Invited Paper

Neuromorphic microelectronics from devices to hardware systems and applications

*Alexandre Schmid*\(^1\)\(^a\)

\(^1\) Microelectronic Systems Laboratory, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland

\(^a\) alexandre.schmid@epfl.ch

Received April 9, 2016; Revised May 20, 2016; Published October 1, 2016

**Abstract:** Neuromorphic systems aiming at mimicking some characteristics of the nervous systems of living humans or animals have been developed since the late 1980s’, taking benefit of intrinsic properties and increasing performances of the successive silicon fabrication technologies. A regain of interest has been observed in the middle of the 2010s’, which manifests itself from the emergence of large-scale projects integrating various computational and hardware perspectives, by the increased interest and involvement of industry and the growth of the volume of scientific publications. This paper reviews research directions and methods of neuromorphic microelectronics hardware, the developed hardware and its performance, and discusses current issues and potential future developments.

**Key Words:** neuromorphic engineering, neuromorphic computing, neuromorphic devices, neuromorphic circuits, neuromorphic architectures

**1. Introduction**

Nature has ever been a source of inspiration of humanity in domains as various as arts, medicine, science. Understanding the concepts of natural evolution has been a prerequisite and the motivation of engineers to replicating solutions that had emerged as the most suitable after millions of years of natural selection. In their turn, engineers in electronics and microelectronics fields have turned their attention to nature and biology as a potential source of inspiration to solving problems that turned out intricate to tackling using established concepts and methods.

The central nervous system of mammals, its operation and the operation of its fundamental constructing units, its construction have been a major inspiration to microelectronics mimicking the living. The reason lies in the general target of microelectronic systems to be the media supporting data processing, storing, and complex system control, which the various subsystems of the central nervous system appear to be doing in a manner that is more efficient than electronic computing systems, in many respects.

Even though, and to very rare exceptions, computing based on sequential arithmetic operations is carried out in a significantly faster speed by classical arithmetic and logic units (ALUs) than by hu-
mans, some higher-level tasks which these complex computing units are devoted to solve are executed by humans in a faster and more reliable way. For example, face recognition that requires multiple and adaptive feature extraction to be compared to short and in some cases long-term memory patterns is still performed by humans in a more reliable way than computing systems; similarly, smell recognition that requires long-term memory, as well as many relatively simple control tasks such as grasping, high-definition image creation using eyes as relatively low-density image sensors, high-dynamic range image vision, reliable sound source separation are some example where the superiority of mammals is still established, though challenged by artificial systems in the recent times. Some complex control tasks based on the fusion of data originating from multiple sources such as various sensors, memory, are carried out in a better way by animals, for example enabling them the complex control of their body into equilibrium, reflex as well as a multitude of involuntary control of internal muscles and bio-chemical processes. Finally, complex tasks such as reasoned decision making, deductive processes, emotions appear be the prerogative of mammals or humans.

Observing that the goal of efficient computing is shared by the central nervous system of mammals as well as artificial computing systems, computer science and electronic engineers among others have considered understanding and replicating the operation of the central nervous systems as potentially offering solutions to otherwise intractable problems. Specifically, electronic engineers have considered mimicking some operation characteristics of parts of the central nervous system into silicon-based components, circuits and systems, considering various levels of abstraction over the years.

First described by Carver Mead in the 1990s' as microelectronic information processing systems mimicking the operation of their biology counterpart [1, 2], these systems have developed into a variety of implementations, taking benefit of merging the latest fabrication progresses along with the efficiency that millions of years of evolution offer biological systems into a novel paradigm. The way biological entities harvest, manage and save energy is a fundamental source of inspiration for electronic systems. Moreover, the robustness of biological systems to the failure of one individual component which is a member of a vast amount of identical components, collaboratively processing a global function is known.

After years of research that has seen many algorithms implemented into VLSI, some architectures commercialized, and also a decline of interest in the 2000s'-2010s', [3], neuromorphic hardware appears to experience a “revival” following the successful development of new algorithms, a certain interest of large companies for the software applications and also their understanding of the possible benefit from dedicated supporting hardware. An exhaustive detail description of all developments is not the goal of this paper. Rather, a review of selected cornerstone developments, as well as recent developments is considered. A broad perspective is preferred, presenting several abstraction levels. References are given to review manuscripts that focus on specific aspects, architecture or circuits with more detail, as well as research papers that detail one specific realization. Section 2 discusses general principles underlying the microelectronic development of neuromorphic systems, the issues that are faced and expectedly solved. Device-level realizations are presented in Section 3, and circuit-level realizations are presented in Section 4, and architecture-level realization are presented in Section 5, following a bottom-up approach. Section 6 discusses the hardware implementation of learning or training algorithms in silicon. Recently emerging large-scale architectures are presented in their specific context in Section 7. Specific applications that have not been discussed in the previous Sections are presented in Section 8. An emphasis over new concepts spanning over diverse levels of abstraction of neuromorphic hardware realizations is considered in Section 9.

2. General principles and related issues

Many industrial, medical and consumer applications are available nowadays taking benefit of the significant advances in microelectronic developments over the past six decades. The MOSFET has been used as the atomic component of a wide majority of developed microelectronic systems, and is found in multi-billion units in modern microprocessors. A growing range of analog and digital applications have benefited from its stable electrical characteristics, reliability, systematic development
and fabrication processes. Its fabrication technology has been improved and developed to fit Moore's law and satisfy industrial and application needs, which has been referred to as technology scaling.

Technology scaling has been evidenced to face some severe limits, and its phasing out has been predicted. The fixed maximal clock frequency of microprocessors, that would not anymore follow a progressing trend from the 2005s' has been an early sign. More recently and with the advent of multicore processors and systems-on-chip, a power barrier has emerged as a major issue in the further development of processors. In spite of the continued development of classical processor architectures and their expected further existence as the workhorses of computer systems, the necessity of figuring out novel processing paradigms has become an evidence to overcome the limits of current conventional processors, built using a classical CMOS technology.

Neuromorphic systems have been considered form their beginnings as potential alternatives to classical computing architectures. Software developments have been numerous, using a vast range of programming languages, however also clearly evidencing limits in terms of real-time processing specifically, [4]. Nevertheless, the inherent parallelism of neural network algorithms naturally lends to a custom hardware implementation that reflects ANNs processing and topology into an appropriate silicon implementation. The stringent need for hardware accelerators has only manifested itself recently, with the need to handle big data, and also the severe constraints related to real-time processing of complex processing in portable systems. Owing to an established research domain, several reviews of the state-of-the-art of hardware implementations of neuromorphic systems have been presented. [5] published in 2006 discusses different possible implementation models and techniques. An exhaustive review of hardware implementations presented as of 2010 is presented in [6], which also discusses the major implementation techniques, signal encoding possibilities and possible evaluation criteria. Recent developments in deep neural network implementations are reviewed in [7], as of 2015 for example. Commercial hardware has been made available, and a review is presented in [3], though as of 2004.

Neuromorphic algorithms relate to their capacity of mimicking some characteristic of the central nervous system (CNS) or its components, in a very general way referred to as neurons. Neuromorphic systems consist of the implementation of these algorithms into some hardware system. A model that captures some specific characteristic of the CNS or neurons operation at some determined level of abstraction is considered and implemented into hardware, or software. The immense complexity of the CNS and also the operation of single neurons dictates a variety of models, which also reflect into a large spread of different hardware. For example, modeling could consist of:

- synapses and the chemical interaction of neuro-transmitters;
- the spiking behavior of one neuron, which may be modeled from a variety of models, e.g., Hodgkin-Huxley, FitzHugh-Nagumo, etc.
- the learning process that involves one or several neurons, e.g., reinforcement learning, Hebbian rule, Spike-timing-dependent plasticity (STDP), etc.
- from a system perspective, Artificial Neural Networks (ANNs) represent the operation of a group of neuron from an arithmetic model; their learning rule accordingly follows an algorithm, e.g., multi-layer perceptron, error backpropagation learning;
- the specific operation of groups of specialized neurons, e.g., retina, associative memory, etc.
- from an elevated level of abstraction, the operation and training of large-scale neural networks is modeled from algorithmic rules and aims at concrete problem solving tasks, e.g., deep learning.

