

# Neuromorph

## 38. Neuromorphic Engineering

**Giacomo Indiveri**

Neuromorphic engineering is a relatively young field that attempts to build physical realizations of biologically realistic models of neural systems using electronic circuits implemented in very large scale integration technology. While originally focusing on models of the sensory periphery implemented using mainly analog circuits, the field has grown and expanded to include the modeling of neural processing systems that incorporate the computational role of the body, that model learning and cognitive processes, and that implement large distributed spiking neural networks using a variety of design techniques and technologies. This emerging field is characterized by its multidisciplinary nature and its focus on the physics of computation, driving innovations in theoretical neuroscience, device physics, electrical engineering, and computer science.

<b>38.1 The Origins</b> .....	715
<b>38.2 Neural and Neuromorphic Computing</b> ...	716
<b>38.3 The Importance of Fundamental Neuroscience</b> .....	717
<b>38.4 Temporal Dynamics in Neuromorphic Architectures</b> .....	718
<b>38.5 Synapse and Neuron Circuits</b> .....	719
38.5.1 Spike-Based Learning Circuits .....	720
<b>38.6 Spike-Based Multichip Neuromorphic Systems</b> .....	721
<b>38.7 State-Dependent Computation in Neuromorphic Systems</b> .....	722
<b>38.8 Conclusions</b> .....	722
<b>References</b> .....	723

### 38.1 The Origins

Models of neural information processing systems that link the type of information processing that takes place in the brain with theories of computation and computer science date back to the origins of computer science itself [38.1, 2]. The theory of computation based on abstract neural networks models was developed already in the 1950s [38.3, 4], and the development of artificial neural networks implemented on digital computers was very popular throughout the 1980s and the early 1990s [38.5–8]. Similarly, the history of implementing electronic models of neural circuits extends back to the construction of perceptrons in the late 1950s [38.3] and retinas in the early 1970s [38.9]. However, the modern wave of research utilizing very large scale integration technology and emphasizing the nonlinear current characteristics of the transistor to study and implement neural computation began only in the mid-1980s, with the collaboration that sprung up be-

tween scientists such as Max Delbrück, John Hopfield, Carver Mead, and Richard Feynman [38.10]. Inspired by graded synaptic transmission in the retina, Mead sought to use the graded (analog) properties of transistors, rather than simply operating them as on–off (digital) switches, to build circuits that emulate biological neural systems. He developed *neuromorphic* circuits that shared many common physical properties with proteic channels in neurons, and that consequently required far fewer transistors than digital approaches to emulating neural systems [38.11]. Neuromorphic engineering is the research field that was born out of this activity and which carries on that legacy: it takes inspiration from biology, physics, mathematics, computer science, and engineering to design artificial neural systems for carrying out robust and efficient computation using low power, massively parallel analog very large scale integration (VLSI) circuits, that operate with the same

*physics of computation* present in the brain [38.12]. Indeed, this young research field was born both out of the *Physics of Computation* course taught at Caltech by Carver Mead, John Hopfield, and Richard Feynman and with Mead's textbook *Analog Very Large Scale Integration and Neural Systems* [38.11]. Prominent in the early expansion of the field were scientists and engineers such as Christof Koch, Terry Sejnowski, Rodney Douglas, Andreas Andreou, Paul Mueller, Jan van der Spiegel, and Eric Vittoz, training a generation of cross-disciplinary students. Examples of successes in neuromorphic engineering range from the first biologically realistic silicon neuron [38.13], or realistic silicon models of the mammalian retina [38.14], to more recent silicon cochlea devices potentially useful for cochlear implants [38.15], or complex distributed multichip architectures for implementing event-driven autonomous behaving systems [38.16].

It is now a well-established field [38.17], with two flagship workshops (the Telluride Neuromorphic Engineering [38.18] and Capo Caccia Cognitive Neuromorphic Engineering [38.19] workshops) that are

currently still held every year. Neuromorphic circuits are now being investigated by many academic and industrial research groups worldwide to develop a new generation of computing technologies that use the same organizing principles of the biological nervous system [38.15, 20, 21]. Research in this field represents *frontier* research as it opens new technological and scientific horizons: in addition to basic science questions on the fundamental principles of computation used by the cortical circuits, neuromorphic engineering addresses issues in computer-science, and electrical engineering which go well beyond established frontiers of knowledge. A major effort is now being invested for understanding how these neuromorphic computational principles can be implemented using massively parallel arrays of basic computing elements (or *cores*), and how they can be exploited to create a new generation of computing technologies that takes advantage of future (nano)technologies and scaled VLSI processes, while coping with the problems of low-power dissipation, device unreliability, inhomogeneity, fault tolerance, etc.

## 38.2 Neural and Neuromorphic Computing

Neural computing (or neurocomputing) is concerned with the implementation of artificial neural networks for solving practical problems. Similarly, hardware implementations of artificial neural networks (neurocomputers) adopt mainly statistics and signal processing methods to solve the problem they are designed to tackle. These algorithms and systems are not necessarily tied to detailed models of neural or cortical processing. Neuromorphic computing on the other hand aims to reproduce the principles of neural computation by emulating as faithfully as possible the detailed biophysics of the nervous system in hardware. In this respect, one major characteristic of these systems is their use of *spikes* for representing and processing signals. This is not an end in itself: spiking neural networks represent a promising computational paradigm for solving complex pattern recognition and sensory processing tasks that are difficult to tackle using standard machine vision and machine learning techniques [38.22, 23]. Much research has been dedicated to software simulations of spiking neural networks [38.24], and a wide range of solutions have been proposed for solving real-world and engineering problems [38.25, 26]. Similarly, there are projects that focus on software

