with a well-defined excitatory/inhibitory connectivity pattern [106] typically referred as soft winner-take-all (sWTA) network.

In sWTA neural networks, a group of neurons compete with each other in response to an input stimulus. The neurons with highest response suppress all other neurons to win the competition. Competition is achieved through a recurrent pattern of connectivity involving both excitatory and inhibitory connections. Cooperation between neurons with similar response properties (e.g., close receptive fields or stimulus preference) is mediated by excitatory connections. Competition and cooperation make the output of an individual neuron depend on the activity of all neurons in the network and not just on its own input [107]. As a result, sWTAs perform not only common linear operations but also complex nonlinear operations [108]. The linear operations include analog gain (linear amplification of the feedforward input, mediated by the recurrent excitation and/or common mode input), and locus invariance [109]. The nonlinear operations include nonlinear selection [110]–[112], signal restoration [13], [111], and multistability [110], [112].

The computational abilities of these types of networks are of great importance in tasks involving feature extraction, signal restoration, and pattern classification problems [113]. For example, localized competitive interactions have been used to detect elementary image features (e.g., orientation) [114], [115]. In these networks, each neuron represents one feature (e.g., vertical or horizontal orientation); when a stimulus is presented, the neurons cooperate and compete to enhance the response to the features they are tuned to and to suppress background noise. When sWTA networks are used for solving classification tasks, common features of the input space can be learned in an unsupervised manner. Indeed, it has been shown that competition supports unsupervised learning because it enhances the firing rate of the neurons receiving the strongest input, which in turn triggers learning on those neurons [116].

B. Distributed Multichip Networks

The modularity of the cortex described in the theoretical works and suggested by the experimental observations above mentioned, constitutes a property of great importance related to the scalability of the system. If we understood the principles by which such computational modules are arranged together and what type of connectivity allows for coherent communication also at large distances, we would be able to build scalable systems, i.e., systems whose properties are qualitatively reproduced at all scales.

The idea of modularity poses some technological questions as to how the communication between the systems should be implemented. Large VLSI networks of I&F neurons can already be implemented on single chips, using today's technology. However, implementations of pulse-based neural networks on multichip systems offer greater computational power and higher flexibility than single-chip systems and constitute a tool for the exploration of the properties of scalability of the neuromorphic systems. Because interchip connectivity is limited by the small number of input–output connections available with standard chip packaging technologies, it is necessary to adopt time-multiplexing schemes for constructing large multichip networks. This scheme should also allow for an asynchronous type of communication, where information is transmitted only when available and computation is performed only when needed in a distributed, nonclocked manner.

In recent years, we have witnessed the emergence of a new asynchronous communication standard that allows analog VLSI neurons to transmit their activity across chips using pulse-frequency-modulated signals (in the form of events, or spikes). This standard is based on the address–event representation (AER) communication protocol [12]. In AER input and output signals are real-time asynchronous digital pulses (events or spikes) that carry analog information in their temporal relationships (interspike intervals). If the activity of the VLSI neurons is sparse and their firing rates are biologically plausible (e.g., ranging from a few spikes per second to a few hundred spikes per second), then it is possible to trade off space with speed very effectively, by time-multiplexing a single (very fast) digital bus to represent many (very slow) neuron axons. For example, it has been recently demonstrated how these time-multiplexing schemes can sustain more than 60 mega events/s, representing the synchronous activity of one million neurons firing at a rate of 60 Hz [99], [118]. In general, AER communication infrastructures provide the possibility to implement arbitrary custom multichip architectures, with flexible connectivity schemes. Address–events can encode the address of the sending node (the spiking neuron) or of the receiving one (the destination synapse). The connectivity between different nodes is implemented by using external digital components and is typically defined as a lookup table with source and destination pairs of addresses, or by more resource-efficient schemes, e.g., using multicast or multistage routing [6], [119], [120]. This asynchronous digital solution permits flexibility in the configuration (and reconfiguration) of the network topology, while keeping the computation analog and low power at the neuron and synapse level.

To handle cases in which multiple sending nodes attempt to transmit their addresses at exactly the same time (event collisions), on-chip digital asynchronous arbitration schemes have been developed [12], [118], [121]. These circuits work by queuing colliding events, so that only one event is transmitted at a time. Multiple colliding events are therefore delayed by a few nanoseconds or fractions of microseconds. For neuromorphic architectures that use biologically plausible time constants (i.e., of the order of milliseconds), these delays are negligible and do not affect the overall performance of the network. For example, assuming a tolerance of 1-ms jitter [122], it is possible to process up to four thousand coincident input events without introducing sensible delays, even with an outdated 350-nm CMOS technology [102]. On the other hand, in accelerated-time systems, such as those proposed in [7] whose circuits operate at 10^4 the speed of their biological counterpart, communication delays are much more critical, because their duration does not scale. In general, the performance of any AER neuromorphic system will be bound by communication memory and bandwidth constraints, which trade off the speed of the neural processing elements with the size of the network that can be implemented.

C. SW/HW Ecosystem

In order to promptly explore the computational properties of different types of large-scale multichip computational architectures, it is important to develop a dedicated HW and SW infrastructure, which allows a convenient, user-friendly way to define, configure, and control in real time the properties of the HW [123], [124] spiking neural networks, as well as a way to monitor in real time their spiking and nonspiking activity.

The definition of an SW infrastructure for neuromorphic systems pertains to an issue of increasing importance. Indeed, as reconfigurable neuromorphic platforms are scaled to larger sizes, it is necessary to develop efficient tools to interpret the neural network model, e.g., through programming or scripting languages, and configure the hardware parameters correspondingly for the neural and synaptic dynamics and for the events routing. Hence, the SW should provide means to configure, control, interact, and monitor the electronic hardware. Fortunately, while the specific electronic implementation of each neuromorphic system can differ substantially, several common properties can be identified, such as the use of an AER scheme for communication. Therefore, an SW ecosystem can be defined to assemble and control the system in a modular, fully reconfigurable way. In this respect, several SW interfaces for neuromorphic and neurocomputing platforms have already been developed. The scopes of these tools are diverse and so are their peculiarities due to the specificities of the corresponding system. Both digital neurocomputing platforms and analog neuromorphic systems typically require a "neuromorphic compiler" able to parse the network topology and configure correspondingly memories, processors, or digital interfaces to properly simulate the neural and synaptic dynamics and route the spiking events through the network [125]–[128]. On top of the compilers, a number of SW frameworks have been developed as scripting and programming languages for neural networks at the level of the single network elements, e.g., neurons, synapses, and connectivity [123] and also including a system-level description for building large-scale, brain simulators [129].

A promising example of an open-source SW framework that interprets generalized hardware specification files and constructs an abstract representation of the neuromorphic devices compatible with high-level neural network programming libraries is available at http://inincs.github.com/pyNCS/. This framework is based on reconfigurable and extensible application programming interfaces (APIs) and includes a high-level scripting front–end for defining neural networks. It constitutes a bridge between applications using abstract resources (i.e., neurons and synapses) and the actual processing done at the hardware level through the management of the system's resources, much like a kernel in modern computers [130], and it is compatible with most existing software. The HW and SW infrastructure can be complemented with tools for dynamic parameter estimation methods [131], [132] as well as automated methods for measuring and setting circuit-level parameters using arbitrary cost functions at the network level [124].

VII. EXPERIMENTAL RESULTS

The circuits and architectures described in this paper have been designed and developed over the course of several years. Therefore, the experimental data presented in this section have been collected from multiple neuromorphic VLSI devices and systems. The results presented demonstrate the correct behavior of the circuits described in Sections III–VI.

A. Synaptic and Neural Dynamics

To show the combined effect of synaptic and neural dynamics, we stimulated a silicon neuron via an excitatory DPI synapse circuit, while sweeping different short-term depression (STD) parameter settings. The typical phenomenology of STD manifests as a reduction of EPSC amplitude with each presentation of a presynaptic spike, with a slow (e.g., of the order of 100 ms) recovery time [133]. In

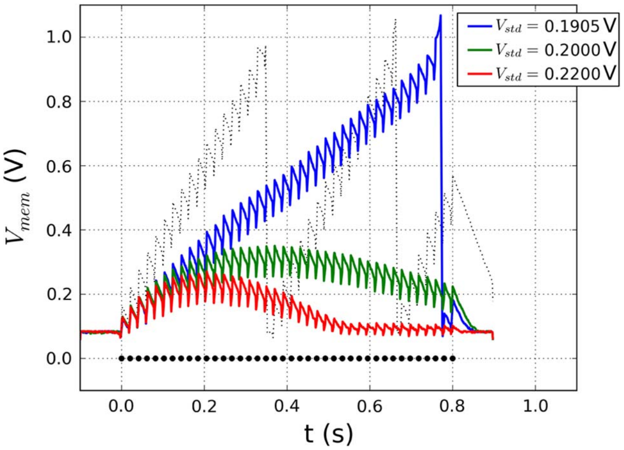


Fig. 7. ***Membrane potential of I&F neuron in response to a 50-Hz presynaptic input spike train for different values of short-term depression adaptation rate, which is controlled by V_{std} bias (see Fig. 3). The dashed trace in background corresponds to the response without STD. Black dots correspond to input spike times.***

Fig. 7, we plot the neuron's membrane potential V_{mem} during the stimulation of one of its excitatory synapses with a regular presynaptic input spike train of 50 Hz, for different STD adaptation settings. Small parameter settings for the STD bias voltage have no or little effect. But for larger settings of this bias voltage the effect of STD is prominent: the synaptic efficacy decreases with multiple input spikes to a point in which the net input current to the soma becomes lower than the neuron's leak current, thus making the neuron membrane potential decrease, rather than increase over time.

