INVITED PAPER

Memory and Information Processing in Neuromorphic Systems

In neuromorphic approaches, memories are distributed. This paper reviews the architectures of cortical and deep neural networks inspired by the brain and raises the issues that need to be tackled for these neuromorphic approaches to reach toward full biological richness.

By Giacomo Indiveri, *Senior Member IEEE*, and Shih-Chii Liu, *Senior Member IEEE*

ABSTRACT | A striking difference between brain-inspired neuromorphic processors and current von Neumann processor architectures is the way in which memory and processing is organized. As information and communication technologies continue to address the need for increased computational power through the increase of cores within a digital processor, neuromorphic engineers and scientists can complement this need by building processor architectures where memory is distributed with the processing. In this paper, we present a survey of brain-inspired processor architectures that support models of cortical networks and deep neural networks. These architectures range from serial clocked implementations of multineuron systems to massively parallel asynchronous ones and from purely digital systems to mixed analog/digital systems which implement more biological-like models of neurons and synapses together with a suite of adaptation and learning mechanisms analogous to the ones found in biological nervous systems. We describe the advantages of the different approaches being pursued and present the challenges that need to be addressed for building artificial neural processing systems that can display the richness of behaviors seen in biological systems.

KEYWORDS | Convolutional networks; deep neural networks (DNNs); event-based computation; learning; massively parallel; memristor; neuromorphic computing; plasticity; VLSI; spike-timing-dependent plasticity (STDP); spiking neural network (SNN); von Neumann bottleneck

I. INTRODUCTION

Neuromorphic information processing systems consist of electronic circuits and devices built using design principles that are based on those of biological nervous systems [1]–[4]. The circuits are typically designed using mixed-mode analog/digital complementary metal–oxide–semiconductor (CMOS) transistors and fabricated using standard very large scale integration (VLSI) processes. Similar to the biological systems that they model, neuromorphic systems process information using energy-efficient asynchronous, event-driven, methods [5]. They are often adaptive, fault tolerant, and can be flexibly configured to display complex behaviors by combining multiple instances of simpler elements. The most striking difference between neuromorphic systems and conventional information processing systems is in their use of memory structures. While computing systems based on the classical von Neumann architecture have one or more central processing units physically separated from the main memory areas, both biological and artificial neural processing systems are characterized by colocalized memory and computation (see Fig. 1): the synapses of the neural network implement at the same time memory storage as well as complex nonlinear operators used to perform collective and distributed computation. Given that memory-related constraints, such as size, access latency, and throughput, represent major performance bottlenecks in conventional

Manuscript received December 8, 2014; revised May 25, 2015; accepted June 8, 2015. Date of current version July 15, 2015. This work was supported by the European Union European Research Council (EU ERC) under Grant "neuroP" (257219), and by the EU Future and Emerging Technologies (FET) under Grant "SI-CODE" (284553) and Grant "RAMP" (612058). The work of S. Liu was supported in part by the Swiss National Foundation Stochastic Event Inference Processor under Grant 200021_135066.
The authors are with the Institute of Neuroinformatics, University of Zurich/ETH Zurich, Zurich 8057, Switzerland (e-mail: giacomo@ini.uzh.ch).

Digital Object Identifier: 10.1109/JPROC.2015.2444094

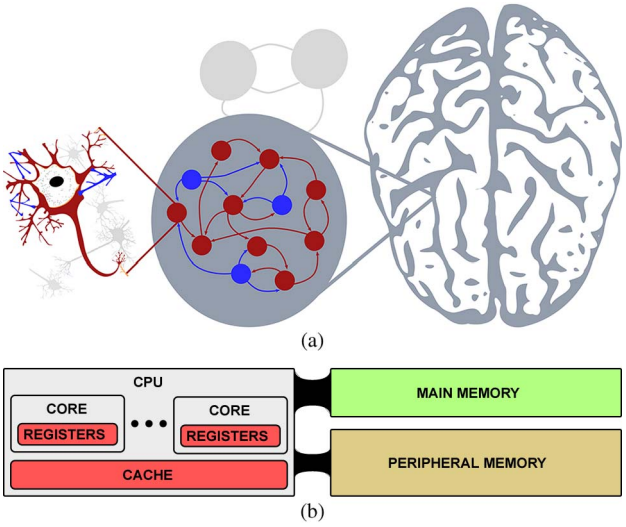

Fig. 1. *Memory hierarchies in brains and computers. In brains, (a) neurons and synapses are the fundamental elements of both neural computation and memory formation. Multiple excitatory and inhibitory neurons, embedded in recurrent canonical microcircuits, form basic computational primitives that can carry out state-dependent sensory processing and computation. Multiple clusters of recurrent networks are coupled together via long-distance connections to implement sensory fusion, inference, and symbolic manipulation. In computers, (b) central processing units (CPUs) containing multiple cores are connected to both main memory and peripheral memory blocks. Each core comprises a microprocessor and local memory (e.g., local registers and cache). The main memory block is the primary storage area, typically larger than the memory blocks inside the CPU, but requiring longer access times. The peripheral memory block requires even longer access times, but can store significantly larger amounts of data.*

computing architectures [6], and given the clear ability of biological nervous systems to perform robust computation, using memory and computing elements that are slow, inhomogeneous, stochastic and faulty [7], [8], neuromorphic brain-inspired computing paradigms offer an attractive solution for implementing alternative non-von-Neumann architectures, using advanced and emerging technologies.

This neuromorphic engineering approach, originally proposed in the late 1980s [9] and pursued throughout the 1990s [2], [10], [11] and early 2000s [12]–[14] by a small number of research labs worldwide, is now being adopted by an increasing number of both academic and industrial research groups. In particular, there have been many recent publications describing the use of new materials and nanotechnologies for building nanoscale devices that can emulate some of the properties observed in biological synapses [15]–[20]. At the network and system level, remarkable brain-inspired electronic multineuron computing platforms have been developed to implement alternative computing paradigms for solving pattern recognition and machine learning tasks [21], [22] and for speeding up the simulation of computational neuroscience models [22], [23]. These latter approaches, however, are only loosely inspired by biological neural processing systems, and are constrained by both precision requirements (e.g., with digital circuits, to guarantee bit-precise equivalence with software simulations, and with analog circuits, to implement as faithfully as possible the equations and specifications provided by the neuroscientists), and bandwidth requirements (e.g., to speed up the simulations by two or three orders of magnitude, or to guarantee that all transmitted signals reach their destinations within some clock cycle duration).

An alternative strategy is to forgo these constraints and emulate biology much more closely by developing new materials and devices, and by designing electronic circuits that exploit their device physics to reproduce the biophysics of real synapses, neurons, and other neural structures [4], [9], [15], [24], [25]. In CMOS, this can be achieved by using field-effect transistors (FETs) operated in the analog "weak-inversion" or "subthreshold" domain [12], which naturally exhibit exponential relationships in their transfer functions, similar, for example, to the exponential dependencies observed in the conductance of sodium and potassium channels of biological neurons [26]. In the subthreshold domain, the main mechanism of carrier transport is diffusion and many of the computational operators used in neural systems (such as exponentiation, thresholding, and amplification) can be implemented using circuits consisting of only a few transistors, sometimes only one. Therefore, subthreshold analog circuits require far fewer transistors than digital for emulating certain properties of neural systems. However, these circuits tend to be slow, inhomogeneous, and imprecise. To achieve fast, robust, and reliable information processing in neuromorphic systems designed following this approach, it is necessary to adopt computational strategies that are analogous to the ones found in nature: for fast processing, low latency, and quick response times these strategies include using massively parallel arrays of processing elements that are asynchronous, real time, and data or event driven (e.g., by responding to or producing spikes). For robust and reliable processing, crucial strategies include both the colocalization of memory and computation, and the use of adaptation and plasticity mechanisms that endow the system with stabilizing and learning properties.

In this paper, we will present an overview of current approaches that implement memory and information processing in neuromorphic systems. The systems range from implementations of neural networks using conventional von Neumann architectures, to custom hardware implementations that have colocalized memory and computation elements, but which are only loosely inspired by biology, to neuromorphic architectures which implement biologically plausible neural dynamics and realistic plasticity mechanisms, merging both computation and memory storage within the same circuits. We will highlight the

advantages and disadvantages of these approaches, pointing out which application domains are best suited to them, and describe the conditions where they can best exploit the properties of new materials and devices, such as oxide-based resistive memories and spin-field effect transistors (spin-FETs).

A. Application Areas for Neuromorphic Systems

Although the origin of the field of neuromorphic engineering can be traced back to the late 1980s, this field is still relatively young when considering the amount of manpower that has been invested in it. Therefore, there are not yet many well-established products and applications in the market that exploit neuromorphic technology to its full potential. However, it has been argued that there are several areas in which neuromorphic systems offer significant advantages over conventional computers [4], [5], [13], [14], [27], such as that of sensory processing [28] or "autonomous systems." An autonomous system can be a simple one, such as a sensory processing system based on environmental sensors or biosensors; or an intermediate-complexity one, such as a brain–machince interface (BMI) making one or two bit decisions based on the real-time online processing of small numbers of signals, sensed continuously from the environment [29]; or a complex one, such as a humanoid robot making decisions and producing behaviors based on the outcome of sophisticated auditory or visual processing [30]. These types of autonomous systems can greatly benefit from the extremely compact and low-power features of neuromorphic hardware technology [1] and can take advantage of the neural style of computation that the neuromorphic hardware substrate supports, to develop new computing paradigms that are better suited to unreliable sensory signals in uncontrolled environments.

Another application domain where dedicated neural processing hardware systems are being used to complement or even to replace conventional computers is that of custom accelerated simulation engines for large-scale neural modeling [22], [31], or of very large-scale spiking neural networks applied to machine learning problems [21]. Also in this application area, the low-power features of the dedicated neuromorphic hardware implementations typically outperform those of general purpose computing architectures used for the same purposes. In Section II, we will describe deep network architectures currently used for machine learning benchmarks and describe examples of neuromorphic hardware network implementations that have been proposed to replace conventional computers used in simulating these architectures. In Section III, we will describe examples of hardware systems that have been proposed to implement efficient simulations of large-scale neural models. Section IV will present an overview of the adaptation, learning, and memory mechanisms adopted by the nervous system to carry out computation, and Section V will present an example of a neuromorphic processor that implements such mechanisms and that can be used to endow autonomous behaving systems with learning abilities for adapting to changes in the environment and interacting with it in real time.

II. FROM CPUs TO DEEP NEURAL NETWORKS

Conventional computers based on the von Neumann architecture typically have one or more central processing units (CPUs) physically separated from the program and data memory elements [see Fig. 1(b)]. The CPUs access both data and program memory using the same shared resources. Since there is a limited throughput between the processors and the memory, and since processor speeds are much higher than memory access ones, CPUs spend most of their time idle. This famous von Neumann bottleneck problem [6] can be alleviated by adding hierarchical memory structures inside the CPUs to cache frequently used data, or by shifting the computational paradigm from serial to parallel. Driven mainly by performance improvement demands, we have been witnessing both strategies in recent years [32]. However, if one takes into account energy consumption constraints, increasing the size or improving the performance of cache memory is not an option. The energy consumption of cache memory is linearly proportional to its size [33]. The alternative strategy is therefore to increase the number of parallel computing elements in the system. Amdahl's law [34] has often been used to evaluate the performance gains of parallel processing in multiprocessor von Neumann architectures [35]. In [36], Cassidy and Andreou demonstrate how performance gained from parallelism is more energy efficient than performance gained from advanced memory optimization and microarchitectural techniques, by deriving a generalized form of Amdahl's law which takes into account communication, energy, and delay factors. A first step toward the implementation of massively parallel neural processing systems is to explore the use of graphical processing units (GPUs), which typically combine hundreds of parallel cores with high memory bandwidths. Indeed several neural simulation tools have been proposed for this purpose [37]–[39]. However, as demonstrated with the cost function derived from the generalized Amdahl's law in [36], even conventional GPU architectures are not optimally suited to running these spiking neural network simulations, when energy consumption is factored in. This is why new custom hardware accelerated solutions, with memory access and routing schemes optimized specifically for neural networks, started to emerge. In particular, a new set of digital architectures have been proposed for implementing "deep" multilayer neural networks, including full custom CMOS solutions [40]–[42] and solutions based on field-programmable gate array (FPGA) devices [43], [44].

