Neuromorphic electronic circuits for building autonomous cognitive systems

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 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.

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.

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 bio-physics of the neural tissue. This approach, on 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 colleagues [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 brainlike 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 the following sections 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 multi-chip 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 co-localized. 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 timescales [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 equations [40], therefore subthreshold log-domain circuits that implement first order low pass filters 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 Low-Pass Filter (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. 1a, 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:

\[ I_{out} = I_{ref} e^{\frac{V_c}{2V_T}} \]
\[ I_C = C \frac{d}{dt} V_C \]

(1)

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

$$I_{th} \cdot (I_n - I_T - I_C) = (I_T + I_C) \cdot I_{out}. \quad (2)$$

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

$$I_C = C \frac{U_T}{\kappa I_{out}} \frac{d}{dt} I_{out} \quad (3)$$

Finally, by replacing $I_C$ from this equation and dividing everything by $I_T$ in eq. (2), we get:

$$\tau \left(1 + \frac{I_{th}}{I_{out}}\right) \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th} I_n}{I_T} - I_{th} \quad (4)$$

where $\tau \triangleq C U_T / \kappa I_{out}$.

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

$$\tau \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th} I_n}{I_T} \quad (5)$$

The general solution of the other two log-domain circuits shown in Fig. 1b and Fig. 1c can be derived analytically following a similar procedure. Table I 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, 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. 1b). 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 $I_T$ and $-2I_T$, 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. 1a). An other 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 implementation of Mihalas-Niebur [47] and Izhikevich [48] neuron models, and the DPI has been used to implement both synapse and neuron models [41], [49]. In the next sections 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 less 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 the previous Section 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 a low-pass filter (M1,−3), a

| Circuit equations | DPI | LPF | Tau-Cell |
|---|---|---|---|
| $I_{out} = \log \frac{V_{out}}{V_{ref}}$ | $I_{out} = \log \frac{V_{out}}{V_{ref}}$ | $I_{out} = \log \frac{V_{out}}{V_{ref}}$ |
| $I_C = C \frac{dV}{dt}$ | $I_C = -C \frac{dV}{dt}$ | $I_C = C \frac{dV}{dt}$ |
| $I_{in} - I_{out} - I_C = I_T + I_C$ | $I_{in} = I_T + I_C$ | $I_{in} = I_T + I_C$ |
| $I_C = C \frac{dV}{dt}$ | $I_C = C \frac{dV}{dt}$ | $I_C = C \frac{dV}{dt}$ |

**Table I:** Characteristic equations of the DPI, LPF, and Tau-Cell log-domain filters.
variable $I_{\text{mem}}$ and the positive-feedback current $I_a$ of Fig. 2:

$$f(I_{\text{mem}}) = \frac{I_a}{I_t}(I_{\text{mem}} + I_h)$$

(7)

In [49] the authors measured $I_{\text{mem}}$ experimentally and showed how $f(I_{\text{mem}})$ could be fitted with an exponential function of $I_{\text{mem}}$. The other parameters of eq. (6) are defined as:

$$\tau \equiv \frac{C_{\text{mem}} U_T}{k I_t}, \quad \tau_{\text{ahp}} \equiv \frac{C_{\text{p}} U_T}{k I_{\text{ahp}}},$$

$$I_t \equiv I_{\text{th}} e^{\frac{-V_{\text{th}}}{V_k}}, \quad I_{\text{ahp}} \equiv I_{\text{th}} e^{\frac{-V_{\text{ahp}}}{V_k}}$$

where $I_{\text{th}}$ and $I_{\text{ahp}}$ represent currents through n-type MOSFETs not present in Fig. 2, and defined as $I_{\text{th}} \equiv I_{0e} e^{\frac{V_{\text{th}}}{V_k}}$, and $I_{\text{ahp}} \equiv I_{0e} e^{\frac{-V_{\text{ahp}}}{V_k}}$ respectively.

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

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

(8)

where $f(I_{\text{mem}}) \approx \frac{I_{\text{th}}}{I_t} I_{\text{mem}}$.