Another important adaptation mechanism discussed in Section III, is that of spike-frequency adaptation. To show the effect of this mechanism, we set the relevant bias voltages appropriately, stimulated the silicon neuron with a constant input current, and measured its membrane potential. Fig. 8 shows an example response to the step input current, in which $V_{lkahp} = 0.05$ V, $V_{thrahp} = 0.14$ V, and $V_{ahp} = 2.85$ V. As shown, we were able to tune the adaptation circuits in a way to produce bursting behavior. This was achieved by simply increasing the gain of the negative feedback adaptation mechanism ($V_{thrahp} > 0$). This is equivalent to going from an asymptotically stable regime to a marginally stable one that produces ringing in the adaptation current I_{ahp}, which in turn produces bursts in the neuron's output firing rate. This was possible due to the flexibility of the DPI circuits, which allow us to take advantage of the extra control parameter V_{thrahp}, in addition to the adaptation rate parameter V_{ahp}, and the possibility of exploiting its nonlinear transfer properties, as described in Section IV, without requiring extra circuits or dedicated resources that alternative neuron models have to use [32], [57], [58].

B. Spike-Based Learning

In this section, we present measurements from the circuits implementing the STDP learning mechanism described in Section V. To stimulate the synapses, we generated presynaptic input spike trains with Poisson distributions. Similarly, the postsynaptic neuron was driven by a current produced via a nonplastic synapse (a DPI circuit with a constant synaptic weight bias voltage) stimulated by software-generated Poisson spike trains. These latter inputs are used to drive the I&F neuron toward different activity regimes which regulate the probabilities of synaptic transitions [25], [134], effectively modulating the learning rate in unsupervised learning conditions, or acting as teacher signals in supervised learning conditions.

The Poisson nature of the spike trains used in this way represents the main source of variability required for implementing stochastic learning [83], [84]. In Fig. 9, we show measurements from a stochastic learning experiment in which the neuron is driven to a regime where both potentiation and depression are possible but depression has a higher probability to occur. As shown, the weight voltage undergoes both positive and negative changes, depending on the timing of the input spike and the state of the postsynaptic neuron (as explained in Section V). In addition, the weight voltage is slowly driven toward one of the two stable states, depending on whether it is above or below the threshold θ [where θ corresponds to the voltage V_{thw} of Fig. 4(a)]. Long-term transitions occur when a series of presynaptic spikes arrive in a short time frame causing the weight to cross the threshold θ. As a consequence, the probability of synaptic state transitions depends on the probability that such events occur, hence it depends on the firing rate of the presynaptic neuron [82], [89]. In the case of the experiment of Fig. 9, an LTD

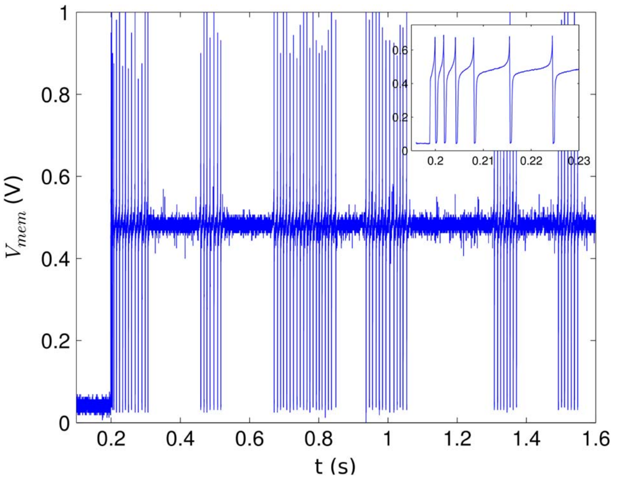


Fig. 8. ***Silicon neuron response to a step input current, with spike-frequency adaptation mechanism enabled and parameters tuned to produce bursting behavior. The figure inset represents a zoom of the data showing the first six spikes. Adapted from [49].***

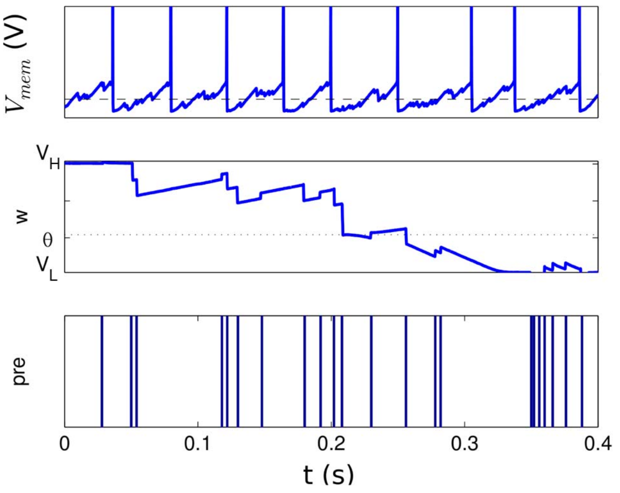
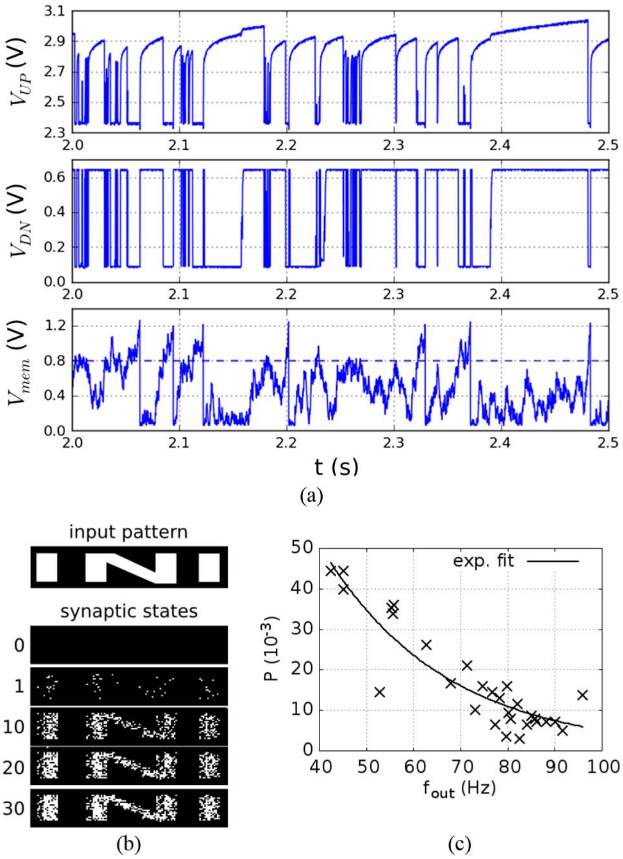


Fig. 9. ***Stochastic transitions in synaptic states. The nonplastic synapse is stimulated with a Poisson-distributed spikes train. The neuron fires at an average rate of 30 Hz. The presynaptic input (V_{pre}) is stimulated with Poisson-distributed spike trains with a mean firing rate of 60 Hz. The updates in the synaptic weight produced an LTD transition that remains consolidated. V_H and V_L show the potentiated and depressed levels, respectively, while w denotes the synaptic weight, and θ is the bistability threshold. Adapted from [85].***

transition has occurred upon the presentation of an input stimulus of 60 Hz for 400 ms. In conclusion, the bistability of the synapses and the spike-based plasticity concur in a mechanism that: 1) ensures that only a random fraction of the stimulated bistable synapses undergo long-term modifications; and 2) that synaptic states are resilient to changes due to spontaneous activity, thus increasing the robustness to noise.

In Fig. 10(a), we show the results of another stochastic learning experiment in which we stimulated the postsynaptic neuron with a high-frequency Poisson-like spike train through a nonplastic excitatory input synapse, in order to produce Poisson-like firing statistics in the output. The dashed line on the V_{mem} plot represents the learning threshold voltage V_{thm} of Fig. 4(b). The V_{UP} (active low) and V_{DN} (active high) signals are the same as shown in Fig. 4(b) and represent the currents that change the synaptic values when triggered by presynaptic spikes. They can be considered as eligibility traces that enable the weight update mechanism when they are active.

