Journal of Neuroscience Methods 210 (2012) 110–118

Contents lists available at SciVerse ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Power-efficient simulation of detailed cortical microcircuits on SpiNNaker

Thomas Sharp*, Francesco Galluppi, Alexander Rast, Steve Furber

School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK

ARTICLE INFO

Article history:
Received 1 July 2011
Received in revised form 1 March 2012
Accepted 7 March 2012

Keywords:
Cortex
Microcircuit
Simulation
Spiking
Real-time
Power
Energy

ABSTRACT

Computer simulation of neural matter is a promising methodology for understanding the function of the brain. Recent anatomical studies have mapped the intricate structure of cortex, and these data have been exploited in numerous simulations attempting to explain its function. However, the largest of these models run inconveniently slowly and require vast amounts of electrical power, which hinders useful experimentation. SpiNNaker is a novel computer architecture designed to address these problems using low-power microprocessors and custom communication hardware. We use four SpiNNaker chips (of a planned fifty thousand) to simulate, in real-time, a cortical circuit of ten thousand spiking neurons and four million synapses. In this simulation, the hardware consumes 100 nJ per neuron per millisecond and 43 nJ per postsynaptic potential, which is the smallest quantity reported for any digital computer. We argue that this approaches fast, power-feasible and scientifically useful simulations of large cortical areas.

1. Introduction

The mammalian cortex is an interesting computer, in that it performs different functions in different areas despite a largely homogeneous circuitry (Kandel et al., 2000). This suggests that cortical function emerges from a common structural template, or *canonical microcircuit* (Douglas et al., 1989), and understanding this circuit may offer insight into the operation of the cortex as a whole. However, it is extremely difficult to observe cortical activity in any great detail, due to the minute nature of neurons and synapses and the destructive properties involved in probing them. Simulations on dedicated computer hardware offer a potential solution to this problem in the form of controllable, reproducible and fully-observable functional models of biologically-plausible neurons and synapses.

The cortex has been simulated at varying degrees of fidelity and scale. Early proponents of a canonical cortical microcircuit showed a simple simulation to reproduce firing-rate-coded dynamics of the visual cortex when stimulated via electrodes in the optic radiation (Douglas et al., 1989). Significant improvements in neuroanatomical techniques have since allowed the cortex to be mapped in great detail (Thomson et al., 2002; Binzegger et al., 2004; Thomson and Lamy, 2007). This has precipitated cortical simulations of corresponding fidelity (Haeusler and Maass, 2006; Binzegger et al., 2009; Symes and Wennekers, 2009) which have demonstrated success in binary classification, feature selection and reproduction of lateral spreading activity in superficial layers following thalamocortical stimulation.

If the cortex is indeed homogeneous and a microcircuit model represents some small (order mm^3) volume of grey matter, then larger cortical structures may be modelled by parallel simulation of interconnected microcircuits. This parallelism in cortical computation is mirrored by the (exponentially growing) parallelism in high-performance computing hardware (de Garis et al., 2010) and has been exploited to produce, in two notable examples using IBM's Blue Gene architecture, simulations of thousands of neurons at ion-channel detail (Markram, 2006) and of billions of neurons represented by simple systems of non-linear differential equations (Ananthanarayanan et al., 2009). The former work concerns the detailed biophysics of neurons and synapses, whereas the latter aims only to reproduce membrane potential dynamics in order to reduce the time and energy required by the computer. This is significant: assuming power consumption of a few watts per neuron (Section 2.2) a real-time simulation of the human cortex on a Blue Gene computer would require support from a dedicated nuclear power plant. Significantly greater energy efficiency and speed is achieved in hardware circuit implementations of neurons and synapses (Mead, 1989; Schemmel et al., 2008) but at the expense of programmability, which is vital in the absence of a definitive description of neural dynamics.

SpiNNaker is a computer architecture intended to address these problems using very many low-power microprocessors and custom interprocessor-communication hardware designed to convey simulated action potentials. In a demonstration of early hardware, we simulate an intricate cortical microcircuit of ten thousand neurons and four million synapses in real-time on four SpiNNaker chips,

* Corresponding author. Tel.: +44 7940 719 564.
E-mail address: thomas.sharp@cs.man.ac.uk (T. Sharp).

0165-0270/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2012.03.001

using less than two watts of power. To our knowledge, this is the most power-efficient simulation of biologically-plausible neural circuits yet performed on digital hardware, and these results suggest the power-feasibility of real-time simulation of billion-neuron models on a planned fifty-thousand-chip machine.

2. Related work

Research into cortical computation follows two threads addressing the structural and emergent functional properties of the cortex, which have been explored in anatomical and simulation studies respectively.

2.1. Anatomical data

Cortical microcircuitry is typically described as an excitatory loop beginning with L4 pyramidal cells (the targets of thalamocortical input) projecting into superficial layer pyramids, which in turn project to deep layer pyramids that close the loop by projecting back into L4. In contrast, inhibitory cells mostly project to cells within their respective layers, thereby modulating the spiking activity in the excitatory loop. Thomson et al. (2002) use paired intracellular recordings to map synaptic interactions between neurons in slices of cat and rat visual cortex; in 998 recordings a presynaptic cell is stimulated and the membrane potential of a candidate postsynaptic target is recorded. A detailed description of cortical composition and connectivity is derived from the presence or absence, type (excitatory or inhibitory), amplitude and decay rate of postsynaptic potentials (PSPs) in each trial from and biocytin labelling of probed cells. The study agrees broadly with earlier descriptions of the cortex while providing much greater detail on synaptic densities and evoked PSPs. It is also observed that there is little difference between the anatomy of the two sampled species, suggesting a common cortical circuit amongst mammals.

Binzegger et al. (2004) draw a map of cat primary visual cortex from 3D reconstructions of 39 neurons and thalamic afferents labelled with horseradish peroxidase and classified according to morphology and location. Average dendrite lengths and output synapse counts are calculated for each neuron class from the reconstructed cells, and the number of neurons of each class is drawn from the literature. A variation of Peters' rule (Peters and Feldman, 1976) is then used in order to predict the number of synapses received in layer u by each neuron of class i from neurons of class j:

$$\bar{s}_{ij}^{u} = n_j \bar{s}_j^{u} \frac{d_i^u}{\sum_k n_k d_k^u} \quad (1)$$

where n_j is the number of neurons of class j, $\bar{s}_j^u$ is the average number of output synapses formed by neurons of that class in layer u, d_i^u is the average length of dendrite formed in layer u by neurons of class i and $\sum_k n_k d_k^u$ is the total length of dendrites in layer u. The result is a detailed map describing the neuron and synapse density of nineteen distinct populations of pyramid, stellate, basket, double bouquet and miscellaneous smooth cells distributed across five cortical layers.

