Neural Networks 45 (2013) 39–49

Contents lists available at SciVerse ScienceDirect

Neural Networks

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

2013 Special Issue

Computing with networks of spiking neurons on a biophysically motivated floating-gate based neuromorphic integrated circuit

S. Brink*, S. Nease, P. Hasler

Georgia Institute of Technology, Technology Square Research Building, Atlanta, GA 30308, USA

ARTICLE INFO

Keywords:
Neuromorphic VLSI
Floating-gate transistor
Single transistor learning synapse
Spiking winner-take-all
Synfire chain

ABSTRACT

Results are presented from several spiking network experiments performed on a novel neuromorphic integrated circuit. The networks are discussed in terms of their computational significance, which includes applications such as arbitrary spatiotemporal pattern generation and recognition, winner-take-all competition, stable generation of rhythmic outputs, and volatile memory. Analogies to the behavior of real biological neural systems are also noted. The alternatives for implementing the same computations are discussed and compared from a computational efficiency standpoint, with the conclusion that implementing neural networks on neuromorphic hardware is significantly more power efficient than numerical integration of model equations on traditional digital hardware.

1. Introduction

It has been said that in order to fully understand something, you must figure out how to build it. While this blanket statement may not be true in all cases, we believe that developing neuromorphic hardware will ultimately lead to a better understanding of biological neural systems and the principles upon which their computation is based. We design neuromorphic systems with the expectation that they will be useful for improving our knowledge of neuroscience and computational science. These sciences in turn inform the design of improved neuromorphic systems. This feedback path is illustrated in Fig. 1, which emphasizes the role of computational experiments performed on neuromorphic hardware. Experiments on neuromorphic hardware not only inform future neuromorphic designs, but also have significantly different constraints than numerical simulations have. Analog implementations model biology using the physics inherent to the devices, which is more power efficient than creating a complex digital system to represent the same equations. Therefore, in signal processing applications where low power and modest precision are required, analog neuromorphic systems are more desirable than their digital equivalents (Douglas, Mahowald, & Mead, 1995). Thus, such experiments are expected to be a stimulant of creative approaches to thinking about neural systems and computation. In this work, we demonstrate results from real-time "simulations" of neural networks with up to 100 neurons. In this context, a "simulation" is an experiment run on our neuromorphic hardware, in which we attempt to simulate the behavior of biological neural networks using our silicon models of neurobiology. These networks perform computationally relevant functions such as arbitrary spatiotemporal pattern generation and recognition, winner-take-all competition, stable generation of rhythmic outputs, and volatile memory.

Previously we reported the design and measurements of a neuron integrated circuit (IC) with 100 neurons, 30,000 synapses, capability to implement spike-timing dependent plasticity (STDP), and an address-event representation (AER) interface (Brink et al., 2013). This study takes this further, not by simply showing a working IC capable of implementing models, but by putting this IC to work as a platform for investigating these models. This demonstration serves several purposes. First, it helps to highlight some interesting approaches to network computation that fit nicely with the neuromorphic circuit models on this particular system. Further, it allows for a kind of benchmarking comparison among the various neuromorphic hardware systems in use today. It also makes a contribution to the case for the neuromorphic very-large-scale integration (VLSI) approach, as it is another instance of a neuromorphic design facilitating the study of spiking networks.

Section 2 gives a description of the system on which the simulations were performed. This section is intended to be accessible to readers that do not have a circuits background. Section 3 covers the basic building blocks that inform the intuition for understanding the results in the networks. Section 4 describes the networks that were simulated and presents the measured data. Section 6 analyzes the computation performed by these networks and by the neuromorphic platform, with a comparison to alternative approaches.

* Corresponding author.
E-mail addresses: Stephen.Brink@gatech.edu (S. Brink), Stephen.Nease@gatech.edu (S. Nease), phasler@ece.gatech.edu (P. Hasler).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.02.011

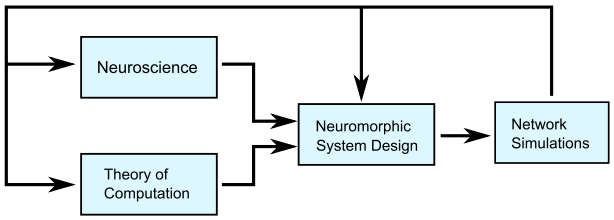

Fig. 1. Closing the loop around neuromorphic systems. Clearly, our understanding of neuroscience and the theory of computation influences the designs of neuromorphic systems. Additionally, by using neuromorphic hardware to perform nontrivial neural modeling, we can gain new insights into those foundational sciences.

2. Neuromorphic platform overview

This section covers the details about the neuromorphic platform that are required for understanding of the system results. Fig. 2 shows the block-level design of the neuromorphic IC as well as the die photo; the IC consists of 100 model neurons, 30,000 model synapses, and an address-event representation (AER) module for digital communication of spiking inputs and outputs.

The model neurons are biophysically inspired channel-based models, which were originally introduced in Farquhar and Hasler (2005), and which exhibit Class 2 excitability (Basu, Petre, & Hasler, 2008). This particular model is unique to our system. Other systems typically implement extensions of integrate-and-fire (IF) dynamics. For example, Gao, Benjamin, and Boahen (2012) implements a quadratic IF model, Yu, Park, Joshi, Maier, and Cauwenberghs (2012) implements a two-compartment leaky IF model, and Indiveri, Chicca, and Douglas (2006) also uses a leaky IF model with features such as spike-frequency adaptation, tunable refractory period, and voltage threshold modulation.

The synapses are implementations of the single transistor learning synapse (STLS) described in Hasler, Diorio, Minch, and Mead (1995). They produce post-synaptic current (PSC) waveforms that approximate those measured in biological neurons, with programmable rise and fall times. Triangle waveform generator circuits, shown in Fig. 2(b), precondition the neurons' spiking outputs to set the time course of these PSCs.