In Fig. 10(b), we show the results of an experiment where we trained a matrix of $28 \times 124 = 3472$ plastic synapses, constituting the total input of a neuron, with multiple presentations of the same input pattern representing the "INI" acronym. Initially, all the neuron's input synaptic weights are set to their low state (black pixels). Then, the postsynaptic neuron is driven by a teacher signal that makes it fire stochastically with a mean rate of 40 Hz. At the same time, input synapses are stimulated according to the image pattern: in the input image (top left image), each white pixel represents a Poisson spike train of 55 Hz, sent to the corresponding synapse; similarly, each black pixel represents a low rate spike train (5 Hz) which is transmitted to its corresponding synapse. Because the probability of LTP depends on the presynaptic firing rate, elements of the input matrix that correspond to a white pixel are more likely to make a transition to the potentiated state compared to the other ones. Because of the stochastic nature of the input patterns, only a random subset of synapses undergoes LTP, leaving room available to store other memories. By repeating the presentation of



(a)

(b)

(c)

Fig. 10. ***Stochastic learning. (a) Single-neuron stochasticity. Traces from a VLSI multineuron chip with I&F neurons and plasticity circuits as in Fig. 4(a). The V_{UP} and V_{DN} signals (top traces) are set by the circuits in Fig. 4(b). A Poisson spike train of high firing rate is sent to the excitatory synapse of an I&F neuron whose V_{mem} trace is reported in the lower trace. The strong input current generated by the synapse has been compensated by a strong leakage current ($V_{\mathrm{leak}} = 0.39$ V). This parameter choice allows to exploit the stochasticity of the input spike trains to produce the highly irregular dynamics of V_{mem}. The nonideal rounding in the rising part of the V_{UP} trace has negligible effects on the synaptic weight given the exponential nature of the current generated through transistor M_{L3} of Fig. 4(a). (b) An image of the "INI" acronym is converted into a series of Poisson spike trains and gradually stored in the memory by repeated presentations. See Section VII-B for details. (c) Normalized frequency of occurrence of LTP transitions during the experiment of (b), fitted by an exponential function (solid line).***

the input pattern multiple times, this pattern gets gradually stored in the synaptic matrix. The bottom left image of Fig. 10(b) represents the synaptic matrix at the end of the experiment. Furthermore, the stop-learning mechanism described in Section V causes a drop in the number of synapses that undergo LTP because as the pattern is stored in the memory, the postsynaptic firing rate increases [Fig. 10(c)].

The above experiments demonstrate the properties of the learning circuits implemented in the VLSI chips. In a feedforward configuration, the neuron can be controlled by an external spiking teacher signal, which indirectly controls the transition probabilities. This "perceptron-like" configuration allows the realization of supervised learning protocols for building real-time classification engines. But, as opposed to conventional perceptron-like learning rules, the spike-triggered weight updates implemented by these circuits overcome the need for an explicit control (e.g., using error backpropagation) on every individual synapse. In "Hopfield-network"-like RNN configurations, the same neuron and plasticity circuits can implement attractor neural network (ANN) learning schemes [9], [135], exploiting the neural network dynamics to form memories through stochastic synaptic updates, without the need for explicit random generators at each synapse.

C. sWTA Networks of I&F Neurons

Two characteristic features of sWTA networks that make them ideal building blocks for cognitive systems are their ability to selectively enhance the contrast between localized inputs and to exhibit activity that persists even after the input stimulus has disappeared. We configured the local hard-wired connectivity of a multineuron chip to implement an sWTA network and carried out test experiments to show both selective amplification and state-dependent computation. Specifically, we configured a chip comprising a network of 128 I&F neurons with local nearest neighbor excitatory connectivity and global inhibition: each neuron was configured to excite its first nearest neighbors, its second neighbors, and a population of four global inhibitory neurons (the top four neurons in the array of 128 neurons). In the first experiment, we calibrated the settings and input stimuli to minimize the effect of device mismatch, following the event-based techniques described in [124] and [131] and stimulated the network with two distinct regions of activation, centered around units 20 and 60 (see shaded areas in Fig. 11). In one case, the top region had a higher mean firing rate than the bottom one, and in the other case, the bottom region had a higher activation (see top and bottom plots in Fig. 11, respectively). As expected from theory [109], [111], [108], the population of silicon neurons receiving the strongest input won the competition, enhancing its activity by means of the local recurrent connections, while suppressing the activity of the competing population via the global inhibitory connections (selective amplification feature).

In the second experiment, we demonstrate the behavior of a sWTA architecture used to construct state-holding elements, which are the basic blocks for building finite state machines (FSMs) using spiking neurons, and in which the FSM states are represented by subpopulations of neurons. The network topology supporting the FSM functionality and used in the following experiments resembles the ones of ANN with discrete or line attractors. As mentioned in Section VI, this type of networks can support a diverse range of functionalities and these networks have been employed in hardware implementations, e.g., for head-direction tracking [137] and memory recall [9]. In particular, we concentrated our experiments

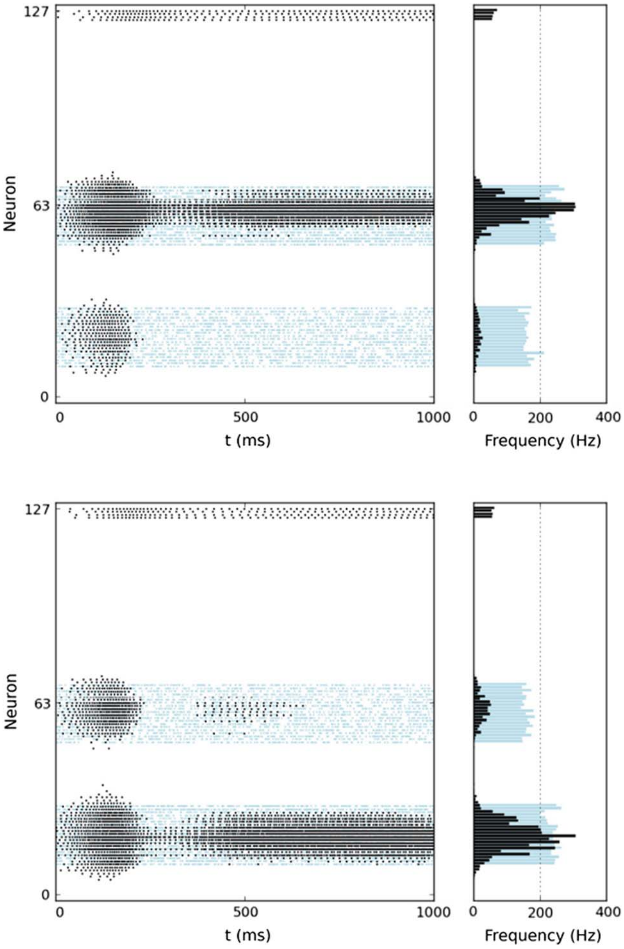


Fig. 11. ***Selective amplification experiments. The network is stimulated in two regions, one centered around unit 20 and the other around unit 60, with Poisson spike trains of mean firing rates 180 and 240 Hz. The figures show the networks response to these inputs (black) and their respective steady-state firing rates on the right panels (calculated for time > 500 ms). Neurons 124–127 are the four inhibitory neurons of the soft WTA network. In the right and left panels, the input amplitudes are swapped. The results show smooth activity profiles that are invariant to input swapping, demonstrating that the mismatch in the local weights has been partially compensated. Adapted from [136].***

on demonstration of two of their main properties useful for implementing the FSM, namely selective amplification and state switching due to external inputs.

In this experiment, we present localized and transient inputs to two groups of neurons using synthetically generated Poisson trains (see Fig. 12). After the presentation of each input stimulus, the activity of the stimulated population persists, reverberating in time, by means of the local recurrent excitatory connectivity. Note that, because of the global competition, only a subset of the stimulated neurons remains active. To obtain the results shown in Fig. 12, we first stimulated the bottom population for 500 ms, and then after subsequent 500 ms, we stimulated the top population. When the second stimulus is applied, a “state transition” is triggered: as the top population becomes active the bottom one is suppressed. When the second stimulus is removed, the bottom population is completely silent, and the top population remains active, in a self-sustained activity regime. In full FSM systems, the state transition signals would be produced by other neuronal populations (transition populations) responding to both incoming input stimuli and to neurons representing the current state. A complete description and analysis of these neural-network-based FSMs is presented in [29], and working examples of multineuron chips implementing spiking FSMs are described in [131] and [132].