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 nanosecond 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 re-configurable 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. While modeling the non-linear 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 Post Synaptic 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 \( V_{in} \)) is applied to both \( M_{D3} \) and \( M_{S3} \). The output current \( I_{syn} \), sourced from \( M_{D6} \) and through \( M_{C6} \), 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 \( V_t \) 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 \( V_{in} \) the synaptic weight voltage \( V_w \) decreases, at a rate set by \( V_{std} \). When no spikes are being received, the \( V_w \) “recovers” toward the resting state set by \( V_{w0} \). In [67] the authors demonstrate that this sub-circuit 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 \( I_h \gg I_{syn} \) at the onset of the stimulation (i.e., during the first spikes), the circuit equation, derived from eq. (4) in the analysis of Section II reduces to:

\[
\tau \frac{d}{dt} I_{syn} + \frac{I_{syn}^2}{I_h} - I_{syn} \left( \frac{I_w}{I_t} + 1 \right) = 0
\]

which can be further simplified to:

\[
\tau \frac{d}{dt} I_{syn} = I_{syn} \left( \frac{I_w}{I_t} + 1 \right)
\]

In other words, the change in circuit response increases with every spike, by an amount greater than one, for as long as the condition \( I_{syn} \ll I_h \) is satisfied. As \( I_{syn} \) increases this condition starts to fail, and eventually the opposite condition \( I_{syn} \gg I_h \) is reached. This is the condition for linearity, under which the circuit starts to behave as a first order low-pass filter, 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 the \( V_{thr} \) 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 trade-off 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. 4a. The circuit comprises three main blocks: an input stage M_{11–2}, a spike-triggered weight update block M_{11–4}, and a bi-stability weight storage/refresh block (see transconductance amplifier in Fig. 4a). The input stage receives spikes from pre-synaptic neurons and triggers increases or decreases in weights, depending on the two signals V_{UP} and V_{DN} generated downstream by the post-synaptic neuron. The bi-stability 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 $V_{ohi}$ or $V_{ohl}$, depending on whether $V_w > V_{thw}$ or $V_w < V_{thw}$ respectively. This bi-stable drive is continuous and its effect is superimposed to the one from the spike-triggered weight update circuit. The analog, bi-stable, 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 bi-stability circuit, its effect on the EPSC produced by the connected DPI synapse is exponential. This non-linearity 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 non-linearity 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. 4a 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 post-synaptic 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 pre- and post-synaptic 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 mecha-
nisms that not only depend on the timing of the pre-synaptic spikes but also on other state variables present at the post-synaptic 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 hardware [9], [85], [88] and in software 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. 4b. The spikes produced by the post-synaptic neuron are integrated by the DPI circuit $M_{D1-5}$ to produce a voltage $V_{Ca}$ which represents a post-synaptic Calcium concentration and is a measure of the recent spiking activity of the neuron. The three current-mode winner-take-all circuits $M_{W1-19}$ 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 pre-synaptic spike $V_{spk}$ reaches the synapse weigh-update block of Fig. 4a:

$$\begin{align*}
V_w &= V_w + \Delta w \text{ if } V_{mem} > V_{nah} \text{ and } V_{thk1} < V_{Ca} < V_{thk3} \\
V_w &= V_w - \Delta w \text{ if } V_{mem} < V_{nah} \text{ and } V_{thk1} < V_{Ca} < V_{thk2}
\end{align*}$$

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

The conditions on $V_{Ca}$ implement a “stop-learning” mechanism that greatly improves the generalization performance of the system by preventing over-fitting 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 post-synaptic neuron will fire with a high rate and $V_{Ca}$ will rise such that $V_{Ca} > V_{thk3}$, and no more synapses will be modified.

In [85], [88] the authors 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 the previous Sections 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. 6a), 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 feed-forward 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 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 memo-
Fig. 5: Silicon neuron diagram. This is a schematic representation of a typical circuitual 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.

Fig. 6: sWTA network topology. (a) Schematic representation of the connectivity pattern of the sWTA network. These connections are implemented by synapses with hardwired connections to pre- and post-synaptic 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 8 excitatory neurons are shown for simplicity. (b) Chip architecture. Squares represent excitatory (E) and inhibitory (I) synapses, 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].

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 a RNN with a well defined excitatory/inhibitory connectivity pattern [106] typically referred as soft Winner-Take-All (sWTA) network.