The microcircuits described by the two studies are similar, with two exceptions: the first finds clusters of output synapses from L4 inhibitory cells hundreds of micrometers from the soma, whereas Binzegger et al. (2004, 2007) find that inhibitory neurons form few distal clusters; and the variation of Peters' rule used to predict connectivity in the latter approach assumes that axons are not class-selective to the dendrites upon which they synapse, which results in 'back' projections from superficial layer pyramidal cells to L4 pyramids that are stronger than those observed by Thomson and Lamy (2007).

In-vivo and in-vitro studies are limited in their ability to investigate the function of the cortex by the scale and complexity of neural computation but it is hoped that detailed anatomical data may be used to simulate the cortex and thereby offer much greater insight into its operation.

2.2. Simulation hardware

Simulations of spiking neurons have been performed on most forms of computers, and many have been proposed as suitable platforms for large-scale models. These proposals have been supported by simulations on conventional supercomputers, field-programmable gate arrays, graphics processing units and analogue circuits, which we discuss subsequently with regards to model flexibility, architectural scalability and power requirements.

2.2.1. Supercomputers

Simulations of the mammalian cortex have been conducted most successfully on conventional supercomputers. Markram (2006) describes the hardware and software architecture of The Blue Brain Project, which intends to use 2^{17} processors in an IBM Blue Gene/L computer and anatomical data produced by the Project itself to simulate ten thousand cortical neurons and their fifty million synapses in great detail. Markram argues that such detailed simulation of all neuron membrane conductances is necessary to produce a valid model of the cortex, against which Izhikevich (2007) argues that membrane potential dynamics may be accurately represented by a simple system of coupled differential equations. Izhikevich and Edelman (2008) exploit the computational efficiency of this approach in work that combines data from Binzegger et al. (2004) with original data on white matter connectivity between (sub) cortical areas to simulate a million neuron, billion synapse thalamocortical system on sixty processors of a Beowulf (Sterling et al., 1995) computer cluster. The study demonstrates network activity comparable to the alpha and beta rhythms evident in human EEG readings and, most interestingly, the model exhibits spontaneous activity that corresponds area-by-area to that seen in the human cortex, suggesting that non-homogeneous white matter connectivity differentiates function on the cortical plane. The primary contribution of the study, however, is a demonstration of the scale and fidelity at which the mammalian thalamocortical system can be simulated. In similar work, Ananthanarayanan et al. (2009) advance this demonstration by using significantly greater computing resources to simulate a billion Izhikevich neurons and ten trillion synapses on 2^{17} processors in an IBM Blue Gene/P machine, and argue that this portends full-scale, real-time simulations of the human cortex.

These studies use general-purpose supercomputers that may incorporate many thousands of processors, so they satisfy the requirements of model flexibility and architectural scalability. Power requirements are not listed in any of the publications, so we are forced to make a gross estimate of the power feasibility of full cortical simulation. For this purpose, we will take the latter work because we suspect that the Blue Gene/P machine is significantly more power-efficient than the Beowulf cluster and because Ananthanarayanan et al. use the most simple and, consequently, most power-efficient network model.

We will estimate the power requirements of full-scale, real-time simulations of the human cortex as

$$P = P_m \cdot \frac{S_c}{S_m} \cdot T_m \quad (2)$$

where P_m is an estimate of the power drawn by the published work, S_c and S_m are the number of neurons in the cortex and the published model respectively and T_m is the number of real seconds taken to complete one second of simulation time. The latter term reflects a generous assumption that the time to compute a second of simulation time is inversely proportional to power expenditure (Salapura et al., 2005).

We were unable to find the power requirements of the Blue Gene/P processor but Gara et al. (2005) report that its predecessor, the Blue Gene/L processor, consumes a constant 10 W during operation. Ananthanarayanan et al. use 2^{16} processors in simulation, suggesting a value $P_m = 10\,\text{W} \cdot 2^{16} \approx 655\,\text{kW}$. They model $1.6 \cdot 10^9$ of the $\approx 10^{10}$ neurons in the cortex (Braitenberg and Schüz, 1991) ($S_c/S_m \approx 10$) at a rate of "643 seconds for one second of simulation per Hz of [spiking] activity" implying $T_m = 1286$ at a global average 2 Hz (Neymotin et al., 2011) neural spiking rate. Consequently

$$P \approx 655\,\text{kW} \cdot 10 \cdot 1286 \approx 8.4\,\text{GW} \tag{3}$$

which is orders of magnitude more power than a research institution may feasibly draw. We readily admit that this very rough assessment says nothing of the biologically fidelity of the simulation, but even the most rudimentary spiking neuron models are no more than three times more computationally efficient than the Izhikevich (2004) equations. We also omit discussion of synapses and dendritic compartments because we believe that optimisation here does not address the fundamental issue: the size of the power problem is such that the promises of Moore's law (which states that transistor size halves every 18 months resulting in an exponential reduction in power requirements) do not permit power-feasible brain-scale simulations on conventional supercomputers before transistor dimensions reach atomic limits.

2.2.2. Field-programmable gate arrays

Customised processors designed primarily to compute membrane potentials and synaptic conductances may significantly outperform general-purpose processors in spiking neuron simulations. Such processors may be prototyped in field-programmable gate arrays (FPGAs) that allow designs to be tested without the difficulty and cost of making application-specific integrated circuits (ASICs). Pearson et al. (2005), Rice et al. (2009) and Cassidy et al. (2011) implement custom processors in FPGAs and the latter achieve real-time simulation of one million neurons. It has also been argued that FPGAs can be used as a reconfigurable substrate for arbitrary axonal wiring between processors, which would address the significant problem (for any simulation platform) of routing the outputs from one simulated neuron to the inputs of another.

However, the typical connectivity degrees of neurons in biological networks are vastly greater than those of logic gates in silicon circuits, and FPGAs are not designed for such wiring densities. Furthermore, as Cassidy et al. admit, an ASIC implementation of any processor design will show significantly better computational performance and energy efficiency than its FPGA counterpart. Indeed, the reconfigurability of FPGAs is generally detrimental to their material cost, computational performance and power requirements so, although they are a useful exploratory tool, successful prototypes are best manufactured in production runs as regular ASIC processors.