A. Deep Networks

Deep networks are neural networks composed of many layers of neurons (see Fig. 2). They are currently the

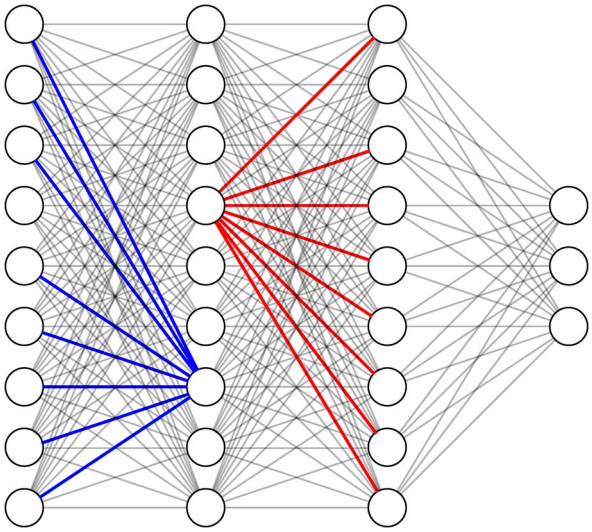

Fig. 2. *Deep neural network example with all-to-all connections. Converging feed-forward connections are highlighted for one example neuron in the second layer; these connections highlight the high fan-in requirements of neuron elements. Diverging feed-forward connections are highlighted for another neuron of layer two; these connections highlight the need for high fan-out support. Connections in these types of networks can be both feed-forward and feed-back.*

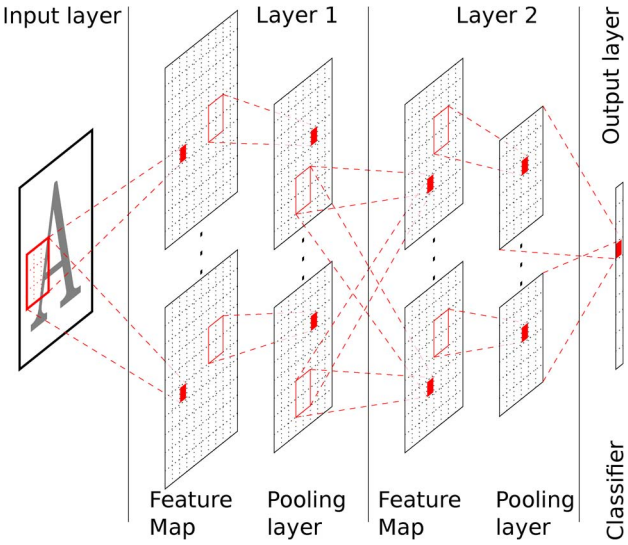

Fig. 3. *Example architecture of a convolutional multilayered network.*

network architecture of choice in the machine learning community for solving a wide range of classification problems, and have shown state-of-the-art performance in various benchmarks tasks such as digit recognition [45]. They include convolutional networks which are being explored intensively within the neuromorphic community for visual processing tasks [46].

B. Convolutional Networks

Convolutional networks consist of a multilayer feedforward network architecture in which neurons in one layer receive inputs from multiple neurons in the previous layer and produce an output which is a thresholded or sigmoidal function of the weighted sum of its inputs (see Fig. 3). The connectivity pattern between the units of one layer and the neuron of the subsequent layer, responsible for the weighted sum operation, forms the convolution kernel. Each layer typically has one or a small number of convolution kernels that map the activity of a set of neurons from one layer to the target neuron of the subsequent layer. These networks were originally inspired by the structure of the visual system in mammals [47], [48], and were used extensively for image processing and machine vision tasks [49]–[51]. They are typically implemented on CPUs and GPUs which consume a substantial amount of power. In recent years, alternate dedicated system-on-chip (SOC) solutions and FPGA platforms have been used to implement these networks for increasing their performance while decreasing their power consumption. Two main approaches are being pursued, depending on the readout scheme of the vision sensor: a frame-based approach, which uses inputs from conventional frame-based cameras, and an event-driven one, which uses inputs from event-driven retina-like vision sensors [52]–[55].

1) Frame-Based Solution: An example of a method proposed for implementing scalable multilayered synthetic vision systems based on a dataflow architecture such as the one shown in Fig. 3 is the "neuFlow" system [56]. The dataflow architecture of this system relies on a 2-D grid of processing tiles (PTs) where each PT has a bank of operators such as multiply, divide, add, subtract, and max; a multiplexer based on-chip router; and a configurable memory mapper block. The architecture is designed to process large streams of data in parallel. It uses a smart direct memory access (DMA) block which interfaces with off-chip memory and provides asynchronous data transfers with priority management. The DMA can be configured to read or write a particular chunk of data, and sends its status to another block called the flow-CPU. The flow-CPU works as a central control unit that can reconfigure the computing grid and the smart DMA at runtime. The configuration data from flow-CPU placed on a runtime configuration bus reconfigures most aspects of the grid at runtime. A full-blown compiler, dubbed "LuaFlow," takes sequential tree-like or flow-graph descriptions of algorithms and parses them to extract different levels of parallelism. With the implementation of this architecture on a Xilinx Virtex 6 ML605 FPGA, the authors demonstrate the segmentation and classification of a street scene using a four-layered network running at 12 frames/s. The same architecture was also implemented in a custom 45-nm silicon-on-insulator (SOI) process application-specific

integrated circuit (ASIC) chip, which was predicted to have, by software simulations, a peak performance of 320 giga-operations per second (GOPS) with a 0.6-W power budget [42]. In comparison, the neuFlow architecture implemented on a standard Xilinx Virtex 6 ML605 FPGA has a peak performance of 16 GOPS with 10 W of power consumption.

To cope with networks of larger numbers of layers and with their related memory bandwidth requirements, a scalable low power system called "nn-X" was presented in [44] and [57]. It comprises a host processor, a coprocessor, and external memory. The coprocessor includes an array of processing elements called collections, a memory router, and a configuration bus. Each collection contains a convolution engine, a pooling module, and a nonlinear operator. The memory router routes the independent data stream to the collections and allows nn-X to have access to multiple memory buffers at the same time. The nn-X system, as prototyped on the Xilinx ZC706 platform, has eight collections, each with a 10×10 convolution engine, and has a measured performance of > 200 GOPS while consuming 4 W.

2) Event-Driven Solution: The implementation of event- or spike-based convolutional network chips was first investigated in the spike-based multichip neuromorphic vision system "CAVIAR"[1] [58]. This system consists of a front–end retina chip, a set of spiking convolution chips, a winner-take-all chip, and a set of learning chips. While the first convolutions chips in CAVIAR were designed using mixed-signal analog/digital circuits, a custom digital version with an array of 32×32 pixels was later implemented in a standard 0.35-μm CMOS process, containing an arbitrary programmable kernel size of up to 32×32 [59]. Extending these chips to implement large-scale networks would require an infrastructure for routing spikes between multiple convolutional chips. In CAVIAR, this was done using an asynchronous communication protocol based on the address–event representation (AER) and a set of multipurpose FPGA routing boards. Currently convolutional networks are both implemented on full custom ASIC platforms [40] and on FPGA platforms [41]. The latest FPGA implementation is done on a Xilinx Virtex-6, and supports up to 64 parallel convolutional modules of size 64×64 pixels [60]. Here, memory is used to store the states of the pixels of the 64 parallel modules and the 64 convolutional kernels of up to size 11×11. When a new event arrives, the kernel of each convolution module is projected around a pixel and a maximum of 121 updates are needed to determine the current state of the convolution. The new state is compared to a threshold to determine if an output spike should be generated, and its value is updated in memory. The event-driven convolutional modules have different memory requirements from the frame-based networks. Since the events arrive asynchronously, the states of the convolutional modules need to be stored in memory all the time. However, since events are processed in sequence, only a single computational adder block is needed, for computing the convolution of the active pixel. A four-layered network which recognizes a small subset of digits was demonstrated using this implementation [60]. Scaling up the network further will require more logic gate resources or a custom digital platform which could support a much larger number of units, such as the one described in Section III-A.

[1]The CAVIAR acronym stands for Convolution Address–Event Representation (AER) Vision Architecture for Real Time. This was a four-year project funded by the European Union under the FP5-IST program in June 2002, within the "Lifelike Perception Systems" subprogram. Its main objective was to develop a bioinspired multichip multilayer hierarchical sensing/processing/actuation system where the chips communicate using an AER infrastructure.

C. Deep Belief Networks

Deep belief networks (DBNs), first introduced by Hinton *et al.* [61] are a special class of deep neural networks with generative properties. They are composed of interconnected pairs of restricted Boltzmann machines (RBMs). They have also been used in a variety of bench-marking tasks [38]. An adaptation of the neural model to allow transfer of parameters to a 784-500-500-10 layer spiking DBN was described in [27] with good performance on the MNIST digit database [62]. This network architecture has been implemented on a Xilinx Spartan-6 LX150 FPGA [43] with very similar classification performance results (92%) on the same MNIST database. This FPGA implementation of the DBN (also called Minitaur) contains 32 parallel cores and 128 MB of DDR2 as main memory (see Fig. 4). Each core has 2048 B of state cache, 8192 B of weight cache, and two digital signal processors (DSPs) for performing fixed-point math. Because of the typical all-to-all connection from neurons of one layer to the next projection layer, memory for storing the weights of these connections is critical. The cache locality of each of the 32 cores is critical to optimizing the neuron weights and the state lookups. The connection rule lookup block in Fig. 4 specifies how an incoming spike is projected to a set of neurons and the connection manager block distributes the updates of the neurons using the 32 parallel cores. This system is scalable but with limitations imposed by the number of available logic gates. On Spartan-6, the used system can support up to 65 536 integrate-and-fire (I&F) neurons.

The Minitaur system, as well as other deep network systems, operate by construction in a massively parallel way with each unit processing local data and using local memory resources. In order to efficiently map these architectures onto FPGAs, DSPs, or classical von Neumann systems, it is necessary to develop custom processing and memory optimized routing schemes that take into account these features [43], [56], [63], [64].