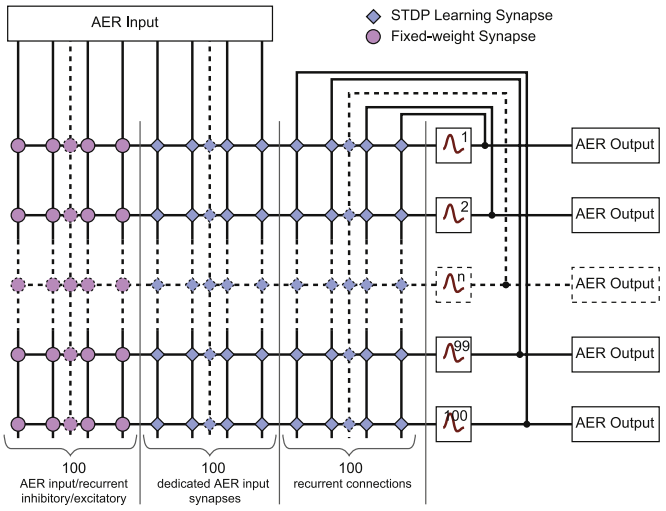

Fig. 3. Schematic depiction of the chip architecture with the partition of synapses into recurrent connections, AER inputs, learning-enabled, fixed weight, and configurable excitatory/inhibitory synapses. All of the STDP synapses are excitatory.

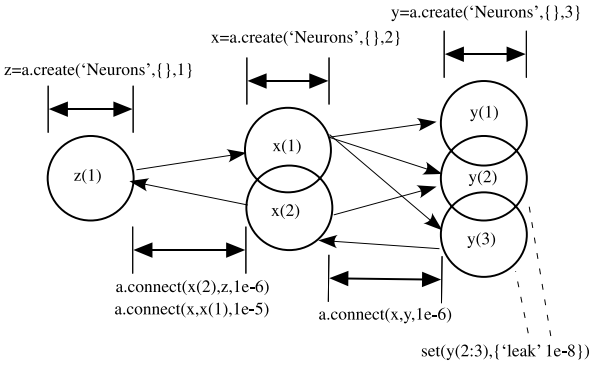

Fig. 4. PyNN-based Matlab code for setting up a network simulation.

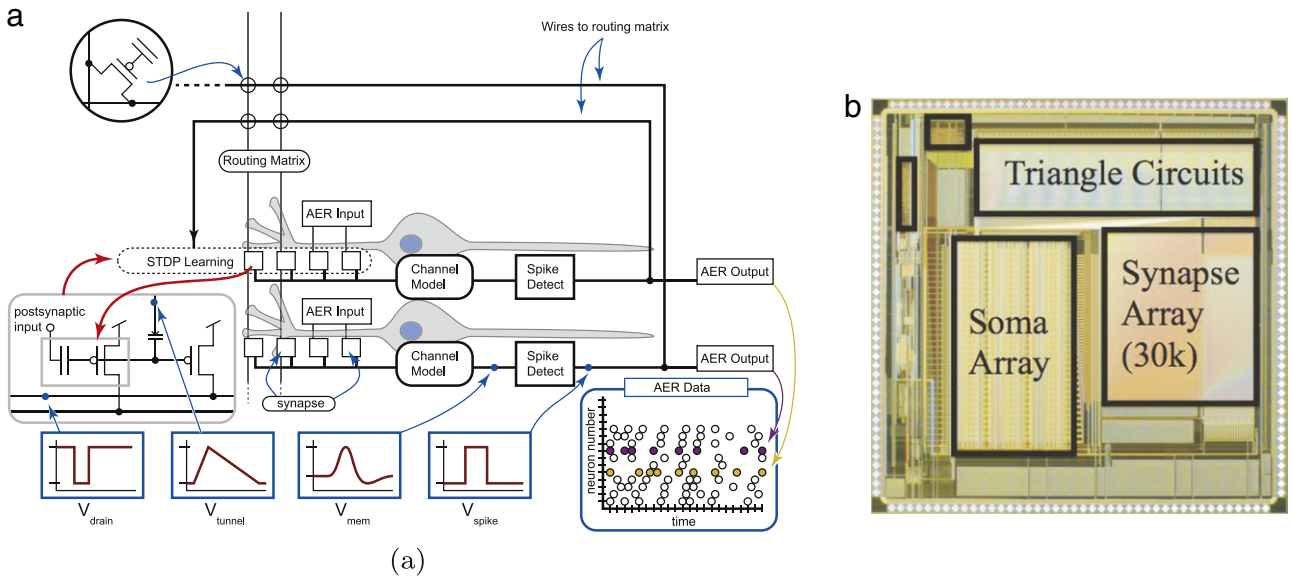

Fig. 2. Overview of neuron IC architecture. (a) Illustration of the correspondence between elements in the integrated circuit with those in a biological neural network. The waveforms that occur in a single neuron block during and after an action potential are shown. Note that the connectivity allows for recurrent connections among the neurons as well as synaptic inputs that are controlled by the AER receiver, and that all spiking activity may be read from the AER transmitter. (b) Die photo of the neuron IC. The synapse and neuron arrays as well as the array of triangle waveform generator circuits for the synapse array are labeled, and boxes denote the locations of the AER receiver and transmitter.

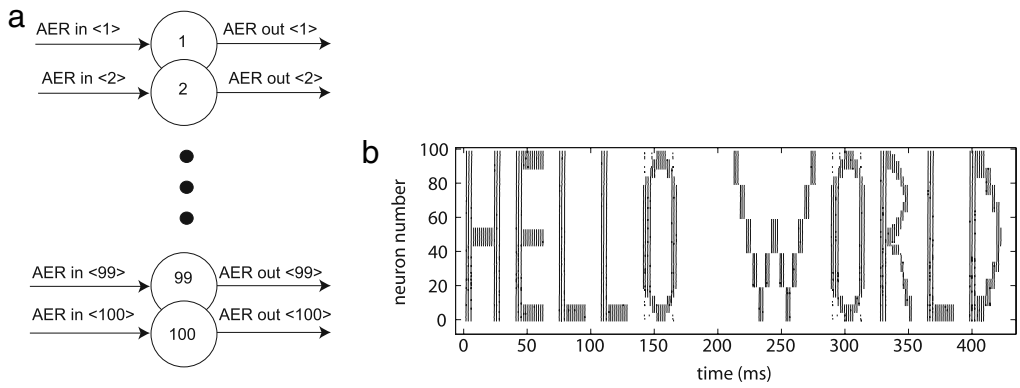

Fig. 5. Results from the dot matrix experiment. (a) Diagram of the connectivity. The synaptic weights are set sufficiently high that a single input reliably elicits a spike from the target neuron. (b) Measured outputs from this circuit. Only the spiking output activity of the neurons is depicted, but the pattern of AER inputs is essentially the same.

