The Human Brain Project and neuromorphic computing

Andrea Calimera, PhD
Enrico Macii, PhD
Massimo Poncino, PhD

Department of Control and Computer Engineering,
Politecnico di Torino, Italy

Correspondence to: Massimo Poncino
E-mail: massimo.poncino@polito.it

Summary

Understanding how the brain manages billions of processing units connected via kilometers of fibers and trillions of synapses, while consuming a few tens of Watts could provide the key to a completely new category of hardware (neuromorphic computing systems). In order to achieve this, a paradigm shift for computing as a whole is needed, which will see it moving away from current "bit precise" computing models and towards new techniques that exploit the stochastic behavior of simple, reliable, very fast, low-power computing devices embedded in intensely recursive architectures.
In this paper we summarize how these objectives will be pursued in the Human Brain Project.

KEY WORDS: analog hardware, digital hardware, high-performance computing, neural networks, neural models, neuromorphic computing

Introduction

The meaning of neuromorphic computing (NC) has evolved significantly since the term was first coined by Carver Mead in the 1980s. While the term was initially meant to describe the use of electronic systems that operate using the same physics of computation used by the nervous system, it now represents a wider concept that bridges computing systems and neural systems in both directions.

The original concept was essentially concerned with the construction of electronic systems using existing technologies in order to emulate neural ones, mostly for brain simulation purposes (D'Angelo et al., 2013). This use of hardware to emulate the behavior of portions of the brain is represented by the arrow from "hardware" to "brain" in figure 1. This traditional view of NC is still the most widely accepted one in the domain of neuroscience and neural computing.

A slightly different view of NC is, however, possible: in an effort to overcome the limitations of current technologies (in terms of energy per operation and area), many researchers are investigating novel computing *architectures* that mimic biological neural structures with the purpose of achieving the computational capabilities of such systems with similar volume and energy efficiency. This is denoted by the arrow from "brain" to "hardware" in figure 1.

These two interpretations are, in fact, two sides of the same problem, although they can be approached independently. For instance, in the original emulation approach, one can stick to a specific hardware implementation (e.g., a multiprocessor architecture, a custom-designed digital or analog circuit) and try to implement neural structure and functions according to this pre-defined implementation style; benefits of this approach could lie in: (i) the exploitation of the specific excellence of this implementation style in some metrics (e.g., computational speed in a digital design) and (ii) the use of consolidated and highly automated design flows. On the other hand, the same implementation style might encounter bottlenecks in other metrics (e.g., energy consumption).

Conversely, in the reverse approach, novel architectures are sought that tend towards the highly parallel, learning-based computational paradigm of the brain. In the case of programmable systems this implies overcoming the classical von Neumann computational paradigm upon which traditional computing systems are based. For custom-designed circuits, it implies overcoming digital implementations based on complemen-

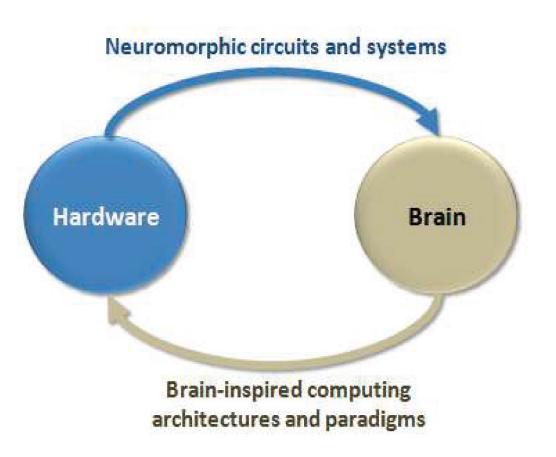
Neuromorphic circuits and systems
Hardware
Brain
Brain-inspired computing architectures and paradigms

Figure 1 - The two views of neuromorphic computing.

tary metal-oxide semiconductor (CMOS) devices and the highly standardized, automated tools and flows used to design them.