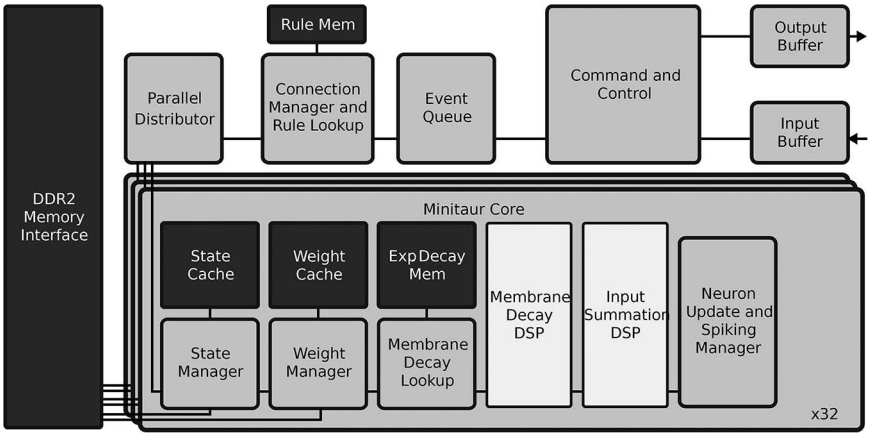

Fig. 4. *Simplified architecture of the Minitaur system. Each core has 2048 B of state cache, 8192 B of weight cache, and two DSPs: one for multiplying the decay and one for summation of the input current. The darker gray blocks indicate use of BRAMs. Events can be streamed from the computer via the input buffer and added to the event queue which also holds the events from the output neurons in the Minitaur core. These events are then streamed out through the output buffer or to the neurons in the core using the connection manager block. Adapted from [43].*

III. LARGE-SCALE MODELS OF NEURAL SYSTEMS

While dedicated implementations of convolutional and deep networks can be extremely useful for specific application domains such as visual processing and pattern recognition, they do not offer a computational substrate that is general enough to model information processing in complex biological neural systems. To achieve this goal, it is necessary to develop spiking neural network architectures with some degree of flexibility, such as the possibility to configure the network connectivity, the network parameters, or even the models of the network's constituent elements (e.g., the neurons and synapses). A common approach that allows a high degree of flexibility and that is closely related to the ones used for implementing convolutional and deep networks is to implement generic spiking neural network architectures using off-the-shelf FPGA devices [64]–[66]. Such devices can be extremely useful for relatively rapid prototyping and testing of neural model characteristics because of their programmability. However, these devices, developed to implement conventional logic architectures with small numbers of input ports (low fan-in) and output ports (low fan-out), do not allow designers to make dramatic changes to the system's memory structure, leaving the von Neumann bottleneck problem largely unsolved. The next level of complexity in the quest of implementing brain-like neural information processing systems is to design custom ASICs using a standard digital design flow [67]. Further customization can be done by combining standard design digital design flow for the processing elements, and custom asynchronous routing circuits for the communication infrastructure.

A. SpiNNaker

This is the current approach followed by the SpiNNaker[2] project [22]. The SpiNNaker system is a multicore computer designed with the goal of simulating the behavior of up to a billion neurons in real time. It is planned to integrate 57 600 custom VLSI chips, interfaced among each other via a dedicated global asynchronous communication infrastructure based on the AER communication protocol [68]–[70] that supports high fan-in and high fan-out connections, and that has been optimized to carry very large numbers of small packets (e.g., representing neuron spikes) in real time. Each SpiNNaker chip is a "system-in-package" device that contains a VLSI die integrating 18 fixed-point advanced RISC machine (ARM) ARM968 cores together with the custom routing infrastructure circuits, and a 128-MB dynamic random access memory (DRAM) die. In addition to the off-chip DRAM, the chip integrates the router memory, consisting of a 1024×32 three-state content-addressable memory (CAM) and a 1024×24 bit random access memory (RAM) module. Furthermore, each ARM core within the chip comprises 64 kB of data memory and 32 kB of instruction memory.

SpiNNaker represents a remarkable platform for fast simulations of large-scale neural computational models. It can implement networks with arbitrary connectivity and a wide variety of neuron, synapse, and learning models (or other algorithms not necessarily related to neural networks). However, the more complex are the models used,

[2]SpiNNaker is a contrived acronym derived from Spiking Neural Network Architecture. The SpiNNaker project is lead by Prof. S. Furber at Manchester University. It started in 2005 and was initially funded by a U.K. Government grant until early 2014. It is currently used as the "many-core" Neuromorphic Computing Platform for the EU FET Flagship Human Brain Project.

the fewer elements that can be simulated in the system. In addition, this system is built using standard computing blocks, such as the ARM cores in each chip. As a consequence, it uses, to a large extent, the same memory hierarchies and structures found in conventional computers [as in Fig. 1(b)], and does not provide a computing substrate that can solve the von Neumann bottleneck problem [6].

B. TrueNorth

The recent implementation of a full custom spiking neural network ASIC by IBM named "TrueNorth"[3] represents a radical departure from classical von Neumann architectures [21]. Although the electronic circuits of TrueNorth use transistors as digital gates, they are fully asynchronous and communicate using event-driven methods. The overall architecture consists of 4096 cores of spiking neural networks integrated into a single CMOS chip. Each core comprises 256 virtual digital leaky I&F neuron circuits, 256×256 binary programmable synaptic connections, and asynchronous encoding, decoding, and routing circuits. Synaptic events can be assigned one of three possible strengths (e.g., to model one type of inhibitory synapse and two excitatory ones with different weights), but they are instantaneous pulses with no temporal dynamics. The dynamics of the neurons is discretized into 1-ms time steps set by a global 1-kHz clock. Depending on the core's synaptic matrix, the source neuron can target from one up to 256 neurons of a destination core. These routing schemes are not as flexible as in the SpiNNaker system, but as opposed to SpiNNaker, this architecture distributes the system memory, consisting of the core synaptic matrix and the routing table entries, across the whole network. The system is inherently parallel, distributed, modular, and (by construction) fault tolerant. The cost of this very high parallelism, however, is relative density inefficiency: the chip, fabricated using an advanced 28-nm CMOS process occupies an area of 4.3 cm^2 and all unused synapses in a given application represent "dark silicon" (silicon area occupied by unused circuits). Note that since also in biology space is precious real estate, unused synapses are typically removed by a dynamic process (see structural plasticity in Section IV-A). In the TrueNorth chip, the synapses do not implement any plasticity mechanism, so they cannot perform online learning or form memories. As a consequence, the goal of colocalizing memory and computation to mitigate the von Neumann bottleneck problem is only partially solved.

C. NeuroGrid

Similar to SpiNNaker and TrueNorth, the goal of the NeuroGrid[4] system [25] is to implement large-scale neural models and to emulate their function in real time. Unlike the previous two approaches, NeuroGrid follows the original neuromorphic engineering vision [1], [2] and uses analog/digital mixed-signal subthreshold circuits to model continuous-time neural processing elements. In particular, important synapse and neuron functions, such as exponentiation, thresholding, integration, and temporal dynamics, are directly emulated using the physics of FETs biased in the subthreshold regime [12].

NeuroGrid consists of a board with 16 standard CMOS "NeuroCore" chips connected in a tree network, with each NeuroCore consisting of a 256×256 array of two-compartmental neurons (see Fig. 5). Synapses are "shared" among the neurons by using the same synapse circuit for different spike sources. Multiple spikes can be superimposed in time onto a single synapse circuit, because it has been designed as a linear integrator filter, and no nonlinear effects are modeled. Each neuron in the array can target multiple destinations thanks to an asynchronous multicast tree routing digital infrastructure. The number of target destinations that a neuron can reach is limited by the size of the memory used in external routing tables and by its access time [71]. However, NeuroGrid increases the fan-out of each neuron by connecting neighboring neurons with local resistive networks or diffusors that model synaptic gap junctions [72]. This structured synaptic organization is modeled after the layered organization of neurons within cortical columns [73]. The full NeuroGrid board, therefore, can implement models of cortical networks of up to one million neurons and billions of synaptic connections with sparse long range connections and dense local connectivity profiles. Like TrueNorth, NeuroGrid represents a radical departure from the classical

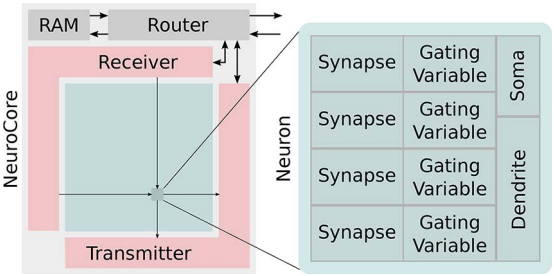

Fig. 5. *NeuroCore chip block diagram (adapted from [25]). The chip comprises a 256×256 array of neuron elements, an asynchronous digital transmitter for sending the events generated by the neurons, a receiver block for accepting events from other sources, a router block for communicating packets among chips, and a memory block for supporting different network configurations. The neuron block comprises four different types of analog synapse circuits that integrate the incoming digital events into analog currents over time, four analog gating variable circuits that model the ion channel population dynamics, a soma circuit that generates the neuron output spikes, and a dendritic circuit that integrates the synaptic currents over space, from neighboring neurons.*

[3]The development of the TrueNorth IBM chip was funded by the U.S. "SyNAPSE"DARPA program, starting in November 2008.

[4]The NeuroGrid project was developed by the group of Prof. K. Boahen in Stanford University, Stanford, CA, USA, and was funded by the U.S. National Institutes of Health (NIH) Pioneer Award granted to Boahen in 2006.

von Neumann computing paradigm. Different memory structures are distributed across the network (e.g., in the form of routing tables, parameters, and state variables). The ability of the shared synapses to integrate incoming spikes reproducing biologically plausible dynamics provides the system with computational primitives that can hold and represent the system state for tens to hundreds of milliseconds. However, the design choice to use linear synapses in the system excluded the possibility of implementing synaptic plasticity mechanisms at each synapse, and therefore the ability of NeuroGrid to model online learning or adaptive algorithms without the aid of additional external computing resources.

NeuroGrid has been designed to implement cortical models of computation that run in real time, and has been used in a closed-loop brain–machine application [74] and to control articulated robotic agents [75]. In this system, time represents itself [9], and data is processed on the fly, as it arrives. Computation is data driven and signals are consumed as they are received. Unlike in conventional von Neumann architectures, time is not "virtualized": signals are not time-stamped and there are no means to store the current state of processing or to transfer time-stamped partial results of signal processing operations to external memory banks for later consumption. Memory and computation are expressed in the dynamics of the circuits, and in the way they are interconnected. So it is important that the system's memory and computing resources have time constants that are well matched to those of the signals they process. As the goal is to interact with the environment and process natural signals with biological time scales, these circuits use biologically realistic time constants which are extremely long (e.g., tens of milliseconds) if compared to the ones used in typical digital circuits. This long time-constant constraint is not easy to achieve using conventional analog VLSI design techniques. Achieving this goal, while minimizing the size of the circuits (to maximize density), is possible only if one uses extremely small currents, such as those produced by transistors biased in the subthreshold domain [4], [12], as is done in NeuroGrid.

D. BrainScales

Another approach for simulating large-scale neural models is the one being pursued in the BrainScales[5] project [76]. BrainScales aims to implement a wafer-scale neural simulation platform, in which each 8-in silicon wafer integrates 50×106 plastic synapses and 200 000 biologically realistic neuron circuits. The goal of this project is to build a custom mixed-signal analog/digital simulation engine that can accurately implement the differential equations of the computational neuroscience models provided by neuroscientists, and reproduce the results obtained from numerical simulations run on standard computers as faithfully as possible. For this reason, in an attempt to improve the precision of the analog circuits, the BrainScales engineers chose to use the above-threshold, or strong-inversion, regime for implementing models of neurons and synapses. However, in order to maximize the number of processing elements on the wafer, they chose to implement relatively small capacitors for modeling the synapse and neuron capacitances. As a consequence, given the large currents produced by the above-threshold circuit and the small capacitors, the BrainScales circuits cannot achieve the long time constants required for interacting with the environment in real time. Rather, their dynamics are "accelerated" with respect to typical biological times by a factor of 10^3 or 10^4. This has the advantage of allowing very fast simulation times which can be useful, e.g., to investigate the evolution of network dynamics over long periods of time, once all the simulation and network configuration parameters have been uploaded to the system. But it has the disadvantage of requiring very large bandwidths and fast digital, high-power, circuits for transmitting and routing the spikes across the network.