simulations of large-scale spiking neural networks for exploring the computational properties of models of cortical circuits [38.27, 28]. Recently, several research projects have been established worldwide to develop large-scale hardware implementations of spiking neural systems using VLSI technologies, mainly for allowing neuroscientists to carry out simulations and virtual experiments in real time or even faster than real-time scales [38.29–31]. Although dealing with hardware implementations of neural systems, either with custom VLSI devices or with dedicated computer architectures, these projects represent the conventional neurocomputing approaches, rather than neuromorphic-computing ones. Indeed, these systems are mainly concerned with fast and large simulations of spiking neural networks. They are optimized for speed and precision, at the cost of size and power consumption (which ranges from megawatts to kilowatts, depending on which approach is followed). An example of an alternative large-scale spiking neural network implementation that follows the original neuromorphic engineering principles (i. e., that exploits the characteristics of VLSI technology to directly emulate the biophysics and the connectivity of cortical circuits) is represented by the *Neurogrid* sys-

tem [38.32]. This system comprises an array of 16 VLSI chips, each integrating mixed analog neuromorphic neuron and synapse circuits with digital asynchronous event routing logic. The chips are assembled on a  $16.5 \times 19 \text{ cm}^2$  printed circuit board, and the whole system can model over one million neurons connected by billions of synapses in real time, and using only about  $\approx 3\text{W}$  of power [38.32].

Irrespective of the approach followed, these projects have two common goals: On one hand they aim to advance our understanding of neural processing in

the brain by developing models and physically building them using electronic circuits, and on the other they aim to exploit this understanding for developing a new generation of radically different non-von Neumann computing technologies that are inspired by neural and cortical circuits. In this interdisciplinary journey neuroscience findings will influence theoretical developments, and these will determine specifications and constraints for developing new neuromorphic circuits and systems that can implement them optimally.

### 38.3 The Importance of Fundamental Neuroscience

The neocortex is a remarkable computational device [38.33]. It is the neuronal structure in the brain that most expresses biology's ability to implement perception and cognition. Anatomical and neurophysiological studies have shown that the mammalian cortex with its laminar organization and regular microscopic structure has a surprisingly uniform architecture [38.34]. Since the original work of *Gilbert* and *Wiesel* [38.35] on the neural circuits of visual cortex it has been argued that this basic architecture, and its underlying computational principles computational principles can be understood in terms of the laminar distribution of relatively few classes of excitatory and inhibitory neurons [38.34]. Based on these slow, unreliable and inhomogeneous computing elements, the cortex easily outperform today's most powerful computers in a wide variety of computational tasks such as vision, audition, or motor control. Indeed, despite the remarkable progress in information and communication technology and the vast amount of resources dedicated to information and communication technology research and development, today's most fastest and largest computers are still not able to match neural systems, when it comes to carrying out robust computations in real-world tasks. The reasons for this performance gap are not yet fully understood, but it is clear that one fundamental difference between the two types of computing systems lies in the style of computation. Rather than using Boolean logic, precise digital representations, and clocked operations, nervous systems carry out robust and reliable computation using hybrid analog/digital unreliable components; they emphasize distributed, event driven, collective, and massively parallel mechanisms, and make extensive use of adaptation, self-organization and learning. Specifically, the patchy organization of the neurons in

the cortex suggests a computational machine where populations of neurons perform collective computation in individual clusters, transmit the results of this computation to neighboring clusters, and set the local context of the cluster by means of feedback connections from/to other relevant cortical areas. This overall graphical architecture resembles graphical processing models that perform Bayesian inference [38.36, 37]. However, the theoretical knowledge for designing and analyzing these models is limited mainly to graphs without loops, while the cortex is characterized by massive recurrent (loopy) connectivity schemes. Recent studies exploring loopy graphical models related to cortical architectures started to emerge [38.33, 38], but issues of convergence and accuracy remain unresolved, hardware implementations in a cortical architectures composed of spiking neurons have not been addressed yet.

Understanding the fundamental computational principles used by the cortex, how they are exploited for processing, and how to implement them in hardware, will allow us to develop radically novel computing paradigms and to construct a new generation of information and communication technology that combine the strengths of silicon technology with the performance of brains. Indeed fundamental research in neuroscience has already made substantial progress in uncovering these principles, and information and communication technologies have advanced to a point where it is possible to integrate almost as many transistors in a VLSI system as neurons in a brain. From the theoretical standpoint of view, it has been demonstrated that any Turing machine, and hence any conceivable digital computation, can be implemented by a noise-free network of spiking neurons [38.39]. It has also been

shown that networks of spiking neurons can carry out a wide variety of complex state-dependent computations, even in the presence of noise [38.40–44]. However, apart from isolated results, a general insight into which computations can be carried out in a robust manner by networks of unreliable spiking elements is still missing. Current proposals in state-of-the-art computational and theoretical neuroscience research represent mainly approximate functional models and are implemented as abstract artificial neural networks [38.45, 46]. It is less clear how these functions are realized by the actual networks of neocortex [38.34], how these networks are interconnected locally, and how percep-

tual and cognitive computations can be supported by them.

Both additional neurophysiological studies on neuron types and quantitative descriptions of local and inter-areal connectivity patterns are required to determine the specifications for developing the neuromorphic VLSI analogs of the cortical circuits studied, and additional computational neuroscience and neuromorphic engineering studies are required to understand what level of detail to use in implementing spiking neural networks, and what formal methodology to use for synthesizing and programming these non-von Neumann computational architectures.

## 38.4 Temporal Dynamics in Neuromorphic Architectures

Neuromorphic spiking neural network architectures typically comprise massively parallel arrays of simple processing elements with memory and computation co-localized (Fig. 38.1). Given their architectural constraints, these neural processing systems cannot process signals using the same strategies used by the conventional von Neumann computing architectures, such as digital signal processor or central processing unit, that time-domain multiplex small numbers of highly complex processors at high clock rates and operate by transferring the partial results of the computation from and to external memory banks. The synapses and neurons in these architectures have to process input spikes and produce output responses as the input signals arrive, in real time, at the rate of the incoming data. It is not possible to virtualize time and transfer partial results in memory banks outside the architecture core, at higher rates. Rather it is necessary to employ resources that compute with time constants that are well matched to those of the signals they are designed to process. Therefore, to interact with the environment and process signals with biological timescales efficiently, hardware neuromorphic systems need to be able to compute using biologically realistic time constants. In this way, they are well matched to the signals they process, and are inherently synchronized with the real world events.