Hardware implementations of neuromorphic systems are attempted at these various levels of representation using different techniques that also relate to the way signals are encoded, for example:

- analog designs exploit the electrical properties of devices or the synthesized behavior of groups of devices to mimic some natural phenomena. Signals are generally encoded in a continuous domain, e.g., voltage or current domains. Designs using MOSFETs operated in subthreshold attempt to significantly reduce power dissipation;
digital designs lend themselves to the implementation of ANN algorithms, though some digital hardware supporting spiking neurons exist. Issues related to the format and bit-precision of number representation arise;

mixed mode designs generally use some type of pulse representation of signals, where analog and/or digital electronics is used that generates information that is coded with respect to pulse width, pulse density, inter-spike (pulse) temporal information, time-to-first spike, for example;

architectural digital developments aim at emulating large-scale artificial neural networks; in addition to the constraints posed to digital designs, issues related to mapping algorithms to multicore systems, memory and access bus managements are of crucial importance.

A bottom-up approach is taken in the remainder of the paper, where selected approaches are presented, showing the evolution of the hardware developments towards the modern systems.

3. Device-based realizations

Exploiting intrinsic electrical characteristics of silicon devices is considered in order to construct neuromorphic systems. In most existing cases, the considered devices must be supported by some passive elements or service circuits, for example carrying out biasing or filtering functionality. Nevertheless, the neuromorphic operation originates from the specific behavior of a certain type of device, that is appropriately tailored in its sizing and fabrication materials and recipes. Several devices have been studied, and some potential applications proposed.

Single-electron transistors (SETs) are constructed from tunneling junctions which obey the Coulomb blockade phenomenon under the condition of sufficiently low temperature, and sufficiently small size of

---

**Fig. 1.** Device-level neuromorphic realizations. (a) Single-electron neuron made from multiple single-electron oscillators. (b) Schematic of a NPN SEBAT device, also including a passive quenching circuit and output amplifier, and conceptual view of the transient SEBAT output voltage at $V_{CB}$. (c) Schematic of a neuron MOSFET, and (d) two-stage charge-based neuron.

Single-electron transistors (SETs) are constructed from tunneling junctions which obey the Coulomb blockade phenomenon under the condition of sufficiently low temperature, and sufficiently small size of
the junctions. Single-electron circuits can be used to mimic an action potential and its propagation in an array of nanodots, [8]. A SET oscillator circuit is formed from a tunneling junction \( C_1 \), a resistance \( R \) and a bias voltage \( V_{dd} \). A nanodot node \( n_i \) is the location from or where electrons tunnel, and thus are stored as an excess electron. If \( n_i = V_{dd} \) then the circuit is in a stable state. Increasing \( n_i \) by the means of an external bias voltage will cause one electron to tunnel from ground into the nanodot, which creates an abrupt decrease of the nanodot voltage. Then, the nanodot is loaded through the resistance until reaching \( V_{dd} \), in a slow manner, with respect to the tunneling event. Connecting several such circuits using capacitors and opposite bias polarities creates a chain that has the capacity of transferring the spiking event further. Using bypasses and feed-forward processing enables the creation of single-electron synapses and neuron circuits as presented in Fig. 1 (a), [8]. Applications of the neuromorphic single-electron fundamental principles have been shown in the context of stochastic resonance from an ensemble of single-electron devices [9], and neural network aiming at synchrony detection [10], for example.

The SEBAT device consists of an integrated bipolar transistor operated in a classical common-base amplifier configuration, as depicted in Fig. 1 (b), where \( E \) denotes the emitter; \( B \), the base; \( C \), the collector; \( C_{CC} \), the collector-base junction capacitance; \( R_Q \), the quenching resistor; \( V_{CC} \), the SEBAT supply voltage; \( I_E \), the emitter (input) current; \( I_A \), the avalanche current; \( I_R \), the recharging current; \( V_{CB} \), the collector-base voltage, and \( V_{BD} \), the base-collector breakdown voltage. The biasing of the circuit is however specific of the device, and dictates its electrical characteristics, the consequent operation modes, as well as fabrication constraints to establish proper operation, [11].

The static collector-base biasing of the SEBAT is maintained at a voltage \( V_{CC} \) which exceeds the collector-base breakdown voltage, \( V_{BD} \). This mode of operating a bipolar transistor is named Geiger mode, and is not used in classical analog circuit design. At rest and neglecting leakage current through the reversed-biased collector-base junction, no current flows through the device, i.e., \( I_R = I_A = 0 \), and \( V_{CB} = V_{CC} \). Upon applying a negative voltage on the emitter, the base-emitter junction is forward-biased which creates a current \( I_E \). Part of the electrons overcome the emitter-base potential barrier and reach the base-collector junction; in the depletion region, these minority carriers are in a strong electric field which accelerates them. Subsequent impact ionization events cause the creation of new electron-holes which eventually triggers an avalanche current. This exponential current is potentially destructive for the device. \( R_Q \) is sized to guarantee proper quenching of the avalanche current which can not sustain itself when \( V_{CB} \) reaches to approximately \( V_{BD} \). In regime, time constant \( \tau = R_Q C_{CB} \).

In the presented configuration, the SEBAT naturally operates as a single-device spiking neuron emulation hardware, where the input voltage dictates the probability of an output spike, as conceptually depicted in Fig. 1 (b). The injection of minority carriers through the base-emitter junction, as well as the avalanche generation by impact ionization in the base-collector depletion region are probabilistic events. Eventually, the operation of the SEBAT as a single-electron detector obeys a Poisson distribution. The input quantities and the output signal are linked through a stochastic process, and the output signal is intrinsically pulsed, each detected electron causing an avalanche to fire. This behavior mimics some of the properties of physiological neurons. In particular, the SEBAT can be described with the well established noisy neuron with exponential escape model, [12]. The SEBAT has been fabricated using commercial CMOS processes in 0.35 \( \mu \)m and 0.18 \( \mu \)m technologies. Early measurements of the SEBAT input-output characteristics in the frequency domain confirm its potential as a neuromorphic device, and specifically as a spiking neuron coincidence detector. Moreover, the intrinsic stochastic nature of the spiking occurrence enables a direct implementation of stochastic resonance-based applications.

The so-called neuron MOSFET (\( \nu \)MOS) is a unique device that has the capacity of calculating the weighted sum of all inputs, [13]. Floating-gate MOSFETs are used and the operation is performed in the voltage domain based on the capacitive coupling, Fig. 1 (c). The transistor controlled this way turns on under the condition that \( \Phi_F = (C_1 V_1 + C_2 V_2 + \ldots + C_n V_n)/C_{TOT} > V_{TH} \), where \( \Phi_F \) is the floating-gate potential that implements the weighted sum, \( V_i \) through \( V_n \) are the input voltages, \( C_1 \) through \( C_n \) are the capacitive couplings between each respective input and the floating gate, and \( C_{TOT} \) is the sum of \( C_1 \) through \( C_n \). Applications of the devices have been demonstrated as a 2-bit digital-to-
analog converter, multi-valued logic circuits, and associative memory, among others, [14]. The $\nu$MOS technology is still developed using recent fabrication technologies, e.g., thin-film transistors, [15].

Charge-based techniques have been used to realize the weighted-sum with thresholding operation, for example in [16], where a two-stage neuron is presented. The circuit is operated from a two-phase non-overlapping clock scheme, Fig. 1(d), where a first phase consists of the precharge of the capacitive nodes, which is followed by the evaluation phase that computes the result. The condition of charge conservation on the row node dictates that any change of the voltage applied to the converging capacitances external node during the evaluation phase results into a perturbation of the row voltage equal to $\Delta V_{\text{row}} = (V_{\text{IN}} C_W - V_{\text{DD}} C_T + V_{\text{DD}} C_P)/(C_{\text{TOT}} W + C_{\text{TOT}} T + C_P + C_S)$, where $V_{\text{IN}}$ is the analog input voltage applied to the $C_W$ capacitance, $C_T$ implements a capactively programmable threshold value, $C_P$ implements a capacitive perturbation value used for perturbation learning, $C_S$ is a stray capacitance to ground, $C_{\text{TOT}} W$ is the upper limit of the capacitive weight range and $C_{\text{TOT}} T$ is the upper limit of the capacitive threshold range. The charge-based node device is implemented as an overlap of two polysilicon layers, and has been measured in various two-polysilicon technologies. Following the charge-based principles, realizations including a flash ADC, [17], and a Hamming artificial neural network have been presented, [18].