Like NeuroGrid, the synaptic circuits in BrainScales express temporal dynamics, so they form memory elements that can store the state of the network (even if only for a few hundreds of microseconds). In addition, the BrainScales synapses also include circuits endowed with spike-based plasticity mechanisms that allow the network to learn and form memories [77]. Therefore, BrainScales implements many of the principles that are needed to build brain-inspired information processing systems that can replace or complement conventional von Neumann computing systems. However, given the circuit design choices made for maximizing precision in reproducing numerical simulation results of given differential equations, the system is neither low power (e.g., when compared to the other large-scale neural processing systems previously described), nor compact. To build neural information processing systems that are at the same time compact, low power, and robust, it will be necessary to follow an approach that can use extremely compact devices (such as nanoscale memristive synapse elements), very low-power circuit design approaches (e.g., with subthreshold current-mode designs), and by adopting adaptation and learning techniques that can compensate for the variability and inhomogeneity present in the circuits at the system level.

IV. ADAPTATION, LEARNING, AND WORKING MEMORY

Adaptation and learning mechanisms in neural systems are mediated by multiple forms of "plasticity," which operate

[5]The BrainScales acronym stands for "Brain-inspired multiscale computation in neuromorphic hybrid systems." This project was funded by the European Union under the FP7-ICT program, and started in January 2011. It builds on the research carried out in the previous EU Fast Analog Computing with Emergent Transient States (FACETS) project, and is now part of the EU FET Flagship Human Brain Project (HBP). The mixed-signal analog/digital system being developed in this project currently represents the "physical model" Neuromorphic Computing Platform of the HBP.

TABLE 1 Memory Structures in Computers Versus Memory Structures in Brains and Neuromorphic Systems

Relative time scale	Computers	Brains and Neuromorphic Systems
Short	Fast local memory (e.g., registers and cache memory banks inside the CPU)	Synapse and neural dynamics at the single neuron level (short-term plasticity, spike-frequency adaptation, leak, etc.)
Medium	Main memory (e.g., dynamic RAM)	Spike-based plasticity (STDP, Hebbian learning, etc.) Formation of working memory circuits (e.g., recurrent networks, attractors, etc.)
Long	Peripheral memory (e.g., hard-disks)	Structural plasticity, axonal growth, long-term changes in neural pathways.

on a wide range of time scales [78]. The most common forms are structural plasticity, homeostatic plasticity, long-term potentiation (LTP), and long-term depression (LTD) mechanisms, and short-term plasticity (STP) mechanisms. While these mechanisms are related to the ability of single neurons and synapses to form memories, the term "working memory" is often used to refer to the ability of full networks of neurons to temporarily store and manipulate information. A common model that has been proposed to explain the neural basis of working memory is that based on "attractor networks" [79]–[82]. In this section, we give a brief overview of single neuron and network level mechanisms observed in biology that subserve the function of memory formation, and of the neuromorphic engineering approaches that have been followed to implement them in electronic circuits. Examples of analogies between the memory structures in conventional von Neumann architectures and the plasticity and memory mechanisms found in neural and neuromorphic systems are shown in Table 1.

A. Plasticity

Structural plasticity refers to the brain's ability to make physical changes in its structure as a result of learning and experience [83], [84]. This mechanism, which typically operates on very long time scales, ranging from minutes to days or more, is important for the formation and maintenance of long-term memories. Homeostatic plasticity is a self-stabilizing mechanism that is used to keep the activity of the neurons within proper operating bounds [85]. It is a process that typically operates on relatively long time scales, ranging from hundreds of milliseconds to hours. From the computational point of view, this mechanism plays a crucial role for adapting to the overall activity of the network, while controlling its stability. Short-term plasticity on the other hand is a process that typically operates on time scales that range from fractions of milliseconds to hundreds of milliseconds. It can manifest itself as both short-term facilitation or short-term depression, whereby the strength of a synapse connecting a presynaptic (source) neuron to a postsynaptic (destination) one is up-regulated or down-regulated, respectively, with each pre-synaptic spike [86]. It has been demonstrated that this mechanism can play a fundamental role in neural computation [81], [87], [88], e.g., for modulating the neuron's sensitivity to its input signals. Finally, long-term plasticity is the mechanism responsible for producing long-lasting, activity-dependent changes in the synaptic strength of individual synapses [89]. A popular class of LTP mechanisms that has been the subject of widespread interest within the neuroscience community [90], the neuromorphic community [91], and more recently in the material science and nanotechnology community [20], [92], is based on the spike-timing-dependent plasticity (STDP) rule [93], [94]. In its simplest form, the relative timing between the presynaptic and postsynaptic spikes determines how to update the efficacy of a synapse. In more elaborate ones, other factors are taken into account, such as the average firing rate of the neurons [95], their analog state (e.g., the neuron's membrane potential) [96], [97], and/or the current value of the synaptic weights [98], [99]. The time scales involved in the STDP timing window for the single weight update are of the order of tens to hundreds of milliseconds. But the LTP and LTD changes induced in the synaptic weights last for much longer time scales, ranging from hours to weeks [100].

B. Attractor Networks

Mechanisms operating at the network level can also allow neural processing systems to form short-term memories, consolidate long-term ones, and carry out nonlinear processing functions such as selective amplification (e.g., to implement attention and decision making). An example of such a network-level mechanism is provided by "attractor networks." These are networks of neurons that are recurrently connected via excitatory synapses, and that can settle into stable patterns of firing even after the external stimulus is removed. Different stimuli can elicit different stable patterns, which consist of specific subsets of neurons firing at high rates. Each of the high-firing rate attractor states can represent a different memory [80]. To make an analogy with conventional logic structures, a small attractor network with two stable states would be equivalent to a flip-flop gate in CMOS.

A particularly interesting class of attractor networks is the one of soft winner-take-all (sWTA) neural networks [102]. In these networks, groups of neurons both cooperate and compete with each other. Cooperation takes place between groups of neurons spatially close to each other, while competition is typically achieved through global recurrent patterns of inhibitory connections [103]. When stimulated by external inputs, the neurons excite their

neighbors and the ones with highest response suppress all other neurons to win the competition. Thanks to these competition and cooperation mechanisms, the outputs of individual neurons depend on the activity of the whole network and not just on their individual inputs [104]. Depending on their parameters and input signals, sWTA networks can perform both linear and complex nonlinear operations [104] (see Fig. 6), and have been shown to posses powerful computational properties for tasks involving feature extraction, signal restoration, and pattern classification [105]. Given their structure and properties, they have been proposed as canonical microcircuits that can explain both the neuroanatomy and the neurophysiology data obtained from experiments in the mammalian cortex [106]. Such networks have also been linked to the dynamic neural fields (DNFs) typically used to model behavior and cognition in autonomous agents [107], [108].

C. Neuromorphic Circuit Implementations

The neuromorphic engineering community has been building physical models of sWTA networks [109]–[111], attractor networks [112], [113], and plasticity mechanisms [91] that cover the full range of temporal and spatial scales described in Section IV-A for many years. For example, several circuit solutions have been proposed to implement short-term plasticity dynamics, using different types of devices and following a wide range of design techniques [114]–[119]; a large set of spike-based learning circuits have been proposed to model long-term plasticity [77], [91], [120]–[128]; multiple solutions have been proposed for implementing homeostatic plasticity mechanisms [129], [130]; impressive demonstrations have been made showing the properties of VLSI attractor networks [4], [23], [112], [113]; while structural plasticity has been implemented both at the single chip level, with morphology learning mechanisms for dendritic trees [131] and at the system level, in multichip systems that transmit spikes using the AER protocol, by reprogramming or "evolving" the network connectivity routing tables stored in the digital communication infrastructure memory banks [132], [133]. While some of these principles and circuits have been adopted in the deep network implementations of Section II and in the large-scale neural network implementations of Section III, many of them still remain to be exploited, at the system and application level, for endowing neuromorphic systems with additional powerful computational primitives.

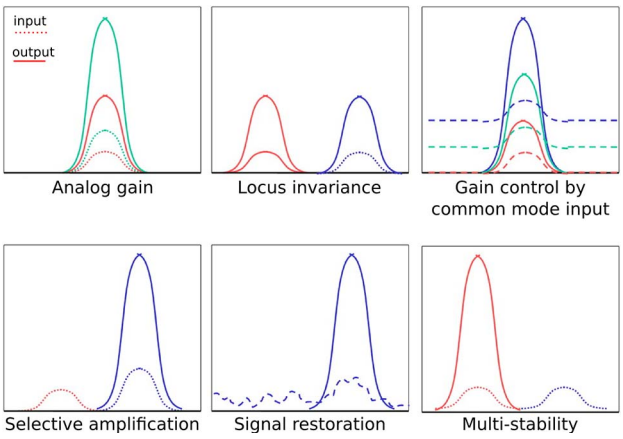

Fig. 6. *Soft winner-take-all network behaviors: linear (top row) and nonlinear (bottom row). The horizontal axis of each trace represents the spatial location of the neurons in the network, while the vertical axis represents the neurons response amplitude. Figure adapted from [101].*

V. NEUROMORPHIC PROCESSOR

An example of a recently proposed neuromorphic multineuron chip that integrates all of the mechanisms described in Section IV is the ROLLS neuromorphic processor[6] [134]. This device implements a configurable spiking neural network using slow subthreshold neuromorphic circuits that directly emulate the physics or real neurons and synapses, and fast asynchronous digital logic circuits that manage the event-based AER communication aspects as well as the properties of the network. While the analog circuits faithfully reproduce the neural dynamics and the adaptive and learning properties of neural systems, the asynchronous digital circuits provide a flexible means to configure both parameters of the individual synapse and neuron elements in the chip, as well as the connectivity of the full network. The goal of the approach followed in designing this device was not to implement large numbers of neurons or large-scale neural networks, but to integrate many nonlinear synapses for exploring their distributed memory and information processing capabilities. Although the analog circuits in the ROLLS neuromorphic processor are characterized by susceptibility to noise, variability, and inhomogeneous properties (mainly due to device mismatch) [134], the multiple types of plasticity mechanisms and the range of temporal dynamics present in these circuits endow the system with a set of collective and distributed computational operators that allow it to implement a wide range of robust signal processing and computing functions [134]. The device chip micrograph is depicted in Fig. 7. As evidenced from Fig. 7, most of the area of the device is dedicated to the synapses, which represent the site of both memory and computation.

The chip, fabricated using a standard six-metal 180-nm CMOS process, occupies an area of 51.4 mm^2 and has approximately 12.2 million transistors. It comprises 256 neurons and 133 120 synapses, equivalent to 130-kB

[6]The ROLLS neuromorphic processor device was designed at the Institute of Neuroinformatics, University of Zurich, ETH Zurich, Zurich, Switzerland. Its development was funded by the EU ERC "Neuromorphic Processors" (NeuroP) project, awarded to Giacomo Indiveri in 2011.

