Neuromorphic Systems
G Indiveri, S-C Liu, T Delbrück, and R Douglas,
University of Zurich, Zurich, Switzerland
© 2009 Elsevier Ltd. All rights reserved.

Neuromorphic Engineering
Neuromorphic engineering is concerned with the design and fabrication of artificial neural systems whose architecture and design principles are based on those of biological nervous systems. Neuromorphic systems of neurons and synapses can be implemented in the electronic medium CMOS (complementary metal oxide semiconductor) using hybrid analog/digital VLSI (very large-scale integrated) technology.

The notion of neuromorphic engineering has appeared in various forms during the last few centuries: From the French and Swiss mechanical automata of the eighteenth to nineteenth centuries (e.g., Vaucanson’s duck) through the biological robots of the 1950s (e.g., Grey Walter’s turtle) to the early neural electrical circuits of the 1960s and 1970s (e.g., Fukushima in Japan, Mueller at the University of Pennsylvania, and Lettvin at MIT). The concept took root again at Caltech during the mid-1980s, this time in the research of Carver Mead, who had already made major conceptual contributions to the design and construction of digital VLSI systems. He recognized that the use of transistors for computation had changed very little from the time when John Von Neumann first proposed the architecture for the programmable serial computer. The bit-perfect, strictly synchronous, and largely serial processing strategy required by present computers is untenable in the long run because it does not scale. Moreover, this design implies an exponentially greater power consumption for larger systems and exponentially higher costs of constructing, testing, and maintaining elements that must be fault-free in operation.

The design of biological neural computation is very different from that of modern computers. Neuronal networks process information using energy-efficient, asynchronous, event-based methods. Biology uses self-construction, self-repair, and self-programming, and it has learned how to flexibly compose complex behaviors from simpler elements. Of course, these biological abilities are not yet understood. But they offer an attractive alternative to conventional technology and have enormous consequences for future artificial information processing and behavior systems.

The challenge for neuromorphic engineering is to explore the methods of biological information processing in a practical electrical engineering context. Can existing CMOS VLSI technology be deployed in a novel way that reflects more closely the neuronal approach to computation? And can we thereby reap some of the computational facility, speed, and efficiency that biological nervous systems bring to the solution of real-world problems? Conversely, can we, by engineering artifacts, gain new insights into how biology organizes computation?

Digital and Analog in Neuromorphic VLSI Systems
The majority of integrated circuits represent numbers as binary digits. Binary digits are used because it is possible to standardize the behavior of transistors so that their state can be determined reliably to a single bit of accuracy. The reliable bits can then be combined to encode variables to an arbitrarily high precision. It is the combination of this precision and the synchronous, serial model of processing that has been the foundation of the long boom in digital electronics. One unnecessary consequence of this exuberance is that modern computers use deterministic high precision to deal even with real-world tasks whose variables can often be encoded with only a few bits.

For many such problems, particularly those in which the input data are ill-conditioned and the computation can be specified in a relative matter, biological solutions are many orders of magnitude more effective than those we have been able to implement using digital methods. This advantage is due principally to biology’s use of elementary physical phenomena as computational primitives and to the representation of information by the relative values of analog signals rather than by the absolute values of digital signals. Typically, it is this style of processing that neuromorphic engineers explore. Their systems are large collections of communicating computational primitives implemented either in analog or, more commonly, in hybrid analog–digital circuits.

Analog computing has a number of interesting features. First, analog processors are inherently more dense than digital ones because the individual electrical nodes of analog circuits can represent multiple bits of information. Second, unlike conventional digital circuits, whose coordination is governed by a global clock, analog circuits are naturally synchronized in physical time (as opposed to being synchronized against clock time). This property is particularly useful for systems that model and interact with real-world processes because, if the time
constants of the analog circuits are appropriately set, their processing can be scaled to match those of the real-world processes with which they interact.

Unfortunately, an analog process is much more difficult to implement than its digital counterpart. Usually an analog processor must be purpose-built and does not have the general character of digital systems. The analog circuits are difficult to implement because the physics of the material used to construct the circuits plays an important role in the solution of the problem. It is also difficult to control the physical properties of micron-size devices such that their analog characteristics are well matched. This matching of analog device characteristics is a major difficulty facing an analog designer. And digital machines have an advantage over analog ones when high precision is required. The analog approach usually requires adaptive techniques to mitigate the effects of component differences. However, this kind of adaptation leads naturally to systems that learn about their environment.