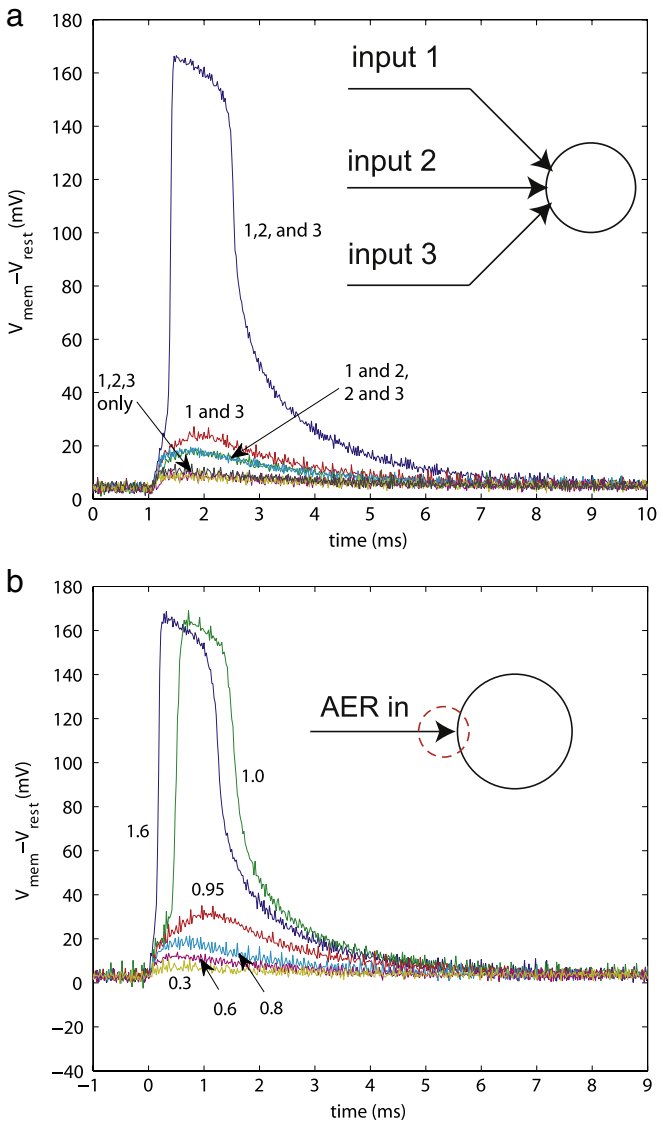

Fig. 6. Measurements of membrane voltage response to synaptic inputs, illustrating that the circuit model produces an action potential when the spiking threshold is exceeded. All waveforms in this figure are averaged over about 20 captures in order to reduce the effect of power supply noise that is present in the measurement system. (a) Three input synapses with approximately equal conductance are connected, and the responses to various subsets of them are shown. (b) The response to synaptic inputs of varying amplitudes is shown (the synaptic weight, normalized to the minimum weight that produced a spike, is shown).

The STLS structure provides compact local storage of continuously variable synaptic weights (i.e., PSC amplitudes) by using floating-gate transistors, the core technology in electrically erasable programmable read-only memory (EEPROM). This approach differs from other systems. Rather than explicitly changing the conductance of synapses, Choudhary et al. (2012) uses a constant synaptic conductance and encodes the weight as the probability of a spike's being transmitted to a neuron (as described in Goldberg, Cauwenberghs, & Andreou, 2001). While other systems time-multiplex synaptic inputs (Yu & Cauwenberghs, 2010), we have one STLS per neuron pair.

Our STLS structure leads to a highly flexible simulation platform with a very dense storage of synaptic weights. A comparison of synapse density for other neuromorphic network simulator ICs is given in Table 1. The use of floating-gate transistors also allows for a fairly simple implementation of synaptic plasticity via STDP, and significantly mitigates the effects of process variation that increasingly plague IC technologies as we scale to processes with smaller feature sizes.

Two thirds of the synapses in the array are able to implement STDP according to the scheme described in Ramakrishnan, Hasler, and Gordon (2011). The large number of synapses allows the simulation of any network of 100 or fewer neurons (i.e., an inhibitory or excitatory synapse may be made between any pair of neurons). In addition to the synapses between neurons, our system includes synapses from the digital AER interface onto each neuron. This allows for a wide range of complex input stimulation patterns to the networks being simulated. Fig. 3 shows the connectivity of the synapse array in more detail.

We designed the IC according to the paradigm of using floating-gate transistor technology to allow greater reconfigurability than other analog systems. As mentioned above, these devices enable us to create configurable synaptic connections. However, floating gates also allow us to program and store analog parameters for other circuits used in the chip. For example, biases in the neuron channel models and gate waveform shapers can be precisely trimmed using floating-gate programming techniques. This allows us to remove many of the offsets which result from manufacturing variation, which is an increasingly important topic in subthreshold circuit design.

We run simulations on the IC by using the benchtop testing system described in Koziol et al. (2010). The system consists of a custom circuit board with multi-channel digital to analog converters (DACs) and analog to digital converters (ADCs), as well as an Atmel microcontroller. This allows data acquisition, programming of the floating-gate transistors, and biasing of the circuits. All of these functions are controlled by Matlab commands, which are sent from a computer to the board via a USB connection (which can be the sole power source for the system). The busy

Table 1
Comparison of synapse density and function. The FACETS IC, Stanford STDP, INI IC 1, and ISS chips are described in detail in (Arthur & Boahen, 2006; Camilleri et al., 2007; Indiveri et al., 2006; Schemmel et al., 2008, 2006). The normalized synapse area is computed by dividing the synapse area by the square of the process feature size.