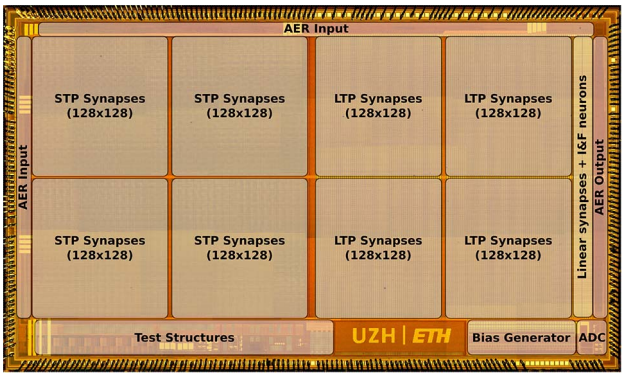

Fig. 7. ***ROLLS neuromorphic processor: micrograph of a neuromorphic processor chip that allocates most of its area to nonlinear synapse circuits for memory storage and distributed massively parallel computing.***

"memory" elements. The synapse circuits are of three different types: linear time-multiplexed (shared) synapses, STP synapses, and LTP synapses. The linear synapses are subdivided into blocks of four excitatory and four inhibitory synapse integrator circuits per neuron, with shared sets of synaptic weights and time constants. The STP synapses are arranged in four arrays of 128 × 128 elements. Each of these elements has both analog circuits that can reproduce short-term adaptation dynamics and digital circuits that can set and change the programmable weights. The LTP synapses are subdivided into four arrays of 128 × 128 elements which contain both analog learning circuits and digital state-holding logic. The learning circuits implement the stochastic plasticity STDP model proposed in [135] to update the synaptic weight upon the arrival of every presynaptic input spike. Depending on the analog value of the weight, the learning circuits also drive the weight to either a high LTP state, or a low LTD state on very long time scales (i.e., hundreds of milliseconds), for long-term stability and storage of the weights (see [134] for a thorough description and characterization of these circuits). The digital logic in the LTP synapse elements is used for configuring the network connectivity. The silicon neuron circuits on the right-hand side of the layout of Fig. 7 implement a model of the adaptive exponential I&F neuron [136] that has been shown to be able to accurately reproduce electrophysiological recordings of real neurons [137], [138]. Additional circuits are included in the I&F neuron section of Fig. 7 to drive the learning signals in the LTP arrays, and to implement self-tuning synaptic scaling homeostatic mechanism [139] on very long time scales (i.e., seconds to minutes) [130]. The currents produced by the synapse circuits in the STP and LTP arrays are integrated by two independent sets of low-power log-domain pulse integrator filters [140] that can reproduce synaptic dynamics with time constants that can range from fractions of microseconds to hundreds of milliseconds. The programmable digital latches in the synapse elements can be used to set the state of the available all-to-all network connections, therefore allowing the user to configure the system to implement arbitrary network topologies, with the available 256 on-chip neurons, ranging from multilayer deep networks, to recurrently connected reservoirs, to winner-take-all networks, etc. All analog parameters of synapses and neurons can be configured via a temperature compensated programmable bias generator [141].

To demonstrate the neuromorphic processor's memory and information processing abilities we trained the network to encode memory patterns into four different attractor networks, such as those described in Section IV-B (see Fig. 8). The protocol followed to train the silicon neurons to form these associative memories is the following: we created a recurrent competitive network, by connecting the chip's 256 neuron outputs to 90% of their 256 × 256 plastic LTP synapses, via recurrent excitatory connections, and to a random subset of 50% of the nonplastic STP inhibitory synapses, via recurrent inhibitory connections; we initialized the LTP bistable plastic synapses to a low state with 90% probability, and to a high state with 10% probability [see the first plot of Fig. 8(a)]; we configured the parameters of the learning circuits to induce LTP and LTD transitions in an intermediate range of firing rates (e.g., between 50 and 150 Hz), and to stop changing the weights for higher frequencies (see [134] for a detailed description of the algorithm and circuits that implement these features); we then stimulated four separate groups of neurons repeatedly by stimulating the nonplastic synapses with Poisson-distributed input spike trains for one second each [see Fig. 8(b)]. With each stimulus presentation, the plastic synapses of the neurons receiving both feedforward input from the Poisson input spike trains and recurrent feedback from the output neurons tended to potentiate, while the plastic synapses of the neurons that did not receive feedback spikes correlated with feedforward inputs tended to depress [see second, third, and fourth plots of Fig. 8(a)]. As the number of potentiated plastic synapses increased, the populations of recurrently connected neurons started to produce sustained activity, with higher firing rates and more structured response properties [see second, third, and fourth plots of Fig. 8(a)]. The attractors are fully formed when enough recurrent connections potentiate, such that the firing rate of the stimulated population is high enough to stop the learning process and the population activity remains sustained even after the input stimulus is removed [e.g., see activity during $t = 1$–2 s, $t = 3$–4 s, $t = 5$–6 s, and $t = 7$–9 s in Fig. 8(b)]. When neurons belonging to different attractors are stimulated, they suppress activity of all other attractors via the recurrent inhibitory connections. Fig. 8(c) shows an example of a silicon neuron output trace measured at the beginning of the experiment, when no attractors exist, in the middle, as the attractors are being formed, and at the end of the experiment, during sustained activity of a fully formed

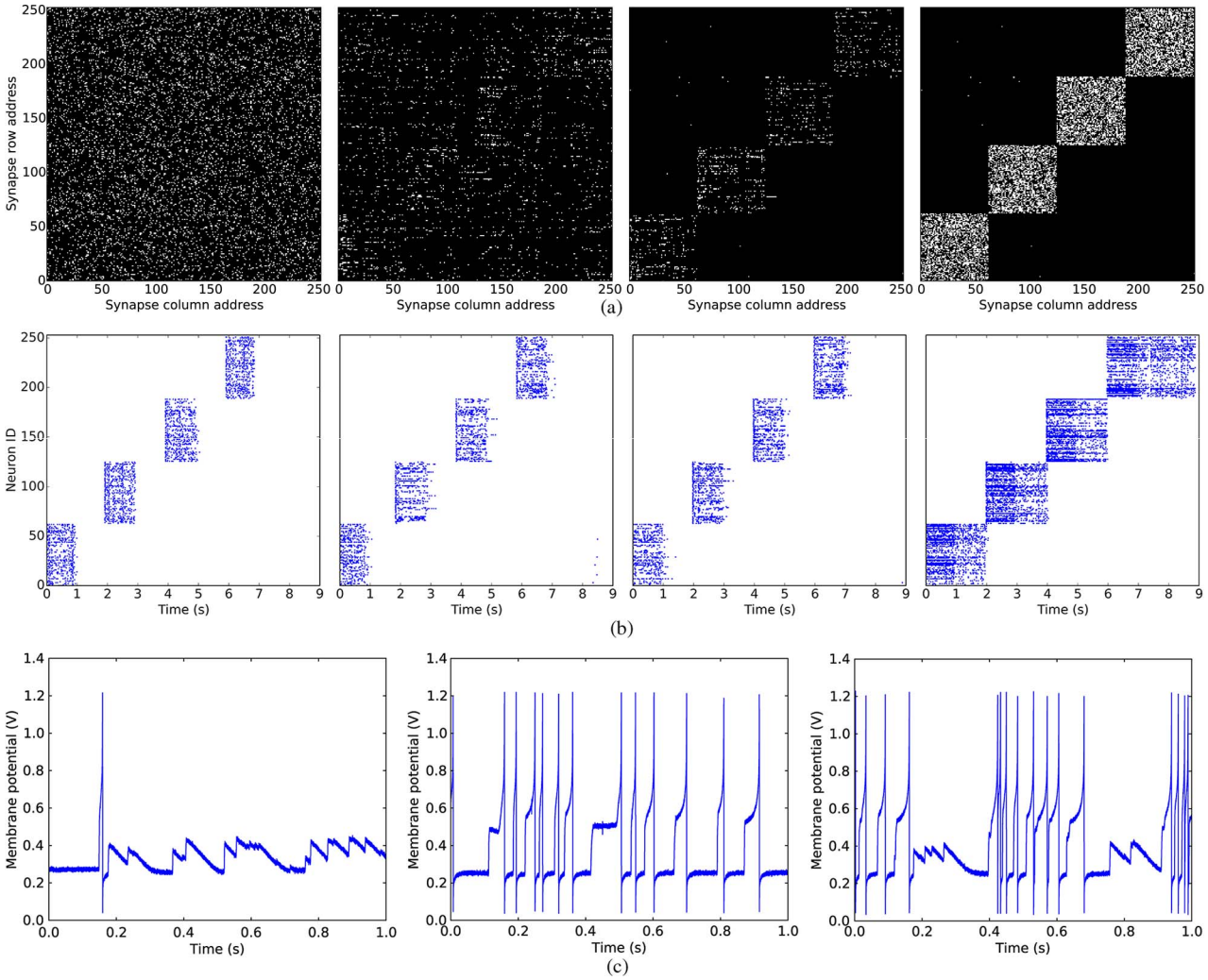

Fig. 8. *Forming stable attractors in the ROLLS neuromorphic processor. (a) State of the bistable LTP synapses at the beginning, during, and at the end of the experiment. (b) Raster plots representing the response of the neurons, both during input stimulus presentation (for one second at $t = 0$ s, 2 s, 4 s, 6 s), and after stimulus is removed. (c) Example of a measured neuron membrane potential at the beginning, during, and at the end of the training sessions, during stimulus presentation.*

attractor. Note that although the analog circuits in the device have a high degree of variability (with a coefficient of variation of about 10% [134]), and that the learning process is stochastic [135], the attractors are formed reliably, and their population response is sustained robustly. Furthermore, the learning process is online and always on. Thanks to the spike-based perceptron-like features of the learning circuits [134], [135], the synapses stop increasing their weights once the attractors are fully formed, and can start to change again and adapt to the statistics of the input, should the input signals change. As discussed in Section IV-B, these attractor networks represent an extremely powerful computational primitive that can be used to implement sophisticated neuromorphic processing modules. In previous work [4], [101], we showed how the neuron, synapse, and learning circuits of the ROLLS neuromorphic processor can be used to implement other types of powerful computational primitives, such as the sWTA network of Section IV. In [142], we showed how these primitives can be combined to synthesize fully autonomous neuromorphic cognitive agents able to carry out context-dependent decision making in an experiment analogous to the ones that are routinely done with primates to probe cognition. By properly defining the types of spiking neural networks in the ROLLS neuromorphic processor, and by properly setting its circuit parameters, it is therefore possible to build already now small-scale embedded systems that can use its distributed memory resources to learn about the statistics of the input signals and of its internal state, while interacting with the environment in real time, and to provide state- and context-dependent information processing.

VI. EMERGING NANOTECHNOLOGIES

An additional resource for building complex brain-like cognitive computing systems that are compact and low power is provided by the large range of emerging nanoscale devices that are being proposed to replace the functionality of larger and bulkier CMOS circuits currently deployed for modeling synapses and neurons. Recent research in nanoscale materials is revealing the possibility of using novel devices to emulate the behavior of real synapses in artificial neural networks, and in particular to reproduce their learning and state-holding abilities. The general goal is to exploit the nonvolatile memory properties of these devices and their ability to keep track of their state's past dynamics to implement massively parallel arrays of nanoscale elements integrated into neuromorphic VLSI devices and systems. For example, in [143], we showed how it is possible to integrate memristive devices in CMOS synapse arrays of the type used in the ROLLS neuromorphic processor of Section V. A promising technology is resistive random access memories (RRAMs) [144], which exploit resistance switching phenomena [145], and are very attractive due to their compatibility with CMOS technology. The base element of an RRAM device is a two-terminal element with a top electrode, a bottom one, and a thin film, sandwiched between the electrodes. By applying a voltage across the electrodes, the electrical conductivity of the thin film material can be reversibly changed, from a high conductive to a high resistive state and *vice versa*, and the corresponding conductance value can be stored for a long period. Several proposals have been made for leveraging basic nanoscale RRAM attributes in synapse circuits in neuromorphic architectures [15], [18], [92], [143]; many of these proposals do not use these devices as conventional RAM cells, but distribute them within and across the synapse circuits in the neuromorphic architectures. It has been shown that these RRAM-based neuromorphic approaches can potentially improve density and power consumption by at least a factor of 10, as compared with conventional CMOS implementations [24].