2.2.3. Graphics processing units

Spiking neuron simulations exhibit significant data parallelism in that an identical stream of arithmetic instructions is used to compute the membrane potential of every neuron. Graphics processing units (GPUs, see Fatahalian and Houston (2008) for an overview) contain tens or hundreds of arithmetic units that may execute a single instruction stream on many data elements simultaneously, thereby computing the state of many neurons in parallel. The capabilities of GPUs have generated significant interest in the neural modelling community (Nageswaran et al., 2009; Bhuiyan et al., 2010; Fidjeland and Shanahan, 2010; Han and Taha, 2010; Pallipuram et al., 2011) and almost all researchers report 10× to 1000× improvement in simulation performance on GPUs over conventional central processing units (CPUs). However, evaluations of GPU performance typically suffer from methodological shortcomings and expose (quietly reported) limitations in the GPUs themselves.

Lee et al. (2010) assess the claim of superlative GPU performance by running fourteen common scientific computing algorithms on a multi-core CPU and a GPU. They find that when each algorithm is carefully optimised for both platforms the GPU performs on average only 2.5× better than the CPU, despite the fact that the former is more than three times more powerful in terms of peak data-parallel floating-point operations per second.

Memory bandwidth dictates the rate at which data may be read from or written to memory, and limited bandwidth presents a significant problem to GPUs due to the immense rate at which the numerous arithmetic units consume or produce data. Nageswaran et al. (2009) report that the memory bottleneck is the ultimate performance limitation of their work; Pallipuram et al. (2011) note that certain computations cannot be usefully performed on the GPU because of the cost of transferring data from the CPU; Bhuiyan et al. (2010) show that the examined GPU only outperforms comparable multi-core processors when the number of floating-operations performed greatly exceeds the number of bytes transferred from or to memory; and Han and Taha (2010) find that a cluster of sixteen GPUs achieves only fourteen times more computations per second than a single GPU, presumably due to the cost of communicating spikes between neurons.

GPUs are relatively cheap commodity platforms which show excellent performance on certain algorithms, particularly those with high ratios of computation to communication. However, simulations of computationally-efficient neuron models have the opposite property, so GPUs are essentially unsuitable for the task and reports of their excellent performance relative to CPUs do not adequately control for differences in peak floating-point operations per second or power consumption.

2.2.4. Analogue circuits

Digital computers simulate continuous neural dynamics by numerical approximation, but neurons can also be emulated by analogue integrated circuits that use subthreshold transistor states to mimic transmembrane ion channels (Mead, 1989; Indiveri et al., 2011). Analogue circuits typically require few transistors and little power per neuron or synapse and may compute many tens or thousands of simulated seconds in a single real second, greatly outperforming their digital counterparts in most regards. However, digital computers are typically programmable, in that they execute instructions from a mutable memory, whereas the operation of analogue hardware is largely determined during fabrication. This may be a problem in the absence of a universal definition of neural dynamics because the design and tooling costs of integrated circuit manufacture are enormous, so the manufactured neurons must be 'correct' according to both current *and future* understandings of neural operation.

In practice, this problem is addressed by neural models that can reproduce many known spiking activities (Brette and Gerstner, 2005) but the problem of spike transmission, which so frustrates FPGAs and GPUs, has typically limited analogue implementations to a handful of emulated neurons connected by a crossbar switch. The BrainScaleS project intends to solve this problem (amongst others) by combining analogue circuitry for neural emulation with digital packet-switched routers for spike transmission (Schemmel et al., 2008, 2010). As with most integrated circuits, some hundreds of *analogue network chips* will be manufactured on a single silicon wafer but, in radical contrast to the usual process of slicing the wafer and packaging each chip separately, a communications fabric will be deposited directly on the wafer to form a single package. Inter-wafer communication will be handled by a backplane to which multiple wafers may be connected. This approach is a promising complement to SpiNNaker, in that it aims to support a similar scale and complexity of simulation at much greater speed for significantly less power, but at the expense of general-purpose programmability. We suspect that the ability to simulate arbitrary phenomena (hormone and neuromodulator diffusion, interaction with glial cells, neural development and death, and so on) in parallel with spiking neurons will prove to be a useful advantage of the SpiNNaker architecture, but we maintain that wafer-scale integration of analogue circuits may provide a powerful simulation tool.

3. Methodology

To investigate the capabilities and power requirements of SpiNNaker hardware, we simulated ten thousand neurons and four million synapses in a circuit inspired by detailed anatomical data from cat visual cortex. We recorded membrane potential traces from selected neurons, spike times from all neurons, and detailed power-consumption figures. From these data, we computed key power and energy metrics for spiking neuron simulation.

3.1. Hardware

SpiNNaker is a computer architecture designed to simulate very many neurons and synapses in real-time. By emulating the structure and function of neural computation (Furber and Temple, 2006) the architecture aims to address the 'Grand Challenge' of understanding the brain and mind (Hoare and Milner, 2005). This has motivated design of a machine which contains very many processors connected by a flexible, asynchronous communications network (Plana et al., 2007). An exhaustive description of architectural details would be excessive here, but we identify two features that are vital for large-scale, power-efficient simulation.

3.1.1. Processing

Each of the 2^{16} chips (Fig. 1) in a machine contains eighteen homogeneous processors: one for administration, sixteen for simulation, and one 'spare' to account for faults in the other seventeen. Each simulation processor is responsible for computing the dynamics of some number of neurons and their associated synapses, and has 32 kilobytes of instruction memory in which programs are stored, and 64 kilobytes of data memory to store program and neuron states; synapse states and any other simulation data are stored in a 128 megabyte off-chip memory. Each processor has a communications controller, through which neural spikes are communicated to and from the on-chip router, and a timer peripheral that generates periodic signals to prompt computation of neuron states.

As argued in Section 2.2.1, power consumption is a leading concern in large-scale spiking neuron simulation. A microprocessor dissipates *static power* as a function of supply voltage and transistor leak current, and *dynamic power* as a function of supply voltage and transistor switching frequency. So, increasing processor clock frequency increases dynamic power, but very high clock frequencies are only possible with low-threshold, high-leakage transistors that also increase static power dissipation. Consequently, for workloads such as spiking neuron simulation that may be efficiently shared, multiple low-performance processors may outperform a single high-performance processor in terms of both power consumption and computational throughput. For this reason, we use ARM968 processors, typically found in low-power embedded computers, to simulate populations of neurons and communicate spikes through a dedicated communications network.

3.1.2. Communication

Each chip contains a packet-switched router that forms links with all eighteen processors on-board and the routers of the six neighbouring chips. These routers may be programmed to emulate axons along which action potentials are transmitted. More formally: each neuron is uniquely identified by a 32 bit routing key; when a processor simulates an action potential, it sends the corresponding neuron's routing key to the on-chip router; the router compares this key against a routing table and delivers duplicate keys to one or more of the neighbouring routers or on-board processors; neighbouring routers repeat the process of look-up and delivery according to their own routing tables; and, finally, processors that receive routing keys induce synaptic currents in a subset of their own neurons selected according to the presynaptic neuron identified by the routing key. In this manner, SpiNNaker is specialised to communicate action potentials from any neuron to any subset of neurons in the machine. Furthermore, routers are programmable during simulation to account for axonal growth and decay, and are capable of processing packets at a rate which promises scalability to billion-neuron simulations (Navaridas et al., 2009, 2010).