This article is organized as follows. It opens with a survey of NC approaches, providing, in this section, an overview of the approaches to NC proposed in the literature, focusing in particular on their distinctive features. It then looks at NC and the Human Brain Project (HBP), describing the ways in which NC is approached in the HBP, indicating its medium- and long-term objectives.

Survey of neuromorphic computing approaches

While the first attempts to implement electronic models of neural circuits were focused mainly on "brain-centered" strategies, e.g., the construction of perceptrons (Rosenblatt, 1958) and retinas (Fukushima et al., 1970), the advent of NC in the late '80s shifted the research paradigm to a more "hardware-centered" strategy under which engineers try to exploit the characteristics made available by electronic devices, circuits and systems to emulate or simulate the brain models proposed by neuroscientists.

Neuromorphic computing hardware has undergone rapid development in the last two decades, with the introduction of a large variety of designs, implementation methodologies and prototype chips. All of these share a common objective, namely, to mimic the functional behavior of the human brain within the same budget of energy, but there are substantial differences in the way in which this goal is pursued. It would be difficult to provide a fair and comprehensive description of such a huge number of solutions, and maybe not even particularly useful; a more constructive endeavor, rather, is to group them in two main classes, namely, "emulative" and "simulative" solutions.

Emulative strategies focus on physical emulations of neural models using inherently noisy and unreliable micro- or nano-scale electronic components, i.e., transistors, with feature sizes approaching the atomic structure of matter. Circuits resulting from this approach are typically referred to as "neurochips" (the left branch of the taxonomy tree in figure 2). These solutions have the potential to exploit the non-linear current characteristics of silicon-based transistors to naturally replicate the electrochemical functions of human neurons.

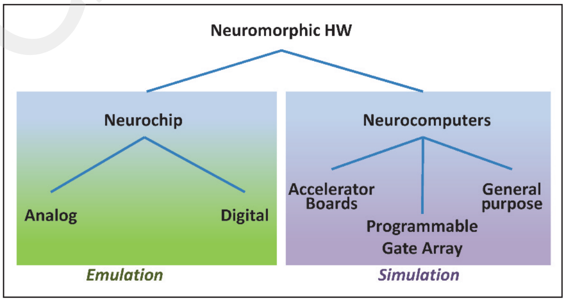

Figure 2 - Classification of neuromorphic hardware.

The choice of "analog" or "digital" neural primitives constitutes the main factor distinguishing between different neurochips. For instance, the pioneering work conducted by Carver Mead (Mead, 1989) belongs to the class of analog neurochips, as it integrates biologically inspired electronic sensors with analog circuits, introducing an address-event-based asynchronous, continuous time communications protocol. Today, the Mead approach is adopted by many ongoing research studies (e.g., Indiveri, 2000). Analog circuits are very compact and offer high speed at low energy dissipation as they naturally perform neuron-like functions, such as integration and summation of currents and charges. This comes at a cost, namely, high sensitivity to noise, process parameter variations, and, most important, greater design and verification efforts, all of which increase the implementation cycle. On the other hand, digital circuits offer high computational power, high reliability and faster prototyping thanks to the availability of powerful computer-aided design tools coming from the very large-scale integration (VLSI) domain. Disadvantages are the relatively large circuit size compared to analog implementations as many elementary functions (like integration) are not available in digital. The 1990s saw pioneering works on digital circuits (Murre, 1995; Ramacher et al., 1993; Jahnke et al., 1996).

Simulative approaches are those that focus on simulation of neural models rather than precise emulation of neural signals. Such methods, referred to as "neurocomputers" (right branch in figure 2), exploit the large availability of low-price, reliable integrated circuits to speed up the execution of neural models. More specifically, they aim at reproducing large systems that abstract away the biological details of the brain and focus on the brain's larger-scale structure and architecture, on how its elements receive sensory input and on how they connect to each other, adapt these connections, and transmit motor output.

Three different types of neurocomputers can be envisaged: "accelerator boards", "programmable arrays" and "general-purpose". Even though they belong to the same class of strategies, their implementation reflects the existence of different requirements that have emerged as neurocomputers have evolved.