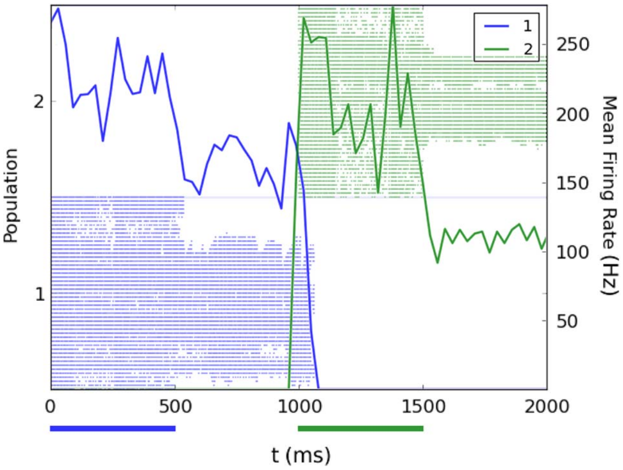


Fig. 12. ***FSM state-holding behavior using a VLSI sWTA architecture. States are represented by two recurrently connected populations of I&F neurons using the hard-wired, on-chip connectivity. Population 1 (bottom half of the raster plot) is stimulated by synthesized Poisson spike trains for the initial 500 ms. Its activity persists due to the recurrent excitatory connectivity, until population 2 (top half of the raster plot) is stimulated. The width and position of the subpopulations depend on the properties of the local connectivity and on their variability. Line plots superimposed to the raster plot represent the mean firing rates computed across each population. The colored bars below the plot represent input stimulus presentations. Input stimuli are composed of Poisson spike trains of 200 Hz lasting for 500 ms, and are applied to all the neurons of one population. The higher variability in the output, e.g., compared with Fig. 11, is due to the absence of mismatch compensation techniques, deliberately omitted to highlight the differences.***

VIII. DISCUSSION

The set of low-power hybrid analog/digital circuits presented in Sections III–V can be used as basic building blocks for constructing adaptive fully parallel, real-time neuromorphic architectures. While several other projects have already developed dedicated hardware implementations of spiking neural networks, using analog [4], digital [23], [138], and mixed mode analog/digital [2], [8] approaches, few [5], [14], [139]–[141] follow the neuromorphic approach originally proposed in the early 1990s [11]. The foundations of this neuromorphic approach were established by pointing out that the implementation of compact and low-power hardware models of biological systems requires the use of transistors in the subthreshold analog domain and the exploitation of the physics of the VLSI medium. We argue that the circuits and architectures presented here adhere to this approach and can, therefore, be used to build efficient biophysically realistic real-time neural processing architectures and autonomous behaving systems.

A. Device Mismatch and Noise

One common criticism to this subthreshold analog VLSI design approach is that circuits operating in this domain have a high degree of noise. However, subthreshold current-mode circuits have lower noise energy (noise power times bandwidth) and superior energy efficiency (bandwidth over power) than above-threshold ones [142], [143]. Another common criticism is that device mismatch in subthreshold circuits is more prominent than in above-threshold circuits. While this observation is correct, device mismatch is a critical problem in any analog VLSI implementation of neural networks (e.g., see the postcalibration neuronal variability measurements of above-threshold accelerated time silicon neuron circuits, presented in [10]). In principle, it is possible to minimize the effect of device mismatch following standard electrical engineering approaches and adopting appropriate analog VLSI design techniques, however we argue that it is not necessary to adopt aggressive mismatch reduction techniques in the type of neuromorphic systems we propose: these techniques would lead to very large transistor or circuit designs, which could, in turn, significantly reduce the number of neurons and synapses integrated onto a single chip (see, for example, [31], where a whole VLSI device was used to implement a single synapse). Rather than attempting to minimize mismatch effects using brute-force engineering techniques at the circuit design level, the neuromorphic engineering approach we promote in this work aims to address these effects at the network and system level, with collective computation, adaptation, and feedback mechanisms. For example, the plasticity mechanisms presented in

Section V are intrinsically robust to mismatch by design, and do not require precisely matched transistors. Moreover, it has been shown how both short- and long-term plasticity mechanisms can be effectively used to reduce the effects of device mismatch in VLSI circuits [68], [144], and how homeostatic plasticity mechanisms can be used to compensate for large changes in the signals affecting the operation of the neurons in multineuron VLSI systems [75]. In addition, the approach of building distributed multichip systems interfaced among each other via the AER protocol (e.g., see Section VI-B) lends itself well to the adoption of event-based mismatch reduction techniques, such as the one proposed in [136], that can be effective even for very large-scale systems (e.g., comprising 1 million silicon neurons) [145]. In addition to being useful for compensating mismatch effects across neurons, homeostatic synaptic scaling circuits, such as the ones described in Section IV-C, can provide another approach to compensating the effects of temperature drifts, complementing dedicated subthreshold bias generator approaches [146], [147]. In summary, this neuromorphic approach makes it possible to tolerate noise, temperature, and mismatch effects at the single device level by exploiting the adaptive features of the circuits and architectures designed, leading to robustness at the system level.

B. Exploiting Variability and Imprecision

The strategy proposed by this approach essentially advocates the construction of distributed and massively parallel computing systems by integrating very compact, but inaccurate and inhomogeneous circuits into large dense arrays, rather than designing systems based on small numbers of very precise, but large and homogeneous computing elements. Indeed, intrinsic variability and diverse activation patterns are often identified as fundamental aspects of neural computation for information maximization and transmission [30], [148]–[150]. The strategy of combining large numbers of variable and imprecise computing elements to carry out robust computation is also followed by a wide set of traditional machine learning approaches. These approaches work on the principle of combining the output of multiple inaccurate computational modules that have slightly different properties, to optimize classification performances and achieve or even beat the performances of single accurate and complex learning systems [151], [152]. A set of similar theoretical studies showed that the coexistence of multiple different time scales of synaptic plasticity (e.g., present due to mismatch in the time constants of the DPI synapse circuits) can dramatically improve the memory performance of ANN [153]. The coexistence of slow and fast learning processes has been shown to be crucial for reproducing the flexible behavior of animals in context-dependent decision-making (i.e., cognitive) tasks and the corresponding single-cell recordings in a neural network model [154].

C. Toward Autonomous Cognitive Systems

Building cognitive systems using noisy and inhomogeneous subthreshold analog VLSI circuits might appear as a daunting task. The neural circuits and architectures presented in this paper represent a useful set of building blocks paving the way toward this goal. These circuits, as well as the analogous one proposed in the literature [155], have been used to build compact, low-power, scalable, computing systems that can interact with the environment [3], [145], [156], learn about the input signals they have been designed to process [85], and exhibit adaptive abilities analogous to those of the biological systems they model [75], [157], [158]. We showed in this paper how the sWTA networks and circuits presented can implement models of working memory and decision making, thanks to their selective amplification and reverberating activity properties, which are often associated to high-level cognitive abilities [21]. Multichip systems employing these architectures can reproduce the results of a diverse set of theoretical studies based on models of sWTA and ANN to demonstrate cognitive properties: for example, Schöner and Sandamirskaya [28], [159] link the types of neural dynamics described in Section VI to cognition by applying similar network architectures to sensory-motor processes and sequence generation; Rutishauser and Douglas [29] show how the sWTA networks described in this paper can be configured to implement FSMs and conditional branching between behavioral states [160]; Rigotti *et al.* [30], [161] describe neural principles, compatible with the ones implemented by the circuits described in Section V, for constructing recurrent neural networks able to produce context-dependent behavioral responses; Giulioni *et al.* [9] demonstrate working memory in a spiking neural network implemented using the same type of silicon neuron circuits and plasticity mechanisms [135] described in Sections III and V.

We recently demonstrated how the circuits and networks presented in Sections III, IV, and VI can be used to synthesize cognition on neural processing systems [20]. Specifically, the neuromorphic multichip system proposed was used to carry out a context-dependent task selection procedure, analogous to the sensory-motor tasks adopted to probe cognition in primates. This is a concrete example showing how neuromorphic systems, built using variable and imprecise circuits, can indeed be configured to express cognitive abilities comparable to those described in [21] and [30].

D. Challenges and Progress in Neuromorphic Engineering

Many years have passed since the first publication on neuromorphic electronic systems [11], and remarkable progress has been made by the small but vibrant neuromorphic engineering (NE) community [162], [163]. For example, the NE community has mastered the art of building real-time sensory-motor reactive systems, by interfacing circuits and networks of the type described in this paper with neuromorphic event-based sensors [164]; new promising neural-based approaches have been proposed that link neuromorphic systems to machine learning [165]–[169]; substantial progress has been made in the field of neuromorphic robots [170]; and we are now able to engineer both large-scale neuromorphic systems (e.g., that comprise the order of 10^6 neurons [171]) and complex multichip neuromorphic systems (e.g., that can exhibit cognitive abilities [20]). However, compared to the progress made in more conventional standard engineering and technology fields, the rate of progress in NE might appear to be disappointingly small. On the one hand, this is due to the fact that NE is still a small community involving a small number of research groups worldwide [e.g., compared to the number of engineers that are assigned to the industrial development of new graphical processing units (GPUs) or central processing units (CPUs)], which lacks the technological infrastructure for automatized design, verification, and configuration tools available for conventional digital integrated circuit (IC) development. On the other hand, scaling and engineering challenges are not the main issue: the major limiting factor that hinders the fast development of neuromorphic engineering is related to our limited understanding of brain function and neural computation, a concept that Carver Mead himself highlighted already over 20 years ago in a video interview (that we transcribe here):

> "I think at the present time we have enough technology to build anything we could imagine. Our problem is, we do not know what to imagine. We don't understand enough about how the nervous system computes to really make more complete thinking systems."

Progress on theoretical and computational neuroscience is accelerating dramatically, also thanks to large-scale funding initiatives recently announced in both Europe and the United States [172], [173]. At the same time, an increasing number of companies are beginning to support research and development in brain-inspired computing technologies [174]–[177]. Supported by these new initiatives, progress in NE is beginning to accelerate as well [178]. In this perspective, reaching the ambitious goal of building autonomous neuromorphic systems able to interact with the environment in real time and to express cognitive abilities is within the realm of possibility. To reach this goal, however, it is important to follow a truly multidisciplinary approach where neuromorphic engineering serves as a medium for the exploration of robust principles of brain computation and not only as a technology platform for the simulation of neuroscience models.