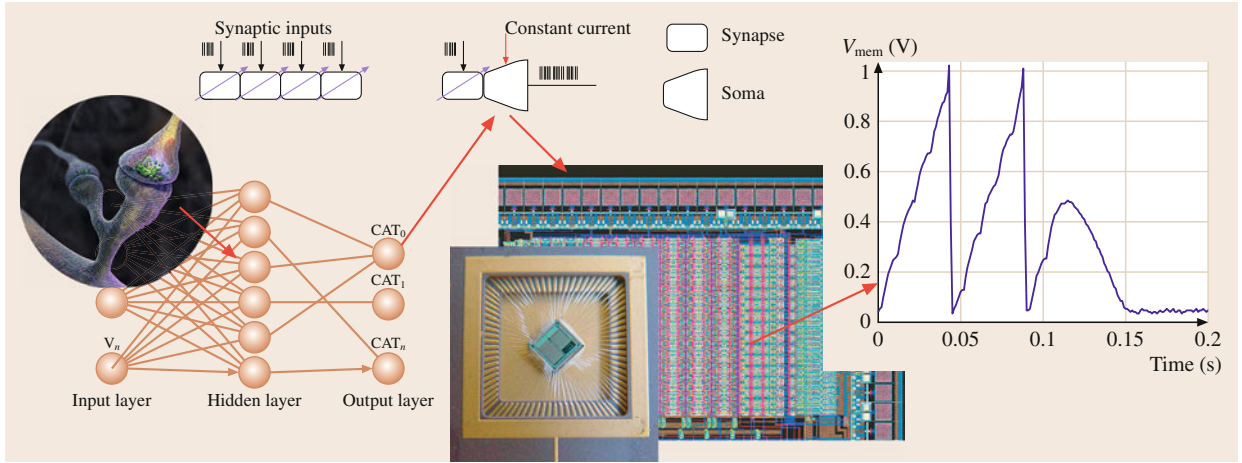
This constraint is not easy to satisfy using analog VLSI technology. Standard analog circuit design techniques either lead to bulky and silicon-area expensive solutions [38.47] or fail to meet this condition, resorting to modeling neural dynamics at *accelerated* unrealistic timescales [38.48–50].

One elegant solution to this problem is to use current-mode design techniques [38.51] and log-domain circuits operated in the weak-inversion regime [38.52]. When metal oxide semiconductor field effect transistors are operated in this regime, the main mechanism of carrier transport is that of diffusion, as it is for ions flowing through proteic channels across neuron membranes. In general, neuromorphic VLSI circuits operate in this domain (also known as the *subthreshold* regime), and this is why they share many common physical properties with proteic channels in neurons [38.52]. For example, metal oxide semiconductor field effect transistor have an exponential relationship between gate-to-source voltage and drain current, and produce currents that range from femto- to nanoampere resolution. In this domain, it is therefore possible to integrate relatively small capacitors in VLSI circuits, to implement temporal filters that are both compact and have biologically realistic time constants, ranging from tens to hundreds of milliseconds.

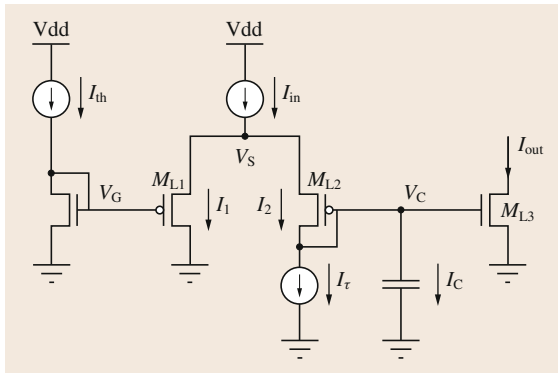
A very compact subthreshold log-domain circuit that can reproduce biologically plausible temporal dynamics is the differential pair integrator circuit [38.53], shown in Fig. 38.2. It can be shown, by log-domain circuit analysis techniques [38.54, 55] that the response of this circuit is governed by the following first-order differential equation

$$\tau \left( 1 + \frac{I_{th}}{I_{out}} \right) \frac{d}{dt} I_{out} + I_{out} = \frac{I_{th} I_{in}}{I_{\tau}} - I_{th}, \quad (38.1)$$

where the time constant  $\tau \triangleq CU_T/\kappa I_{\tau}$ , the term  $U_T$  represents the thermal voltage and  $\kappa$  the subthreshold slope factor [38.52].



**Fig. 38.1** Neuromorphic spiking neural network architectures: detailed biophysical models of cortical circuits are derived from neuroscience experiments; neural networks models are designed, with realistic spiking neurons and dynamic synapses; these are mapped into analog circuits, and integrated in large numbers on VLSI chips. Input spikes are integrated by synaptic circuits, which drive their target postsynaptic neurons, which in turn integrate all synaptic inputs and generate action potentials. Spikes of multiple neurons are transmitted off chip using asynchronous digital circuits, to eventually control in real-time autonomous behaving systems



**Fig. 38.2** Schematic diagram of neuromorphic integrator circuit. Input currents are integrated in time to produce output currents with large time constants, and with a tunable gain factor

Although this first-order *nonlinear* differential equation cannot be solved analytically, for sufficiently large input currents ( $I_{in} \gg I_{\tau}$ ) the term  $-I_{th}$  on the right-hand side of (38.1) becomes negligible, and eventually when the condition  $I_{out} \gg I_{th}$  is met, the equation can be well approximated by

$$\tau \frac{d}{dt} I_{out} + I_{out} = \frac{I_{in} I_{th}}{I_{\tau}}. \quad (38.2)$$

Under the reasonable assumptions of nonnegligible input currents, this circuit implements therefore a compact linear integrator with time constants that can be set to range from microseconds to hundreds of milliseconds. It is a circuit that can be used to build neuromorphic sensory systems that interact with the environment [38.56], and most importantly, is a circuit that reproduces faithfully the dynamics of synaptic transmission observed in biological synapses [38.57].