Digital systems depend on a small set of primitives that can support universal logical computation. For example, any logic function can be implemented using only NAND gates. In principle, a similar mathematical universality is possible in analog circuits by using primitive circuits for addition and multiplication. However, the neuromorphic engineer is usually more concerned with identifying the primitives needed to emulate neuronal processing than with mathematics per se.

For many crucial neuronlike operations, dedicated analog processors can be constructed much more compactly than their digital counterparts. Some neuromorphic primitives are obvious, such as the simple addition of currents by joining their conductors. Others provide useful generic functions, such as logistic and tanh circuits. Yet others are simple but sophisticated processors (e.g., Mahowald and Mead’s elementary contrast reporting pixel and Delbrück’s correlation circuit) (Figure 1).

These circuits are usually based on CMOS field effect transistors (FETs), whose conductivity can be altered by an applied electric field. A FET is a device in which the flow of current between its source and drain terminals is controlled by a voltage applied to its gate terminal. The relationship between the gate voltage and the current that flows across the transistor’s channel is rather subtle. There are two primary regimes of behavior. In the subthreshold regime, in which small voltages are applied between the gate and the source, the transistor current grows exponentially with the gate voltage. In above-threshold regime, in which larger voltages are applied, the current grows more slowly. Conventionally, transistors are operated in the above-threshold regime, and in the extreme case of digital circuits only very large gate voltages are used so that the transistors are either fully off or fully conducting.

By contrast, neuromorphic circuits typically exploit the otherwise unpopular subthreshold regime because this regime offers some striking advantages for non-digital circuits. The exponential increase in transistor current with gate voltage means that the gain (loosely defined as the change in current for a given change in voltage) of the subthreshold transistor is very high, and gain is the lever of information processing. The exponential property can be used to compose interesting analog circuits with natural exponential and logarithmic properties that are useful for emulating neuronal properties. A further feature of subthreshold circuits is that the absolute currents flowing through the transistors are very small, so these circuits operate at very low power. The downside is that the signals are very small, so the circuits are susceptible to noise and fabrication variations.

All these interesting properties offer the possibility of constructing very low-power brainlike electronic systems.
circuits that operate in real time. An important strategic consideration in this quest is that the basic CMOS VLSI technology does not have to be developed by the neuromorphic engineering community. The development of VLSI technology is fueled by the digital information technology (IT) market, so those developments need only be applied to the needs of subthreshold analog neuromorphic engineering. The very same technology used to construct digital computers can also be used (in subthreshold) to construct computational primitives for systems of neuronlike processors. Unlike digital systems, which have a relatively small number of general-purpose processors that process a command stream sequentially, the performance of neuromorphic systems depends on the parallel configuration of large numbers of these special-purpose primitives.

**Emulation versus Simulation**

What advantages do these neuromorphic circuits have? One advantage is that they offer a medium in which neuronal networks can be emulated directly in hardware rather than simply simulated on a general-purpose computer. However, it should be noted that the uses of emulation are not the same as those of simulation. Digital simulation is generally used to explore the quantitative behavior of neuronal systems. Because they are composed of large numbers of nonlinear elements and have a wide range of time constants, their mathematical behavior can rarely be solved analytically. But the speed of simulation is limited by the shortest time constant of the problem, so simulation performance slows dramatically as the number and degree of coupling of elements increase.

By contrast, neuromorphic emulations operate in real time, and the speed of the network is independent of the number of neurons or their coupling. However, analog circuits provide only a good qualitative approximation of the exact performance of simulated neurons. Moreover, the design of special-purpose hardware is a significant investment, particularly if it is analog hardware because analog VLSI (aVLSI) design remains very much an art form. Designing robust novel circuits depends on engineers with considerable experience, preferably working in small groups with sufficient collective knowledge to monitor and criticize one another’s design assumptions and circuit layouts. So, in their present form, neuromorphic circuits are not suitable for quantitative simulation.