IX. CONCLUSION

In this paper, we proposed circuit and system solutions following the neuromorphic approach originally proposed in [11] for building autonomous neuromorphic cognitive systems. We presented an in-depth review of such types of circuits and systems, with tutorial demonstrations of how to model neural dynamics in analog VLSI. We discussed the problems that arise when attempting to implement spike-based learning mechanisms in physical systems and proposed circuit solutions for solving such problems. We described examples of recurrent neural network implementations that can be used to implement decision making and working-memory mechanisms, and argued how, together with the circuits described in the previous sections, they can be used to implement cognitive architectures. We discussed about the advantages and disadvantages of the approach followed (e.g., for the subthreshold regime of operation or for mismatch in analog subthreshold circuits), and proposed system-level solutions that are inspired by the strategies used in biological nervous systems. Finally, we provided an assessment of the progress made in the NE field so far and proposed strategies for accelerating it and reaching the ambitious goal of building autonomous neuromorphic cognitive systems. ■

Acknowledgment

Many of the circuits and concepts presented here were inspired by the ideas and work of R. Douglas, M. Mahowald, K. Martin, M. Cook, and S. Fusi. The HW/SW infrastructure used to characterize the chips throughout the years and build multichip systems was developed in collaboration with P. Del Giudice, V. Dante, A. Whatley, E. Neftci, D. Fasnacht, and S. Sheik. The authors would also like to thank T. Delbruck, S.-C. Liu and all their other colleagues at the Institute of Neuroinformatics for fruitful discussions and collaborations. They would like to thank the reviewers for their constructive comments.

REFERENCES

[1] X. Jin, M. Luján, L. A. Plana, S. Davies, S. Temple, and S. B. Furber, "Modeling spiking neural networks on SpiNNaker," *Comput. Sci. Eng.*, vol. 12, no. 5, pp. 91–97, Sep.–Oct. 2010.

[2] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 431–438.

[3] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. Olsen, "Neurotech for neuroscience: Unifying concepts, organizing principles, emerging tools," *J. Neurosci.*, vol. 27, no. 44, pp. 11807–11819, 2007.

[4] J. Wijekoon and P. Dudek, "VLSI circuits implementing computational models of neocortical circuits," *J. Neurosci. Methods*, vol. 210, no. 1, pp. 93–109, 2012.

[5] S. Brink, S. Nease, and P. Hasler, "Computing with networks of spiking neurons on a biophysically motivated floating-gate based neuromorphic integrated circuit," *Neural Netw.*, vol. 45, pp. 39–49, Sep. 2013.

[6] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.

[7] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Front. Neurosci.*, vol. 7, no. 11, 2013, DOI: 10.3389/fnins.2013.00011.

[8] J. M. Cruz-Albrecht, T. Derosier, and N. Srinivasa, "A scalable neural chip with synaptic electronics using CMOS integrated memristors," *Nanotechnology*, vol. 24, no. 38, 2013, 384011.

[9] M. Giulioni, P. Camilleri, M. Mattia, V. Dante, J. Braun, and P. Del Giudice, "Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI," *Front. Neurosci.*, vol. 5, no. 149, 2012, DOI: 10.3389/fnins.2011.00149.

[10] M. Schmuker, T. Pfeil, and M. Nawrot, "A neuromorphic network for generic multivariate data classification," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 6, pp. 2081–2086, 2014.

[11] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[12] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, 1992.

[13] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue VLSI," *Annu. Rev. Neurosci.*, vol. 18, pp. 255–281, 1995.

[14] T. Horiuchi and C. Koch, "Analog VLSI-based modeling of the primate oculomotor system," *Neural Comput.*, vol. 11, no. 1, pp. 243–265, Jan. 1999.

[15] G. Indiveri and R. Douglas, "Robotic vision: Neuromorphic vision sensor," *Science*, vol. 288, pp. 1189–1190, May 2000.

[16] G. Indiveri, "A neuromorphic VLSI device for implementing 2-D selective attention systems," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1455–1463, Nov. 2001.

[17] C. Bartolozzi and G. Indiveri, "Selective attention in multi-chip address-event systems," *Sensors*, vol. 9, no. 7, pp. 5076–5098, 2009.

[18] M. Lewis, R. Etienne-Cummings, M. Hartmann, A. Cohen, and Z. Xu, "An in silico central pattern generator: Silicon oscillator, coupling, entrainment, physical computation and biped mechanism control," *Biol. Cybern.*, vol. 88, no. 2, pp. 137–151, 2003.

[19] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45 k neuron, 5 M synapse, 12G connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[20] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. J. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 37, pp. E3468–E3476, 2013.

[21] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[22] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B.-D. Rubin, F. Akopyan, E. McQuinn, W. P. Risk, and D. S. Modha, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2013, DOI: 10.1109/IJCNN.2013.6707077.

[23] A. Cassidy, J. Georgiou, and A. Andreou, "Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Netw.*, vol. 45, pp. 4–26, Sep. 2013.

[24] W. Senn and S. Fusi, "Learning only when necessary: Better memories of correlated patterns in networks with bounded synapses," *Neural Comput.*, vol. 17, no. 10, pp. 2106–2138, 2005.

[25] J. Brader, W. Senn, and S. Fusi, "Learning real world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, pp. 2881–2912, 2007.

[26] A. Renart, P. Song, and X.-J. Wang, "Robust spatial working memory through homeostatic synaptic scaling in heterogeneous cortical networks," *Neuron*, vol. 38, pp. 473–485, May 2003.

[27] G. Deco and E. Rolls, "Neurodynamics of biased competition and cooperation for attention: A model with spiking neurons," *J. Neurophysiol.*, vol. 94, pp. 295–313, 2005.

[28] G. Schöner, *Dynamical Systems Approaches to Cognition*. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 101–126.

[29] U. Rutishauser and R. Douglas, "State-dependent computation using coupled recurrent networks," *Neural Comput.*, vol. 21, pp. 478–509, 2009.

[30] M. Rigotti, D. B. D. Rubin, S. Morrison, C. Salzman, and S. Fusi, "Attractor concretion as a mechanism for the formation of context representations," *NeuroImage*, vol. 52, no. 3, pp. 833–847, 2010.

[31] G. Rachmuth, Z. Shouval, M. Bear, and C.-S. Poon, "A biophysically-based neuromorphic model of spike rate- and timing-dependent plasticity," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 49, pp. E1266–E1274, Dec. 2011.

[32] J. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Netw.*, vol. 21, no. 2–3, pp. 524–534, Mar.–Apr. 2008.

[33] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2007, pp. 3367–3370.

[34] C. Tomazou, F. Lidgey, and D. Haigh, Eds., *Analogue IC Design: The Current-Mode Approach*. Stevenage, U.K.: Peregrinus, 1990.

[35] E. Drakakis, A. Payne, and C. Toumazou, "'Log-domain state-space': A systematic transistor-level approach for log-domain filtering," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 3, pp. 290–305, Mar. 1999.

[36] R. Edwards and G. Cauwenberghs, "Synthesis of log-domain filters from first-order building blocks," *Int. J. Analog Integr. Circuits Signal Process.*, vol. 22, pp. 177–186, 2000.

[37] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI: Circuits and Principles*. Cambridge, MA, USA: MIT Press, 2002.

[38] T. Yu and G. Cauwenberghs, "Log-domain time-multiplexed realization of dynamical conductance-based synapses," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2010, pp. 2558–2561.

[39] S. Mitra, G. Indiveri, and R. Etienne-Cummings, "Synthesis of log-domain integrators for silicon synapses with global parametric control," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 97–100.

[40] A. Destexhe, Z. Mainen, and T. Sejnowski, "Kinetic Models of Synaptic Transmission," in *Methods in Neuronal Modelling, From Ions to Networks*. Cambridge, MA, USA: MIT Press, 1998, pp. 1–25.

[41] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Comput.*, vol. 19, no. 10, pp. 2581–2603, Oct. 2007.

[42] C. Bartolozzi, S. Mitra, and G. Indiveri, "An ultra low power current-mode filter for neuromorphic systems and biomedical signal processing," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2006, pp. 130–133.

[43] J. Arthur and K. Boahen, "Recurrently connected silicon neurons with active dendrites for one-shot learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, vol. 3, pp. 1699–1704.

[44] A. van Schaik and C. Jin, "The tau-cell: A new method for the implementation of arbitrary differential equations," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2003, pp. 569–572.

[45] B. Gilbert, "Translinear circuits: An historical review," *Analog Integr. Circuits Signal Process.*, vol. 9, no. 2, pp. 95–118, Mar. 1996.

[46] J. Arthur and K. Boahen, "Synchrony in silicon: The gamma rhythm," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1815–1825, Nov. 2007.

[47] A. van Schaik, C. Jin, T. Hamilton, S. Mihalas, and E. Niebur, "A log-domain implementation of the Mihalas-Niebur neuron model," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 4249–4252.

[48] A. van Schaik, C. Jin, and T. Hamilton, "A log-domain implementation of the Izhikevich neuron model," in *Proc. Int. Symp. Circuits Syst.*, 2010, pp. 4253–4256.

[49] G. Indiveri, F. Stefanini, and E. Chicca, "Spike-based learning with a generalized integrate and fire silicon neuron," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1951–1954.