## 38.5 Synapse and Neuron Circuits

Synapses are fundamental elements for computation and information transfer in both real and artificial neural systems. They play a crucial role in neural coding and learning algorithms, as well as in neuromorphic neural network architectures. While modeling the nonlinear properties and the dynamics of real synapses

can be extremely onerous for software simulations in terms of computational power, memory requirements, and simulation time, neuromorphic synapse circuits can faithfully reproduce synaptic dynamics using integrators such as the differential pair integrator shown in Fig. 38.2. The same differential pair integrator cir-

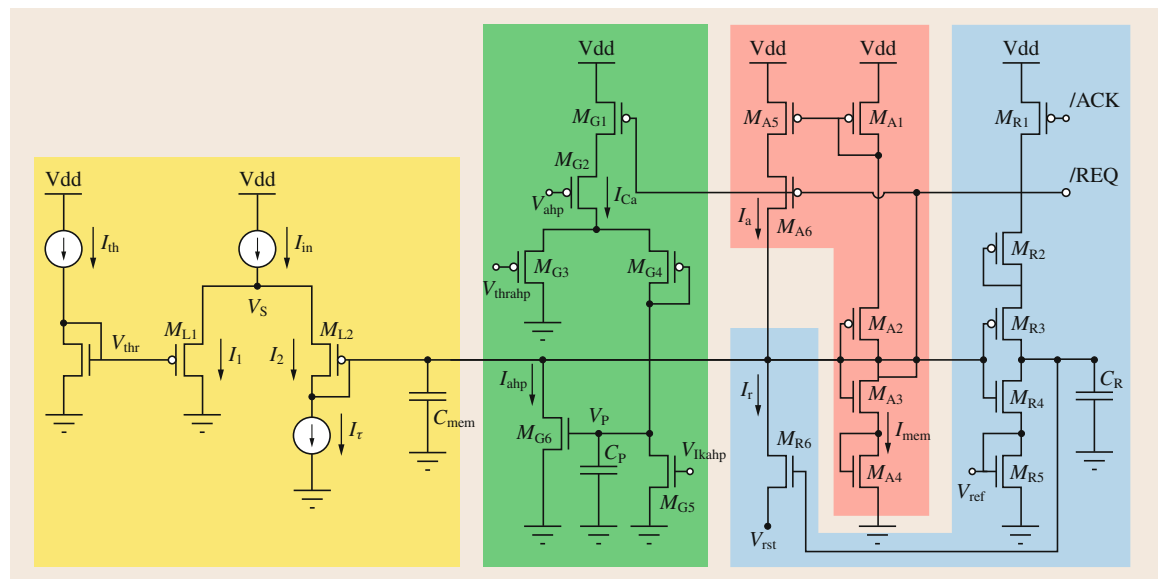
circuit can be used to model the passive leak and conductance behavior in silicon neurons. An example of a silicon neuron circuit that incorporated the differential pair integrator is shown in Fig. 38.3. This circuit implements an *adaptive exponential integrate-and-fire neuron* model [38.58]. In addition to the conductance-based behavior, it implements a spike-frequency adaptation mechanisms, a positive feedback mechanism that models the effect of sodium activation and inactivation channels, and a reset mechanism with a free parameter that can be used to set the neuron's reset potential. The neuron's input differential pair integrator integrates the input current until it approaches the neuron's threshold voltage. As the positive feedback circuit gets activated, it induces an exponential rise in the variable that represents the model neuron membrane potential, which in the circuit of Fig. 38.3 is the current  $I_{\text{mem}}$ . This quickly causes the neuron to produce an action potential and make a request for transmitting a spike (i. e., the *REQ* signal of Fig. 38.3 is activated). Once the digital request signal is acknowledged, the membrane capacitance  $C_{\text{mem}}$  is reset to the neuron's tunable reset potential  $V_{\text{rst}}$ . These types of neuron circuits have been shown to be extremely low power, consuming about 7 pJ per

spike [38.59]. In addition, the circuit is extremely compact compared to alternative designs [38.58], while still being able to reproduce realistic dynamics.

As synapse and neuron circuits integrate their corresponding input signals in parallel, the neural network emulation time does not depend on the number of elements involved, and the network response always happen in real time. These circuits can be therefore used to develop low-power large-scale hardware neural architectures, for signal processing and general purpose computing [38.58]

### 38.5.1 Spike-Based Learning Circuits

As large-scale very large scale integration (VLSI) networks of spiking neurons are becoming realizable, the development of robust spike-based learning methods, algorithms, and circuits has become crucial. Spike-based learning mechanisms enable the hardware neural systems they are embedded in to adapt to the statistics of their input signals, to learn and classify complex sequences of spatiotemporal patterns, and eventually to implement general purpose state-dependent computing paradigms. Biologically plausible spike-driven



**Fig. 38.3** Schematic diagrams of a conductance-based integrate-and-fire neuron. An input differential pair integrator low-pass filter ( $M_{L1-3}$ ) implements the neuron leak conductance. A noninverting amplifier with current-mode positive feedback ( $M_{A1-6}$ ) produces address events at extremely low-power operation. A reset block ( $M_{R1-6}$ ) resets the neuron to the reset voltage  $V_{rst}$  and keeps it reset for a refractory period, set by the  $V_{ref}$  bias voltage. An additional differential pair integrator low-pass filter ( $M_{G1-6}$ ) integrates the output events in a negative feedback loop, to implement a *spike-frequency adaptation* mechanism