Chip description	Process node (nm)	No. of synapses	Synapse area (μm^2)	Normalized syn. area	Syn. storage resolution
GT Neuron IC	350	30 000	133	**1 088**	$>$10 bit, STDP
FACETS IC	180	98 304	108	3 338	4 bit, STDP
Stanford STDP	250	21 504	238	3 810	STDP, no storage
INI IC 1	800	256	4495	7 023	1 bit w/learning
ISS	350	16 384	3200	26 122	2.5 bit w/learning

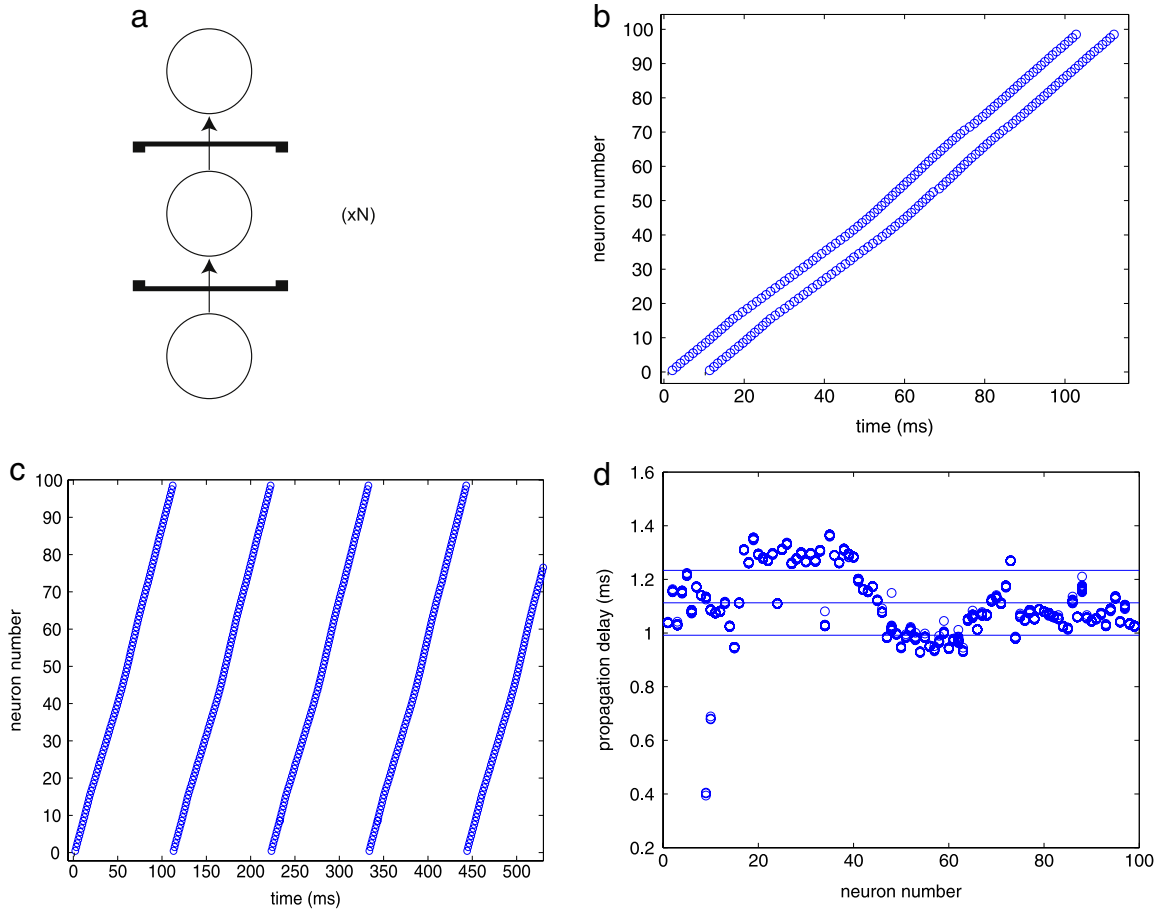

Fig. 7. Simulation results for 100-neuron synfire chain. (a) Depiction of synfire chain network. For the data shown here there are 100 neurons, so N is 98. (b) Raster plot of spiking output resulting from two sequential inputs initiated at the beginning of the chain. This results in two simultaneous waves propagating along the chain. (c) Raster plot of periodic behavior in the synfire chain when the neuron at the end of the chain is connected back to the one at the beginning. (d) Measured propagation delays for each neuron in the chain. The mean and $+/-1\sigma$ levels are shown by horizontal lines. In (b) and (c), input spikes are denoted by vertical tick marks, while outputs are denoted by open circles.

neuroscientist need only be equipped with a laptop and USB cable in order to use this system for running simulations on the subway, on a train, or in an airplane.

The system uses Matlab-based tools for defining the network topologies and parameters. The syntax of this system closely follows PyNN, a network description language developed with the goal of facilitating interoperability of neuromorphic hardware (Davison et al., 2008). Some example code for specifying a network is depicted along with a graphical interpretation in Fig. 4.

3. Building blocks

3.1. Dot matrix network

In order to demonstrate the basic functionality of all of the components of the signal flow, a dot matrix network was simulated, wherein each neuron can be caused to spike by applying an AER input to it. All 100 neurons in the chip were given a stream of input events designed to result in a desired pattern when viewed in a raster plot format. Fig. 5 shows the network topology and the measured result.

A few aberrations in the result can be seen. For example, a few neurons around the edges of the "O" seem to fire even though they were not stimulated. This is due to some crosstalk between firing neurons and those which were not stimulated. We can mitigate the effects of crosstalk by reducing the sensitivity of the neurons by modifying one of the floating gates in the Na circuit responsible for fast activation. For the same amount of crosstalk at its input, a neuron with lower sensitivity will not spike.