On the other hand, where the electronic neural circuits do provide a tangible advantage is in the investigation of questions concerning the strict real-time interaction of the system with its environment, possibly taking sensory input from neuromorphic sensors such as silicon retinas or silicon cochleas. Neuromorphic retina and cochlea sensors represent examples of extremely successful systems built with these design principles. Neuromorphic retinas are vision sensors that can adapt in real time to local changes in illumination over a wide range of intensities and that reproduce many of the functionalities of the retina’s outer plexiform layer. Thanks to their adaptation and real-time response properties, they are currently being applied to machine vision tasks that are difficult to solve using conventional approaches. These include, for example, traffic monitoring in open uncontrolled environments such as freeways or fast robotics, in which the low latency and sparsity of address-event representation (AER) sensor data are very advantageous. Similarly, silicon cochleas convert auditory signals into sequences of spikes generated by populations of silicon neurons arranged tonotopically along the frequency axis, similar to the inner-ear hair cells located along the basilar membrane of real cochleas. These devices have promising application domains, both as alternative low-power solutions for cochlear implants and as computational devices for engineering applications.

**Neurons in Silicon**

One of the central goals of neuromorphic engineering is to capture the computational principles of neurons and their networks in hardware. A cornerstone of this quest has been the development over the last decade of hybrid analog–digital VLSI neurons, together an infrastructure for composing networks of these neurons. It is now possible to assemble quite complex systems of such neurons.

**Integrate-and-Fire Models**

Neurons communicate by pulses (or spikes) that propagate along electrically lossy point-to-point wires that are the axons. Real neurons have a complex morphology and even more complex biophysics, whose full emulation is beyond the reach of present electronic technology. Nevertheless, the integrate-and-fire neuron (I&F), which is a bold simplification of real neurons, has proved to have significant explanatory power in understanding the behavior of neuronal networks both in theory and simulation. So far, it is this simplified model that has received the most attention in neuromorphic circles.

I&F neurons integrate presynaptic input currents and generate a voltage pulse analogous to an action potential when the integrated voltage reaches a threshold. Many variants of these circuits were built during the 1950s and 1960s using discrete electronic components. The first simple VLSI version was probably the
Axon Hillock circuit, built by Mead in the late 1980s. In this circuit, a capacitor that represents the somatic membrane capacitance integrates current input to the neuron. When the capacitor voltage crosses a threshold, it is reset by a positive feedback loop. More recent models include additional neural characteristics, such as spike-frequency adaptation properties and refractory period mechanisms. These features can be combined in a compact low-power circuit such as the one shown in Figure 2.

Conductance-Based Models

These VLSI I&F neurons provide convenient approximations of the behavior of neuronal somata without committing to the overhead of emulating the plethora of voltage-dependent conductances and currents present in real neurons. But, if necessary, these conductances can be emulated using subthreshold CMOS circuits. For example, the silicon neurons of Mahowald and Douglas are composed of connected compartments, each of which is populated by modular subcircuits that emulate particular ionic conductances. The dynamics of these types of circuits is qualitatively similar to the Hodgkin–Huxley mechanism without implementing their specific equations. An example of this type of silicon neuron circuit is shown in Figure 3.

In this circuit, the membrane capacitance $C_{\text{mem}}$ is connected to a transconductance amplifier that implements a conductance term, whose magnitude is modulated by the bias voltage $G_{\text{leak}}$. This passive leak conductance couples the membrane potential to the potential of the ions to which the membrane is permeable ($E_{\text{leak}}$). Similar strategies are used to implement the active sodium and potassium conductance circuits. In these cases, transconductance amplifiers configured as simple first-order low-pass filters are used to emulate the kinetics of the conductances. A current mirror is used to subtract the sodium activation and inactivation variables ($I_{\text{Naon}}$ and $I_{\text{Naoff}}$) rather than multiply them, as in the Hodgkin–Huxley formalism. Additional current mirrors half-wave rectify the sodium and potassium conductance signals, so that they are never negative.

Several other conductance modules have been implemented using these principles. For example,

![Figure 2](image-url)
there are modules for the persistent sodium current, various Ca$^{2+}$ currents, Ca$^{2+}$-dependent potassium current, potassium A current, nonspecific leak current, and an exogenous (electrode) current source. The prototypical circuits can be modified in various ways to emulate the particular properties of a desired ion conductance. For example, some conductances are sensitive to Ca$^{2+}$ concentration rather than membrane voltage and require a separate voltage variable representing free Ca$^{2+}$ concentration.

Synaptic conductance circuits have also been developed using similar principles. But synaptic conductances are sensitive to ligand concentrations rather than membrane voltage, so these circuits require a voltage variable that represents neurotransmitter concentration. This set of ionic conductances, with their different dependencies and time constants, gives rise to state-dependent dynamics when embedded in the resistive capacitive network that represents the passive electrical structure of a neuron. These circuits can be composed in this way to model in detail the electrophysiological behavior of several types of neurons, for example, pyramidal cells.