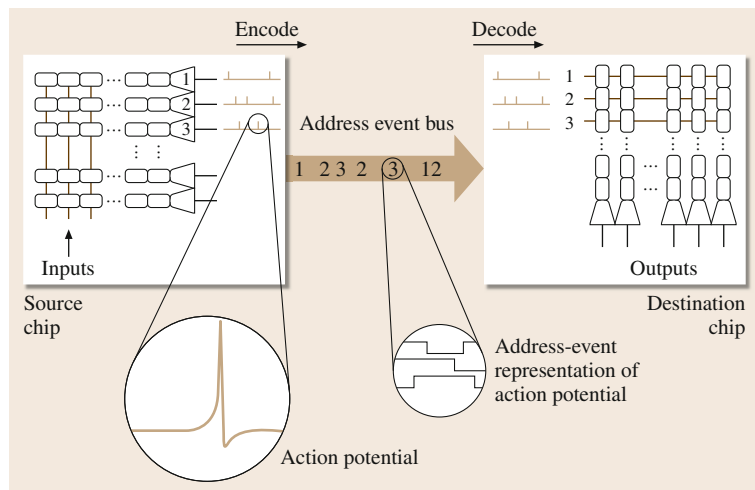
synaptic plasticity mechanisms have been thoroughly investigated in recent years. It has been shown, for example, how spike-timing dependent plasticity (STDP) can be used to learn to encode temporal patterns of spikes [38.42, 60, 61]. In spike-timing dependent plasticity the relative timing of pre- and postsynaptic spikes determine how to update the efficacy of a synapse. Plasticity mechanisms based on the timing of the spikes map very effectively onto silicon neuromorphic devices, and so a wide range of spike-timing dependent plasticity models have been implemented in VLSI [38.62–67]. It is therefore possible to build

large-scale neural systems that can carry out signal processing and neural computation, and include adaptation and learning. These types of systems are, by their very own nature, modular and scalable. It is possible to develop very large scale systems by designing basic neural processing *cores*, and by interconnecting them together [38.68]. However, to interconnect multiple neural network chips among each other, or to provide sensory inputs to them, or to interface them to conventional computers or robotic platforms, it is necessary to develop efficient spike-based communication protocols and interfaces.

## 38.6 Spike-Based Multichip Neuromorphic Systems

In addition to using spikes for signal efficient processing and computations, neuromorphic systems can use spiking representations also for efficient communication. The use of asynchronous spike- or event-based representations in electronic systems can be energy efficient and fault tolerant, making them ideal for building modular systems and creating complex hierarchies of computation. In recent years, a new class of neuromorphic multichip systems started to emerge [38.69–71]. These systems typically comprise one or more neuromorphic sensors, interfaced to general-purpose neural network chips comprising spiking silicon neurons and dynamic synapses. The strategy used to transmit signals across chip boundaries in these types of systems is based on asynchronous *address-events*: output events are represented by the addresses of the neurons that spiked, and transmitted in real time on a digital bus

(Fig. 38.4). The communication protocol used by these systems is commonly referred to as address event representation [38.72, 73]. The analog nature of the AER (address event representation) signals being transmitted is encoded in the mean frequency of the neurons spikes (spike rates) and in their precise timing. Both types of representations are still an active topic of research in neuroscience, and can be investigated in real time with these hardware systems. Once on a digital bus, the address events can be translated, converted or remapped to multiple destinations using the conventional logic and memory elements. Digital address event representation infrastructures allow us to construct large multichip networks with arbitrary connectivity, and to seamlessly reconfigure the network topology. Although digital, the asynchronous real-time nature of the AER protocol poses significant technological challenges that are still



**Fig. 38.4** Asynchronous communication scheme between two chips: when a neuron on the source chip generates an action potential, its address is placed on a common digital bus. The receiving chip decodes the address events and routes them to the appropriate synapses

being actively investigated by the electrical engineering community [38.74]. But by using analog processing in the neuromorphic *cores* and asynchronous digital communication outside them, neuromorphic systems can exploit the best of both worlds, and implement compact

low-power brain inspired neural processing systems that can interact with the environment in real time, and represent an alternative (complementary) computing technology to the more common and the conventional VLSI computing architectures.

## 38.7 State-Dependent Computation in Neuromorphic Systems

General-purpose cortical-like computing architectures can be interfaced to real-time autonomous behaving systems to process sensory signals and carry out event-driven state-dependent computation in real time. However, while the circuit design techniques and technologies for implementing these neuromorphic systems are becoming well established, formal methodologies for *programming* them, to execute specific procedures and solve user defined tasks, do not exist yet. A first step toward this goal is the definition of methods and procedures for implementing state-dependent computation in networks of spiking neurons. In general, state-dependent computation in autonomous behaving systems has been a challenging research field since the advent of digital computers. Recent theoretical findings and technological developments show promising results in this domain [38.16, 43, 44, 75, 76]. But the computational tasks that these systems are currently able to perform remain rather simple, compared to what can be achieved by humans, mammals, and many other animal species. We know, for instance, that nervous systems can exhibit context-dependent behavior, can execute *programs* consisting of series of flexible iterations, and can conditionally branch to alternative behaviors. A general understanding of how to configure artificial neural systems to achieve this sophistication of processing, including also adaptation, autonomous learning, interpretation of ambiguous input signals, symbolic manipulation, inference, and other characteristics that we could regard as effective cognition is still missing. But progress is being made in this direction by studying the computa-

tional properties of spiking neural networks configured as *attractors* or winner-take-all networks [38.33, 44, 77]. When properly configured, these architectures produce persistent activities, which can be regarded as computational states. Both software and VLSI event-driven soft-winner-take-all architectures are being developed to couple spike-based computational models among each other, using the asynchronous communication infrastructure, and use them to investigate their computational properties as neural finite-state machines in autonomous behaving robotic platforms [38.44, 78].