When the objective is to speed up the simulation of a stable, already available neural model, accelerator boards are definitively the best option. Accelerators were also the first kind of NC hardware seen in the early '90s because they were relatively cheap, widely available, and simple to connect to a PC using expansion slots, and were typically provided with user-friendly software tools. These accelerators are typically based on artificial neural networks, but some prototypes that use digital signal processors (DSPs) for fast signal processing have been proposed. The speed-up they can achieve is in the order of 10x with respect to pure software simulations running on single workstations. The first examples of such boards were proposed around two decades ago (Lindsey et al., 1995; Trealeaven, 1989; Arif et al., 1993).

Accelerator boards are specialized for certain tasks, hence, they lack flexibility, and when a new model has to be implemented, the design time they require is typically very long. For this reason they are not particularly suitable for fast prototyping of new computing paradigms. To overcome this issue, the second class of neurocomputers, i.e., "programmable arrays" (better known as field programmable gate arrays - FPGAs), has emerged as a more effective solution. FPGAs offer hardware-like performance gains with greatly reduced design time (Roggen et al., 2003) as they can be electronically reprogrammed in real time (Maguire et al., 2007) using commercial development tools. Nevertheless, FPGAs suffer from severe power and routability issues which limit their use for scaling a neural model to large sizes.

The third class of solution relies on the use of "general-purpose" processors and software, which offer enough programmability to explore different neural models. The accuracy they guarantee is far below that of the above approaches, but they are particularly useful when the research question is not "how" to scale a model, but "which" model to scale (Rast et al., 2010); in this sense they offer the option of "what-if" analysis that can serve as feedback for the optimization of other NC hardware, like accelerators and neurochips, or for the refinement of neural models. One of the main limitations of these approaches is the overhead introduced by software, which is typically stored in memory units that are physically far from the computational engines (CPUs). Hence, the "flight time" of data exchanged between CPUs and memory may introduce large delay penalties that seriously compromise the performance.

Implementations range from architectures of simple, low-cost elements – the first examples are described by Heemskerk et al. (1994) and Speckman et al. (1993) – to architectures with rather sophisticated processors like transputers, which are unique for their parallel I/O lines (Foo et al., 1993), or DSPs, which were primarily developed for correlators and discrete Fourier transforms (Onuki et al., 1993). A more recent, and efficient strategy is to exploit the availability of multicore architectures. This is the approach adopted within the SpiNNaker platform, which will subsequently be described in detail.

Among all the aforementioned techniques, none is systematically better than the others; different strategies can be adopted orthogonally in order to achieve the final goal of understanding the human brain. This is the goal of the NC division of the HBP.

Neuromorphic computing and the Human Brain Project

Neuromorphic computing is a fundamental pillar of the HBP and one of the six platforms implemented within it (the *Neuromorphic Computing Platform,* hereafter NCP for brevity) (Markram, 2013). The term *platform* emphasizes the fact that, like the other HBP platforms, the NCP will provide users with access to specific "services". In the case of the NCP, these services correspond to the two *neuromorphic computing systems* (NCSs), i.e., specific and custom-designed neuromorphic "hardware" systems that, described later in this section, are the concrete outcome of the project in the context of NC research. Access to these NCSs implies not just usage of the hardware, but also availability of software tools for their configuration, operation and the analysis of generated data as well as user support through documentation.

Thanks to the possibility of accessing these services, researchers will be relieved of the need to develop and maintain basic hardware and software, and allowed to focus on experiments and applications directly relevant to their field. We expect that the NCP, allowing researchers to work with state-of-the-art design tools and with two advanced NCSs that implement simplified models of actual neuronal circuitry, will enable a huge acceleration in the current research of brain simulation and emulation.

The NCP is closely linked with two other HBP platforms: the *Brain Simulation Platform* and the *High-Performance Computing Platform* (Fig. 3). The former feeds the NCP with brain models, whereas the latter provides supercomputing, and cloud capabilities as well as the system software, middleware, and visualization support necessary to create, simulate and analyze multiscale brain models.