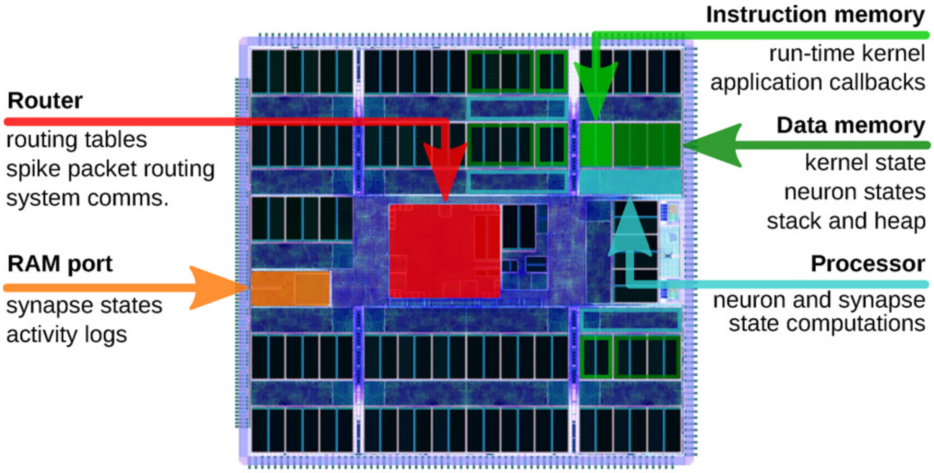

Fig. 1. A SpiNNaker chip containing eighteen processors and communication hardware specialised for spiking neural network simulation.

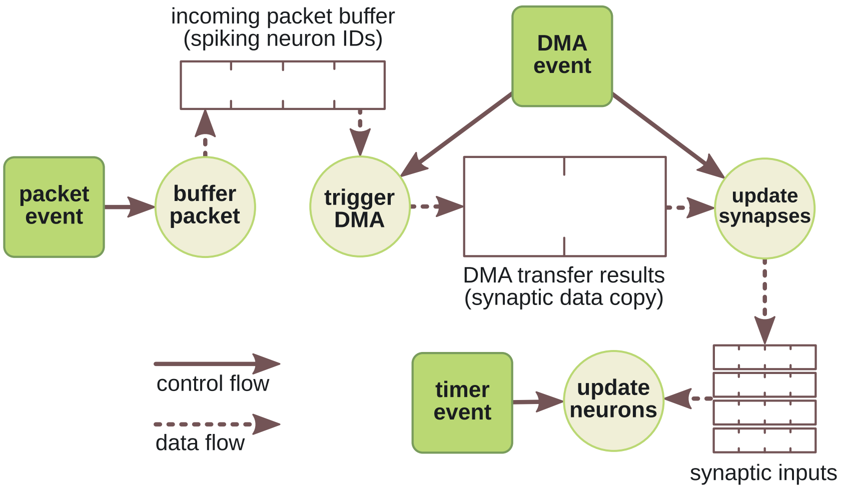

Fig. 2. Events and corresponding tasks in a typical SpiNNaker SNN simulation.

3.2. Software

We use two distinct software packages for the simulation and specification of SNNs.

3.2.1. SpiNNaker-side

SpiNNaker programs are *event-driven* (Fig. 2) in that all computational *tasks* follow from events in hardware (Sharp et al., 2011). These events should not be confused with those discussed by Brette (2006) in the analytical solution of membrane potential equations. On each processor, neuron states are computed in millisecond timesteps initiated by a local periodic *timer event*; at each timestep processors evaluate the membrane potentials of all of their neurons given prior synaptic inputs and deliver a packet to the router for each neuron that spikes. Spike packets are routed to all processors that model neurons efferent to the spiking neuron. Receipt raises a *packet event* that prompts the efferent processor to retrieve the appropriate synaptic weights from off-chip RAM using a background direct memory access (DMA) transfer. The processor is then free to perform other computations during the DMA transfer and is notified of its completion by a *DMA done event* that prompts calculation of the synaptic inputs to subsequent membrane potential evaluations. In this framework, a programmer may implement arbitrary neuron and synapse dynamics by writing standard sequential C code for the tasks corresponding to each event.

3.2.2. Host-side

A SpiNNaker machine is controlled by a conventional desktop computer (the *host*) which is used to specify models, trigger simulations and download results. We use the PyNN language (Davidson et al., 2009) to specify simulations in terms of populations (groups of homogeneous neurons and their dynamics) and projections (distributions or sets of homogeneous synapses and their dynamics, between two populations) without reference to the target simulator, which allows researchers to disregard the complexities of the underlying hardware or software.

We compile PyNN programs into simulation data structures (arrays of neuron and synapse states) in three main steps (Jin et al., 2010; Galluppi et al., 2010) using a program we call the Partitioning and Configuration Manager (PACMAN). Firstly, each population is allocated to one or more processors, depending on the population size and the maximum number of neuron states a processor can compute in one millisecond, which is in turn determined by the complexity of the neural dynamics; one binary file is then written for each processor, containing neuron data structures and population meta-data. Secondly, routes are computed to convey spikes between processors along the specified projections and routing tables for each chip are written to binary files. Thirdly, the synapses arising from each projection are computed from the specified distribution or set and corresponding binary files are produced for each chip.

Program code and data structures are finally loaded into a SpiNNaker machine via Ethernet and simulations are triggered through the same medium. Membrane potential traces, spikes, and aggregated data (such as computed local field potentials) are then either sent to the host over Ethernet in real-time during simulation, or stored in memory for later retrieval.

3.3. Microcircuit model

We constructed a cortical microcircuit model of around ten thousand neurons and four million synapses according to composition and connectivity data from Binzegger et al. (2004) reproduced in detail by Izhikevich and Edelman (2008, supporting material). We preserved the proportional sizes of each population and the proportions of synapses received by each population from each population, but were unable to preserve exact projection probabilities between populations of the canonical microcircuit: the number of synapses received by each neuron according to Peters' rule (Eq. 1) is independent of the number of neurons in the microcircuit, so changing the number of neurons inevitably changes the projection probabilities. We translated the absolute numbers of synapses per neuron given by Izhikevich and Edelman into projection probabilities P_{ij} from population j to population i for use in PyNN:

$$P_{ij} = \frac{C \cdot S_i \cdot Sp_{ij}}{N_j} \tag{4}$$

where N_j is the number of neurons in population j, S_i is the total number of synapses received by a neuron in population i, Sp_{ij} is the proportion of those synapses received from neurons in population j, and $C = 0.1$ is an arbitrary coefficient we selected to prevent absurdly dense connectivity in a microcircuit of just ten thousand neurons. Where synapse numbers and proportions were listed for multiple dendritic compartments of the same cell, we simply summed across these values to create a single compartment. Projections with a probability of <0.01 were pruned. Table 1 lists the total numbers of neurons and synapses in the model and Table 2 contains the matrix of projection probabilities between populations.