The theoretical, modeling, and VLSI design interdisciplinary activities is carried out with tight interactions, in an effort to understand:

1. How to use the analog, unreliable, and low-precision silicon neurons and synapse circuits operated in the weak-inversion regime [38.52] to carry out reliable and robust signal processing and pattern recognition tasks;
2. How to compose networks of such elements and how to embody them in real-time behaving systems for implementing sets of prespecified desired functionalities and behaviors; and
3. How to formalize these theories and techniques to develop a systematic methodology for configuring these networks and systems to achieve arbitrary state-dependent computations, similar to what is currently done using high-level programming languages such as Java or C++ for conventional digital architectures.

## 38.8 Conclusions

In this chapter, we presented an overview of the neuromorphic engineering field, focusing on very large scale integration implementations of spiking neural networks and on multineuron chips that comprise synapses and neurons with biophysically realistic dynamics, nonlinear properties, and spike-based plasticity mechanisms. We argued that the multineuron chips built using these

silicon neurons and synaptic circuits can be used to implement an alternative brain inspired computational paradigm that is complementary to the conventional ones based on von Neumann architectures.

Indeed, the field of neuromorphic engineering has been very successful in developing a new generation of computing technologies implemented with design prin-



ciples based on those of the nervous systems, and which exploit the physics of computation used in biological neural systems. It is now possible to design and implement complex large-scale artificial neural systems with elaborate computational properties, such as spike-based plasticity and soft winner-take-all behavior, or even complete artificial sensory-motor systems, able to robustly process signals in real time using neuromorphic VLSI technology.

Within this context, neuromorphic VLSI technology can be extremely useful for exploring neural processing strategies in real time. While there are clear advantages of this technology, for example, in terms of power budget and size requirements, there are also restrictions and limitations imposed by the hardware implementations that limit their possible range of applications. These constraints include for example limited resolution in the state variables or bounded parameters (e.g., bounded synaptic weights that cannot grow indefinitely or become negative). Also the presence of noise and inhomogeneities in all circuit components, place severe limitations on the precision and reliability of the computations performed. However, most, if not all, the limitations that neuromorphic hardware implementations face, (e.g., in maintaining stability, in achieving robust computation using unreliable components, etc.)

are often the same one faced by real neural systems. So these limitations are useful for reducing the space of possible artificial neural models that explain or reproduce the properties of real cortical circuits. While in principle these *features* could be simulated also in software (e.g., by adding a noise term to each state variable, or restricting the resolution of variables to 3, 4, or 6 bits instead of using the floating point representations), they are seldom taken into account. So in addition to representing a technology useful for implementing hardware neural processing systems and solving practical applications, neuromorphic circuits can be used as an additional tool for studying and understanding basic neuroscience.

As VLSI technology is widespread and readily accessible, it is possible to easily learn (and train new generations of students) to design neuromorphic VLSI neural networks for building hardware models of neural systems and sensory-motor systems. Understanding how to build real-time behaving neuromorphic systems that can work in real-world scenarios, will allow us to both gain a better understanding of the principles of neural coding in the nervous system, and develop a new generation of computing technologies that extend and that complement current digital computing devices, circuits, and architectures.

## References

- 38.1 W.S. McCulloch, W. Pitts: A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5**, 115–133 (1943)
- 38.2 J. von Neumann: *The Computer and the Brain* (Yale Univ. Press, New Haven 1958)
- 38.3 F. Rosenblatt: The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol. Rev.* **65**(6), 386–408 (1958)
- 38.4 M.L. Minsky: *Computation: Finite and Infinite Machines* (Prentice-Hall, Upper Saddle River 1967)
- 38.5 J.J. Hopfield: Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA* **79**(8), 2554–2558 (1982)
- 38.6 D.E. Rumelhart, J.L. McClelland: Foundations, parallel distributed processing. In: *Explorations in the Microstructure of Cognition*, ed. by D.E. Rumelhart, J.L. McClelland (MIT, Cambridge 1986)
- 38.7 T. Kohonen: *Self-Organization and Associative Memory*, Springer Series in Information Sciences, 2nd edn. (Springer, Berlin Heidelberg 1988)
- 38.8 J. Hertz, A. Krogh, R.G. Palmer: *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading 1991)
- 38.9 K. Fukushima, Y. Yamaguchi, M. Yasuda, S. Nagata: An electronic model of the retina, *Proc. IEEE* **58**(12), 1950–1951 (1970)
- 38.10 T. Hey: Richard Feynman and computation, *Contemp. Phys.* **40**(4), 257–265 (1999)
- 38.11 C.A. Mead: *Analog VLSI and Neural Systems* (Addison-Wesley, Reading 1989)
- 38.12 C. Mead: Neuromorphic electronic systems, *Proc. IEEE* **78**(10), 1629–1636 (1990)
- 38.13 M. Mahowald, R.J. Douglas: A silicon neuron, *Nature* **354**, 515–518 (1991)
- 38.14 M. Mahowald: The silicon retina, *Sci. Am.* **264**, 76–82 (1991)
- 38.15 R. Sarpeshkar: Brain power – borrowing from biology makes for low power computing – bionic ear, *IEEE Spectrum* **43**(5), 24–29 (2006)
- 38.16 R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, A. Linares-Barranco, G. Jiménez-Moreno, A. Civit-Balcells, B. Linares-Barranco: Spike events processing for vision systems, *Int. Symp. Circuits Syst. (ISCAS, Piscataway)* (2007)
- 38.17 G. Indiveri, T.K. Horiuchi: Frontiers in neuromorphic engineering, *Front. Neurosci.* **5**(118), 1–2 (2011)
- 38.18 Telluride neuromorphic cognition engineering workshop, <http://ine-web.org/workshops/workshops-overview>
- 38.19 The Capo Caccia Workshops toward Cognitive Neuromorphic Engineering. <http://capocaccia.ethz.ch>.