Other approaches that also store memory state as resistance, but that exhibit a range of different behaviors include spin-transfer torque magnetic random access memories (STT–MRAMs) [146]–[148], ferroelectric devices [149], and phase change materials [150]–[152]. In general, oxide-based RRAMs, STT–MRAMs, and phase-change memories are under intense industrial development. Although, these technologies are currently difficult to access for nonindustrial applications, basic research in this domain has very high potential, because neuromorphic circuits can harness the interesting physics being discovered in these new devices to extend their applicability: in addition to developing nanoscale materials and devices that can emulate the biophysics of real synapses and neurons, this research can lead to understanding how to exploit their complex switching dynamics to reproduce relevant computational primitives, such as state-dependent conductance changes, multilevel stability, and stochastic weight updates, for use in large-scale neural processing systems [20], [143].

VII. DISCUSSION

The array of possible neuromorphic computing platforms described in this work illustrates the current approaches used in tackling the partitioning of memory and information processing blocks on these systems. Here we discuss the advantages and disadvantages of the different approaches and summarize the features of the neuromorphic computing platforms presented.

A. Large-Scale Simulation Platforms

Neural network simulation frameworks based on C or Python [37], [153], [154] which can run on conventional CPU- and GPU-based systems, offer the best flexibility and quickest development times for simulating large-scale spiking networks. These simulations however can still take a long time on conventional computing platforms with the additional disadvantage of very high power consumption figures (e.g., up to tens of megawatts). Dedicated hardware solutions have been built to support fast simulations of neural models, and to reduce their power-consumption figures. Hardware platforms such as SpiNNaker, BrainScales, and NeuroGrid fall into this category.

Questions still remain whether large-scale simulations are necessary to answer fundamental neuroscience questions [155], [156]. It is also not clear whether the compromises and tradeoffs made with these custom hardware implementations will restrict the search for possible solutions to these fundamental neuroscience questions or dissuade neuroscientists from using such platforms for their work. For example, the models of neurons and synapses implemented on Neurogrid and BrainScales are hardwired and cannot be changed. On the SpiNNaker platform they are programmable, but there are other constraints on the type of neurons, synapses, and networks that can be simulated imposed by the limited memory, the limited resolution, and by the fixed-point representation of the system.

Another critical issue that affects all large-scale simulator platforms is the input/output (I/O) bottleneck. Even if these hardware systems can simulate neural activity in real time, or accelerated time (e.g., 1 ms of physical time simulated in 1 μs), the time required to load the configuration and the parameters of a large-scale neural network can require minutes to hours: for example, even using the latest state-of-the-art technology with transfer rates of 300 Gb/s (e.g., with 12x EDR InfiniBand links), the time required to configure a single simulation run of a network comprising 10^6 neurons with a fan-out of 1000, and a fan-in of 10 000 synapses with 8-b resolution weights would require at least 45 min.

B. General Purpose Computing Platforms

In addition to dedicated simulation engines for neuroscience studies, neuromorphic information processing

systems have also been proposed as general purpose non von Neumann computing engines for solving practical application problems, such as pattern recognition or classification. Example platforms based on FPGA and ASIC designs using the standard logic design flow have been described in Section II. Systems designed using less conventional design techniques or emerging nanoscale technologies include the IBM TrueNorth system (see Section III-B for the former), and the memristor- and RRAM-based neuromorphic architectures (see Section VI for the latter). The TrueNorth system however does not implement learning and adaptation. Therefore, it can only be used as a low-power neural computing engine once the values of the synaptic weights have been computed and uploaded to the network. The complex learning process that determines these synaptic weights is typically carried out on power-hungry standard- or super-computers. This rules out the possibility of using this and similar architectures in dynamic situations in which the system is required to adapt to the changes of the environment or of its input signals. Endowing these types of architectures with learning mechanisms, e.g., using memristive or RRAM-based devices, could lead to the development of non von Neumann computing platforms that are more adaptive and general purpose. The state of development of these adaptation and learning mechanisms and of the nanoscale memristive technologies, however, is still in its early stages, and the problems related to the control of learning dynamics, stability, and variability are still an active area of research [20], [157].

C. Small-Scale Special Purpose Neuromorphic Systems

Animal brains are not general purpose computing platforms. They are highly specialized structures that evolved to increase the chances of survival in hazardous environments with limited resources and varying conditions [158]. They represent an ideal computing technology for implementing robust feature extraction, pattern recognition, associative learning, sequence learning, planning, decision making, and ultimately for generating behavior [159]. The original neuromorphic engineering approach [1], [2] proposed to develop and use electronic systems precisely for this purpose: to build autonomous cognitive agents that produce behavior in response to multiple types of varying input signals and different internal states [107], [142]. As argued in Section I, the best way to reach this goal is to use electronic circuits biased in the subthreshold regime [4], [12], and to directly emulate the properties of real neural systems by exploiting the physics of the silicon medium. Examples of systems that follow this approach are the ones described in Sections III-A and V. The types of signals that these systems are optimally suited to process include multidimensional auditory and visual inputs, low-dimensional temperature and pressure signals, biosignals measured in living tissue, or even real-time streaming digital bit strings, e.g., obtained from internet, WiFi, or telecommunication data. Examples of application domains that could best exploit the properties of these neuromorphic systems include wearable personal assistants, coprocessors in embedded/mobile devices, intelligent brain–machine interfaces for prosthetic devices, and sensory-motor processing units in autonomous robotic platforms.

D. Memory and Information Processing

Classical von Neumann computing architectures face the von Neumann bottleneck problem [6]. We showed in Section II how current attempts to reduce this problem, e.g., by introducing cache memory close to the CPU or by using general purpose GPUs, are not viable, if energy consumption is factored in [36]. We then described dedicated FPGA and full custom ASIC architectures that carefully balance the use of memory and information processing resources for implementing deep networks [40], [42], [57] or large-scale computational neuroscience models [67]. While these dedicated architectures, still based on frames or graded (nonspiking) neural network models, represent an improvement over CPU and GPU approaches, the event-based architectures described in Sections II-B1 and II-C improve access to cache memory structures even further, because of their better use of locality in both space and time.

Also the SpiNNaker system, described in Section III-A, exploits event-based processing and communication to optimize the use of memory and computation resources: computation is carried out by the system's parallel ARM cores, while memory resources have been carefully distributed within each core (e.g., for caching data and instructions), across in-package DRAM memory chips (e.g., for storing program variables encoding network parameters) and in routing tables (e.g., for storing and implementing the network connectivity patterns). SpiNNaker and the event-based architectures of Section II-B1 and C however still separate, to a large extent, the computation from memory access, and implement them in physically different circuits and modules. Conversely, TrueNorth, NeuroGrid, BrainScales, and ROLLS neuromorphic processor architectures described in Sections III and V represent a radical departure from the classical von Neumann computer design style. In TrueNorth (Section III-B) for example, the synapses (i.e., memory elements) are physically adjacent to the neuron circuits (i.e., computation elements), and multiple neurosynaptic cores are distributed across the chip surface. Synapses in this architecture are used as basic binary memory elements and computation is mostly relegated to the neuron circuits (memory and computation elements are distributed and physically close to each other, but not truly colocalized). As discussed in Section III-C, the NeuroGrid architecture follows a substantially different approach. Rather than implementing binary memory circuits in each synapse, it uses circuits with global shared parameters that emulate the temporal dynamics of real synapses with biologically realistic time constants. Computation is therefore carried out in both

neurons and synapses, and the main memory requirements for programming and reconfiguring the network's topology and function are in the routing tables and in the shared global parameters. Since a large part of the network topology is hardwired, the range of possible neural models and functions that can be best emulated by NeuroGrid is restricted (by design) to models of cortical structures (e.g., parts of visual cortex). The BrainScales project (Section III-D), on the other hand, aims to support the simulation of large-scale networks of arbitrary topology, and which include nonlinear operations at the synapse level (e.g., such as spike-timing-dependent plasticity). Therefore, the memory structures that set the network parameters are truly distributed and colocated with the computational elements. These include floating gate devices that set neuron and synapse parameter values as well as static random access memory (SRAM) and digital-to-analog converter (DAC) circuits that store the synaptic weights. The BrainScales memory elements that are used to store the network connectivity patterns, on the other hand, are distributed across multiple devices, and interfaced to very fast routing circuits designed following the conventional digital communication approach [31]. A compromise between the highly flexible reconfigurable but energy consuming approach of BrainScales, and the ultralow power but with restricted degree of configurability approach of NeuroGrid is the one followed with the ROLLS neuromorphic processor (Section V). In this system all memory resources, both for routing digital events and for storing synaptic and neural circuit parameters, are tightly integrated with the synapse and neuron computing circuits. Since the memory of the events being processed is stored in the dynamics of the circuits, which have time constants that are well matched to the type of computation being carried out (see also the NeuroGrid real-time arguments in Section III-C), memory and computation are colocalized. The strategies used by the ROLLS neuromorphic processor for implementing multiple types of memory structures analogous to those used in conventional von Neumann architectures are the ones summarized in Table I. For example, we showed in Section V how to build associative memories, by training a network of plastic neurons to memorize different patterns in four different attractors. The long-term memory changes were made in the network's synaptic weights, via the chip's spike-timing-based learning mechanisms. Bistable or short-term memory structures can be made using the learned network attractors that represent state-holding elements which emulate working-memory structures in cortical circuits, and which can be employed for state-dependent computation, for example, implementing neural analogs of finite-state machines (FSMs) [142], [160]. Alternative training protocols and network connectivity patterns can be used in the same chip to carry out different types of neural information processing tasks, such as binary classification, e.g., for image recognition tasks [134]. Admittedly, the ROLLS neuromorphic processor comprises fewer neurons than those implemented in the large-scale neural systems surveyed in Section III, so the problem of allocating on-chip memory resources for routing events and configuring different connectivity patterns is mitigated. To build more complex neuromorphic systems that can interact with the world in real time and express cognitive abilities it will be necessary to consider more complex systems, which combine multiple neural information processing modules with diverse and specialized functionalities, very much like the cortex uses multiple areas with different sensory, motor, and cognitive functional specifications [161], [162], and which restrict the possible connectivity patterns to a subset that maximizes functionality and minimizes routing memory usage, very much like cortex uses patchy connectivity patterns with sparse long-range connections and dense short-range ones [163], [164]. Within this context, the optimal neuromorphic processor would be a multicore device in which each core could be implemented following different approaches, and in which the routing circuits and memory structures would be distributed (e.g., within and across cores), heterogeneous (e.g., using CAM, SRAM, and/or even memristive devices) and hierarchical (e.g., with intracore level-one routers, intercore level-two routers, interchip level-three routers, etc.) [165]. The systems surveyed in this paper represent sources of inspiration for choosing the design styles of the neural processing modules in each core and the memory structures to use, for different application areas.