In sWTA neural networks, 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 non-linear operations [108]. The linear operations include analog gain (linear amplification of the feed-forward input, mediated by the recurrent excitation and/or common mode input), and locus invariance [109]. The non-linear operations include non-linear selection [110]–[112], signal restoration [13], [111], and multi-stability [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 multi-chip 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 multi-chip 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 inter-chip 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 multi-chip 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, non-clocked 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 (inter-spike 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 M events/sec, representing the synchronous activity of 1 M neurons firing at a rate of 60 Hz [99], [118]. In general, AER communication infrastructures provide the possibility to implement arbitrary custom multi-chip 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 look-up table with source and destination pairs of addresses, or by more resource-efficient schemes e.g., using multicast or multi-stage routing [6], [119], [120]. This asynchronous digital solution permits flexibility in the configuration (and re-configuration) 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 nano-seconds 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 4 K coincident input events without introducing sensible delays, even with an outdated 350nm 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. A SW/HW echo-system

In order to promptly explore the computational properties of different types of large-scale multi-chip 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 non-spiking activity.

The definition of a 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 whole system. 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 a SW echo-system can be defined to assemble and control the system in a modular, fully reconfigurable way. In this respect, several SW interfaces for neuromorphic and neuro-computing 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 neuro-computing 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 has been collected from multiple neuromorphic VLSI devices and systems. The results presented demonstrate the correct behavior of the circuits described in the previous Sections.

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 pre-synaptic spike, with a slow (e.g., of the order of 100 ms) recovery time [133]. In Fig. 7 we plot the neuron’s membrane potential $V_{mem}$ during the stimulation of one of its excitatory synapses with a regular pre-synaptic 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 it’s membrane potential. Figure 8 shows an example response to the step input current, in which $V_{thrahp} = 0.05 V$, $V_{thrahp} = 0.14 V$, $V_{ahp} = 2.85 V$. 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 non-linear 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 pre-synaptic input spike trains with Poisson distributions. Similarly, the post-synaptic neuron was driven by a current produced via a non-plastic 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 towards 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 post-synaptic 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 \( \theta \) (where \( \theta \) corresponds to the voltage \( V_{\text{th}} \) of Fig. 4a). Long-term transitions occur when a series of pre-synaptic spikes arrive in a short time-frame causing the weight to cross the threshold \( \theta \). 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 pre-synaptic neuron [82], [89]. In the case of the experiment of Fig. 9 an LTD transition has occurred upon the presentation of an input stimulus of 60 Hz for 400 ms. In conclusion, the bi-stability of the synapses and the spike-trains with a high-frequency Poisson-like spike train through a non-plastic excitatory input synapse, in order to produce Poisson-like firing statistics in the output. The dashed line on the \( V_{\text{mem}} \) plot represents the learning threshold voltage \( V_{\text{th}} \) of Fig. 4b. The \( V_{\text{IP}} \) (active low) and \( V_{\text{DN}} \) (active high) signals are the same shown in Fig. 4b and represent the currents that change the synaptic values when triggered by pre-synaptic spikes. They can be considered as eligibility traces that enable the weight update mechanism when they are active.

In Fig. 10b 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 post-synaptic 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. The probability of LTP depends on the pre-synaptic firing rate, elements of the input matrix that correspond to a white pixel have 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 the input pattern multiple times, this pattern gets gradually stored in the synaptic matrix. The bottom left image of Fig. 10b represents the synaptic matrix at the end of the experiment. Furthermore, the stop-learning mechanism described in Sec. V causes a drop in the number of synapses that undergo LTP because as the pattern is stored in the memory the post-synaptic firing rate increases (Fig. 10c).