Table 1
Neurons and synapses per population. Layer identifiers in brackets indicate the primary target of innervation for that population thereby distinguishing, for example, L5 pyramids that project into L2/3 from those which project into L5 and L6.

Population	Neurons	Synapses
nb1	150	24,150
p2/3	2600	1,666,599
b2/3	310	103,850
nb2/3	420	116,760
ss4(L4)	920	401,120
ss4(L2/3)	920	331,200
p4	920	430,560
b4	540	114,479
nb4	150	37,500
p5(L2/3)	480	221,760
p5(L5/6)	130	98,540
b5	60	15,780
nb5	80	21,040
p6(L4)	1360	522,240
p6(L5/6)	450	125,100
b6	200	22,000
nb6	200	22,000
Total	9890	4,284,568

Excitatory pyramidal cells (denoted p*l* for pyramids in layer *l*) and stellate cells (ss*l*) were simulated as regular spiking neurons, and inhibitory basket cells (b*l*) and non-basket (nb*l*) cells were simulated as fast spiking neurons, all according to the simple model of spiking neurons by Izhikevich (2003)

$$\dot{\upsilon} = 0.04\upsilon^2 + 5\upsilon + 140 - u + I \tag{5}$$

$$\dot{u} = a(b\upsilon - u) \tag{6}$$

$$\text{if } \upsilon \geq 30\,\text{mV} \quad \upsilon \leftarrow c,\, u \leftarrow u + d \tag{7}$$

which was numerically approximated at a 0.5 ms timestep using an approach similar to that of Jin et al. (2008).

The synaptic response to an action potential was simulated as an instantaneous rise and exponential decay of input current *I* to the postsynaptic neuron; synaptic weights (the amount of the instantaneous rise) and synaptic time constants between excitatory and inhibitory neurons are listed in Table 3. As with synaptic densities, synaptic weights were chosen arbitrarily to preserve stable network activity; time constants, however, are approximately those suggested by Thomson et al. (2002).

To induce spiking activity in the network, we simulated background synaptic currents using a gross approximation of an Ornstein–Uhlenbeck process inspired by Destexhe et al. (2001).

Table 3
Synaptic weights (w) and time constants (τ) between excitatory and inhibitory cells.

		Pre	
		Exci.	Inhi.
Post	Exci.	$w = 0.05$, $\tau = 30$	$w = 0.05$, $\tau = 30$
	Inhi.	$w = 0.2$, $\tau = 15$	$w = 0.1$, $\tau = 15$

Each neuron received a time-varying background current $I_b(t)$ which was computed as

$$I_b(t+1) = \overline{I_b^C} + \tau\left(I_b(t) - \overline{I_b^C}\right) + A \cdot U(-0.5, +0.5) \tag{8}$$

where $\overline{I_b^C}$ is the mean background current for cell class C, τ is proportional to the process time constant, and A is the amplitude of the fluctuations drawn from the uniform random distribution U. We set $\overline{I_b^p} = 2.8$ for pyramids, $\overline{I_b^b} = 2.8$ for baskets, $\overline{I_b^{nb}} = 0.5$ for nonbaskets, $\tau = 0.9$, $A = 0.85$, and generated pseudo-random numbers from U with linear feedback shift register. Again, we chose $\overline{I_b^C}$ to produce stable network activity, not to reflect known values of biological neurons.

3.4. Simulation and recording

We simulated the cortical microcircuit on a four-chip test board containing a total of seventy-two processors: four for administration, sixty-four for simulation, and four spares. PACMAN allocated neural populations to fifty processors and assigned a further four, one on each chip, to aggregate spikes from neighbouring processors and forward these data to the host in real-time; the remaining ten simulation processors were left idle.

We used an oscilloscope (LeCroy 9344CM, LeCroy, NY, USA) to record the power consumed during simulation by measuring the voltage drops across 0.1 Ohm resistors placed on the 1.2 V and 1.8 V power rails, which supply power to the SpiNNaker chips and their RAMs respectively. We were limited by the oscilloscope to forty five thousand samples per trial, so we chose to measure at one million samples per second as the maximum temporal resolution at which we could observe any significant period of network activity.

In the first experiment, we observed the average power drawn during simulation of the microcircuit, and took readings from $t = 5100$ ms to $t = 5145$ ms; this period was chosen arbitrarily, other than to allow the network to settle into regular activity. To better understand the power consumption of the simulation, we conducted two further experiments across the sample period: in the

Table 2
Projection probability (the chance of a synapse occurring between a pair of cells) for each pair of populations in the model.

		Presynaptic population																
		nb1	p2/3	b2/3	nb2/3	ss4(L4)	ss4(L2/3)	p4	b4	nb4	p5(L2/3)	p5(L5/6)	b5	nb5	p6(L4)	p6(L5/6)	b6	nb6
Postsynaptic population	nb1	.600	.022	.016	.024	.	.	.	.	.	.	.	.	.	.	.	.	.
	p2/3	.087	.137	.171	.064	.	.043	.049	.	.033	.090	.	.	.	.010	.	.	.230
	b2/3	.033	.077	.132	.031	.	.024	.027	.	.020	.050	.	.	.	.	.	.	.015
	nb2/3	.040	.062	.123	.026	.	.020	.023	.	.020	.042	.	.	.	.	.	.	.
	ss4(L4)	.	.	.	.	.075	.023	.026	.076	.080	.096	.	.	.	.139	.	.	.170
	ss4(L2/3)	.	.011	.	.	.061	.021	.024	.067	.073	.013	.	.	.	.114	.	.	.135
	p4	.053	.031	.016	.017	.063	.026	.030	.067	.073	.027	.	.	.	.118	.	.	.185
	b4	.	.	.	.	.039	.013	.015	.050	.053	.	.	.	.	.072	.	.	.085
	nb4	.	.	.	.	.047	.014	.016	.056	.060	.	.	.	.	.087	.	.	.105
	p5(L2/3)	.013	.087	.032	.	.020	.014	.040	.	.033	.113	.031	.067	.062	.015	.020	.	.275
	p5(L5/6)	.387	.135	.052	.031	.032	.026	.057	.	.047	.144	.054	.067	.062	.035	.029	.	.320
	b5	.	.052	.023	.	.011	.	.024	.	.020	.073	.023	.050	.050	.	.013	.	.170
	nb5	.	.052	.023	.	.011	.	.024	.	.020	.073	.023	.050	.050	.	.013	.	.170
	p6(L4)	.	.027	.032	.	.032	.014	.024	.019	.027	.031	.131	.017	.025	.052	.100	.125	.245
	p6(L5/6)	.	.017	.	.	.	.	.014	.	.020	.013	.215	.	.025	.012	.164	.215	.240
	b6	.	.	.	.	.	.	.	.	.	.	.123	.	.013	.	.096	.125	.125
	nb6	.	.	.	.	.	.	.	.	.	.	.123	.	.013	.	.096	.125	.125