VIII. CONCLUSION

In this work, we presented a survey of state-of-the-art neuromorphic systems and their usability for supporting deep network models, cortical network models, and brain-inspired cognitive architectures. We outlined the tradeoffs that these systems face in terms of memory requirements, processing speed, bandwidth, and their ability to implement the different types of computational primitives found in biological neural systems. We presented a mixed-signal analog/digital neuromorphic processor and discussed how that system, as well as analogous ones being developed by the international research community, can be used to implement cognitive computing. Finally, in Section VII, we highlighted the advantages and disadvantages of the different approaches being pursued, pointing out their strengths and weaknesses. In particular, we argued that while there is currently a range of very interesting and promising neuromorphic information processing systems available, they do not yet provide substantial advantages over conventional computing architectures for large-scale simulations, nor are they complex enough to implement specialized small-scale cognitive agents that can interact autonomously in any environment.

The tremendous progress in microelectronics and nanotechnologies in recent years has been paralleled by remarkable progress in both experimental and theoretical

neuroscience. To obtain significant breakthroughs in neuromorphic computing systems that can demonstrate the features of biological systems such as robustness, learning abilities, and possibly cognitive abilities, continuing research and development efforts are required in this interdisciplinary approach which involves neuroscientists, computer scientists, technologists, and material scientists. This can be achieved by training a new generation of researchers with interdisciplinary skills and by encouraging the communities specializing on these different disciplines to work together as closely as possible, as is currently being done in several computational neuroscience academic institutions, such as, for example, the Institute of Neuroinformatics, University of Zurich/ETH Zurich, or at neuroscience and neuromorphic engineering workshops, such as the Telluride and CapoCaccia Neuromorphic Engineering Workshops [166], [167]. ■

Acknowledgment

The authors would like to thank the reviewers for their constructive comments and helpful suggestions. Many of the concepts presented here were inspired by the discussions held at the CapoCaccia Cognitive Neuromorphic Engineering Workshop and by the work of R. Douglas, M. Mahowald, K. Martin, M. Cook, and S. Fusi. The authors would like to thank especially R. Douglas and K. Martin for their support and for providing an outstanding intellectual environment at the Institute of Neuroinformatics, Zurich, Switzerland. The ROLLS neuromorphic processor was designed and characterized together with N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, and D. Sumislawska. The attractor network and associative memory experimental results presented in Section V were obtained by F. Corradi at the 2015 CapoCaccia Workshop.

REFERENCES

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

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

[3] G. Indiveri and T. Horiuchi, "Frontiers in neuromorphic engineering," *Front. Neurosci.*, vol. 5, no. 118, 2011, DOI: 10.3389/fnins.2011.00118.

[4] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proc. IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep. 2014.

[5] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. New York, NY, USA: Wiley, 2014.

[6] J. Backus, "Can programming be liberated from the von Neumann style?: A functional style and its algebra of programs," *Commun. ACM*, vol. 21, no. 8, pp. 613–641, 1978.

[7] S. Habenschuss, Z. Jonke, and W. Maass, "Stochastic computations in cortical microcircuit models," *PLoS Comput. Biol.*, vol. 9, no. 11, 2013, Art. ID. e1003311.

[8] W. Maass, "Noise as a resource for computation and learning in networks of spiking neurons," *Proc. IEEE*, vol. 102, no. 5, pp. 860–880, May 2014.

[9] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA, USA: Addison-Wesley, 1989.

[10] E. Fragnière, A. van Schaik, and E. Vittoz, "Design of an analogue VLSI model of an active cochlea," *J. Analog Integr. Circuits Signal Process.*, vol. 13, no. 1/2, pp. 19–35, May 1997.

[11] 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.

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

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

[14] 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.

[15] S. H. Jo *et al.*, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.

[16] K. Kim *et al.*, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Lett.*, vol. 12, no. 1, pp. 389–395, 2011.

[17] S. Yu *et al.*, "A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation," *Adv. Mater.*, vol. 25, no. 12, pp. 1774–1779, 2013.

[18] M. Suri *et al.*, "Bio-inspired stochastic computing using binary CBRAM synapses," *IEEE Trans. Electron Devices*, vol. 60, no. 7, pp. 2402–2409, Jul. 2013.

[19] S. Kim *et al.*, "Experimental demonstration of a second-order memristor and its ability to biorealistically implement synaptic plasticity," *Nano Lett.*, vol. 15, no. 3, pp. 2203–2211, 2015.

[20] S. Saighi *et al.*, "Plasticity in memristive devices," *Front. Neurosci.*, vol. 9, no. 51, 2015, DOI: 10.3389/fnins.2015.00051.

[21] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[22] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[23] T. Pfeil *et al.*, "Six networks on a universal neuromorphic computing substrate," *Front. Neurosci.*, vol. 7, 2013, DOI: 10.3389/fnins.2013.00011.

[24] B. Rajendran *et al.*, "Specifications of nanoscale devices and circuits for neuromorphic computational systems," *IEEE Trans. Electron Devices*, vol. 60, no. 1, pp. 246–253, Jan. 2013.

[25] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

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

[27] 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, no. 178, 2013, DOI: 10.3389/fnins.2013.00178.

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

[29] F. Corradi and G. Indiveri, "A neuromorphic event-based neural recording system for smart brain-machine-interfaces," *IEEE Trans. Biomed. Circuits Syst.*, 2015, to be published.

[30] C. Bartolozzi *et al.*, "Embedded neuromorphic vision for humanoid robots," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 129–135.

[31] J. Schemmel *et al.*, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 1947–1950.

[32] R. Cavin, J. Hutchby, V. Zhirnov, J. Brewer, and G. Bourianoff, "Emerging research architectures," *Computer*, vol. 41, no. 5, pp. 33–37, May 2008.

[33] M. Kamble and K. Ghose, "Energy-efficiency of VLSI caches: A comparative study," in *Proc. 10th Int. Conf. VLSI Design*, Jan. 1997, pp. 261–267.

[34] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. AFIPS Spring Joint Comput. Conf.*, 1967, pp. 483–485.

[35] M. Hill and M. Marty, "Amdahl's law in the multicore era," *IEEE Computer*, vol. 41, no. 7, pp. 33–38, Jul. 2008.

[36] A. Cassidy and A. Andreou, "Beyond Amdahl's law: An objective function that links multiprocessor performance gains to delay and energy," *IEEE Trans. Comput.*, vol. 61, no. 8, pp. 1110–1126, Aug. 2012.

[37] R. Brette and D. Goodman, "Simulating spiking neural networks on GPU," *Network, Comput. Neural Syst.*, vol. 23, no. 4, pp. 167–182, 2012.

[38] D. Cirean, U. Meier, L. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.

[39] J. Nageswaran, N. Dutt, J. Krichmar, A. Nicolau, and A. Veidenbaum, "A configurable simulation environment

for the efficient simulation of large-scale spiking neural networks on graphics processors," *Neural Netw.*, vol. 22, no. 5/6, pp. 791–800, 2009.

[40] F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, 2015, pp. 683–688.

[41] L. Camunas-Mesa *et al.*, "An event-driven multi-kernel convolution processor module for event-driven vision sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 2, pp. 504–517, Feb. 2012.

[42] P.-H. Pham *et al.*, "Neuflow: Dataflow vision processing system-on-a-chip," in *Proc. IEEE 55th Int. Midwest Symp. Circuits Syst.*, Aug. 2012, pp. 1044–1047.

[43] D. Neil and S.-C. Liu, "Minitaur, an event-driven FPGAbased spiking network accelerator," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2621–2628, Dec. 2014.

[44] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 g-ops/s mobile coprocessor for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 696–701.

[45] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[46] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[47] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.

[48] K. Fukushima, "Artificial vision by multi-layered neural networks: Neocognitron and its advances," *Neural Netw.*, vol. 37, pp. 103–119, 2013.

[49] D. Marr, *Vision, A Computational Investigation Into the Human Representation and Processing of Visual Information*. San Francisco, CA, USA: Freeman, 1982.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[51] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, 1999.

[52] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 dB 3μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[53] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

[54] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 × 128 1.5% contrast sensitivity 0.9% FPN 3 μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.

[55] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[56] C. Farabet *et al.*, "Neuflow: A runtime reconfigurable dataflow processor for vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 109–116.

[57] A. Dundar, J. Jin, V. Gokhale, B. Martini, and E. Culurciello, "Memory access optimized scheduling scheme for DCNNs on a mobile processor," in *Proc. IEEE High Performance Extreme Comput. Conf.*, 2014, DOI: 10.1109/HPEC.2014.7040963.

[58] T. Serrano-Gotarredona, A. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision processing systems," *IEEE Trans. Circuits Syst. I, Fund. Theory Appl.*, vol. 46, no. 9, pp. 1064–1071, Sep. 1999.

[59] L. Camunas-Mesa, A. Acosta-Jimenez, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 32× 32 pixel convolution processor chip for address event vision sensors with 155 ns event latency and 20 Meps throughput," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 777–790, Apr. 2011.

[60] C. Zamarreño-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 82–102, Feb. 2013.

[61] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[62] Y. LeCun, "The MNIST database of handwritten digits," May 2012. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[63] A. Majumdar, S. Cadambi, M. Becchi, S. Chakradhar, and H. Graf, "A massively parallel, energy efficient programmable accelerator for learning and classification," *ACM Trans. Architect. Code Optim.*, vol. 9, no. 1, pp. 6:1–6:30, Mar. 2012.

[64] 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, 2013.

[65] R. Wang *et al.*, "An FPGA implementation of a polychronous spiking neural network with delay adaptation," *Front. Neurosci.*, vol. 7, 2013, DOI: 10.3389/fnins.2013.00014.

[66] L. Maguire *et al.*, "Challenges for large-scale implementations of spiking neural networks on FPGAs," *Neurocomputing*, vol. 71, no. 1, pp. 13–29, 2007.

[67] N. Farahini, A. Hemani, A. Lansner, F. Clermidy, and C. Svensson, "A scalable custom simulation machine for the Bayesian confidence propagation neural network model of the brain," in *Proc. 19th Asia South Pacific Design Autom. Conf.*, Jan. 2014, pp. 578–585.

[68] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Boston, MA, USA: Kluwer, 1994.

[69] S. Deiss, R. Douglas, and A. Whatley, "A pulse-coded communications infrastructure for neuromorphic systems," in *Pulsed Neural Networks*, W. Maass and C. Bishop, Eds. Cambridge, MA, USA: MIT Press, 1998, pp. 157–178.

[70] 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.

[71] S. Choudhary *et al.*, "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.

[72] K. Boahen and A. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 4, J. Moody, S. Hanson, and R. Lippman, Eds. Cambridge, MA, USA: MIT Press, 1992, pp. 764–772.

[73] T. Binzegger, R. Douglas, and K. Martin, "A quantitative map of the circuit of cat primary visual cortex," *J. Neurosci.*, vol. 24, no. 39, pp. 8441–8453, 2004.

[74] J. Dethier *et al.*, "A brain-machine interface operating with a real-time spiking neural network control algorithm," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 24, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Cambridge, MA, USA: MIT Press, 2011, pp. 2213–2221.

[75] S. Menon, S. Fok, A. Neckar, O. Khatib, and K. Boahen, "Controlling articulated robots in task-space with spiking silicon neurons," in *Proc. 5th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.*, Aug. 2014, pp. 181–186.

[76] *Brain-inspired Multiscale Computation in Neuromorphic Hybrid Systems (BrainScaleS)*, FP7 269921 EU Grant, 2011–2015.

[77] 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.