The above experiments demonstrate the properties of the learning circuits implemented in the VLSI chips. In a feed-forward 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 back-propagation) 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
Fig. 10: Stochastic learning. (a) Single neuron stochasticity. Traces from a VLSI multi-neuron chip with I&F neurons and plasticity circuits as in Fig. 4a. The $V_{UP}$ and $V_{DN}$ signals (top traces) are set by the circuits in Fig. 4b. A Poisson spike-train of high firing rate is sent to the excitatory synapse of a 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 ($V_{leak} = 0.39$ V). This parameter choice allows to exploit the stochasticity of the input spike-trains to produce the highly irregular dynamics of $V_{mem}$. The non-ideal 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. 4a. (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 text for details. (c) Normalized frequency of occurrence of LTP transitions during the experiment of Fig. b, fitted by an exponential function (dashed line).

inputs and to exhibit activity that persists even after the input stimulus has disappeared. We configured the local wired connectivity of a multi-neuron 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 a first experiment, we calibrated the settings and input stimuli to minimize the effect of device mismatch, following the event-based techniques described in [124], [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 [108], [109], [111], 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 a 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 sub-populations 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 the previous sections, this type of networks can support a diverse range of functionalities and have employed in hardware implementation, e.g., for head-direction tracking [137] and memory recall [9]. In particular we concentrated our experiments 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 sub-set 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 multi-neuron chips implementing spiking FSMs are described in [131], [132].

VIII. DISCUSSION

The set of low-power hybrid analog/digital circuits presented in the previous sections 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 nineties [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 sub-threshold 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 sub-threshold analog VLSI design approach is that circuits operating in this domain have a high degree of noise. However sub-threshold 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 sub-threshold 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 post-calibration 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 multi-neuron VLSI systems [75]. In addition, the approach of building distributed multi-chip 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 sub-threshold 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. Towards 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 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]. Multi-chip 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 finite state machines and conditional branching between behavioral states [160]; Rigotti and colleagues [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; Giuliani and colleagues [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 multi-chip 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], [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 of the order of 10⁶ neurons [171]) and complex multi-chip 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 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 is 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 multi-disciplinary 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. Conclusions

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.

ACKNOWLEDGMENTS

Many of the circuits and concepts presented here were inspired by the ideas and work of Rodney Douglas, Misha Mahowald, Kevan Martin, Matthew Cook, and Stefano Fusi. The HW/SW infrastructure used to characterize the chips throughout the years and build multi-chip systems was developed in collaboration with Paolo Del Giudice, Vittorio Dante, Adrian Whatley, Emre Neftci, Daniel Fasnacht, and Sadique Sheik. We acknowledge also Tobi Delbruck, Shih-Chii Liu and all our other colleagues at the Institute of Neuroinformatics for fruitful discussions and collaborations. We would like to thank the reviewers for their constructive comments. This work was supported by the EU ERC Grant “neuroP” (257219), the EU FET Grant “SI-CODE” (284553), and by the Excellence Cluster 227 (CITEC, Bielefeld University).