3.2. Spiking threshold and spatial summation

One of the key features of the biological neurons is the thresholding of inputs. The input currents are summed at the

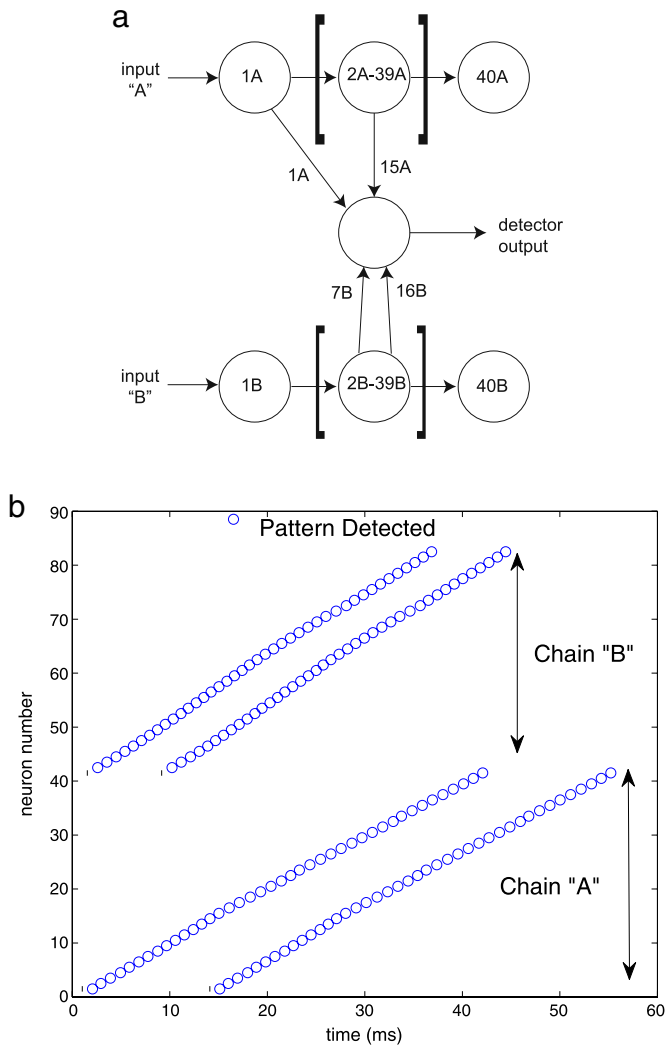

Fig. 8. Simulation results for a spatiotemporal pattern detector network. (a) Diagram showing network connectivity. Brackets indicate repeated units connected sequentially to form a chain. (b) Raster plot of activity in the network when the tuned ABBA pattern is given with the appropriate timings of 0.5, 7.5, and 4.5 ms separating successive inputs. Input spikes are denoted by vertical tick marks, while outputs are denoted by open circles. The synfire chain A corresponds to neuron numbers 2–42, chain B is numbers 43–82, and the output neuron is 89.

soma, and if their sum exceeds a certain threshold, the neuron will spike. This behavior can be illustrated by measuring the neuron's responses to an excitatory PSC of varying amplitudes, or similarly by measuring the response to a number of synaptic inputs that all add current to the neuron simultaneously. The results of these simulations using a single neuron are depicted in Fig. 6. In both of these simulations, subthreshold depolarizations can be observed in response to weaker stimuli, while stronger stimuli cause an action potential on the target neuron, as expected.

It is important to identify how system variations will affect the spiking threshold of a neuron and the amount of input current generated by its synapses. The spiking threshold is determined by the membrane capacitance and the cutoff frequencies of the bandpass and lowpass filters in the channel model circuits. Any variations in these parameters can affect the threshold. However, floating-gate technology does allow us to trim out a significant portion of the mismatch effects.

The size of the input current is determined by two factors: the synaptic efficacy and the shape of the triangle waveform impinging upon the synapse. The synaptic efficacy is controlled by a single floating-gate transistor, so it is very simple to tune. The waveform shaping circuitry is more complex, as it is controlled by three floating-gate transistors and consists of many non-floating-gate transistors and a charging capacitor. This means that the majority of the input current mismatch is typically due to mismatch in gate waveform circuitry, rather than problems with the synapse itself. We have developed methods to trim offsets from the gate waveforms, but they are not yet as precisely tuned as the synapses.

3.3. 100-neuron synfire chain

Pools of neurons that are connected serially into a chain, called synfire chains, can produce sequences of spikes with consistent timings each time the chain is stimulated. This has been proposed as an explanation for some electrophysiological findings (Abeles, Bergman, Margalit, & Vaadia, 1993), and several contexts for computational significance of this type of network have been suggested (Abeles, Hayon, & Lehmann, 2004; Arnoldi, Englmeier, & Brauer, 1999).

In order to make the most of the limited neurons available, the simplified network topology depicted in Fig. 7(a) (wherein each pool in the chain is modeled by a single neuron) was simulated. A raster plot of the result of the simulation is also shown in Fig. 7(b). The plot depicts the response of two successive spikes at the input of the chain. Each spike starts a wave of activity down the chain. Note that the second wave is initiated while the first wave is still propagating. The speed of the wave propagation, which is determined by the delay in synaptic transmission and the strength of the synaptic connections, averages about 1 ms per synapse.

The measured propagation delays are shown in Fig. 7(d). In this system, the propagation delay is determined by the shape of the triangle waveform arriving at the gate of the synapse, since the synaptic efficacies are all greater than 1. Device mismatch is apparent here because there is a small variance around the mean propagation delay. The triangle waveform circuits are particularly susceptible to mismatch because they have many analog parameters. Mismatch is a common problem in analog systems, and its effects are especially obvious in the subthreshold regime. However, compared to the variance of a non-configurable system, the propagation delay seen here is fairly consistent across all of the neurons in the chain. This kind of uniformity would not be possible without the facility for trimming the synaptic waveform shaping circuitry by programming floating-gate transistors. Some preliminary results about this topic were shown in Brink et al. (2013), and we have since fully integrated the approach into the system for using the IC.