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

[51] D. Dupeyron, S. Le Masson, Y. Deval, G. Le Masson, and J.-P. Dom, "A BiCMOS implementation of the Hodgkin-Huxley formalism," in *Proc. 5th IEEE Int. Conf. Microelectron. Neural Fuzzy Bio-inspired Syst.*, Feb. 1996, pp. 311–316.

[52] L. Alvado, J. Tomas, S. Säighi, S. Renaud, T. Bal, A. Destexhe, and G. Le Masson, "Hardware computation of conductance-based neuron models," *Neurocomputing*, vol. 58–60, pp. 109–115, 2004.

[53] M. Simoni, G. Cymbalyuk, M. Sorensen, and R. D. S. Calabrese, "A multiconductance silicon neuron with biologically matched

dynamics," *IEEE Trans. Biomed. Circuits Syst.*, vol. 51, no. 2, pp. 342–354, Feb. 2004.

[54] T. Yu and G. Cauwenberghs, "Analog VLSI biophysical neurons and synapses with programmable membrane channel kinetics," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 139–148, Jun. 2010.

[55] E. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[56] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *J. Neurophysiol.*, vol. 94, pp. 3637–3642, 2005.

[57] S. Mihalas and E. Niebur, "A generalized linear integrate-and-fire neural model produces diverse spiking behavior," *Neural Comput.*, vol. 21, pp. 704–718, 2009.

[58] F. Folowosele, R. Etienne-Cummings, and T. Hamilton, "A CMOS switched capacitor implementation of the Mihalas-Niebur neuron," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2009, pp. 105–108.

[59] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 2898–2901.

[60] B. Connors, M. Gutnick, and D. Prince, "Electrophysiological properties of neocortical neurons *in vitro*," *J. Neurophysiol.*, vol. 48, no. 6, pp. 1302–1320, 1982.

[61] R. Jolivet, T. Lewis, and W. Gerstner, "Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy," *J. Neurophysiol.*, vol. 92, pp. 959–976, 2004.

[62] L. Badel, S. Lefort, R. Brette, C. C. Petersen, W. Gerstner, and M. J. Richardson, "Dynamic I-V curves are reliable predictors of naturalistic pyramidal-neuron voltage traces," *J. Neurophysiol.*, vol. 99, pp. 656–666, 2008.

[63] R. Naud, T. Berger, B. Bathellier, M. Carandini, and W. Gerstner, "Quantitative single-neuron modeling: Competition 2009," *Front. Neur. Conf. Abstract: Neuroinformatics 2009*, pp. 1–8, 2009.

[64] D. Buonomano, "Decoding temporal information: A model based on short-term synaptic plasticity," *J. Neurosci.*, vol. 20, pp. 1129–1141, 2000.

[65] R. Zucker and W. Regehr, "Short-term synaptic plasticity," *Annu. Rev. Physiol.*, vol. 64, pp. 355–405, 2002.

[66] C. Rasche and R. Hahnloser, "Silicon synaptic depression," *Biol. Cybern.*, vol. 84, no. 1, pp. 57–62, 2001.

[67] M. Boegerhausen, P. Suter, and S.-C. Liu, "Modeling short-term synaptic depression in silicon," *Neural Comput.*, vol. 15, no. 2, pp. 331–348, Feb. 2003.

[68] J. Bill, K. Schuch, D. Brüderle, J. Schemmel, W. Maass, and K. Meier, "Compensating inhomogeneities of neuromorphic VLSI devices via short-term synaptic plasticity," *Front. Comput. Neurosci.*, vol. 4, 2010, DOI: 10.3389/fncom.2010.00129.

[69] M. Noack, C. Mayr, J. Partzsch, and R. Schuffny, "Synapse dynamics in CMOS derived from a model of neurotransmitter release," in *Proc. IEEE Eur. Conf. Circuit Theory Design*, 2011, pp. 198–201.

[70] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, and M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Mater.*, vol. 10, no. 8, pp. 591–595, 2011.

[71] T. Dowrick, S. Hall, and L. Mcdaid, "Silicon-based dynamic synapse with depressing response," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1513–1525, Oct. 2012.

[72] L. Abbott, K. Sen, J. Varela, and S. Nelson, "Synaptic depression and cortical gain control," *Science*, vol. 275, no. 5297, pp. 220–223, 1997.

[73] M. Tsodyks and H. Markram, "The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability," *Proc. Nat. Acad. Sci. USA*, vol. 94, no. 2, pp. 719–723, Jan. 1997.

[74] G. Turrigiano, K. Leslie, N. Desai, L. Rutherford, and S. Nelson, "Activity-dependent scaling of quantal amplitude in neocortical neurons," *Nature*, vol. 391, pp. 892–896, Feb. 1998.

[75] C. Bartolozzi and G. Indiveri, "Global scaling of synaptic efficacy: Homeostasis in silicon synapses," *Neurocomputing*, vol. 72, no. 4–6, pp. 726–731, Jan. 2009.

[76] L. Abbott and S. Nelson, "Synaptic plasticity: Taming the beast," *Nature Neurosci.*, vol. 3, pp. 1178–1183, Nov. 2000.

[77] D. Amit and S. Fusi, "Constraints on learning in dynamic synapses," *Network, Comput. Neural Syst.*, vol. 3, no. 4, pp. 443–464, 1992.

[78] S. Fusi and L. Abbott, "Limits on the memory storage capacity of bounded synapses," *Nature Neurosci.*, vol. 10, pp. 485–493, 2007.

[79] D. Amit, *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[80] J. Nadal, G. Toulouse, J. Changeux, and S. Dehaen, "Networks of formal neurons and memory palimpsests," *Europhys. Lett.*, vol. 1, no. 10, 1986, DOI: 10.1209/0295-5075/1/10/008.

[81] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," *Neural Comput.*, vol. 6, no. 5, pp. 957–982, 1994.

[82] S. Fusi, "Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates," *Biol. Cybern.*, vol. 87, pp. 459–470, 2002.

[83] S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. Amit, "Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation," *Neural Comput.*, vol. 12, pp. 2227–2258, 2000.

[84] E. Chicca and S. Fusi, "Stochastic synaptic plasticity in deterministic aVLSI networks of spiking neurons," in *Proc. World Congr. Neuroinf.*, F. Rattay, Ed., 2001, pp. 468–477.

[85] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 1, pp. 32–42, Feb. 2009.

[86] J. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha, and D. J. Friedman, "A 45 nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2011, DOI: 10.1109/CICC.2011.6055293.

[87] S. Sheik, M. Coath, G. Indiveri, S. L. Denham, T. Wennekers, and E. Chicca, "Emergent auditory feature tuning in a real-time neuromorphic VLSI system," *Front. Neurosci.*, vol. 6, no. 17, 2012, DOI: 10.3389/fnins.2012.00017.

[88] M. Giulioni, M. Pannunzi, D. Badoni, V. Dante, and P. Del Giudice, "Classification of correlated patterns with a configurable analog VLSI neural network of spiking neurons and self-regulating plastic synapses," *Neural Comput.*, vol. 21, no. 11, pp. 3106–3129, 2009.

[89] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of 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.

[90] A. Bofill-i-Petit and A. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1296–1304, Sep. 2004.

[91] P. Häfliger, M. Mahowald, and L. Watts, "A spike based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems*, vol. 9, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA, USA: MIT Press, 1997, pp. 692–698.

[92] M. R. Azghadi, S. Al-Sarawi, D. Abbott, and N. Iannella, "A neuromorphic VLSI design for spike timing and rate based synaptic plasticity," *Neural Netw.*, vol. 45, pp. 70–82, 2013.

[93] J. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA, USA: MIT Press, 2006.

[94] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing-based decisions," *Nature Neurosci.*, vol. 9, pp. 420–428, 2006.

[95] W. Senn, "Beyond spike timing: The role of nonlinear plasticity and unreliable synapses," *Biol. Cybern.*, vol. 87, pp. 344–355, 2002.

[96] J. Lisman and N. Spruston, "Postsynaptic depolarization requirements for LTP and LTD: A critique of spike timing-dependent plasticity," *Nature Neurosci.*, vol. 8, no. 7, pp. 839–841, Jul. 2005.

[97] M. Beyeler, N. Dutt, and J. Krichmar, "Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule," *Neural Netw.*, vol. 48, pp. 109–124, Dec. 2013.

[98] J. Lazzaro, S. Ryckebusch, M. Mahowald, and C. Mead, "Winner-take-all networks of$O(n)$ complexity," in *Advances in Neural Information Processing Systems*, vol. 2, D. Touretzky, Ed. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 703–711.

[99] D. Fasnacht and G. Indiveri, "A PCI based high-fanout AER mapper with 2 GiB RAM look-up table, 0.8 μs latency and 66 mhz output event-rate," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 2011, pp. 1–6.

[100] S. Scholze, S. Schiefer, J. Partzsch, S. Hartmann, C. G. Mayr, S. Höppner, H. Eisenreich, S. Henker, B. Vogginger, and R. Schüffny, "VLSI implementation of a 2.8 gevent/s packet based AER interface with routing and event sorting functionality," *Front. Neurosci.*, vol. 5, 2011, DOI: 10.3389/fnins.2011.00117.