[78] D. Feldman, "Synaptic mechanisms for plasticity in neocortex," *Annu. Rev. Neurosci.*, vol. 32, pp. 33–55, 2009.

[79] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, 1982.

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

[81] 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.

[82] E. Rolls, "Attractor networks," *Wiley Interdisciplinary Rev., Cogn. Sci.*, vol. 1, no. 1, pp. 119–134, 2010.

[83] R. Lamprecht and J. LeDoux, "Structural plasticity and memory," *Nature Rev. Neurosci.*, vol. 5, no. 1, pp. 45–54, 2004.

[84] E. Maguire *et al.*, "Navigation-related structural change in the hippocampi of taxi drivers," *Proc. Nat. Acad. Sci. USA*, vol. 97, no. 8, pp. 4389–4403, Apr. 2000.

[85] G. Turrigiano and S. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Rev. Neurosci.*, vol. 5, pp. 97–107, Feb. 2004.

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

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

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

[89] D. Hebb, *The Organization of Behavior*. New York, NY, USA: Wiley, 1949.

[90] H. Markram, W. Gerstner, and P. Sjöström, "Spike-timing-dependent plasticity: A comprehensive overview," *Front. Synaptic Neurosci.*, vol. 4, no. 2, 2012, DOI: 10.3389/fnsyn.2012.00002.

[91] M. R. Azghadi, N. Iannella, S. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, challenges," *Proc. IEEE*, vol. 102, no. 5, pp. 717–737, May 2014, 0018-9219.

[92] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Front. Neurosci.*, vol. 7, no. 2, 2013, DOI: 10.3389/fnins.2013.00002.

[93] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, pp. 213–215, 1997.

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

[95] J. Gjorgjieva, C. Clopath, J. Audet, and J.-P. Pfister, "A triplet spike-timing-dependent plasticity model generalizes the Bienenstock-Cooper-Munro rule to higher-order spatiotemporal correlations," *Proc. Nat. Acad. Sci.*, vol. 108, no. 48, pp. 19 383–19 388, 2011.

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

[97] M. Graupner and N. Brunel, "Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, dendritic location," *Proc. Nat. Acad. Sci.*, vol. 109, pp. 3991–3996, 2012.

[98] P. Sjöström, G. Turrigiano, and S. Nelson, "Rate, timing, cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, Dec. 2001.

[99] B. Nessler, M. Pfeiffer, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLoS Comput. Biol.*, vol. 9, no. 4, 2013, Art. ID. e1003037.

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

[101] G. Indiveri, E. Chicca, and R. Douglas, "Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition," *Cogn. Comput.*, vol. 1, pp. 119–127, 2009.

[102] A. Yuille and G. Geiger, *Winner-Take-All Networks*. Cambridge, MA, USA: MIT Press, 2003, pp. 1228–1231.

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

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

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

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

[107] Y. Sandamirskaya, "Dynamic neural fields as a step toward cognitive neuromorphic architectures," *Front. Neurosci.*, vol. 7, 2013, DOI: 10.3389/fnins.2013.00276.

[108] G. Schöner, "Dynamical systems approaches to cognition," in *Cambridge Handbook of Computational Cognitive Modeling*, R. Sun, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2007, pp. 101–126.

[109] G. Indiveri, T. Horiuchi, E. Niebur, and R. Douglas, "A competitive network of spiking VLSI neurons," in *World Congress on Neuroinformatics*, F. Rattay, Ed. Vienna, Austria: ARGESIM/ASIM–Verlag, 2001, pp. 443–455.

[110] E. Chicca, "A Neuromorphic VLSI System for modeling spike-based cooperative competitive neural networks," Ph.D. dissertation, Dept. Phys., ETH Zürich, Zürich, Switzerland, Apr. 2006.

[111] M. Oster, R. Douglas, and S.-C. Liu, "Computation with spikes in a winner-take-all network," *Neural Comput.*, vol. 21, pp. 2437–2465, 2009.

[112] E. Chicca *et al.*, "A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long term memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1297–1307, Sep. 2003.

[113] M. Giulioni *et al.*, "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.

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

[115] 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.

[116] J. Bill *et al.*, "Compensating inhomogeneities of neuromorphic VLSI devices via short-term synaptic plasticity," *Front. Comput. Neurosci.*, vol. 4, 2010, DOI: 10.3389/fncom.2010.00129.

[117] 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.

[118] T. Ohno *et al.*, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Mater.*, vol. 10, no. 8, pp. 591–595, 2011.

[119] 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.

[120] G. Indiveri, "Neuromorphic bistable VLSI synapses with spike-timing-dependent plasticity," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15. Cambridge, MA, USA: MIT Press, Dec. 2003, pp. 1091–1098.

[121] 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.

[122] H. Riis and P. Hafliger, "Spike based learning with weak multi-level static memory," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. 393–396.

[123] 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.

[124] J. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA, USA: MIT Press, 2008, pp. 1401–1408.

[125] 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.

[126] 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.

[127] S. Bamford, A. Murray, and D. Willshaw, "Spike-timing-dependent plasticity with weight dependence evoked from physical constraints," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 4, pp. 385–398, Aug. 2012.

[128] S. Ramakrishnan, R. Wunderlich, and P. Hasler, "Neuron array with plastic synapses and programmable dendrites," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2012, pp. 400–403.

[129] 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.

[130] G. Rovere, Q. Ning, C. Bartolozzi, and G. Indiveri, "Ultra low leakage synaptic scaling circuits for implementing homeostatic plasticity in neuromorphic architectures," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014, pp. 2073–2076.

[131] S. Hussain, S.-C. Liu, and A. Basu, "Improved margin multiclass classification using dendritic neurons with morphological learning," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 2640–2643.

[132] D. Fasnacht and G. Indiveri, "A PCI based high-fanout AER mapper with 2 GiB RAM look-up table, 0.8μs latency and 66 MHz output event-rate," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 2011, DOI: 10.1109/CISS.2011.5766102.

[133] E. Chicca *et al.*, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 981–993, May 2007.

[134] N. Qiao *et al.*, "A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Front. Neurosci.*, vol. 9, no. 141, 2015, DOI: 10.3389/fnins.2015.00141.

[135] 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.

[136] 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.

[137] C. Rossant, D. Goodman, J. Platkiewicz, and R. Brette, "Automatic fitting of spiking neuron models to electrophysiological recordings," *Front. Neuroinf.*, 2010, DOI: 10.3389/neuro.11.002.2010.

[138] 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.

[139] G. Turrigiano, "The self-tuning neuron: Synaptic scaling of excitatory synapses," *Cell*, vol. 135, no. 3, pp. 422–435, 2008.

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

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

[142] E. Neftci *et al.*, "Synthesizing cognition in neuromorphic electronic systems," *Proc. Nat. Acad. Sci.*, vol. 110, no. 37, pp. E3468–E3476, 2013.

[143] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, 2013, Art. ID. 384010.

[144] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. Wong, "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2729–2737, Aug. 2011.

[145] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Mater.*, vol. 9, no. 5, pp. 403–406, 2010.

[146] K. C. Chun *et al.*, "A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMs for high-density cache memory," *IEEE J. Solid-State Circuits*, vol. 48, no. 2, pp. 598–610, Feb. 2013.

[147] A. Vincent *et al.*, "Spin-transfer torque magnetic memory as a stochastic memristive synapse," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2014, pp. 1074–1077.

[148] N. Locatelli, V. Cros, and J. Grollier, "Spin-torque building blocks," *Nature Mater.*, vol. 13, no. 1, pp. 11–20, 2014.

[149] A. Chanthbouala *et al.*, "A ferroelectric memristor," *Nature Mater.*, vol. 11, no. 10, pp. 860–864, 2012.

[150] M. Suri *et al.*, "Physical aspects of low power synapses based on phase change memory devices," *J. Appl. Phys.*, vol. 112, no. 5, 2012, Art. ID. 054904.

[151] D. Kuzum, R.-G.-D. Jeyasingh, B. Lee, and H.-S. P. Wong, "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano Lett.*, vol. 12, no. 5, pp. 2179–2186, 2012.

[152] M. Wimmer and M. Salinga, "The gradual nature of threshold switching," *New J. Phys.*, vol. 16, no. 11, 2014, Art. ID. 113044.

[153] R. Brette *et al.*, "Simulation of networks of spiking neurons: A review of tools and strategies," *J. Comput. Neurosci.*, vol. 23, no. 3, pp. 349–398, Dec. 2007.

[154] M. Richert, J. Nageswaran, N. Dutt, and J. Krichmar, "An efficient simulation environment for modeling large-scale cortical processing," *Front. Neuroinf.*, vol. 5, 2011, DOI: 10.3389/fninf.2011.00019.

[155] H. Markram, "The blue brain project," *Nature Rev. Neurosci.*, vol. 7, pp. 153–160, 2006.

[156] Y. Frégnac and G. Laurent, "Neuroscience: Where is the brain in the Human Brain Project?" *Nature*, vol. 513, 2014, DOI: 10.1038/513027a.

[157] M. Prezioso *et al.*, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[158] J. M. Allman, *Evolving Brains*. New York, NY, USA: Scientific American Library, 2000.

[159] P. Churchland and T. Sejnowski, *The Computational Brain*. Cambridge, MA, USA: MIT Press, 1992.

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

[161] D. Van Essen, C. Anderson, and D. Felleman, "Information processing in the primate visual system—An integrated systems perspective," *Science*, vol. 255, pp. 419–423, 1992.

[162] R. Douglas and K. Martin, "Behavioral architecture of the cortical sheet," *Current Biol.*, vol. 22, no. 24, pp. R1033–R1038, 2012.

[163] D. Muir *et al.*, "Embedding of cortical representations by the superficial patch system," *Cerebral Cortex*, vol. 21, no. 10, pp. 2244–2260, 2011.

[164] N. Markov and H. Kennedy, "The importance of being hierarchical," *Current Opinion Neurobiol.*, vol. 23, no. 2, pp. 187–194, 2013.

[165] S. Moradi, G. Indiveri, N. Qiao, and F. Stefanini, "Networks and hierarchical routing fabrics with heterogeneous memory structures for scalable event-driven computing systems," Eur. Patent Appl. EP 15/165272, Apr. 2015.

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

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

ABOUT THE AUTHORS

Giacomo Indiveri (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from the University of Genoa, 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 *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 ETH Zurich, Zurich, Switzerland, in 2006.

He is an Associate Professor at the Faculty of Science, University of Zurich, Zurich. He 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 Postdoctoral Researcher at the Institute of Neuroinformatics, University of Zurich/ETH Zurich. 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.

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.

Shih-Chii Liu (Senior Member, IEEE) studied electrical engineering as an undergraduate at the Massachusetts Institute of Technology, Cambridge, MA, USA and received the Ph.D. degree from the computation and neural systems program at the California Institute of Technology, Pasadena, CA, USA, in 1997.

She worked at various companies, including Gould American Microsystems, LSI Logic, and Rockwell International Research Labs. She is currently a group leader at the Institute of Neuroinformatics, University of Zurich/ETH Zurich, Zurich, Switzerland. Her research interests include neuromorphic visual and auditory sensors, cortical processing circuits, and event-driven circuits, networks, and algorithms.

Dr. Liu is past Chair of the IEEE Circuits and Systems Society Sensory Systems and Neural Systems and Applications Technical Committees. She is current Chair of the IEEE Swiss Circuits and Systems/Electron Devices Society and an Associate Editor of the IEEE TRANSACTIONS OF BIOMEDICAL CIRCUITS AND SYSTEMS and the *Neural Networks* journal.