4. Network simulation results

4.1. Spatiotemporal pattern generation and detection

Spatiotemporal processing in neural networks has been an area of interest in neuromorphic engineering for many years (Liu & Douglas, 2004). The repeatable timings demonstrated in our synfire chain make it useful for creating some networks wherein spike timing is important in the information encoding scheme. For instance, arbitrary spatiotemporal patterns can be detected by a network that has a distinct synfire chain for each distinct input channel. The approach is illustrated for a sequence of four spikes total on two inputs, A and B, with the sequence ABBA, and intervals of 0.5 ms, 7.5 ms, and 4.5 ms separating sequential spikes.

The network that performs this detection is depicted with the raster plot of its response to the input pattern of interest in Fig. 8. If any of the four spikes in this pattern are omitted or shifted by more than about 1 ms, the output neuron does not spike. Getting the synaptic weights to a value such that the coincidence of all four inputs is necessary and sufficient to produce a spike in the output neuron required an iterative process of programming weights and testing the response to inputs.

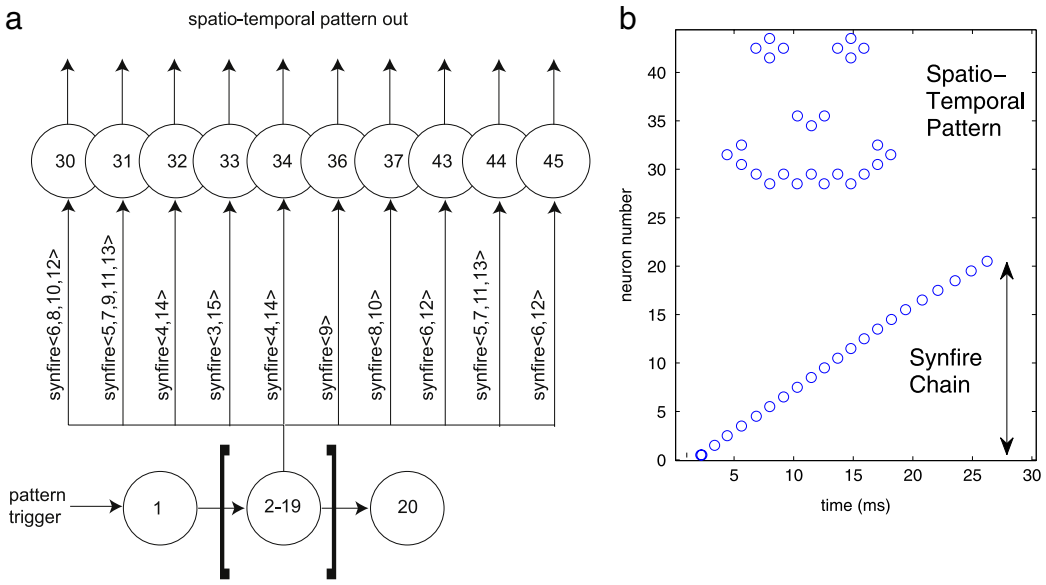

Fig. 9. Simulation results for a spatiotemporal pattern generator network. (a) Diagram showing network connectivity. Brackets indicate repeated units connected sequentially to form a chain, and a signal bus notation is used to denote the various connections from the synfire chain to the output neurons. (b) Raster plot of activity in the network in response to a single pattern trigger spike. Input spikes are denoted by vertical tick marks, while outputs are denoted by open circles.

It is worth noting that sound localization in animals requires the detection of delays in two input lines (the inputs coming from the two ears), and that a neural structure that operates on a similar principle to the one presented here has been characterized in the barn owl (Knudsen, 1981).

Arbitrary spatiotemporal patterns can also be generated using a similar approach. The network shown in Fig. 9(a) generates the sequence of spikes depicted in the raster plot in Fig. 9(b) when its input is stimulated with a single spike. This kind of functionality could be useful in generating a precisely timed stereotyped motor output from a neural system.

4.2. Ring winner-take-all (WTA) network

Spiking WTA models have also been important in neuromorphic engineering for several years (Oster & Liu, 2006). The WTA network models competition through lateral inhibition. It is a functional block that is useful for performing classification tasks (see, for example, Robinson, Yoneda, & Sanchez-Sinencio, 1992), and it is an integral part of many models of neural phenomena, especially regarding attention (Niebur & Koch, 1998). An example of such a network that has six input neurons is shown in Fig. 10.

A simulation with a 30-input winner-take-all network was run, and the results are depicted in Fig. 11. The duration of the post-synaptic currents of the inhibitory synapses was set to 20 ms in order to increase the effect of inhibition, and thus maximize the competitive dynamics. The inputs were Poisson spike trains at a fixed average spike rate of 1 Hz, except for one input channel (the "winner"), whose input rates were varied from 1 to 240 Hz.

In Fig. 11(b), the resulting firing rates of all the neurons (averaged over 100 s of measurement) are shown for each input rate on the "winner" channel. The average firing rate of the "winner" neuron strongly increases when its input rate is increased, while the average firing rates of the others are suppressed. This illustrates the competitive behavior that is the essential feature of this network. Fig. 11(a) depicts a sample raster plot of network activity for a case in which all neurons receive inputs at an average rate of 1 Hz for 10 s, and then a single input rate goes to 240 Hz for 10 s. The increase in activity of the "winner" neuron and the decrease of the others is apparent. The symmetry of the structure was verified, i.e., all of the features pointed out here do not depend upon which neuron is selected to receive the faster input stimulation.

It is interesting to note that there is a rather gradual suppression of the "non-winning" inputs as the firing rate of the "winner" is increased. This effect is explained by the method of competition used in this experiment. Here, the efficacies of all neurons are greater than 1, so, whenever an excitatory neuron receives an input, it will cause global inhibition to occur and prevent all others from firing. So even when the "winner" is firing at a relatively high rate, the "non-winners" can still cause global inhibition if they happen to receive an input before the winner. They are simply less likely to do so because they have a lower spike rate than the winner. This explains why there is a gradual suppression of the "non-winning" inputs — they can still cause output spikes, but they are increasingly less likely to do so as the "winning" rate is increased.