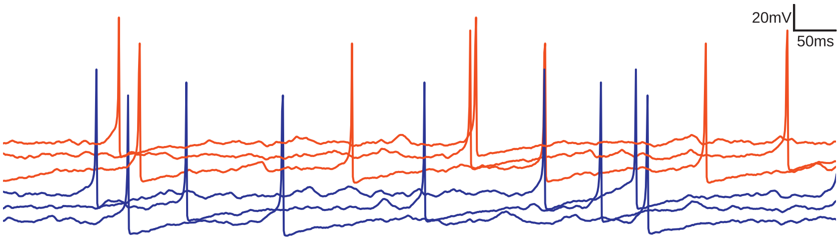

Fig. 3. Membrane potential traces from three spiny stellate cells (blue) and three basket cells (red) in layer 4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

first, we disabled transmission of spikes from all neurons; in the second, we enabled spikes but did not include synaptic currents in the membrane potential computations. Thus, the two experiments showed identical patterns of spiking, but the former allowed us to observe the amount of power drawn only by neuron state computation, and we were then able to attribute the additional power consumed by the latter to synaptic activity. From these figures, we were able to derive two key metrics: energy-per-neuron-millisecond and energy-per-synaptic-event.

4. Results

4.1. Network activity

We simulated a detailed cortical microcircuit primarily to understand the power requirements of biologically plausible models on SpiNNaker. Nevertheless, we present two recordings of simulation activity to demonstrate both the model sanity and the hardware capabilities. Fig. 3 shows one second of membrane potential traces from three spiny stellate cells (blue) and three basket cells (red) in layer 4. Membrane potential spikes in Fig. 3 are each represented by single dots in Fig. 4, in which one second of network activity, stratified by population, clearly shows the different spiking rates of excitatory and inhibitory cells and weak oscillations arising from recurrent inhibitory connections. Trivial analysis of these data (Fig. 5(a)) shows global-average firing rate to oscillate around 4 Hz, with the majority of cells (Fig. 5(b)) firing somewhat more frequently than suggested by Neymotin et al. (2011) but nevertheless close to biologically-plausible rates. The firing rate for the simulation without weights (not shown) was largely constant over time but showed some variation due to the low-quality pseudo-random numbers used for the background currents.

We counted the number of spikes produced by each population in each millisecond and used model connectivity data to calculate the number of *synaptic events* (of which a single instance is one action potential hitting one synapse) that occurred each period. From these data we were able to reason about the energy cost of each event.

4.2. Power consumption

We found an average power consumption of 1.951 W over the 45 ms of observed simulation in the first experiment, in which all spikes were delivered and synaptic currents were effected. When we disabled synaptic currents the simulation drew 1.882 W, and disabling spikes gave a power draw of 1.096 W.

So, we can estimate the energy required by SpiNNaker to simulate one of the ten thousand Izhikevich neurons in the model for one millisecond as

$$\frac{1.096\,\mathrm{W}\cdot 10^{-3}\,\mathrm{s}}{10,000\,\mathrm{neurons}} \approx 100\,\mathrm{nJ/neuron/ms} \tag{9}$$

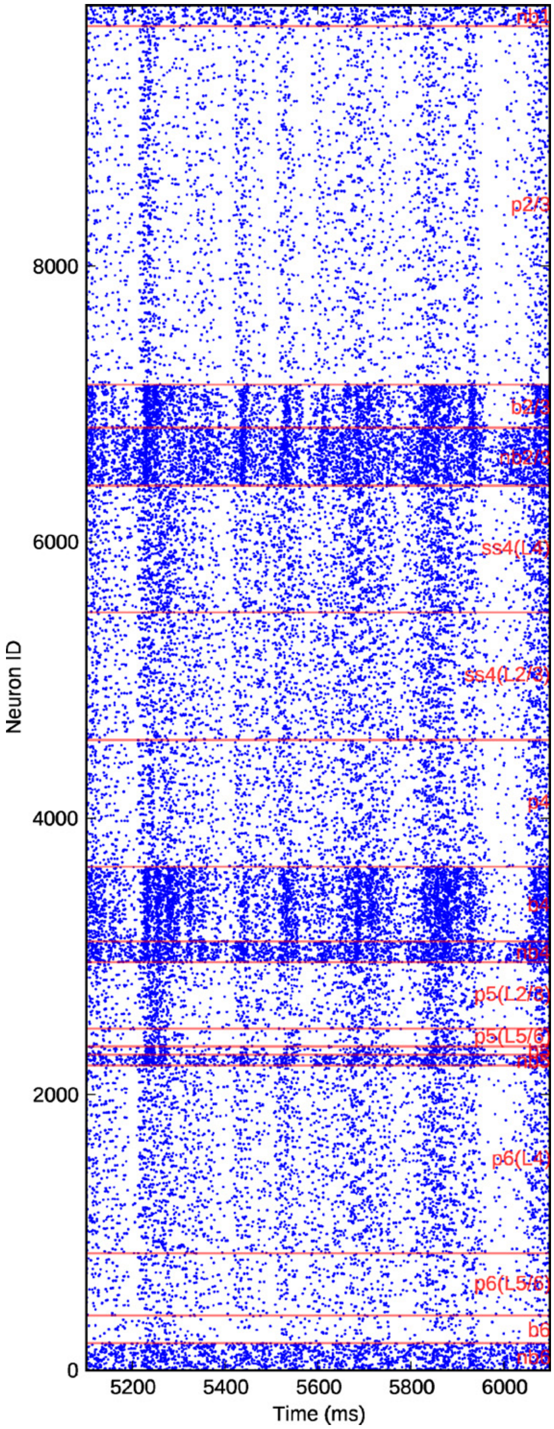

Fig. 4. Spiking activity of the network in a one-second period.