As mentioned before, the NCP overlaps almost entirely with the two NCSs that are provided. These NCSs implement two different conceptual approaches to NC as described in the survey section.

The specific contribution of the POLITO research unit to the NC sub-project in the HBP will relate to the development of new and more efficient tools for software development on the neuromorphic multicore (NM-MC) systems, specifically the SpiNNaker one: in particular, tools for the automated parallelization of source code starting from algorithms. Using software terminology, this *middleware* will ease the task of programmers in writing applications for systems of this type.

Neurochip-like computing system

The first type of NCS provided by the HBP is based on the European FACETS project (*http://facets.kip.uni-heidelberg.de*), which has pioneered an approach combining local analog computation in neurons and synapses with binary, asynchronous, continuous time spike communication. This NCS is termed NM-PM in the project, where PM stands for physical model.

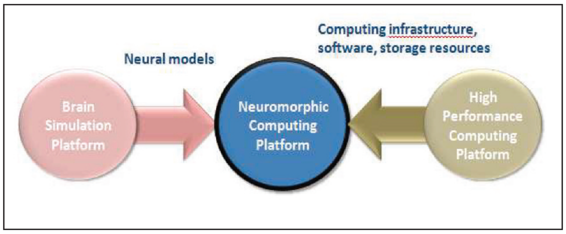
Computing infrastructure,
software, storage resources
Neural models
Brain Simulation Platform
Neuromorphic Computing Platform
High Performance Computing Platform

Figure 3 - Relationship of the NCP with other platforms.

Current versions of the NM-PM system incorporate 5x107 plastic synapses and 200,000 biologically realistic neuron models on a single 8-inch silicon wafer.

In terms of technology the large-scale FACETS system is based on a mixed analog/digital VLSI implementation in a standard 180nm CMOS process. Local computation in neurons and synapses is mostly performed by compact custom-designed analog circuits, which communicate by exchanging spikes in an asynchronous fashion. The neuron and synapse models implement state-of-the-art results from neuroscience; the models include features such as plasticity mechanisms and a complex neuron model with up to 16,000 synaptic inputs per neuron, spike frequency adaptation and various firing modes as observed in biology.

As the substrate represents a typical non-von Neumann system architecture, the memory required for synaptic weights and cell parameters is distributed in the computing fabric and employs technologies like small SRAM memory cells as well as analog units.

The various silicon wafers with the analog components, which implement the true neuromorphic computations, are stacked in a crate and attached to a motherboard that contains the digital portion of the system used to interface the analog chips on the wafers with several FPGAs on the backplane interconnecting the wafer boards in the crate. These FPGAs implement the necessary communication protocols to exchange neural events between the different network wafers and the host computer. Figure 4 shows an abstract block diagram of the NM-PM adapted from (http://facets.kip.uni-heidelberg.de).

A key characteristic of the NM-PM computing system is that *it does not execute a programmed code* but evolves according to the physical properties of the electronic devices. In this sense NM-PM truly implements the neuromorphic hardware paradigm, in which a specific hardware architecture is used to implement a brain model. Another feature of the original FACETS project, which has been inherited by the NM-PM computing system, is the use of a network description language (PyNN) that provides platform-independent access to software simulators and neuromorphic systems and will be used throughout the HBP. The NM-PM paradigm will evolve in three versions of increasing complexity within the timeframe of the HBP.

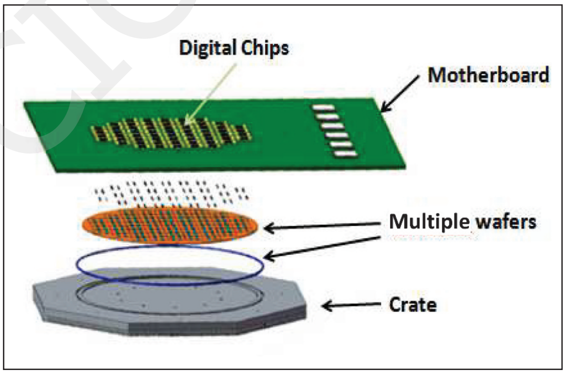

Figure 4 - Structure of the basic meuromorphic device used in the NM-PM computing system.