Generally, circuits combining several devices need to be implemented in order to synthesize functionality with a higher level of complexity or richness of behavior. Also, interfacing devices with their surrounding environment or the outside world is carried out using CMOS circuits.

4. Circuit-level realizations

This Section presents realizations at the level of relatively “small” circuits that achieve functionalities that could be used as part of a larger computational system. CMOS has been a preferred implementation media, enabling designers to benefit from industry-proven fabrication methodologies, MOSFET transistor models and experience in the development of neuromorphic basic circuits and to the level of small-scale applications. The various possible types of signaling that the usage of MOSFETs offers have been exploited, including analog voltage and current, frequency, semi-digital pulse-based and pulse modulation based to full digital-domain representations. Analog realizations generally dominate small-scale realizations, which take benefit of the richness of the transistor dynamics to realize complex operations using only few transistors, where a digital realization would require tens to hundreds of transistors to achieve an identical result. The variety of circuits is very vast and some relevant building blocks are presented in the following, as an overview of the field.

Winner-Take-All (WTA) circuits accommodate many input signals and discriminate one as the winner upon a selection criterion. The selection generally originates from the amplitude of currents or voltages, and the winner is presented as the input with higher value. The Loser-Take-All (LTA) circuits are a variation that discriminates the loser, e.g., the input with smaller amplitude. A further development of these circuits consists of the k-WTA or k-LTA circuits which extract the k values of higher amplitudes, smaller amplitudes respectively. The circuit in Fig. 2(a) shows a WTA with current input $I_k$ and voltage result as $V_k$, [19].

Voter circuits also accommodate several inputs, and determine the consensual input value as the output result. The differential circuit presented in Fig. 2(b) carries out a weighted average and dynamic compression of the output voltage, [20]. The digital inputs may be incorrect or sustain a large amount of voltage variation. The analog output voltage is subsequently thresholded to recover the correct binary voted result. The voter is used in an artificial neural network topology to recover correct binary operation in cases of massive defect density.

The sigmoid operation is fundamental to artificial neural network operation, creating the nonlinearity that follows the sum-of-products. Diverse analog circuits have been proposed in literature. Some simple voltage-domain realizations consist of a properly sized inverter, while complex realizations make use of an integrated operational transconductance amplifier and following output stages. Digital realizations have also been presented, generally using a significant amount of resources in terms of algorithmic operators or Lookup-Table (LUT) hardware, [21]. The potential programmability of some parameters of the sigmoid faces severe issues in analog and digital implementations, requiring
additional area to store and route data.

Highly non-linear, bell-shaped functions can be implemented in the voltage domain using the bump and antibump circuits, [22]. The circuit presented in Fig. 2 (c) has two inputs, \( I_1 \) and \( I_2 \) and one output \( I_{\text{out}} \) which implements a “bump” of current at \( V_2-V_1 \) equals 0 V.

Resistive networks are used to implement the diffusion function to next neighbor. Figure 2 (d) shows the example of a two-dimensional resistive array where each node connects to its four first-neighbors as well as to the ground, [23]. Resistors can be suitably replaced by MOSFETs operated in weak inversion and used as pseudo-conductances, which offers the advantage of a compact integrated realization, and also enables the electric control of the diffusion length, and hence the creation of anisotropic diffusion. The network may be of higher dimensionality, or implement a hexagonal topology.

![Fig. 2. Classical nature-inspired circuit blocks. (a) Continuous-time WTA, (b) analog differential voter, (c) example of a bump-function synthesizing circuit, and (d) concept of a two-dimensional resistive network.](image)

The analog design of neurons consists of creating circuits that implement equation models typically consisting of first-order differential equations, in an accurate manner. The main system blocks of a silicon neuron are expressed in [24] as the temporal integration block, the spike or event generation block, the refractory period mechanism, the adaptation of spike frequency and threshold blocks. Temporal integration is used to implement the synaptic dynamics. The circuit in Fig. 3 (a) integrates spikes presented in the voltage domain at \( V_{\text{in}} \), and generates \( I_{\text{syn}} \) which has an exponential rise and decay, [24–26]. An integrate-and-fire neuron is shown in Fig. 3 (b) where spikes are generated when the central node exceeds a threshold, following the Volterra model, [27]. Complex spiking patterns can be achieved in hardware, to the cost of increasing the complexity of the analog circuits. For example, the circuit in Fig. 3 (c) realizes the hardware implementation of Izhikevich spiking neuron model, [28]. A variety of synapse and neuron circuits exist and most demonstrate silicon implementation and operation of a single unit, focusing on the accuracy of the electronics to implementing non-linear equations that govern diverse neural dynamics.

Circadian rhythms regulate the general cycles of the metabolism. Rhythmic signals need to be locally generated to control the short-term activity of organs or limbs. Biological spinal central pattern generators (CPGs) consist of a group of interconnected neurons that are in charge of generating such cycles, for example the generation of the limb control patterns that support locomotion. The Amari-
Hopfield neuron model has been used in [29] to generate diverse quadruped locomotion dynamics. To this purpose, four neurons operate in a coupled manner using excitatory and inhibitory connections. Each neuron is designed as an analog system (Fig. 4 (a)) operated in subthreshold mode to limit the application power consumption. The efficiency of silicon CPGs has been demonstrated in [30] where the rear limbs of an anesthetized cat are controlled by a CPG driving constant-current stimulator. The neural network that implements the CPG consists of four integrate-and-fire neurons that are connected to each other, also including reciprocal inhibition, Fig. 4 (b). In this schematic, L and R related to left and right, and E and F related to extension and flexion, respectively. Inputs consist of sensory limb feedback (e.g., $L_{E,EXT}$, tonic bias input, tonic bias (e.g., $B_1$), as well as lateral connections where a filled circle marks inhibition. The neuron output generates periodic stimulation signals. A digital implementation of a leech heartbeat CPG has been demonstrated in [31] using an FPGA platform. An Izhikevic neuron model is adapted to a digital implementation, and implemented in a pipeline processing.

Additional neuromimetic circuits and circuit techniques are presented for instance in [32, 33].
5. Architecture-level realizations

Numerous neuromorphic models have been implemented into hardware, demonstrating clear benefits obtained from analog design techniques over digital techniques, specifically in terms of power consumption, and operation speed. Nevertheless, the integration of neural networks into small architectures with the ability to carry out an application has proven quite detrimental to the analog implementation, which requires a one-to-one mapping of the network topology into the hardware. Digital systems organized into a classical processor architecture naturally support virtualizing the network, where all parameters and variable data are stored in a memory, and all calculations operated from local register banks and customized arithmetic units. In such a context, the development challenges consist of defining a processor architecture that optimizes the processing throughput. Memory and caches access strategies must be defined that minimizes transfer in consideration of the implemented algorithm. Parallelism of operators may increase the global processing speed. Data format and its impact to the results may be optimized. Artificial neural algorithms and learning algorithms as well as the implementation architecture must be adapted to optimize processing speed while also keeping power constrained to a reasonable level.

Digital hardware implementations of ANNs have been proposed in the 1990’s using VLSI technology. The general concept underlying most presented developments relates to supporting a virtual network in the sense that a limited amount of physical resources should be integrated, which are expected to emulate neural networks of arbitrary sizes. A certain level of programmability should be offered by software reconfigurability of the hardware. A possible taxonomy of digital ANN emulators is presented in Fig. 5, which is loosely based on [34].