- 38.20 K.A. Boahen: Neuromorphic microchips, *Sci. Am.* **292**(5), 56–63 (2005)
- 38.21 R.J. Douglas, M.A. Mahowald, C. Mead: Neuro-morphic analogue VLSI, *Annu. Rev. Neurosci.* **18**, 255–281 (1995)
- 38.22 W. Maass, E.D. Sontag: Neural systems as nonlinear filters, *Neural Comput.* **12**(8), 1743–1772 (2000)
- 38.23 A. Belatreche, L.P. Maguire, M. McGinnity: Advances in design and application of spiking neural networks, *Soft Comput.* **11**(3), 239–248 (2006)
- 38.24 R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J.M. Bower, M. Diesmann, A. Morrison, P.H. Harris Jr., F.C. Goodman, M. Zirpe, T. Natschlager, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A.P. Davison, S. El Boustani, A. Destexhe: Simulation of networks of spiking neurons: A review of tools and strategies, *J. Comput. Neurosci.* **23**(3), 349–398 (2007)
- 38.25 J. Brader, W. Senn, S. Fusi: Learning real world stimuli in a neural network with spike-driven synaptic dynamics, *Neural Comput.* **19**, 2881–2912 (2007)
- 38.26 P. Rowcliffe, J. Feng: Training spiking neuronal networks with applications in engineering tasks, *IEEE Trans. Neural Netw.* **19**(9), 1626–1640 (2008)
- 38.27 *The Blue Brain Project*. EPFL website. (2005) <http://bluebrain.epfl.ch/>
- 38.28 E. Izhikevich, G. Edelman: Large-scale model of mammalian thalamocortical systems, *Proc. Natl. Acad. Sci. USA* **105**, 3593–3598 (2008)
- 38.29 *Brain-Inspired Multiscale Computation in Neuromorphic Hybrid Systems (BrainScaleS)*. FP7 269921 EU Grant 2011–2015
- 38.30 Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SynAPSE). US Darpa Initiative (<http://www.darpa.mil/dso/solicitations/baa08-28.html>) (2009)
- 38.31 R. Freidman: Reverse engineering the brain, *Biomed. Comput. Rev.* **5**(2), 10–17 (2009)
- 38.32 B.V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J.M. Bussat, R. Alvarez-Icaza, J.V. Arthur, P.A. Merolla, K. Boahen: Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations, *Proc. IEEE* **102**(5), 699–716 (2014)
- 38.33 R.J. Douglas, K. Martin: Recurrent neuronal circuits in the neocortex, *Curr. Biol.* **17**(13), R496–R500 (2007)
- 38.34 R.J. Douglas, K.A.C. Martin: Neural circuits of the neocortex, *Annu. Rev. Neurosci.* **27**, 419–451 (2004)
- 38.35 C.D. Gilbert, T.N. Wiesel: Clustered intrinsic connections in cat visual cortex, *J. Neurosci.* **3**, 1116–1133 (1983)
- 38.36 G.F. Cooper: The computational complexity of probabilistic inference using bayesian belief networks, *Artif. Intell.* **42**(2/3), 393–405 (1990)
- 38.37 D.J.C. MacKay: *Information Theory, Inference and Learning Algorithms* (Cambridge Univ. Press, Cambridge 2003)
- 38.38 A. Steimer, W. Maass, R. Douglas: Belief propagation in networks of spiking neurons, *Neural Comput.* **21**, 2502–2523 (2009)
- 38.39 W. Maass: On the computational power of winner-take-all, *Neural Comput.* **12**(11), 2519–2535 (2000)
- 38.40 W. Maass, P. Joshi, E.D. Sontag: Computational aspects of feedback in neural circuits, *PLOS Comput. Biol.* **3**(1), 1–20 (2007)
- 38.41 L.F. Abbott, W.G. Regehr: Synaptic computation, *Nature* **431**, 796–803 (2004)
- 38.42 R. Güti, H. Sompolinsky: The tempotron: A neuron that learns spike timing-based decisions, *Nat. Neurosci.* **9**, 420–428 (2006)
- 38.43 T. Wennekers, N. Ay: Finite state automata resulting from temporal information maximization and a temporal learning rule, *Neural Comput.* **10**(17), 2258–2290 (2005)
- 38.44 U. Rutishauser, R. Douglas: State-dependent computation using coupled recurrent networks, *Neural Comput.* **21**, 478–509 (2009)
- 38.45 P. Dayan, L.F. Abbott: *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT, Cambridge 2001)
- 38.46 M. Arbib (Ed.): *The Handbook of Brain Theory and Neural Networks*, 2nd edn. (MIT, Cambridge 2002)
- 38.47 G. Rachmuth, H.Z. Shouval, M.F. Bear, C.-S. Poon: A biophysically-based neuromorphic model of spike rate- and timing-dependent plasticity, *Proc. Natl. Acad. Sci. USA* **108**(49), E1266–E1274 (2011)
- 38.48 J. Schemmel, D. Brüderle, K. Meier, B. Ostendorf: Modeling synaptic plasticity within networks of highly accelerated I & F neurons, *Int. Symp. Circuits Syst. (ISCAS, Piscataway)* (2007) pp. 3367–3370
- 38.49 J.H.B. Wijekoon, P. Dudek: Compact silicon neuron circuit with spiking and bursting behaviour, *Neural Netw.* **21**(2/3), 524–534 (2008)
- 38.50 D. Brüderle, M.A. Petrovici, B. Vogginger, M. Ehrlich, T. Pfeil, S. Millner, A. Grübl, K. Wendt, E. Müller, M.-O. Schwartz, D.H. de Oliveira, S. Jeltsch, J. Fieres, M. Schilling, P. Müller, O. Breitwieser, V. Petkov, L. Muller, A.P. Davison, P. Krishnamurthy, J. Kremkow, M. Lundqvist, E. Muller, J. Partzsch, S. Scholze, L. Zühl, C. Mayr, A. Destexhe, M. Diesmann, T.C. Potjans, A. Lansner, R. Schüffny, J. Schemmel, K. Meier: A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems, *Biol. Cybern.* **104**(4), 263–296 (2011)
- 38.51 C. Tomazou, F.J. Lidgely, D.G. Haigh (Eds.): *Analogue IC Design: The Current-Mode Approach* (Peregrinus, Stevenage, Herts., UK 1990)
- 38.52 S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, R.J. Douglas: *Analog VLSI: Circuits and Principles* (MIT Press, Cambridge 2002)