[101] D. Fasnacht, A. Whatley, and G. Indiveri, "A serial communication infrastructure for multi-chip address event system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 648–651.

[102] E. Chicca, A. M. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R. J. Douglas, and G. Indiveri, "A multi-chip

pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 5, no. 54, pp. 981–993, May 2007.

[103] F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miro, S. Vicente, G. Jimenez, and A. Civit, "AER tools for communications and debugging," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3253–3256.

[104] G. Mongillo, D. Amit, and N. Brunel, "Retrospective and prospective persistent activity induced by Hebbian learning in a recurrent cortical network," *Eur. J. Neurosci.*, vol. 18, no. 7, pp. 2011–2024, 2003.

[105] R. Douglas, K. Martin, and D. Whitteridge, "A canonical microcircuit for neocortex," *Neural Comput.*, vol. 1, pp. 480–488, 1989.

[106] R. Douglas and K. Martin, "Neural circuits of the neocortex," *Annu. Rev. Neurosci.*, vol. 27, pp. 419–451, 2004.

[107] R. Douglas, C. Koch, M. Mahowald, K. Martin, and H. Suarez, "Recurrent excitation in neocortical circuits," *Science*, vol. 269, pp. 981–985, 1995.

[108] R. Douglas and K. Martin, "Recurrent neuronal circuits in the neocortex," *Current Biol.*, vol. 17, no. 13, pp. R496–R500, 2007.

[109] D. Hansel and H. Sompolinsky, "Modeling feature selectivity in local cortical circuits," in *Methods in Neuronal Modeling*. Cambridge, MA, USA: MIT Press, 1998, pp. 499–567.

[110] S. Amari and M. Arbib, "Competition and cooperation in neural nets," in *Systems Neuroscience*, J. Metzler, Ed. New York, NY, USA: Academic, 1977, pp. 119–165.

[111] P. Dayan and L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA, USA: MIT Press, 2001.

[112] R. Hahnloser, R. Sarpeshkar, M. Mahowald, R. Douglas, and S. Seung, "Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[113] W. Maass, "On the computational power of winner-take-all," *Neural Comput.*, vol. 12, no. 11, pp. 2519–2535, 2000.

[114] R. Ben-Yishai, R. Lev Bar-Or, and H. Sompolinsky, "Theory of orientation tuning in visual cortex," *Proc. Nat. Acad. Sci. USA*, vol. 92, no. 9, pp. 3844–3848, Apr. 1995.

[115] D. Somers, S. Nelson, and M. Sur, "An emergent model of orientation selectivity in cat visual cortical simple cells," *J. Neurosci.*, vol. 15, pp. 5448–5465, 1995.

[116] A. Bennett, "Large competitive networks," *Network*, vol. 1, pp. 449–462, 1990.

[117] E. Chicca, G. Indiveri, and R. Douglas, "Context dependent amplification of both rate and event-correlation in a VLSI network of spiking neurons," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hofmann, Eds. Cambridge, MA, USA: MIT Press, Dec. 2007, pp. 257–264.

[118] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[119] S. Carrillo, J. Harkin, L. J. McDaid, S. Pande, S. Cawley, B.McGinley McGinley, and F. Morgan, "Hierarchical network-on-chip and traffic compression for spiking neural network implementations," in *Proc. 6th IEEE/ACM Int. Symp. Netw. Chip*, 2012, pp. 83–90.

[120] S. Moradi, N. Imam, R. Manohar, and G. Indiveri, "A memory-efficient routing method for large-scale spiking neural networks," in *Proc. IEEE Eur. Conf. Circuit Theory Design*, 2013, DOI: 10.1109/ECCTD. 2013.6662203.

[121] K. Boahen, "A burst-mode word-serial address-event link—I: Transmitter design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.

[122] N. Hatsopoulos, S. Geman, A. Amarasingham, and E. Bienenstock, "At what time scale does the nervous system operate?" *Neurocomputing*, vol. 52, pp. 25–29, 2003.

[123] A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "PyNN: A common interface for neuronal network simulators. front. neuroinform," *Front. Neuroinf.*, vol. 2, 2008, DOI: 10.3389/neuro.11.011.2008.

[124] S. Sheik, F. Stefanini, E. Neftci, E. Chicca, and G. Indiveri, "Systematic configuration and automatic tuning of neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 873–876.

[125] C. Patterson, J. Garside, E. Painkras, S. Temple, L. A. Plana, J. Navaridas, T. Sharp, and S. Furber, "Scalable communications for a million-core neural processing architecture," *J. Parallel Distrib. Comput.*, vol. 72, no. 11, pp. 1507–1520, 2012.

[126] F. Galluppi, S. Davies, A. Rast, T. Sharp, L. A. Plana, and S. Furber, "A hierarchical configuration system for a massively parallel neural hardware platform," in *Proc. ACM 9th Conf. Comput. Front.*, 2012, pp. 183–192.

[127] K. Minkovich, N. Srinivasa, J. Cruz-Albrecht, Y. Cho, and A. Nogin, "Programming time-multiplexed reconfigurable hardware using a scalable neuromorphic compiler," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 889–901, Jun. 2012.

[128] R. Preissl, T. M. Wong, P. Datta, M. Flickner, R. Singh, S. K. Esser, W. P. Risk, H. D. Simon, and D. S. Modha, "Compass: A scalable simulator for an architecture for cognitive computing," in *Proc. IEEE Int. Conf. High Performance Comput. Netw. Storage Anal.*, 2012, article 54.

[129] T. C. Stewart, B. Tripp, and C. Eliasmith, "Python scripting in the nengo simulator," *Front. Neuroinf.*, vol. 3, 2009, DOI: 10.3389/neuro.11.007.2009.

[130] W. Wulf, E. Cohen, W. Corwin, A. Jones, R. Levin, C. Pierson, and F. Pollack, "Hydra: The kernel of a multiprocessor operating system," *Commun. ACM*, vol. 17, no. 6, pp. 337–345, 1974.

[131] E. Neftci, E. Chicca, G. Indiveri, and R. Douglas, "A systematic method for configuring VLSI networks of spiking neurons," *Neural Comput.*, vol. 23, no. 10, pp. 2457–2497, Oct. 2011.

[132] E. Neftci, B. Toth, G. Indiveri, and H. Abarbanel, "Dynamic state and parameter estimation applied to neuromorphic systems," *Neural Comput.*, vol. 24, no. 7, pp. 1669–1694, Jul. 2012.

[133] H. Markram and M. Tsodyks, "Redistribution of synaptic efficacy between neocortical pyramidal neurons," *Nature*, vol. 382, pp. 807–810, 1996.

[134] S. Fusi and M. Mattia, "Collective behavior of networks with linear (VLSI) integrate and fire neurons," *Neural Comput.*, vol. 11, pp. 633–652, 1999.

[135] M. Giulioni, P. Camilleri, V. Dante, D. Badoni, G. Indiveri, J. Braun, and P. Del Giudice, "A VLSI network of spiking neurons with plastic fully configurable 'stop-learning synapses," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, 2008, pp. 678–681.

[136] E. Neftci and G. Indiveri, "A device mismatch compensation method for VLSI spiking neural networks," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2010, pp. 262–265.

[137] T. Massoud and T. Horiuchi, "A neuromorphic VLSI head direction cell system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 1, pp. 150–163, Jan. 2011.

[138] S. Furber and S. Temple, "Neural systems engineering," *J. Roy. Soc. Interface*, vol. 4, no. 13, pp. 193–206, 2007.

[139] K. Boahen, "Neuromorphic microchips," *Sci. Amer.*, vol. 292, no. 5, pp. 56–63, May 2005.

[140] R. Sarpeshkar, "Brain power—Borrowing from biology makes for low power computing—Bionic ear," *IEEE Spectrum*, vol. 43, no. 5, pp. 24–29, May 2006.

[141] K. Hynna and K. Boahen, "Nonlinear influence of T-channels in an in silico relay neuron," *IEEE Trans. Biomed. Circuits Syst.*, vol. 56, no. 6, pp. 1734–1743, Jun. 2009.

[142] R. Sarpeshkar, T. Delbruck, and C. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 9, no. 6, pp. 23–29, Nov. 1993.

[143] B. Shi, "The effect of mismatch in current-versus voltage-mode resistive grids," *Int. J. Circuit Theory Appl.*, vol. 37, pp. 53–65, 2009.

[144] K. Cameron and A. Murray, "Minimizing the effect of process mismatch in a neuromorphic system using spike-timing-dependent adaptation," *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 899–913, May 2008.

[145] S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen, "Silicon neurons that compute," in *Artificial Neural Networks and Machine Learning—ICANN 2012*, vol. 7552, A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 121–128.

[146] T. Delbruck and A. Van Schaik, "Bias current generators with wide dynamic range," *Analog Integr. Circuits Signal Process.*, vol. 43, no. 3, pp. 247–268, 2005.

[147] T. Delbruck, R. Berner, P. Lichtsteiner, and C. Dualibe, "32-bit configurable bias current generator with sub-off-current capability," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1647–1650.