![Fig. 5. Digital realizations of ANNs.](image)

The emulation of ANNs can be undertaken with the use of serial computers (CISC, RISC architectures); this is generally referred to as a software implementation that is executed using a general purpose processor. The efficiency of the processor versus a conventional CPU is increased by dedication to the exclusive processing of neural algorithms, whereas a conventional CPU would share its processing time among several tasks including serving of the operating system, peripheral units, and other software executions.

In order to further increase the efficiency of a serial processor, it is possible to add some dedicated instructions that are suitable for hardware acceleration. Referred to as optimized RISC processors, these units exploit the locality of neural algorithms, which have an efficient implementation taking advantage of the load/store architecture, the large register file, and the pipelining mechanism. Parallel implementations have emerged as a result of efforts to reflect the inherent parallelism of neural algorithm into the hardware. Hence, a careful study of the different neural algorithms and their learning schemes allows the determination of a set of common operations which therefore have an advantageous hardware integration. These appear to be dominated by operations involving vectors and matrices, [35]. The documented realizations differ in their architectural organization (topology/connectivity, but also the way that the information is transmitted up to the processing element), as well as in the nature of the atomic processors. Three sorts of these latter are reported to be implemented, namely DSPs, RISC and optimized RISC processors, as well as dedicated neuro-chips. A so-called slice organization of processors is intended to split out the computational burden of one
given operation into several identical processors, producing their result all at the same time (SIMD architecture, Single Instruction Multiple Data). The recognized ability of the systolic architecture to implement multiplication in an area-saving manner is exploited with systolic arrays of processors. More recently, exploiting the inherent parallel operation of graphic processing units (GPUs) and their efficiency to processing matrix operations has been recognized. The established fact of excessive complexity, power and flexibility of conventional DSP or RISC processor architectures has opened the door to their replacement by specialized processing elements (PEs) in parallel systems, to be fully integrated on the same die. SIMD arrays (central control) and systolic arrays (local control) in the form of linear systolic arrays, also referred to as ring systolic arrays or two-dimensional systolic arrays, have been applied as parallel integrated implementations of dedicated hardware. Dedicated neuro-processors include all digital systems which are designed for the purpose of exclusive acceleration of a small class of neural algorithms. Within this family we find ICs exhibiting structures and architectures very different from the conventional previously mentioned ones, which inherently exhibit small flexibility. Also we include the relatively new family of reconfigurable designs based on FPGA technology into this family. The design-flow consisting of a high-level description of a system using a hardware description language such as VHDL/Verilog, to be synthesized, and then almost automatically placed and routed by software tools, has attracted large interest in the community of digital neural networks. Several realizations have emerged, taking advantage of this fast design-flow in order to implement specific ANN hardware, which can dynamically be tailored to the actual task to be solved. A Hopfield ANN with on-chip learning based on an evolution of the neuron is demonstrated in [36].

Finding the optimal precision of all input, output, variables and weights that are required for a successful processing, yet trying to minimize the area cost is a major trade-off designers of digital hardware systems are facing. Two architectures are shown as typical examples in the following.

The CNAPS neurocomputer architecture, [37] depicted in Fig. 6 (a), consists of a SIMD parallel configuration of elementary processing units (PNs for Processor Nodes), linked together with two 8-bit I/O busses and one 31-bit command bus. Each of the PN (Fig. 6 (b)) is linked by the means of a bus broadcast interconnection scheme. The IC contains 13.6 M transistors for 64 PNs. The architecture of one processing element concentrates several registers and memory devices, including a 4 kB local memory, as well as ALU integer operators composed of an 8×16-bit two's complement multiplier, a 32-bit adder, and a shifting unit aside two 16-bit busses. The division has to be software emulated.

The MANTRA 1 architecture, [38] depicted in Fig. 6 (c), implements a 2-D systolic array of processing elements called GENES IV, Fig. 6 (d), as a hardware support to several connectionist models and learning algorithms. The GENES IV unit is a 4×40 array of processing elements, with I/O access to the diagonal elements, implementing systolic flow of data as well as instruction, which reduces the latency in case of a modification in the operation mode. A processing element consists of several temporary registers, an instruction and weight unit, and an arithmetic unit, each of which rely on software simulations for an optimal choice of the required precision. A TMS320C40 processor is used to control the MANTRA1 board, and to generate sequences for the GENES IV to process the neural algorithms. A GENES IV processor housing a 2×2 array of PEs has been constructed, totaling 71,690 transistors.

The recent revival of ANNs in the context of deep learning has also been supported by hardware accelerators that implement supervised and unsupervised models. Real-time scene analysis from a video stream is a challenging task that deep convolutional networks have been assigned to attempt. The so-called “neuFlow” system is a scalable dataflow architecture that has the capacity of efficiently emulating deep convolutional networks, [39, 40]. The architecture of a neuFlow processor consists of a two-dimensional array of Processing Tiles (PTs), Fig. 7 (a). Each tile can be reconfigured in run-time to operate as FIFOs, arithmetic operators or a combination of arithmetic operators. Each PT communicates with its neighbors and external memory. The architecture has been implemented onto a Xilinx Virtex 6 FPGA platform approximately consuming 10 W, and is supported by a custom dataflow compiler. Street view analysis at 12 fps has been proposed as the target demonstration application. Big data processing poses severe constraints on the processing load and time, and several
deep-learning algorithms have been considered potential solutions to accelerating the performances. An other FPGA-based accelerator of deep convolutional network is proposed in [42], where data servers operations acceleration is aimed as the final application. The architecture is shown in Fig. 7 (b), and consists of an expandable array of processing elements that receive data from a system of buffers that limits off-chip communication. Run-time software reconfiguration gives the system the flexibility that is required to support the multi-layer convolution calculations. Stratix V FPGAs are intentionally selected as mid-range FPGAs in a system consuming 25 W. An ASIC implementation capable of supporting convolutional neural network is presented in [43]. The custom processor is fabricated in a 65 nm technology, consumes an average of 185 mW, and is suitable to low-power portable and battery-powered applications. The system architecture consists of a multicore two-dimensional mesh network-on-chip connecting four deep-learning and two deep-inference processors. An additional ASIC for portable image (three-dimensional gesture) and sound (speech) processing and implementing multiple deep-learning cores is presented in [44], which is fabricated in a 65 nm technology, and consumes 126.1 mW. An extreme learning machine system is presented in [45], with the goal of decoding neural codes in signals acquired from the brain, and subsequently generating control signals for a neuroprosthetic device. A mixed-mode ASIC is developed as a coprocessor in a 0.35 μm technology which consumes 0.4 μW from a double 1.2 V and 0.6 V supply. Restricted Bolzmann Machines implement an unsupervised model of deep learning algorithm. A Stratix III FPGA implementation is shown in [46] which aims at exploiting internal computational resources of modern FPGA efficiently, i.e., internal control processor, bus and arithmetic operators. An additional implementation considering an FPGA prototyping approach is presented in [47]. The latency of computation is kept in linear relation to the number of neurons using a parallel and scalable architecture consisting of processing data in blocks of four pairs of visible and hidden neurons. One processing block is shown in Fig. 7 (c), which is repeated in a fully scalable manner to accommodate large network sizes, Fig. 7 (d).

The necessary bit-precision for training and running trained networks implemented in a digital ar-
Architecture has constantly been studied over several decades. Several quantitative results have yielded from these studies, without any consensus on precise values to be widely accepted as universal directives. The reason for this discrepancy lies in the variety of analysis methods applied, as well in the use of problem dependent benchmarks. Moreover, these studies restrict to some specific algorithms, for example, learning is mostly assumed to be done with the backpropagation algorithm. The forward (recall) step generally requires some 6 to 8-bit of precision at least [48], whereas the backward step (backpropagation algorithm) requires 14 to 20-bit of precision at least [49], although some authors are satisfied with less, as to say 10 to 12-bit [50]. More recently, studies have focused on the necessary precision in the case of deep-learning algorithms. In [51, 52], selected artificial neural networks (e.g., the Maxout networks) are trained using selected classical datasets (e.g., MNIST, CIFAR10), and considering different number formats. Relatively low bit-precision equal to 10-bit to 12-bit are assessed sufficient to keep performance for arithmetics and storage.