**Axons, Action Potentials, and the Address–Event Representation**

Biological neurons communicate with one another using dedicated point-to-point axons. The all-or-nothing action potential can be translated into a discrete level signal, that is robust against noise and interchip variability, and can be conveniently transmitted between chips and easily interfaced to standard logic and computer systems. In the AER method developed by Mahowald and others, the action potentials generated by a particular neuron are transformed into an address that identifies the source neuron and then broadcast on a common data bus. Many silicon neurons can share the same bus because switching times in CMOS and on the bus are much faster than the switching times of neurons. Events generated by silicon neurons can be broadcast and removed from a data bus at frequencies greater than a megahertz. Therefore, more than 1000 address events could be transmitted in the time it takes one neuron to complete a single action potential. The addresses are detected by the target synapses, which then initiate their local synaptic action (see Figure 4). In this way, synapses that should be connected to the source neuron initiate a synaptic input whenever the source neuron spikes. These multiplexing strategies are most effective if, like their biological counterparts, the neurons of a network are only sparsely active.

Event-based digital encoding methods are convenient for configuring large networks because the network connectivity can be implemented by a programmable digital address mapper. The mapper receives an address event from a source neuron and translates it into one or more target addresses that are transmitted on the presynaptic bus. Because the mapping table is programmable, arbitrary network topologies can be set and modified dynamically. This method greatly facilitates the configuration and testing of large multichip systems.
Synapses in Silicon

Silicon synapses come in many forms, depending on the underlying model and the circuit implementation. The models range from simple current source models to more complicated conductance-based models including approximations of α-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid (AMPA) and N-methyl-D-aspartate (NMDA) excitatory synapses, and potassium-mediated and chloride-mediated (shunting) inhibitory synapses. An example of a synaptic circuit that can reproduce the exponential dynamics observed in real synapses is shown in Figure 5.

Plasticity and Learning

One of the key properties of biological synapses is their short- and long-term plasticity. Short-term plasticity produces a dynamic modulation of the synaptic strength by the timing of the input spikes alone, whereas long-term plasticity produces sustained changes in synaptic strength that are induced by the correlations in the spiking activity of the pre- and postsynaptic processes.

Circuits have been developed that emulate the short-timescale synaptic depression; also, various learning circuits that implement long-term plasticity of synapses use spike-timing information and/or rate information. Spike-timing-dependent plasticity (STDP) mechanisms are particularly well suited to the VLSI neuromorphic networks described here because these networks process AER signals. In STDP, the precise timing of spikes generated by the pre- and postsynaptic neurons have an important role in shaping the synaptic efficacy. If a presynaptic spike arrives at the synaptic terminal before a postsynaptic spike is emitted, during a window of causality, the synaptic efficacy is increased. Conversely, if the postsynaptic spike is emitted soon, before the presynaptic one arrives, the synaptic efficacy is decreased. Several modeling studies have developed learning algorithms based on STDP and have demonstrated how systems that use these types of algorithms can carry out complex information processing tasks. The excitatory synaptic circuit of Indiveri and colleagues implements both long- and short-term plasticity.

Memory and Synaptic Weight Storage

There are a number of practical problems that slow down the development of large-scale, distributed, massively parallel networks of VLSI I&F neurons. Three of the most important ones are (1) how to access the individual synapses of the network to provide input signals, and how to read from each neuron to generate output signals; (2) how to set and/or store the weights of individual synapses in the network; and (3) how to (re)configure the network topology on the same chip. Of course, these problems arise directly from the fundamental difference in architecture between conventional computers and neural networks. Conventional computers have a single processor, or small number of processors, connected to a random access memory. This global access...
means that the state of the machine can be conveniently loaded and examined. By contrast, in neuronal networks the memory and processing is massively distributed and co-localized at the synapses. In this case, direct loading and inspection are not possible, except at the huge cost of providing duplicate access lines. Long-term plasticity circuits alleviate these problems slightly (for both biology and neuromorphic VLSI) by allowing the weights of synapses to be set automatically, without requiring dedicated access to individual synapses. But the difficulty of experimental observation and control remains for these circuits.