The first version, NM-PM-1, will be delivered within the first 18 months of the HBP, and will scale the system up to a size of 20 wafers corresponding to 4 million neurons and 1 billion synapses operating at an acceleration factor of 10,000 compared to biological real time. It will be installed in a dedicated building on the Science Campus at Heidelberg University in Germany. The NM-PM-1 system will be complemented by a dedicated conventional compute cluster for network configuration, data analysis and execution of closed perception-action loop experiments.

NM-PM-2, expected three years after the start, will contain 1000 modules implemented in 65nm circuit technology and will pioneer the use of wafers embedded in printed circuit boards. In the second phase of the project, the HBP will build a third-generation NM-PM-3 system, exploring options for systems that can shift between different speeds of operation, from real time (a pre-condition for robotics and many other applications) to 10,000 times faster than real time. It is planned that NM-PM-3 will incorporate 10,000 wafer modules with 1016 components. This will give the system the ability to emulate a substantial fraction of the human brain.

Neurocomputer-like neuromorphic computing system

The second NCS provided for in the HBP is a programmable system based on massively parallel multicore architectures, in turn based on ARM processors. This NCS is termed NM-MC in the project, where, as already indicated, MC denotes multicore.

The NM-MC system is based on the approach adopted by the UK SpiNNaker group (Furber and Temple, 2007). The basic building block is the SpiNNaker chip, which contains, in its original version, 18 ARM cores and a shared local 128Mbyte memory. It allows real-time simulation of networks implementing complex, non-linear neuron models. A single chip can simulate 16,000 neurons with 8 million plastic synapses running in real time within an energy budget of 1W.

One of the processors acts as a monitor processor, and runs the operating system functions on the chip. The other ones act as *fascicle* processors, each modeling a group of up to a thousand individual neurons. Each fascicle processor receives spike events from, and issues spike events into, a packet-switching communications system, with each spike event encoded as a single packet. Within a chip, these spike events converge through the network-on-chip (NoC) communications to an arbiter, where they are selected and sent in sequence to a router, which uses internal tables to identify which fascicle processors should receive each event (determined from the connectivity netlist of the neural network that is being modeled) and passes the event on accordingly.

The overall system consists of multiple interconnected chips: connectivity is provided by six transmit and receive bidirectional interfaces to six neighboring chips (Fig. 5). Local memory is not shown in the picture.

This interface implies a fully connected system as an NoC where each chip is connected to six others in a toroidal manner (Fig. 6, over).

As such, the NM-MC is a brain-inspired massively parallel computing system that can, in principle, be used for any type of computation. What makes it suitable for simulation of neural systems is clearly the application software running on it. It is evident that considerable effort has gone into providing software development kits that make it possible to compile and execute user programs, as well as proper operating system support to manage communication between the nodes.

Unlike the NM-PM approach, the NM-MC one does not implement a specific algorithm, and it is, rather, to be viewed as a platform on which different algorithms, and thus different types of neurons and connectivity patterns, can be evaluated.

Similar to the NM-PM, the NM-MC will evolve in two versions of increasing complexity. In the first 18 months the system will scale to 1 million ARM cores corresponding to approximately 56,000 SpiNNaker chips with a simulated bisection bandwidth of 109 spikes per second (version NM-MC-1) and a simulation capability of 1 billion neurons in biological real time.

The second version (NM-MC-2), expected after three years of the project, will improve to 4 million cores with a simulated bisection bandwidth of 1011 spikes per second.