![Fig. 7. Example digital deep learning architectures. (a) neuFlow system architecture. (c) Block-diagram of a RBM four-hidden and four-visible neuron realization, and (d) corresponding hardware realization that expands to arbitrary-size networks.](image)

The usage of graphic processing units (GPUs) has extended outside pure image processing as a benefit of very high computing capabilities. Scientific processing makes a wide usage of GPUs, which are used as a parallel processing platform that is software-programmed using dedicated parallel programming platforms, e.g., CUDA, [53]. The efficiency of GPUs in accelerating simulations in the field of bioinformatics has already been recognized, [54]. The suitability of the network topology to the hardware appears to be a relevant issue which is discussed in [55], along with additional important parameters such as the network size and its connectivity, memory alignment and floating precision. Deep unsupervised learning in large-scale networks is presented in [56], evidencing a clear benefit of
GPUs over CPUs.

Some additional recent processor architecture developments include Qualcomm’s Zeroth cognitive platform, [57], Google’s Tensor Processing Unit, [58], Darwin neuromorphic processor implementing spiking neurons, [59].

Architectural analog developments include analog processor or smart sensors into which neuromorphic operators are integrated, at the level of the front-end, and aiming at carrying out signal processing at the closest possible to the source. Classical CMOS image acquisition systems follow the specifications that are demanded by electronic consumer product manufacturers [60]. Nowadays, the major development target of imagers relates to a very high pixel count and density, a high fill factor, a dynamic range that is adapted to the application, low power operation using classical circuit techniques [61], or fabrication process techniques, and in some specific cases high frame rate [62]. Smart image sensors specifically lend themselves to integrating neuromorphic primitives at the level of the focal plane as a benefit of the inherent parallel acquisition of the pixels. Imagers that integrate neuromorphic algorithms in the focal plane aim at realizing a globally complex operation using pixel-local operations as a way to fulfill some of the aforementioned commercial targets. In most cases, trading off some system characteristics must be accepted. Typically, implementing additional electronics at the pixel level will increase the pixel area and likely decrease the fill factor while this may reduce the overall system power consumption by eluding the need of a digital signal processor on-chip that should carry out filtering operations in the digital domain. The use of Cellular Neural Networks (CNN) to process images in the focal plane is described in [63–65]; a 128×128 array of smart pixels is presented which is intended to model the operation of several layers of the visual pathway. In [66], an imager system that is sensitive to changes in consecutive frames is presented, which is based on analog, in-pixel processing. [67] presents an alternate analog neuromorphic imager that focuses on the emulation of the biological signal paths. A neuromorphic analog imager is presented in [68] where image processing is carried out in the analog domain mimicking processes observed in rabbit’s retinas. Additional types of neuromorphic sensor systems have been considered, e.g., neuromorphic cochleas [69, 70].

6. Learning

Learning in hardware is central to the successful implementation of neuromorphic models. Some neuromorphic algorithms do not require any training phase (unsupervised learning), and they adapt internal parameters upon each presentation of a new data. Many neuromorphic algorithms require a training phase (supervised learning) during which selected input-output pairs are presented to the network in turn, each resulting in an error. Diverse learning techniques are applied to adapt internal parameters until the observed error is acceptable. The parameters generally consist of weights or connection strengths. The key to learning in hardware consists of the storage of the parameters, as well as the efficient access to the parameters. Training may be carried out off-chip, if an accurate model of the neuromorphic system is available. The resulting system may target area and power savings; the parameters are stored in a read-only memory (ROM) which prevents further adaptation of the hardware. The chip-in-the-loop training method stipulates that the neuromorphic system should store its parameters into some type of random access memory (RAM), which enables re-training to change the stored values. The training support itself may consists of software that is external to the neuromorphic chip that consequently preserves its low-power and low-area characteristics while keeping flexibility to adapt to a new data set. On-chip training requires that the neuromorphic chip including its training hardware be implemented on-chip.

A review of methods, circuits and applications of learning in silicon are presented in [71, 72]. The most popular supervised learning algorithm for ANNs is the so-called backpropagation of error which is computationally intensive in vector-matrix multiplications, [73]. Hebbian learning, [74] or autoassociative memory, [75] are two popular and classical models of unsupervised learning where no teacher guides the learning which aims at maximizing mutual information between analog input and discrete output classes. Reinforcement learning [76] operates with feedback in discrete-time and is based on
delayed reward-punishment mechanisms. Spike timing dependent plasticity [77] is a recent learning scheme that exploits the precise timing between pre and post synaptic firing events in the learning rule. Numerous adaptations of these major learning rules have been proposed, as well as hardware-friendly rules, such as weight perturbation [78].

The hardware implementation of ANN training requires accessing and storing the weights attached to every connection in the network. A classical processor architecture consisting of a central processing unit that computes the learning rule, and a global memory that stores all weights is generally preferred. Hence, performance improvement originates from the architecture of multiple parallel processing units, parallel memory segments and their communication bus, that should optimally support the learning algorithms. This issue is central in the development of new large-scale neuromorphic systems.

The memorization of the synaptic strength values has a reliable realization in the digital domain, which offers a flexible choice of the resolution, a safe storage and retrieval of the values, [79]. The analog storage of weight values contrasts very much with the digital approach. Long-term and short-term storage can be distinguished in their implementations. In analog hardware, weights or connection strengths are generally distributed, i.e., are stored in the vicinity to the analog circuits that carry out the neural operation. Short-term storage can be carried out using integrated capacitors that store a weight value in the charge domain. Issues that may arise relate to the precision that can be achieved in the stored charge. Charge injection may cause offsets during programming, reading may be destructive. Finally, charge leakage has become a significant issue in modern fabrication technologies. Nevertheless, the simplicity of the technique and its low area consumption make it an attractive candidate for analog storage. Long-term storage of analog weights may require floating-gate fabrication technologies, and memristors [80]. Three-dimensional chip stacking may offer solutions in future developments.

7. Large-scale realizations

The advent of very-deep submicron fabrication technologies, as well as supporting design methodologies and software tools for CAD have promoted the development of advanced digital microelectronic systems. In addition to the development of application-specific integrated circuits (ASICs), field-programmable gate-array (FPGA) system development has emerged as a new and tractable platform for prototyping and also for some specific industrial applications. Research on neuro-mimetic microelectronic systems has taken benefit of progress in the semiconductor fabrication technologies as an enabler of the upscaling of integrated neural network size and complexity, [81].

The turnaround date of this new development cycle may be difficult to clearly state, as the research process has consisted of incremental progresses achieved over a significant time span, and has been shared by many research groups. Nevertheless, some relevant developments may be identified as the foundation of neuromorphic systems upscaling, [82].

The so-called “CAM-Brain Machine,” (CBM) has been developed in the decade from 1993 through the beginning of the 2000s’, [83]. Four CBMs have been built in Japan, Belgium and in America, based on the idea of creating a new research field to be named “brain building.” FPGAs are used to implement a genetic algorithm that is used to evolve a 1,152-neuron neural network that emulates a three-dimensional cellular automata. 64,640 such modules are stored in a RAM along with interconnection information, thereby creating a 74.5 million neuron artificial brain that is updated by the CBM at a rate of 130 billion of CA cells per second. A simplified CA model has been developed under the major constraint of fast processing, [84]. A 1-bit signal encoding is used as a spike interval information encoding. As a specificity, the structure of the neural network is not engineered, but is evolved on the hardware platform. The number of FPGAs in the CBM is equal to 72 which causes a power consumption of 1.5 kW (5 V, 300 A). A kitten robot control has been selected as the target demonstration application.