Basic considerations about the relation between precision and required silicon area suggest that biologically realistic low-resolution weights could also be stored as analog values as voltages across capacitors. But conventional CMOS capacitors are subject to leakage, so their voltages decay over seconds if not refreshed. Alternatively, the analog values can be stored using nonvolatile technology similar to that used in electrically erasable programmable memory (EEPROM). In this case, the analog value is stored as charge on a floating gate, that is the gate of an FET sandwiched between two layers of (perfect) oxide insulator. Charge is added or removed from the floating gate by Fowler–Nordheim tunneling and impact-ionized hot-electron injection. The uncontrolled decay of charge from floating gates is negligible, so learned synaptic weights can be retained for decades, even when the power to the circuits is off. This exciting technology is still under development, but simple structures such as single transistor synapses and examples of networks that implement learning algorithms based on these circuits have been demonstrated. The problem of synaptic weight storage would be simplified considerably if only one binary bit, rather than an analog value, need be stored. In fact, it has been demonstrated that networks of sufficiently large numbers of binary synapses are adequate for any memory task. Circuits for such binary synapses have been proposed that bypass the need to have specialized structures for nonvolatile analog memory within each synapse.

**Multichip Neural Networks**

Several examples of successful multichip networks of spiking neurons have been demonstrated during recent years. Using present CMOS technology, it is possible to implement on the order of hundreds of neurons and thousands synapses per square millimeter of silicon. In principle, networks of this type can be scaled up to any arbitrary size, but in practice the network size is limited by the maximum silicon area and AER bandwidth available. Given the current speed and specifications of the AER-interfacing circuits and the availability of present silicon VLSI technology, the network size could be increased by at least two orders of magnitude. It is likely that large neuronal networks, such as those of the neocortex, are dominated by local connectivity, with only a relatively small fraction of long-range connections. In this case, it may be possible to make similarly large-scale VLSI networks in which multiple regional AER buses with the same address spaces carry local-event traffic and are interconnected by sparser long-range traffic between local domains. However, before testing those connectivity limits, there is more immediate interesting work to be done with these multichip networks. For example, existing methods can be used for investigating complex spike-based learning algorithms in real time. And these studies are all the more

---

**Figure 5** A compact synaptic circuit that exhibits the exponential dynamics observed in real synapses. The circuit’s analog output current $I_{\text{syn}}$ encodes the frequency of the input spikes, represented as digital voltage pulses arriving at the $V_{\text{spk}}$ node. The circuit’s time constant is set by adjusting the value of the $I_c$ current. The $V_g$ bias in the gain module can be used to set the circuit’s gain; lower values increase the gain, whereas higher ones decrease it. Similarly, the synaptic efficacy is controlled by the $V_w$ bias voltage in the weight module. In this case, higher values of $V_w$ lead to higher output currents. Additional circuits can be connected to the $V_w$ node to implement short- and/or long-term plasticity mechanisms and to locally store or refresh the value of the weight.
interesting when we consider problems of adaptation and learning in neuronal networks interfaced to neuromorphic AER sensors, such as silicon retinas and silicon cochleas.

**Impact of Neurobiology on Computer Engineering**

Conventional engineering computing systems are beginning to face challenges that have many points in common with those faced by neuromorphic and biological neural systems. As VLSI technology scales to the submicron feature size, power consumption per square micron starts to become a serious limiting factor and single transistors start to behave like unreliable stochastic devices. Under these conditions, insights obtained from building low-power neuromorphic circuits and biologically inspired methods for computing reliably with unreliable components can bring crucial knowledge to advanced computing technologies. Similarly, the current trend toward multicore digital processors in the IT industry will continue and will require the understanding of massively parallel computation, which is also a central question of neuroscience. The principles of unclocked, real-time asynchronous event-driven computation are still far from clear, but it seems likely that this form of communication and processing lies at the heart of biology’s ability to achieve effective collective behavior from massively distributed, weakly connected, and relatively slow neuronal processors. We expect that the exploration of neuromorphic systems will continue to contribute strongly to solving this intriguing puzzle.

See also: Long-Term Depression (LTD): Metabotropic Glutamate Receptor (mGluR) and NMDAR-Dependent Forms; Long-Term Potentiation (LTP): NMDA Receptor Role; Neuroplasticity: Computational Approaches; NMDA Receptors and Development; Retinal Models; Spike-Timing-Dependent Plasticity Models.

**Further Reading**