Accuracy issues

From this “computational” perspective, details about the accuracy of neural element models (neurons and synapses in particular) against the real elements or biological-level models are not degrees of freedom; each specific approach implements selected models that are decided upfront. Specifically, as regards neuron models, the FACETS system implements an exponential integrate-and-fire (AdExp) neuron model, whereas the SpiNNaker system is optimized for neuron models such as the Leaky Integrate-and-Fire and the Izhikevich ones. SpiNNaker admittedly states that its architecture is “not intended to run models with high biological accuracy, but is much more aimed at exploring the potential of the spiking neuron as a component from which useful systems may be engineered”, which clearly explains the semantics of the

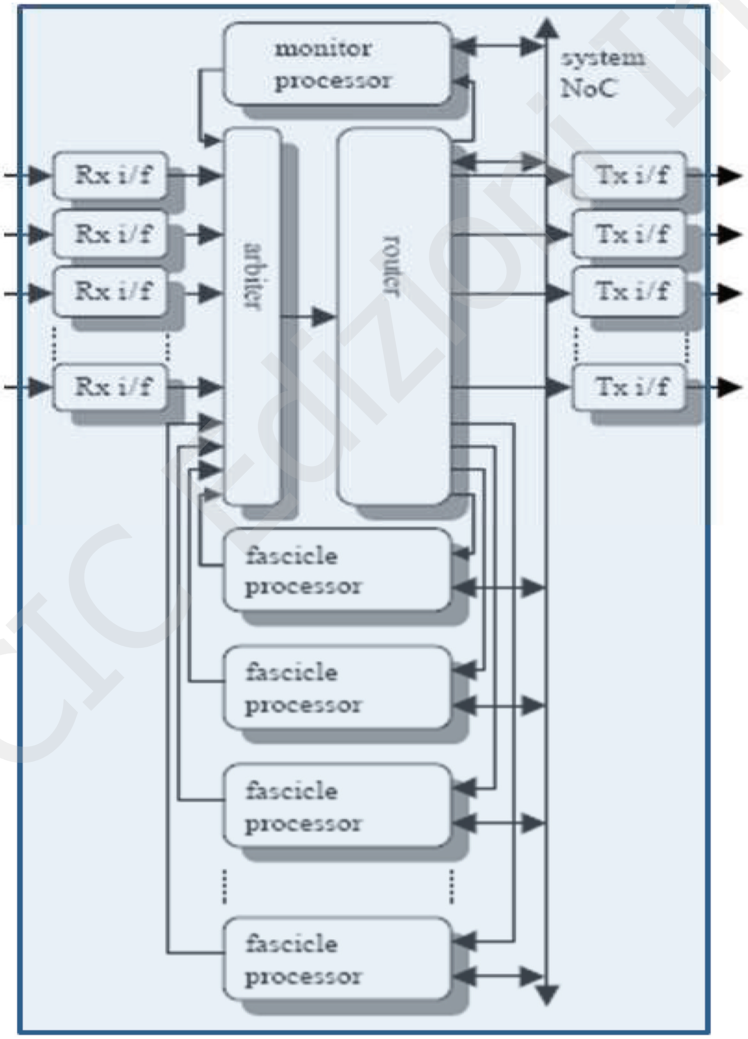

Figure 5 - Structure of the SpiNNaker (and NM-MC) basic chip (adapted from http://www.artificialbrains.com/spinnaker#hardware).

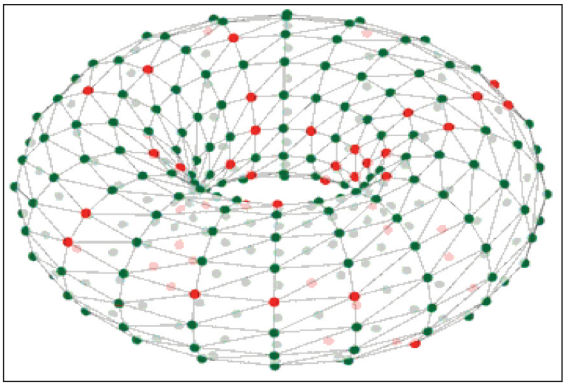

Figure 6 - Interconnection architecture of the NM-MC system (Source: http://apt.cs.man.ac.uk/projects/Spinnaker/).

computational perspective. The procedures and implications of realistic biologically based models are presented elsewhere in this issue (D'Angelo et al. 2013).