Classical processor architectures have been used in a variety of projects aiming at simulating large-scale neural networks. The necessity of very high computational resources dictates that semi-custom systems made from commercial units should be developed. The “Blue Brain Project” has started
in 2005 with the goal to simulate 10,000 neocortical neurons within the dimension of a neocortical
column, [85]. Different types of neurons and interneuron anatomical-electrical types are considered,
and a significant importance is given to including parameters and models obtained from in-vivo
measurements. The hardware supporting the simulations consist of a Blue Gene/L supercomputer
developed by IBM, [86]. One atomic ASIC in the Blue Gene/L (BGL) architecture is a system-
on-chip that includes two PowerPC 440 cores clocked at 770 MHz, and a floating-point processing
unit. Two BGL chips complemented with up to 1 GB of memory are integrated as a card. The
entire supercomputer comprises four system-racks (cabinets), each of which comprises 64 racks into
which node cards are mounted. Thus, the supercomputer comprises 65,536 computing nodes and
32 TB of memory that implement a torus architecture. The peak computing capacity is assessed at
360 TFLOPS (tera floating-point operations per second). A Beowulf cluster comprising 27, 3 GHz
processors was used in 2005 to simulate a large-scale model of the mammalian thalamo-cortical system
comprising 10^{11} neurons and 10^{15} synapses, [87]. Several neuronal dynamics are supported, [88].
One second of biological time was simulated over 50 days of computation. These large-scale simulators
are constructed as custom architectures using classical processors as computing elements. Their
neuromorphic nature may be argued, and may only result in the way the software makes use of
the hardware architecture, while their system architecture results from a classical supercomputing
approach, that is used for the sake of speeding up neuromorphic model simulations.

The “SpiNNaker” system has been developed with the goal of benefiting from massive parallel com-
putation to simulate large-scale spiking neural networks in biological real-time, [89]. One architectural
SpiNNaker node consists of 18, 200 MHz, 32-bit ARM9 processors, each with 96 kB of local memory,
128 MB of shared memory, a dedicated packet router, as well as peripheral support hardware. 16
cores are effectively used to process neural dynamics operations, specifically to simulate the point
neuron model, where all inputs are considered to be directly applied to the soma, regardless of the
dendritic dynamics. Each ARM9 core is expected to simulate the behavior of approximately 1,000
neurons, [90]. The neuron model can be selected among spiking neuron models such as leaky integrate
and fire, or Izhikevich’s model, [91]. Considering that communication between neurons is the bollte-
neck of classical architectures, a specific routing mechanism based on a simple packet representation
based on an adaptation of the AER [92] (Address Event Representation) is implemented. Each node
is packaged using three-dimensional packaging, Fig. 8 (a). The number of parallel chips that can be
implemented is claimed to be only limited by physical and budget constraints. A system consisting
of 48 nodes/chips has been demonstrated, along with the necessary software, Fig. 8 (b). The example
of closed perception-action loop for mobile robot control has been shown.

The “Neurogrid” hardware has been designed at the University of Stanford as a neuromorphic
system enabling the real-time simulation of neural models, [93]. Neurocore chips are the atomic
hardware ASIC which are developed in a 180 nm fabrication technology, and emulate 65,000 neurons,
approximately, Fig. 9 (a). A Neurogrid board integrating 16 Neurocore ASICs has been developed,
that emulates 1 million neurons, approximately, while consuming 3.1 W, Fig. 9 (b). The neural
dynamics is modeled as a quadratic integrate-and-fire, which is implemented in the analog domain.
Spikes are transmitted between chips using AER.

A wafer-scale approach is proposed in the development of the “HICANN” (High Input Count
Analog Neural Network) chips and with the aim of modeling neural tissues. The project has been
consecutively supported by the European Union projects “FACETS,” “BrainScales,” [94], and has more recently been included into the “Human Brain Project” (HBP) research, [95]. Mixed analog-
digital techniques are used to implement an exponential integrate-and-fire neuron model which has 23
analog parameters, each. One HICANN chip [96] implements one analog network core (ANC) as well
as communication and controlling circuitry, Fig. 10 (a). The maximal number of pre-synaptic inputs is
as high as 14,336, and synaptic weights are represented by a current generated by a 4-bit multiplying
digital-to-analog converter (DAC). The membrane capacitance is realized using metal-insulator-metal
(MIM) capacitors, physically located in the upper layers of the chip, and hence area-controlled. A
synchronous serial event protocol has been developed that enables data transfer rates between ANCs up
to 2 Gb/s. A major specificity of the approach resides in the wafer-level integration of 352 HICANN
chips onto 8-inch silicon wafers, Fig. 10 (b). An additional metal layer is deposited on top of the wafer to support inter-chip communication. One wafer contains $4 \cdot 10^7$ synapses and 180,000 neurons. Inter-wafer communication is handled on a system PCB by specialized ASICs and control FPGAs.

The “TrueNorth” architecture has emerged in 2014, as a collaborative work of IBM research centers in America and Japan, and Cornell University, [97]. A 5.4 billion transistor ASIC has been developed in a 28 nm CMOS technology, that physically implements 4,096 ($64 \times 64$) neural cores, Fig. 11. Based on an architecture that integrates (neural) computation, memory and communication in a tight manner in a core, one chip has the capacity of emulating one million neurons and 256 million synapses. Each core consists of 1.2 million transistors that fit into a 240 $\mu$m by 390 $\mu$m size. A time-multiplexed neuron computes various neural dynamics, and is locally connected to a scheduler, memory, router and a controller. An interchip communication facility enables scaling the systems to multi-chip operation.

The operation of each core is based on the digital computation of spiking neurons, and spike events are carried throughout the network using time-multiplexed wires that connect a mesh network of routers. Each chip consumes 65 mW, [98]. The primary application has been determined as multiobject detection from a video input, [99]. The research effort has been supported by DARPA under the acronym “SyNAPSE” standing for Systems of Neuromorphic Adaptive Plastic Scalable Electronics. Multiprocessor boards using the TrueNorth architecture have been presented. For example, a system
integrating 48 TrueNorth chips has been developed and claimed to be representative of a rodent’s brain in term of the number of neural cells, [100]. In spite of a high potential towards applications expressed by its authors, the argument of the possible inappropriate encoding of data (spiking neurons) has been argued against this architecture [101].

Criticism with respect to the aforementioned projects has emerged in various ways. Of course, the methods and applied models result from a project decision that may be argued. Beyond these aspects, arguments have emerged regarding the possibility to quantify the successful modeling obtained by these systems, in true understanding that the brain complexity and nonlinearity is very high, and that all these systems are actually built to obtain a better understanding of the brain, i.e., build a brain to understand its complex operation. Moreover, the level of detail of a model is of critical importance to the obtained results, and also has a significant impact on the simulation time and its tractability [102]. Eventually, the increased size and funding required by such large-scale simulation project appears beyond the capacity of small research groups, and needs the support of government or supra-governmental agencies. The governance and control of such project may become of major importance to its success, regardless of the scientific value, [103].

8. Prospective applications and market
A wide range of applications may take advantage of nature-inspired algorithms to support solving complex problems. The absence of a determined model of some reality application is also a prescriptor of the usage of artificial intelligence algorithms in control or classification tasks. In all evidence,
the vast majority of the applications that have been presented from the early times of neural networks are software based. Certainly, the necessity of verifying the pertinence of any algorithm and assessing its efficacy using a software model and a test set of significant size is a first strong reason to this situation. Neuromorphic application reviews are quite few, in spite of a respectable history of developments. Neural network fundamental and applications are presented in [104–106], in quite early developments. Applications that require specific neuromorphic hardware are found in domains where classical processor architectures face their limits in terms of real-time or high-speed processing of complex algorithms, power limitations as the major factors. Some applications which have not been presented in earlier Sections are discussed in the following.