An encoding scheme that is more compatible with this system would be one in which the inputs arrive in burst that could, for example, correspond to a particular phase of a global oscillation in a neural system. The "winner" in each burst is the neuron whose inputs arrive first. This type of encoding scheme has been proposed as a mechanism by which real biological networks process data efficiently and with low latency (Thorpe, 1990). An illustration of a simulation of such a regime is shown for a six-input WTA network in Fig. 10(b). In this experiment, the inhibitory PSC lasts only 1–2 ms, and the suppression of the later-arriving inputs is complete.

The first to arrive encoding actually has more in common with the rate-encoded system than one might expect. In the rate-encoded system in Figs. 10(b) and 11, the first neuron to receive an input after the end of a given inhibitory PSC usually elicits a spike from the inhibitory interneuron, thus preventing the other neurons in the network from firing for a period of time. If the inputs to the neurons are Poisson spike trains, the higher the average frequency on a given neuron's inputs, the more likely it is to be the first to spike. As the frequency of spikes on the input of the "winner" neuron increases, the "winner" ends up getting this first spike more often, so its spike frequency increases, and the interval between successive inhibitory spikes decreases, which suppresses firing of the other neurons. This is the mechanism of competition in the rate-encoded version of the ring WTA network.

The mechanics of this network are nicely illustrated by examining a histogram of the time since the preceding inhibitory

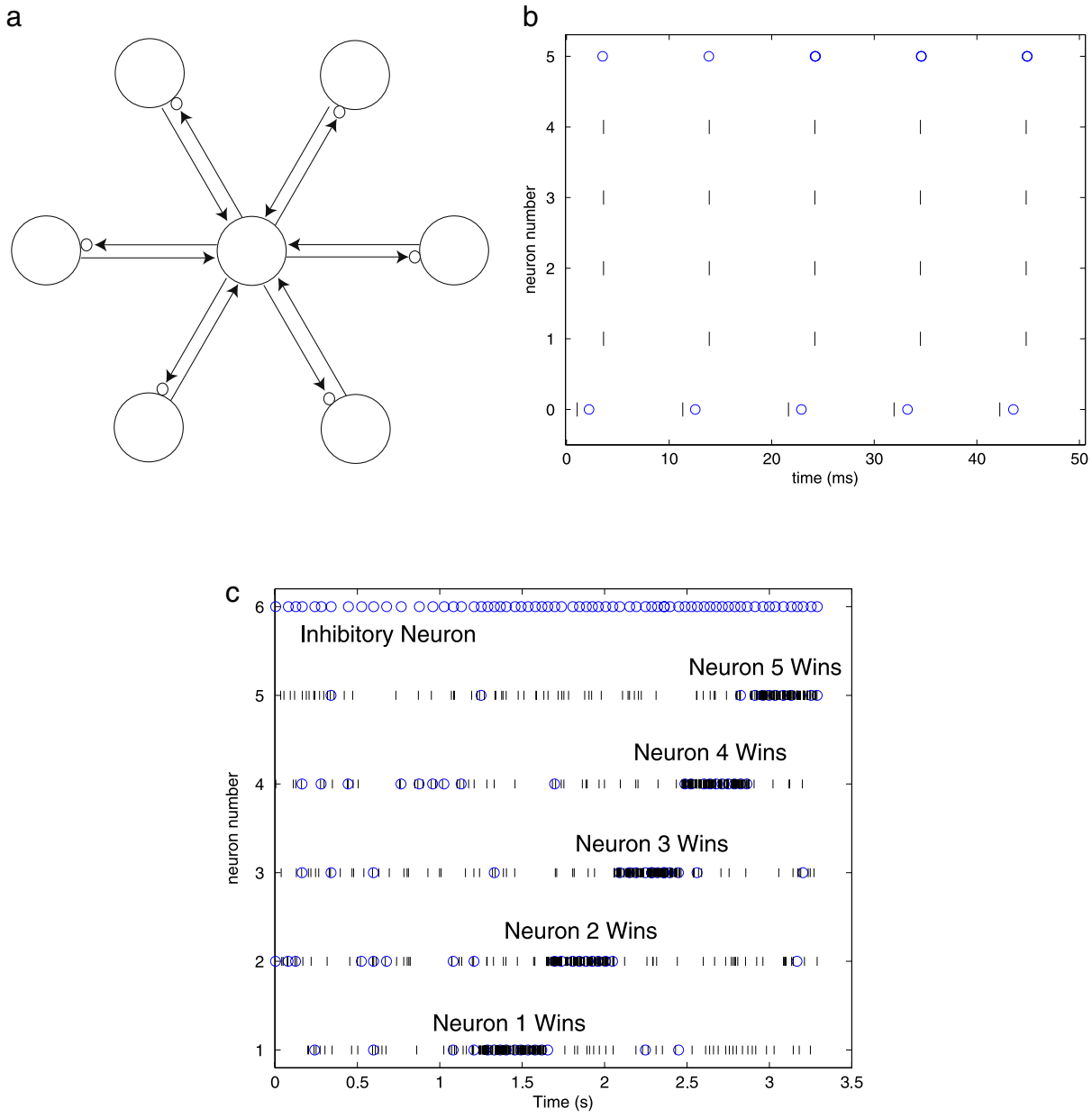

Fig. 10. Results for a six-input WTA network. (a) Diagram showing network connectivity for the ring winner-take-all network. Arrows that terminate in small open circles are used to denote inhibitory connections. (b) Raster plot of this network for the first-to-spike encoding scheme. In this scheme, the first neuron to spike in each burst of inputs prevents the others from spiking. (c) Raster plot of five-input WTA network for the rate encoding scheme. The input rate to all neurons is 20 Hz for the first 1.2 s, after each of the five inputs sequentially receives a burst of stimulation at 320 Hz while the others maintain 20 Hz stimulation. In (b) and (c), input spikes are denoted by vertical tick marks, while outputs are denoted by open circles.

action potential for each excitatory action potential that is recorded. This is shown for the 30-input WTA data in Fig. 12 (this plot is constructed from the same data depicted in Fig. 11(b)). Note that, regardless of the input rates, there are very few excitatory spikes that occur during the inhibitory PSC, which spans a period of about 20 ms following the inhibitory spikes. The distribution of spikes after the end of the PSC is broad if the input rate of all of the neurons is low, but becomes more sharply peaked as the input rate to the "winner" neuron increases.