[148] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[149] W. Shew, H. Yang, S. Yu, R. Roy, and D. Plenz, "Information capacity and transmission are maximized in balanced cortical networks with neuronal avalanches," *J. Neurosci.*, vol. 31, no. 1, pp. 55–63, 2011.

[150] E. Schneidman, W. Bialek, and M. B. II, "Synergy, redundancy, independence in population codes," *J. Neurosci.*, vol. 23, no. 37, pp. 11539–11553, 2003.

[151] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[152] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[153] S. Fusi, P. Drew, and L. Abbott, "Cascade models of synaptically stored memories," *Neuron*, vol. 45, pp. 599–611, 2005.

[154] S. Fusi, W. Asaad, E. Miller, and X.-J. Wang, "A neural circuit model of flexible sensori-motor mapping: Learning and forgetting," *Neuron*, vol. 54, no. 2, pp. 319–333, 2007.

[155] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Front. Neurosci.*, vol. 5, 2011, DOI: 10.3389/fnins.2011.00073.

[156] E. Neftci, E. Chicca, M. Cook, G. Indiveri, and R. Douglas, "State-dependent sensory processing in networks of VLSI spiking neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 2789–2792.

[157] G. Indiveri, "A low-power adaptive integrate-and-fire neuron circuit," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2003, pp. IV-820–IV-823.

[158] R. Mill, S. Sheik, G. Indiveri, and S. Denham, "A model of stimulus-specific adaptation in neuromorphic analog VLSI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 5, pp. 413–419, Oct. 2011.

[159] Y. Sandamirskaya and G. Schöner, "An embodied account of serial order: How instabilities drive sequence generation," *Neural Netw.*, vol. 23, no. 10, pp. 1164–1179, 2010.

[160] E. Neftci, J. Binas, E. Chicca, G. Indiveri, and R. Douglas, "Systematic construction of finite state automata using VLSI spiking neurons," in *Biomimetic and Biohybrid Systems*, vol. 7375, T. Prescott, N. Lepora, A. Mura, and P. Verschure, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 382–383.

[161] M. Rigotti, D. B. D. Rubin, X.-J. Wang, and S. Fusi, "Internal representation of task rules by recurrent dynamics: The importance of the diversity of neural responses," *Front. Comput. Neurosci.*, vol. 4, 2010, DOI: 10.3389/fncom.2010.00024.

[162] Telluride Neuromorphic Cognition Engineering Workshop. [Online]. Available: http://ine-web.org/workshops/workshops-overview

[163] The Capo Caccia Workshops Toward Cognitive Neuromorphic Engineering. [Online]. Available: http://capocaccia.ethz.ch

[164] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opinion Neurobiol.*, vol. 20, no. 3, pp. 288–295, 2010.

[165] B. Nessler, M. Pfeiffer, and W. Maass, "STDP enables spiking neurons to detect hidden causes of their inputs *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. I. Williams, and A. Culotta, Eds. Cambridge, MA, USA: MIT Press, 2009, pp. 1357–1365.

[166] A. Steimer, W. Maass, and R. Douglas, "Belief propagation in networks of spiking neurons," *Neural Comput.*, vol. 21, pp. 2502–2523, 2009.

[167] D. Corneil, D. Sonnleithner, E. Neftci, E. Chicca, M. Cook, G. Indiveri, and R. Douglas, "Real-time inference in a VLSI spiking neural network," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2012, pp. 2425–2428.

[168] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Front. Neurosci.*, vol. 7, 2013, DOI: 10.3389/fnins.2013.00178.

[169] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Front. Neurosci.*, vol. 7, 2014, DOI: 10.3389/fnins.2013.00272.

[170] J. Krichmar and H. Wagatsuma, *Neuromorphic and Brain-Based Robots*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[171] P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, and K. Boahen, "A multicast tree router for multichip neuromorphic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 820–833, Mar. 2014.

[172] A. P. Alivisatos, M. Chun, G. M. Church, R. J. Greenspan, M. L. Roukes, and R. Yuste, "The brain activity map project and the challenge of functional connectomics," *Neuron*, vol. 74, no. 6, pp. 970–974, 2012.

[173] H. Markram, "The human brain project," *Sci. Amer.*, vol. 306, no. 6, pp. 50–55, 2012.

[174] E. McQuinn, P. Datta, M. D. Flickner, W. P. Risk, and D. S. Modha, "Connectivity of a cognitive computer based on the macaque brain," *Science*, vol. 339, no. 6119, pp. 513–513, 2013.

[175] IBM Research, "Cognitive computing—Artifical intelligence meets business intelligence," 2013.

[176] Samsung's SAITSamsung Global Research Outreach (GRO) Program, 2013.

[177] Brain CorporationBuilding artificial nervous systems: Technology, 2013.

[178] G. Indiveri and T. Horiuchi, "Frontiers in neuromorphic engineering," *Front. Neurosci.*, vol. 5, 2011, DOI: 10.3389/fnins.2011.00118.

ABOUT THE AUTHORS

Elisabetta Chicca (Member, IEEE) studied physics at the University of Rome, La Sapienza, Italy, where she graduated in 1999. She received the Ph.D. degree in natural sciences from the Physics Department, Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland, and the Ph.D. degree in neuroscience from the Neuroscience Center Zurich (ZNZ), Zurich, Switzerland, in 2006.

Immediately after her Ph.D., she started a postdoctoral research at the Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, where she continued working as a Research Group Leader from May 2010 to August 2011. Since August 2011, she has been an Assistant Professor at Bielefeld University, Bielefeld, Germany, and is heading the Neuromorphic Behaving Systems Group affiliated with the Faculty of Technology and the Cognitive Interaction Technology—Center of Excellence (CITEC). Her current interests are in the development of very large-scale integration (VLSI) models of cortical circuits for brain-inspired computation, learning in spiking VLSI neural networks, and bio-inspired sensing (olfaction, active electrolocation, audition).

Prof. Chicca is a member of the IEEE Biomedical Circuits and Systems Technical Committee and the IEEE Neural Systems and Applications Technical Committee (currently Secretary).

Fabio Stefanini received the Laurea Triennale degree (B.S.) and the "Laurea Magistrale" degree (M.S.) in physics from La Sapienza University of Rome, La Sapienza, Italy, in 2006 and 2009, respectively, and the Ph.D. degree from the Institute of Neuroinformatics of Zurich, Zurich, Switzerland, in 2013, implementing a brain-inspired, real-time pattern recognition system using neuromorphic hardware with distributed synaptic plasticity.

He has been a Research Collaborator at the Institute for Complex Systems, CNR-INFM, Rome, Italy, developing experimental, software, and theoretical methods for the study of collective behavior in flocking birds. His main research interests are in neuromorphic systems with analog VLSI circuits, learning neural networks and complex systems. He currently has a postdoctoral position at the Institute of Neuroinformatics of Zurich. His research involves the development of cortical-inspired smart processing systems for context-aware, embedded processors for resource management in mobile devices. He is one of the creators of PyNCS, a Python package proposed as a flexible, kernel-like infrastructure for neuromorphic systems.

Chiara Bartolozzi (Member, IEEE) received the Laurea degree (with honors) in biomedical engineering from the University of Genova, Genova, Italy, in 2001, the Ph.D. degree in natural sciences from the Physics Department, Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland, and the Ph.D. degree in neuroscience from the Neuroscience Center Zurich (ZNZ), Zurich, Switzerland, both in 2007.

She then joined the Istituto Italiano di Tecnologia, Genova, Italy, first as a Postdoctoral Researcher in the Robotics, Brain, and Cognitive Sciences Department and then as a Researcher in the iCub Facility, where she has been heading the Neuromorphic Systems and Interfaces group. Her main research interest is the design of event-driven technology and their exploitation for the development of novel robotic platforms. To this aim, she coordinated the eMorph (ICT-FET 231467) project that delivered the unique neuromorphic iCub humanoid platform, developing both hardware integration and computational framework for event-driven robotics.

Dr. Bartolozzi is a member of the IEEE Circuits and Systems Society (CASS) Sensory Systems (SSTC) and Neural Systems and Applications (NSA) Committees.

Giacomo Indiveri (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from the University of Genoa, Genova, Italy in 1992. Subsequently, he was awarded a doctoral postgraduate fellowship within the National Research and Training Program on ''Technologies for Bioelectronics'' from which he graduated *summa cum laude* in 1995. He also received the Ph.D. degree in computer science and electrical engineering from the University of Genoa in 2004, and the ''Habilitation'' certificate in neuromorphic engineering from the Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland, in 2006.

He is an Associate Professor at the Faculty of Science, University of Zurich, Zurich, Switzerland. He carried out research on neuromorphic vision sensors as a Postdoctoral Research Fellow in the Division of Biology, California Institute of Technology, Pasadena, CA, USA, and on neuromorphic selective attention systems as a Postdoctoral Researcher at the Institute of Neuroinformatics, University of Zurich and ETHZ. His current research interests lie in the study of real and artificial neural processing systems, and in the hardware implementation of neuromorphic cognitive systems, using full custom analog and digital very large-scale integration (VLSI) technology.

Dr. Indiveri is a member of several Technical Committees (TCs) of the IEEE Circuits and Systems society and a Fellow of the European Research Council.