The ability of the brain to recover from the continuous death of neurons has inspired neuromorphic research conducted in mimicking this property into fault-tolerant hardware. Neural-inspired Boolean operators are presented that operate over spike sequences. Neural logic blocks are presented in [107], which consist of a hybrid of CMOS and memristor circuit arrangement that can be reconfigured, as a FPGA. The system is presented to be fault-tolerant, and Monte Carlo simulations have been processed to confirm this statement. Clearly, the need of dynamic reconfiguration is established to adapt to soft errors (transient), and hard errors that result from permanent ageing effects. The tolerance to variation of environmental or functional conditions, as well as internal failure of atomic components of biological systems is widely recognized, and a source of inspiration for hardware reliability enhancement. Artificial neural networks based on the perceptron model can be trained to sustain a certain level of internal defectiveness occurring at random in time and space domains; Boolean operators can be synthesized using this principle, showing significant tolerance to random fault, at the cost of a large increase in hardware and thus power dissipation [108]. Low redundancy factors appear of primary importance in the synthesis of fault-tolerant digital systems, where all classical schemes which are used involve very large redundancy factors, typically several tens to several hundred. Based on this observation, a multiple-layer arrangement forming a fault-tolerant digital system inspired by ANNs is devised, as depicted in Fig. 12. In this system, data progresses in a strictly feed-forward manner. The first layer comprises redundant digital clusters. Data provided from the clusters output are analog in essence, reflecting faults that modify the correct transfer functions. Replicating the architecture of the perceptron, analog data is fed to an averaging unit which also rescales voltages to comply with the available power supply range. Finally a thresholding voter restores correct digital outputs. The content of a cluster is variable from a single Boolean gate to a full functional block, and must be adapted to the local defect density, [109].

Big data processing is one of the major modern application of neuromorphic systems. Large data centers are in need of computing power to support data analyzing and sorting in the form of co-processing servers. Several global companies are known to carry out research in the general domain of AI (artificial intelligence), also including hardware developments, [42, 82, 110]. Several large-scale projects have recently been started that relate to the modeling and simulation of the brain operation using neuromorphic hardware. The European Community Flagship Human Brain Project that has been approved in January 2013 also considers neuromorphic computing as a central tool to its aim, [95]. As such, this research area is acknowledged a dynamic area offering promising developments, and research results are expected to strengthen knowledge and propose applications in domains including neuroscience, implantable bioelectronics and healthcare devices.

Interfacing neuromorphic systems with the living has been demonstrated in [111], where a neuronal culture in vitro is stimulated in open-loop depending on the activity of an in silico artificial spiking neural network. This domain of research aims at developing closed-loop systems where implantable microelectronic system communicate in a bidirectional manner, i.e., in closed-loop with excitable living tissue. The interfacing layer between the living and the electronics should consist of an artificial spiking neural network. This domain is still in its infancy, and many fundamental steps must still be taken to the development of smart neuro-prosthetic systems.

According to [112], the global market of neuromorphic chips should be at USD 1.2 billion in 2016, and expected to reach to USD 4.8 billion in 2022, growing at a compound annual growth rate (CAGR) of 26.3 %. The target end-user market segment is expected to be dominated by consumer products at
Fig. 12. (a) Four-layer fault-tolerant architecture, (b) training of a NAND operation showing a short initial training phase, followed by random fault injection and retraining; the mean square error is shown to decrease confirming that the network tolerates random faults by training.

more than 98%. The other market segments should share the remainder at approximately identical segments, consisting of medical, aerospace and military and defense, industrial, automotive. Image recognition applications should dominate the field at 60% followed by data mining at 20%. Consequently, applications of neuromorphic hardware are expected to become prominent in daily life within the next years.

9. Recent developments

Some new concepts have emerged, which appear inspired or derived from the latest VLSI technology, and which reveal that the domain has reached a maturity level that extends beyond proof-of-concept, and requires powerful methods to leverage further developments. Such concepts are presented and the applicability of neuromorphic systems is discussed from a high level perspective.

9.1 Device level, the memristor and crossbar realizations

Material science developments and their application to semiconductor fabrication processes is a source of inspiration for neuromorphic design. One major development consists of the memristor which is considered in many researches. The memristor has been postulated by Leon O. Chua four decades ago, as the fourth fundamental electrical element complementing the resistor, capacitor and inductor quartet, [113]. Theoretical and circuit developments have been presented in the years following the discovery, [114], and identifying the presence of the memristive effect in various systems, including the Hodgkin-Huxley membrane circuit model. Nevertheless, research progresses have been restrained by the absence of proper fabrication technologies enabling the demonstration of the device. Memristor research has significantly regained in interest since its proof-of-concept circuit has been presented by HP in 2008, [115]. Understanding its particular characteristics, many groups have turned their attention to study the memristor, at various levels of abstraction. Still its modeling, circuit and systems and fabrication are in very early stages of their development. Recently, the existence of the memcapacitors and meminductors has been postulated, [116].

Modeling the memristor has been carried out with a first target of understanding its intrinsic physical characteristics, and translating them into a reproducible electrical behavior. The fundamental definition of memristance $M$ states a functional relation between charge and flux: $d\phi = M dq$. Making $M$ a function of $q$, nonlinearly relates $\phi$ to $q$, thus yielding a nonlinear device. The resulting i-v curves in Fig. 13 (a) show the classical bow tie shape (under the electrical parameter and conditions given in [115]). The theoretical model may have a variety of physical implementations.

Behavioral or electrical models on the other hand have been proposed over the past few years and openly published. A strictly theoretical model is presented in [119] which includes some physical effects. In [120], nodal analysis is modified to account for the presence of the inductive device at dc, and the memristor is modeled into first-order differential equations where magnetic flux is the
Fig. 13. Memristor concept and neuromorphic applications. (a) i-v memristor characteristics under a sinusoidal input voltage stimuli, [115, 117], and (b) Conceptual view of HP memristor device and basic operation; adapted from [134]. (c) Small-scale version of the feed-forward hybrid CMOS-memristor array proposed in [118].
A chaotic circuit is presented in [131, 132], making use of the memristor, and which is constructed from discrete elements, i.e., the nonlinear operation of the memristor is emulated using passive and active elements (operational amplifiers). Diverse chaotic circuit models based on configurations of three memristors and switches are presented in [133].

While most existing circuits and systems have been demonstrated in simulation, only, some attempts to demonstrate the usage of the memristor in physical implementation has also been successful, and published in recent literature. The fabrication of the memristor has first been demonstrated in 2008, as a crossbar architecture [134]. The proposed memristor consists of a stack of TiO2 which has a perfect ratio of oxygen versus titanium of 2:1 and acts as insulator, on top of which lies a layer of TiO2-x, which exhibits a lack of 0.5% in oxygen, as depicted in Fig. 13 (b). The switch consists of a cube of 40 nm³ placed in-between two perpendicular stripes of platinum as contacts. Upon application of a voltage to the switch, oxygen vacancies scattered through the TiO2-x layer tend to modify their location, spreading into a larger or smaller portion of the switch, thereby modifying the resistance of the switch. The memory effect of the devices lies in the fact that the state of the system in terms of oxygen vacancies and their global locations remains upon turning bias off.

Along the thin-film based fabrication of memristors, alternate technologies are under study. For example, the spin torque magnetization effect is used in [135] to fabricate a CMOS compatible spintronic memristor; application in multilevel memory, temperature sensing and data security are presented. Stepwise switching of the magnetic tunnel junction is presented in [136] as a novel method to fabricate spin-torque based memristors. A ferroelectric realization of the memristor is presented in [137], targeting the emulation of synapses. Organic materials are considered in the implementation of memristors and synapses in [138].

Memristors are fundamental elements of neuromorphic systems learning as dynamic flexible and distributed memory elements. Several STDP learning models are discussed in [80, 139] including STDP, double-spike STDP, quadratic STDP which are suitable to the usage of memristors, possibly arranged in arrays, Fig. 13 (c).

Recently, the correctness of the CMOS-memristor hybrid design has been presented in literature, based on theoretical study. No large-scale applicable hybrid CMOS-memristor technology exists, and two architectures are independently presented. Stacking a layer of memristors over a CMOS chip is presented in [140]; a quantitative comparison with existing memory technology showing the power-saving superiority of the memristor technology is also presented in this reference. [141] also proposes the hybrid integration of crossbar arrays of memristive devices with CMOS, including in-situ learning. A competing technology consists of full hybrid integration of CMOS and memristors, as computing arrays. This technology exists in research centers and presents issues in terms of excessive area, though it appears more promising in terms of computational power, co-integrating memory and computing. Currently available studies have been conducted along the lines of nanoelectronics, logic, [142], modeling and memory [143].