Concluding remarks

Neuromorphic hardware is an essential instrument to provide feedback on neural models developed by neuroscientists. The Neuromorphic Computing Platform in the Human Brain Project serves this purpose through two different computing paradigms to allow a better coverage of this feedback.

References

Arif AF, Kuno S, Iwata A et al. (1993). A neural network accelerator using matrix memory with broadcast bus. Proceedings of IJCNN-93-Nagoya, pp. 3050-3053, doi: 10.1109/IJCNN.1993.714363.

D'Angelo E, Solinas S, Garrido J, et al. (2013). Realistic modeling of neurons and networks: towards brain simulation. Funct Neurol 28: 153-166.

Foo SK, Saratchandran P, Sundarararjan N (1993). Parallel implementation of backpropagation on transporters. Proceedings of the IJCNN-93-Nagoya, pp, 3058-3061, doi: 10.1145/1787275.1787297.

Fukushima K, Yamaguchi Y, Yasuda M, et al (1970). An electronic model of the retina. Proceedings of the IEEE 58: 1950-1951.

Furber S, Temple S (2007). Neural systems engineering. J R Soc Interface 4: 193-206.

Heemskerk JNH, Hoekstra J, Murre JMJ et al (1994). The BSP400: a modular neurocomputer. Microprocessors and Systems 18: 67-79.

Indiveri G (2000). Modeling selective attention using a neuromorphic analog VLSI device. Neural Comput 12: 2857-2880.

Jahnke A, Roth U, Klar H (1996). A SIMD/dataflow architecture for a nNeurocomputer for spike-processing neural networks (NESPINN). Proceedings of the Fifth International Conference on Microelectronics for Neural Networks (MicroNeuro), 232-237.

Lindsey CS, Lindblad T, Sekniaidze G, et al (1995). Experience with the IBM ZISC neural network chip. International Journal of Modern Physics C 06: 579-584, doi: 10.1142/S012918 3195000460

Maguire L, McGinnity TM, Glackin B, et al (2007). Challenges for large-scale implementations of spiking neural networks on FPGAs. Neurocomputing 71: 13-29.

Markram H (2013). Seven challenges for neuroscience. Funct Neurol 2013; 28:145-151.

Mead CA (1989). Analog VLSI and Neural Systems. Reading, MA, Addison-Wesley.

Murre JMJ (1995). Neuroinformatics In: Arbib MA (Ed.) Handbook of Brain Research and Neural Networks, Cambridge, MA, MIT Press, pp. 741-750.

Onuki J, Maenosono T, Shibata M et al (1993). ANN Accelerator by Parallel Processor Based on DSP. Proceedings of IJCNN-93-Nagoya, pp. 1913-1916, doi: 10.1109/IJCNN.1993.717029.

Ramacher U, Raab W, Anlauf J et al (1993). Multiprocessor and memory architecture of the neurocomputer SYNAPSE-1. Int J Neural Syst 4:333-336.

Roggen D, Hofmann S, Thoma Y, et al (2003). Hardware spiking neural network with run-time reconfigurable connectivity in an autonomous robot. Proceedings 2003 NASA/DoD Conference on Evolvable Hardware. pp. 189-198, doi: 10.1109/EH.2003.1217666.

Rosenblatt F (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65 386-408.

Rast AD, Galluppi F, Jin X et al (2010). The "Leaky Integrate-and-Fire neuron: a platform for synaptic model exploration on the SpiNNaker chip. Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, doi: 10.1109/IJCNN.2010.5596364

Speckman H, Thole P, Rosentiel W (1993). A COprocessor for KOhonen's Self-organizing map (COKOS). Proceedings of ICJNN-93-Nagoya, pp, 1951-1954, doi: 10.1109/IJCNN.1993.717038

The FACETS Project, www.facets-project.org.

Treleaven PC (1989), Neurocomputers. Neurocomputing 1: 4-31.