REFERENCES

[1] X. Jin, et al., “Modeling spiking neural networks on SpiNNaker,” Computing in Science & Engineering, vol. 12, no. 5, pp. 91–97, September-October 2010.
[2] J. Schemmel, J. Fieres, and K. Meier, “Wafer-scale integration of analog neural networks,” in Proceedings of the IEEE International Joint Conference on Neural Networks, 2008.
[3] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. Olsen, “Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools,” Journal of Neuroscience, vol. 27, no. 44, p. 11807, 2007.
[4] J. Wijekoon and P. Dudek, “VLSI circuits implementing computational models of neocortical circuits,” Journal of Neuroscience Methods, vol. 210, no. 1, pp. 93–109, 2012.
[5] S. Brink, S. Neuse, and P. Hasler, “Computing with networks of spiking neurons on a biophysically motivated floating-gate based neuromorphic integrated circuit,” Neural Networks, 2013.
[6] E. Painkras, et al., “SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation,” IEEE Journal of Solid-State Circuits, vol. 48, no. 8, pp.–, August 2013.
[7] T. Pfeil, et al., “Six networks on a universal neuromorphic computing substrate,” Frontiers in neuroscience, vol. 7, 2013.
[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, 384011, 2013.
[9] M. Giuliani, et al., “Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI,” Frontiers in Neuroscience, vol. 5, no. 149, 2012.
[10] M. Schmucker, T. Pfeil, and M. Nawrot, “A neuromorphic network for generic multivariate data classification,” Proceedings of the National Academy of Sciences, vol. 111, no. 6, pp. 2081–2086, 2014.
[11] C. Mead, “Neuromorphic electronic systems,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1629–36, 1990.
[12] M. Mahowald, “VLSI analogs of neuronal visual processing: a synthesis of form and function,” Ph.D. dissertation, Department of Computational and Neural Systems, California Institute of Technology, Pasadena, CA., 1992.
[13] R. Douglas, M. Mahowald, and C. Mead, “Neuromorphic analogue VLSI,” Annu. Rev. Neurosci., vol. 18, pp. 255–281, 1995.
[14] T. Horiiuchi and C. Koch, “Analog VLSI-based modeling of the primate oculomotor system,” Neural Computation, vol. 11, no. 1, pp. 243–265, January 1999.
[15] G. Indiveri and R. Douglas, “ROBOTIC VISION: Neuromorphic vision sensor,” Science, vol. 288, pp. 1189–1190, May 2000.
Elisabetta Chicca (Member, IEEE) studied physics at the University of Rome La Sapienza Italy, where she graduated in 1999. In 2006 she received a PhD in Natural Sciences from the Physics department of the Federal Institute of Technology Zurich (ETHZ), Switzerland, and a PhD in Neuroscience from the Neuroscience Center Zurich (ZNZ). Immediately after the PhD, she started a PostDoc at the Institute of Neuroinformatics at the University of Zurich and ETH Zurich, where she continued working as Research Group Leader from May 2010 to August 2011. Since August 2011, she is an assistant professor at Bielefeld University and is heading the Neuromorphic Behaving Systems Group affiliated to the Faculty of Technology and the Cognitive Interaction Technology - Center of Excellence (CITEC). Her current interests are in the development of VLSI models of cortical circuits for brain-inspired computation, learning in spiking VLSI neural networks, bio-inspired sensing (olfaction, active electrollocation, audition). Elisabetta Chicca is member of the IEEE Biomedical Circuits and Systems TC and IEEE Neural Systems and Applications TC (currently Secretary).

Fabio Stefanini obtained a Laurea Triennale degree (B.S.) and a ”Laurea Magistrale” degree (M.S.) in Physics from La Sapienza University of Rome (Italy) in 2006 and 2009 respectively. He has been a Research Collaborator at the Institute for Complex Systems, CNR-INFN (Rome, Italy), developing experimental, software and theoretical methods for the study of collective behaviour in flocking birds. He obtained a Ph.D. at the Institute of Neuroinformatics in Zurich (Switzerland) implementing a brain-inspired, real-time pattern recognition system using neuromorphic hardware with distributed synaptic plasticity. His main research interests are in neuromorphic systems with analog VLSI circuits, learning neural networks and complex systems. He currently owns a PostDoc 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. Dr. Fabio Stefanini 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 (with honors) degree in biomedical engineering from the University of Genova, Genova, Italy, in 2001 and the Ph.D. degree in Natural Sciences from the Physics department of the Federal Institute of Technology Zurich (ETHZ), Switzerland, and a PhD in Neuroscience from the Neuroscience Center Zurich (ZNZ) in 2007. She then joined the the Istituto Italiano di Tecnologia, Genova, Italy, first as a PostDoc in the Robotics, Brain and Cognitive Sciences Department and then as Researcher in the iCub Facility, where she is 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 the hardware integration and the computational framework for event-driven robotics. She is member of the IEEE Circuits and Systems Society (CASS) Sensory Systems (SSTC) and Neural Systems and Applications (NSA) Committees.

Giaccomo Indiveri (Senior Member, IEEE) is an Associate Professor at the Faculty of Science, University of Zurich, Switzerland. Indiveri received the M.Sc. degree in electrical engineering from the University of Genoa, 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 with “summa cum laude” in 1995. He also obtained a Ph.D. degree in computer science and electrical engineering from the University of Genoa, Italy in 2004, and the “Habilitazione” certificate in Neuromorphic Engineering from ETH Zurich, Switzerland in 2006. Indiveri carried out research on neuromorphic vision sensors as a Postdoctoral Research Fellow in the Division of Biology at the California Institute of Technology, Pasadena, CA, USA, and on neuromorphic selective attention systems as a postdoc at the Institute of Neuroinformatics of the University of Zurich and ETH Zurich, Switzerland. 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 VLSI technology. Indiveri is a member of several Technical Committees (TCs) of the IEEE Circuits and Systems Society and a Fellow of the European Research Council.