9.2 Unconventional computing

Facing the limits of conventional processing architectures has motivated identifying new computing methods based on non-classical concepts. Nature in a general sense is a source of inspiration in the development of unconventional computing algorithms and their subsequent possible efficient implementation into custom microelectronic hardware. Some of these new concepts are not stricto sensu neuromorphic in the sense that the models are not strictly limited to neuronal operation.

Noise is inherent to any real process, and can not be avoided in microelectronic systems in which it has traditionally been considered as an effect that must be combated. Nevertheless, noise has been observed to promote processing in biological systems including populations of neurons, e.g., [144–146]. Hence, non-conventional logic and noise-based operators have been considered potential replacement of standard Boolean operation that have the capacity to absorb faults. Noise-based logic is proposed in [147] as a way to exploit inherent electronic noise. Stochastic resonance generally relates to the possible detection of signals with an intensity normally under the detection threshold as a benefit of the injection of noise into the system. This phenomenon is particularly suitable to neural operations which...
specifically operate from trains of pulses that are generated upon overcoming the neuron threshold. The concept of stochastic electronics is presented in [148] as a way to design reliable microelectronic systems in highly noisy or mismatch fabrication environments.

Collision-based computing is an alternate unconventional processing method. Waves (or spikes) propagate in an active medium composed of oscillators (or neurons) and obey specific rules upon colliding at a specific node. A model based on a media consisting of single-electron oscillators is presented in [149], where circuits of the complexity of a full-adder are simulated.

9.3 Concepts of maturity, roadmap and benchmarking

The community of neuromorphic hardware researchers has recently considered new concepts which appear to reflect a certain level of maturity that the field may have reached. Specifically, topics have emerged that relate to the need of organizing the domain and research.

Controlling how the performance of newly developed neuromorphic systems is assessed has been a recent concern. As is the case in other domains of microelectronics or computer science, the application of some sets of traditional benchmarks (e.g., MNIST database, [150, 151]) has become a common usage, de facto. Benchmarks are of specific importance in the case of neuromorphic systems that have learned an expected correct behavior from a training data set that is necessarily of fixed and limited size. Moreover, some neuromorphic systems are expected to be used in closed-loop, where an interaction with their environment may cause a reaction fed back. [152] discusses issues realted to neuromorphic benchmarks with a specific focus on closed-loop operations.

The immense variety of neuromorphic models combined to the diversity of the possible hardware implementations creates difficulty in classifying the approaches and implementations. A very general classification that covers bio-inspired hardware systems is proposed in [153], that discusses three orthogonal branches named phylogenetic, ontogenetic and epigenetic levels that aside of learning also cover temporal aspects of artificial systems development. [154] reviews existing neuromorphic hardware, as well as the different classification systems that have been proposed between 1992 and 2010. More than ten classification methods have been proposed, which consider very diverse criteria such as the degree of parallelism, the implementation technology (data representation, interconnect strategy, mapping of the algorithms, etc.), the nature of the processor (special purpose, general purpose), and others.

Finally, the necessity of anticipating the future development in terms of scaling of energy efficiency, performance and size is expressed in [155], where the idea of a roadmap is expressed.

9.4 Ethical aspects and controversy

In addition to starting considering neuromorphic hardware systems not only from the specific microelectronic developments, but also from a general, organizational and high level perspective, researchers and also general public increasingly facing amazing applications also understand potential ethical issues or limits of the technology. Interactions between machines and humans have ever raised ethical issues. First used as tools extending the capacity of humans, machines have been applied in agricultural and later industry environment supporting humans in extending productivity and concomitantly replacing humans, [156]. Alongside, machines of increasing complexity have been used as weapons from the beginning of mankind.

The possibility that humans be confronted to alternate sorts of evolved intelligence has ever been a source of curiosity or fear which has created uncountable tales, and later popular culture books, movies, etc. The fact that alternate forms of intelligence may be created by human themselves has been a source of ethical concern and controversy. A milestone in this process certainly consists of Mary Shelley’s literary work “Frankenstein, or the Modern Prometheus,” which may have been inspired by the early electrophysiology experiments of Luiggi Galvani, and which addresses the responsibility of the scientists with respect to their creation, probably one of the first time in modern history and science, [157]. Machines of growing levels of autonomy have been developed in the 20th and 21th centuries, which has favored the emergence of the understanding of the potentiality of autonomous intelligent machines. Models of relationship between humans and so-called “robots” assumed con-
sisting of autonomous machines benefiting of an intelligence equal or higher than mankind has been a matter of growing concern. The publication of the Three Laws of Robotics presented from 1942 by Isaac Asimov in his capacity as an author of science fiction evidenced the emergence of a new understanding of the mutual interactions between humans and robots; though not published as in a scientific context, these rules and their derivatives have been quite influential to generations of researchers, [158]. The possibility of development of machines with sufficient intelligence to have a conscience of themselves, and a capability to construct themselves without intervention or control of humans has conducted to significant ethical and philosophical concerns, as well as fears and phantasms expressed in popular culture. Controversy has followed the statement by artificial intelligence scientist Hugo de Garis of the unavoidable conflict between intelligent machines and mankind, [159].

At the moment of writing, progresses in robotics have enabled the development of many humanoids and animats, though without massive involvement of artificial intelligence in their global control and without successful target applications. Automated or self-driving vehicles have received approval of public or semi-public testing in several countries, also causing a first accident [160]. An AI software-based system has beaten an expert human at the game of go, [110, 161]. On the other hand, AI and its hardware integrated derivatives power specific applications such as big data processing, brain emulation and sensory-systems based control and detection. Integration of AI into a complete autonomous system is not yet performed, though all elements appear to have reached technological maturity to do so. This integration step will necessarily raise philosophical, ethical issues that will to a possible extent require legal texts to define a new context that considers intelligent machines.

10. Conclusions

After three decades of latent development, microelectronic hardware for neuromorphic algorithms implementation or acceleration appears to be in a phase of strong revival and expansion. The reasons are multiple and include the maturity of its fundamental originating fields, namely microelectronic technology and implementation methods, sensory and computational neuroscience. Neuromorphic hardware developments appear to offer solutions to stringent issues faced by conventional computing systems including the excessive power consumption of conventional processing architectures in embedded sensory processing, computing over big data, also including the real-time operation of complex neuro-biology related complex operations. In addition, neuromorphic systems are expected to have the ability of creating models of reality, to some extent by their learning capacity, which may support the understanding of phenomena that are known difficult to model, among which the operation of biological systems including the central nervous system as a part or whole.

In turn, neuromorphic electronics is reaching maturity with its implementation in micro and nanoelectronic systems, which is evidenced by the emergence of new needs such as benchmarking and patenting, the growing interest of industry and the involvement of industry into research and development, the growing investment volume from national and supranational funding entities, as well as a shift of research into higher levels of abstraction based on novel nanoelectronic devices.

At this time, very ambitious targets and expectations are set as goals of neuromorphic hardware systems, thereby also rising some criticism with respect to an excess of optimism, which is also supported by a lack of full understanding of the operation of the central nervous system.

The further development of neuromorphic microelectronic systems will face major challenges emerging at various levels of abstraction, device, circuit and architectures, algorithms. The success of the developments may originate from the concomitant consideration of all these abstraction levels in solving new specific tasks. In addition, low-power, real-time operation may be key factors to demonstrate the real benefits of neuromorphic systems over classical processor architectures.

The aforementioned challenges are of crucial importance to sustain the development of neuromorphic microelectronics towards successful applications. The multidisciplinary nature of the contributions required to this aim as well as the maturity of the domain suggest the possibility to consider neuromorphic hardware systems as a self-sustained field with a high potential of novelty.
References


[57] https://www.qualcomm.com/invention/cognitive-technologies/zeroth


[61] S. Hanson, Z. Foo, D. Blaauw, and D. Sylvester, “A 0.5 V sub-microwatt CMOS image sensor with pulse-width modulation read-out,” IEEE Journal of Solid-State Circuits, vol. 45, no. 4,


[87] Simulation of Large-Scale Brain Models.
http://www.izhikevich.org/human_brain_simulation/Blue_Brain.htm
[103] C.M. Bishop, “Neural networks and their applications,” Review of Scientific Instruments,