- 38.53 C. Bartolozzi, G. Indiveri: Synaptic dynamics in analog VLSI, *Neural Comput.* **19**(10), 2581–2603 (2007)
- 38.54 E.M. Drakakis, A.J. Payne, C. Toumazou: Log-domain state-space: A systematic transistor-level approach for log-domain filtering, *IEEE Trans. Circuits Syst. II* **46**(3), 290–305 (1999)
- 38.55 D.R. Frey: Log-domain filtering: An approach to current-mode filtering, *IEE Proc G* **140**(6), 406–416 (1993)
- 38.56 S.-C. Liu, T. Delbruck: Neuromorphic sensory systems, *Curr. Opin. Neurobiol.* **20**(3), 288–295 (2010)
- 38.57 A. Destexhe, Z.F. Mainen, T.J. Sejnowski: Kinetic models of synaptic transmission. In: *Methods in Neuronal Modelling, from Ions to Networks*, ed. by C. Koch, I. Segev (MIT Press, Cambridge 1998) pp. 1–25
- 38.58 G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, K. Boahen: Neuromorphic silicon neuron circuits, *Front. Neurosci.* **5**, 1–23 (2011)
- 38.59 P. Livi, G. Indiveri: A current-mode conductance-based silicon neuron for address-event neuromorphic systems, *Int. Symp. Circuits Syst. (ISCAS)* (2009) pp. 2898–2901
- 38.60 L.F. Abbott, S.B. Nelson: Synaptic plasticity: Taming the beast, *Nat. Neurosci.* **3**, 1178–1183 (2000)
- 38.61 R.A. Legenstein, C. Näger, W. Maass: What can a neuron learn with spike-timing-dependent plasticity?, *Neural Comput.* **17**(11), 2337–2382 (2005)
- 38.62 S.A. Bamford, A.F. Murray, D.J. Willshaw: Spike-timing-dependent plasticity with weight dependence evoked from physical constraints, *IEEE Trans. Biomed. Circuits Syst.* **6**(4), 385–398 (2012)
- 38.63 S. Mitra, S. Fusi, G. Indiveri: Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI, *IEEE Trans. Biomed. Circuits Syst.* **3**(1), 32–42 (2009)
- 38.64 G. Indiveri, E. Chicca, R.J. Douglas: A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity, *IEEE Trans. Neural Netw.* **17**(1), 211–221 (2006)
- 38.65 A. Bofill, I. Petit, A.F. Murray: Synchrony detection and amplification by silicon neurons with STDP synapses, *IEEE Trans. Neural Netw.* **15**(5), 1296–1304 (2004)
- 38.66 S. Fusi, M. Annunziato, D. Badoni, A. Salamon, D.J. Amit: Spike-driven synaptic plasticity: Theory, simulation, VLSI implementation, *Neural Comput.* **12**, 2227–2258 (2000)
- 38.67 P. Häfliger, M. Mahowald: Weight vector normalization in an analog VLSI artificial neuron using a backpropagating action potential. In: *Neuromorphic Systems: Engineering Silicon from Neurobiology*, ed. by L.S. Smith, A. Hamilton (World Scientific, London 1998) pp. 191–196
- 38.68 P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, A. Chandra, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S.K. Esser, R. Appuswamy, B. Taba, A. Amir, M.D. Flickner, W.P. Risk, R. Manohar, D.S. Modha: A million spiking-neuron integrated circuit with a scalable communication network and interface, *Science* **345**(6197), 668–673 (2014)
- 38.69 R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civi-Ballcells, T. Serrano-Gotarredona, A.J. Acosta-Jiménez, B. Linares-Barranco: CAVIAR: A 45k neuron, 5M synapse, 12G connects/s aer hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking, *IEEE Trans. Neural Netw.* **20**(9), 1417–1438 (2009)
- 38.70 E. Chicca, A.M. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R.J. Douglas, G. Indiveri: A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity, *IEEE Trans. Circuits Syst. I* **5**(54), 981–993 (2007)
- 38.71 T.Y.W. Choi, P.A. Merolla, J.V. Arthur, K.A. Boahen, B.E. Shi: Neuromorphic implementation of orientation hypercolumns, *IEEE Trans. Circuits Syst. I* **52**(6), 1049–1060 (2005)
- 38.72 M. Mahowald: *An Analog VLSI System for Stereoscopic Vision* (Kluwer, Boston 1994)
- 38.73 K.A. Boahen: Point-to-point connectivity between neuromorphic chips using address-events, *IEEE Trans. Circuits Syst. II* **47**(5), 416–434 (2000)
- 38.74 A.J. Martin, M. Nystrom: Asynchronous techniques for system-on-chip design, *Proc. IEEE* **94**, 1089–1120 (2006)
- 38.75 G. Schoner: Dynamical systems approaches to cognition. In: *Cambridge Handbook of Computational Cognitive Modeling*, ed. by R. Sun (Cambridge Univ. Press, Cambridge 2008) pp. 101–126
- 38.76 G. Indiveri, E. Chicca, R.J. Douglas: Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition, *Cogn. Comput.* **1**, 119–127 (2009)
- 38.77 M. Giulioni, P. Camilleri, M. Mattia, V. Dante, J. Braun, P. Del Giudice: Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI, *Front. Neurosci.* **5**, 1–16 (2011)
- 38.78 E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, R. Douglas: Synthesizing Cognition in neuromorphic electronic Systems, *Proc. Natl. Acad. Sci. USA* **110**(37), E3468–E3476 (2013)