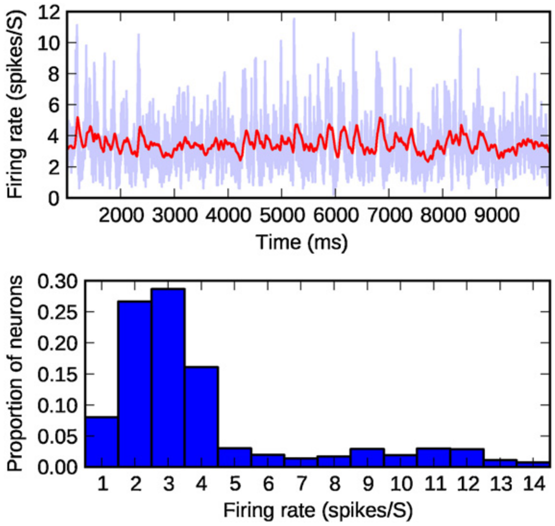

Fig. 5. Top: instantaneous (blue) and smoothed (red, first-order low-pass filter, $\tau = 100$ ms) average firing rate of all neurons. Bottom: proportion of neurons at each firing rate, over the simulation period shown above. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We can attribute 1.882 W − 1.096 W = 0.786 W to spikes and synaptic events, of which there were 1954 and 816,584 respectively in the observed 45 ms period. This gives an energy-per-synaptic-event of

$$\frac{0.786\,\mathrm{W} \cdot 45 \cdot 10^{-3}\,\mathrm{s}}{816,584\,\mathrm{s.e.}} = 43\,\mathrm{nJ/s.e.} \tag{10}$$

Fig. 6 shows power consumption in the three experiments as a function of time. Processor activity is clearly driven by the millisecond timer event that prompts neuron states to be computed: when spikes are not transmitted (green trace) this short period of activity is followed by a passive state in which only static leakage power is consumed; in the other two experiments (without weights in red, full simulation in blue) this period is significantly longer as packet and DMA events interrupt to demand attention from the processor, and overall power consumption is increased by on-going DMA transfers. Full simulation consumes more power than simulation without weights as a result of the greater number of spikes caused by recurrent excitation. The exception is in millisecond 5100, as a trough in full simulation firing rates coincides with a random peak in firing rates in the simulation without weights.

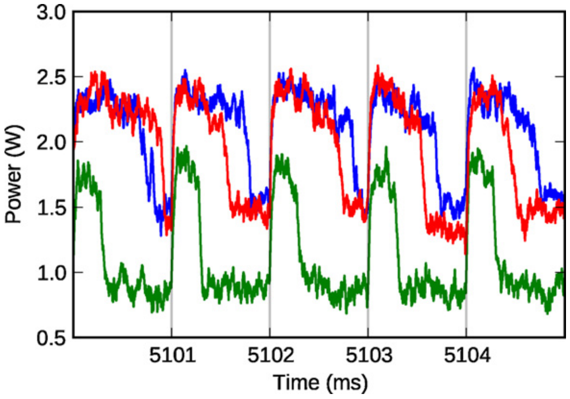

Fig. 6. Power traces from full simulation (blue), minus weights (red) and minus spikes (green), smoothed by first-order low-pass filter, $\tau = 20\,\mu$s. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Discussion

Simulation of the cortex is a promising methodology for investigating higher brain function, and recent developments in anatomical techniques and computer hardware have precipitated models of an unprecedented scale and fidelity. However, these simulations require vast amounts of electrical power and great lengths of time, which hinders scientifically-useful experimentation. We seek to address this problem with a low-power architecture dedicated to real-time simulation of spiking neurons, SpiNNaker. We have simulated an anatomically-inspired cortical microcircuit of ten thousand neurons and four million synapses using four SpiNNaker chips and less than two watts, and calculated energy consumption of 100 nJ per neuron per millisecond and 43 nJ per postsynaptic potential or *synaptic event*. To our knowledge, this is the most power-efficient simulation of biologically-plausible neural circuits on digital hardware to date. Further comparison with related work is difficult because the power problem is rarely discussed, and although we are confident of architectural scalability (Navaridas et al., 2009, 2010) we have yet to address the infrastructural challenges faced by large machines such as Blue Gene; it would be appropriate, for example, to double all of the power and energy figures reported here to account for inevitable losses in voltage regulation, cooling and other infrastructure required by a large computer (Hölzle and Barroso, 2009). Nevertheless, we have demonstrated the considerable capabilities of the first digital, programmable, multiprocessor chip dedicated to spiking neuron simulation, and suggested the power-feasibility of fast simulation of a large cortical volume on a fifty-thousand-chip SpiNNaker machine.

The human cortex is a fascinating computer characterised by massive parallelism superlative power-efficiency. To date, efforts to simulate the cortex have sought only processing parallelism, at such great power costs as to make the ultimate endeavour infeasible. Simulations of the cortex that run quickly and cheaply enough to be scientifically useful must mimic the brain in three key regards: massive parallelism in the form of thousands or millions of processors, hardware designed for intricate communication patterns, and extreme tuning for the power-efficiency of the machine.

Acknowledgements

This work is supported by the Engineering and Physical Sciences Research Council of the United Kingdom and ARM Holdings plc. Generous technical support was given by Ernie Hill, Luis A. Plana and Steve Temple.

References

Ananthanarayanan R, Esser SK, Simon HD, Mohda DS. The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses. In: Conference on high performance computing, networking, storage and analysis; 2009. p. 1–12.

Bhuiyan MA, Pallipuram VK, Smith MC. Acceleration of spiking neural networks in emerging multi-core and GPU architectures. In: International symposium on parallel distributed processing; 2010. p. 1–8.

Binzegger T, Douglas RJ, Martin KAC. A quantitative map of the circuit of cat primary visual cortex. J Neurosci 2004;24(39):8441–53.

Binzegger T, Douglas RJ, Martin KAC. Stereotypical bouton clustering of individual neurons in cat primary visual cortex. J Neurosci 2007;27(45):12242–54.

Binzegger T, Douglas RJ, Martin KAC. Topology and dynamics of the canonical circuit of cat V1. Neural Netw 2009;22(8):1071–8.

Braitenberg A, Schüz A. Anatomy of the cortex: statistics and geometry. 1st ed. Springer-Verlag; 1991.

Brette R. Exact simulation of integrate-and-fire models with synaptic conductances. Neural Comput 2006;18:2004–27.

Brette R, Gerstner W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. J Neurophysiol 2005;94(5):3637–42.

Cassidy A, Andreou AG, Georgiou J. Design of a one million neuron single FPGA neuromorphic system for real-time multimodal scene analysis. In: Annual conference on information sciences and systems; 2011. p. 1–6.

Davidson AP, Brüderle D, Eppler JM, Kremkow J, Muller E, Pecevski D, et al. PyNN: a common interface for neuronal network simulators. Front Neuroinform 2009;2(0):1–10.

Destexhe A, Rudolph M, Fellous JM, Sejnowski TJ. Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. Neuroscience 2001;107:13–24.

Douglas RJ, Martin KAC, Whitteridge D. A canonical microcircuit for neocortex. Neural Comput 1989;1(4):480–8.

Fatahalian K, Houston M. A closer look at GPUs. Commun ACM 2008;51(10):50–7.

Fidjeland AK, Shanahan MP. Accelerated simulation of spiking neural networks using GPUs. In: International joint conference on neural networks; 2010. p. 1–8.

Furber S, Temple S. Neural systems engineering. J Royal Soc Interface 2006;4(13):193–206.

Galluppi F, Rast A, Davies S, Furber S. A general-purpose model translation system for a universal neural chip. In: International conference on neural information processing; 2010. p. 58–65.

Gara A, Blumrich MA, Chen D, Chiu GLT, Coteus P, Giampapa ME, et al. Overview of the Blue Gene/L system architecture. IBM J Res Dev 2005;49(2–3):195–212.

de Garis H, Shuo C, Goertzel B, Ruiting L. A world survey of artificial brain projects, Part I: large-scale brain simulations. Neurocomputing 2010;74(1–3):3–29.

Haeusler S, Maass W. A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. Cereb Cortex 2006;17(1): 149–62.

Han B, Taha TM. Neuromorphic models on a GPGPU cluster. In: International joint conference on neural networks; 2010. p. 1–8.

Hoare T, Milner R. Grand challenges for computing research. Comput J 2005;48(1):49–52.

Hölzle U, Barroso LA. The datacenter as a computer: an introduction to the design of warehouse-scale machines. 1st ed. Morgan & Claypool; 2009.

Indiveri G, Linares-Barranco B, Hamilton TJ, van Schaik A, Etienne-Cummings R, Delbruck T, et al. Neuromorphic silicon neuron circuits. Front Neurosci 2011;5(0):1–21.

Izhikevich EM. Simple model of spiking neurons. IEEE Trans Neural Netw 2003;14(6):1569–72.

Izhikevich EM. Which model to use for cortical spiking neurons? IEEE Trans Neural Netw 2004;15(5):1063–70.

Izhikevich EM. Dynamical systems in neuroscience: the geometry of excitability and bursting. 1st ed. The MIT Press; 2007.

Izhikevich EM, Edelman GM. Large-scale model of mammalian thalamocortical systems. Proc Natl Acad Sci 2008;105(9):3593–8.

Jin X, Furber SB, Woods JV. Efficient modelling of spiking neural networks on a scalable chip multiprocessor. In: International joint conference on neural networks; 2008. p. 2812–9.

Jin X, Galluppi F, Patterson C, Rast A, Davies S, Temple S, et al. Algorithm and software for simulation of spiking neural networks on the multi-chip SpiNNaker system. In: International joint conference on neural networks; 2010. p. 1–8.

Kandel ER, Schwartz JH, Jessell TM. Principles of neural science. 4th ed. McGraw-Hill Medical; 2000.

Lee VW, Kim C, Chhugani J, Deisher M, Kim D, Nguyen AD, et al. Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. In: International symposium on computer architecture; 2010. p. 451–60.

Markram H. The Blue Brain Project. Nat Rev Neurosci 2006;7(2):153–60.

Mead C. Analog VLSI and neural systems. 1st ed. Addison-Wesley; 1989.

Nageswaran JMM, Dutt N, Krichmar JL, Nicolau A, Veidenbaum AV. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. Neural Netw 2009;22(5–6):791–800.

Navaridas J, Luján M, Miguel-Alonso J, Plana LA, Furber S. Understanding the interconnection network of SpiNNaker. In: International conference on supercomputing; 2009. p. 286–95.

Navaridas J, Plana LA, Miguel-Alonso J, Luján M, Furber SB. SpiNNaker: impact of traffic locality, causality and burstiness on the performance of the interconnection network. In: International conference on computing frontiers; 2010. p. 11–20.

Neymotin SA, Heekyung L, Park E, Fenton AA, Lytton WW. Emergence of physiological oscillation frequencies in a computer model of neocortex. Front Comput Neurosci 2011;5(19):1–17.

Pallipuram VK, Bhuiyan MA, Smith MC. Evaluation of GPU architectures using spiking neural networks. In: Symposium on application accelerators in high-performance computing; 2011. p. 93–102.

Pearson M, Gilhespy I, Gurney K, Melhuish C, Mitchinson B, Nibouche M, et al. A real-time, FPGA based, biologically plausible neural network processor. In: International conference on artificial neural networks; 2005. p. 1021–6.

Peters A, Feldman ML. The projection of the lateral geniculate nucleus to area 17 of the rat cerebral cortex. I. General description. J Neurocytol 1976;5(1): 63–84.

Plana LA, Furber SB, Temple S, Khan M, Shi Y, Wu J, et al. A GALS infrastructure for a massively parallel multiprocessor. IEEE Des Test Comput 2007;24(5):454–63.

Rice KL, Bhuiyan MA, Taha TM, Vutsinas CN, Smith MC. FPGA implementation of Izhikevich spiking neural networks for character recognition. In: International conference on reconfigurable computing and FPGAs; 2009. p. 451–6.

Salapura V, Bickford R, Blumrich M, Bright AA, Chen D, Coteus P, et al. Power and performance optimization at the system level. In: International conference on computing frontiers; 2005. p. 125–32.

Schemmel J, Briiderle D, Griibl A, Hock M, Meier K, Millner S. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In: International symposium on circuits and systems; 2010. p. 1947–50.

Schemmel J, Fieres J, Meier K. Wafer-scale integration of analog neural networks. In: International joint conference on neural networks; 2008. p. 431–8.

Sharp T, Plana LA, Galluppi F, Furber S. Event-driven simulation of arbitrary spiking neural networks on SpiNNaker. In: International conference on neural information processing; 2011. p. 424–30.

Sterling T, Becker DJ, Savarese D, Dorband JE, Ranawake UA, Packer CV. BEOWULF: a parallel workstation for scientific computation. In: International conference on parallel processing; 1995. p. 11–4.

Symes A, Wennekers T. Spatiotemporal dynamics in the cortical microcircuit: a modelling study of primary visual cortex layer 2/3. Neural Netw 2009;22(8):1079–92.

Thomson AM, Lamy C. Functional maps of neocortical local circuitry. Front Neurosci 2007;1:19–42.

Thomson AM, West DC, Wang Y, Bannister AP. Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2–5 of adult rat and cat neocortex: triple intracellular recordings and biocytin labelling in vitro. Cereb Cortex 2002;12(9):936–53.