This peaking helps visualize how the network achieves WTA behavior. Recall that an inhibitory spike occurs when any excitatory neuron receives an input. When all rates are low, the time between spikes is relatively high, so the histogram has a broad peak. As the "winning" rate increases, the likelihood of a spike soon after inhibition increases, so we see a sharp peak immediately after inhibition is let up.

4.3. Bistable oscillator

If the final neuron in the synfire chain depicted in Fig. 7(a) is connected back to the input of the chain, the propagating wave of action potentials will be restarted each time it reaches the end of the chain, forming a stable oscillator. Given the refractory period and synaptic delays in the system, this topology was found to produce a stable propagating wave of action potentials using any synfire chain that is at least four neurons long. If the circuit also has a global inhibitory input, as depicted in Fig. 13(a), then the oscillations may be turned on or off by a single excitatory or inhibitory input. This circuit is somewhat reminiscent of a central pattern generator (CPG), with the global inhibition perhaps playing the role of neuromodulators that promote or suppress rhythmic activity.

This network has two stable dynamical states: the propagating wave, and the quiescent state of no spiking (for a longer chain,

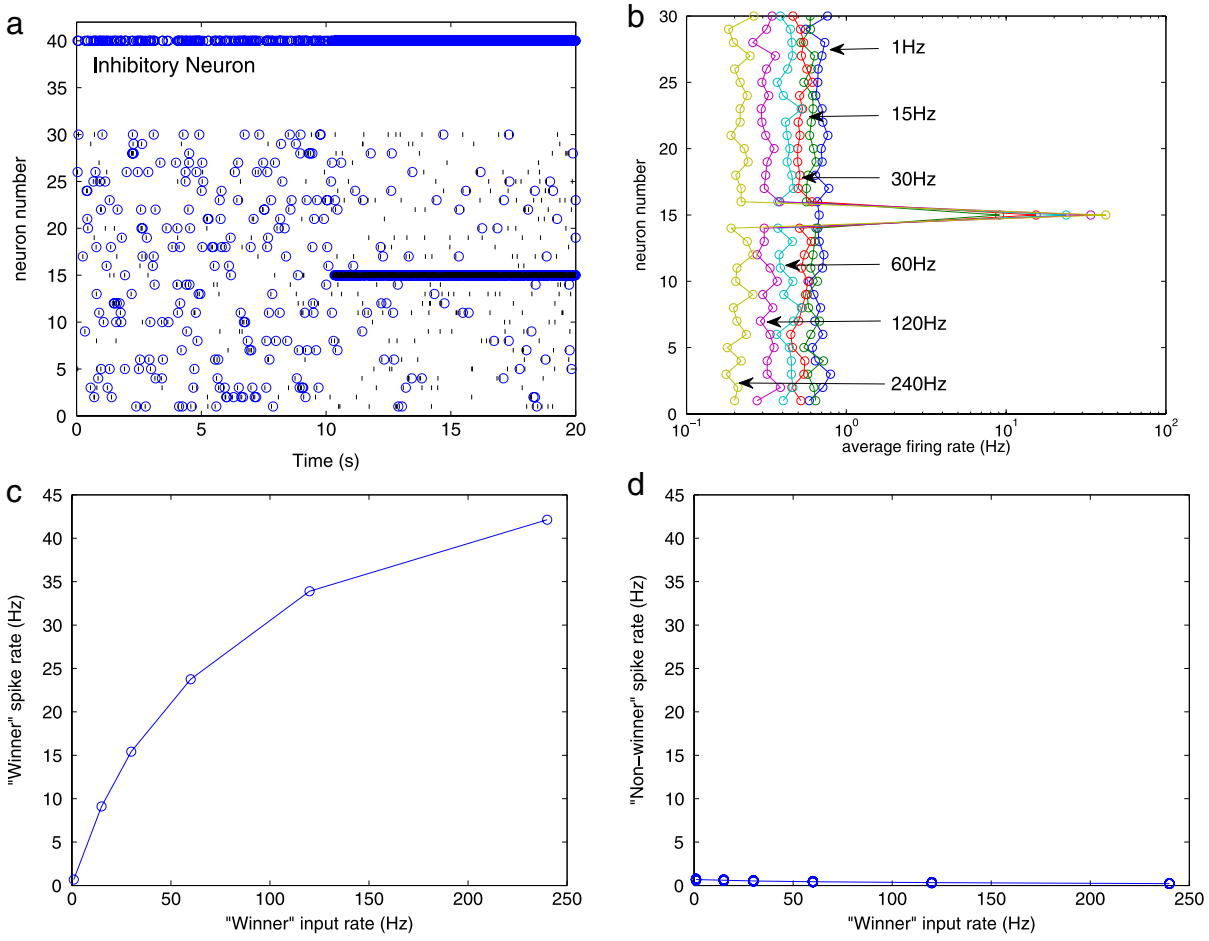

Fig. 11. Simulation results for a 30-input ring winner-take-all (WTA) network. (a) Raster plot depicting a typical example of activity in the network when the input rate for neuron 15 is 1 Hz for 10 s, and is then sustained at 240 Hz for 10 s. The inputs on the other neurons remain at 1 Hz for the duration of the experiment. Input spikes are denoted by vertical tick marks, while outputs are denoted by open circles. (b) Plot of average firing rates (over 100 s of measured data) of each of the 30 neurons, shown for six different input rates on neuron 15 (again, with 1 Hz input rate on all other neurons). Note that the vertical axis is set up to approximately align with the axis in (a), and that a log scale is used for the horizontal axis in order to simultaneously show the changes in rates of the "winner" neuron and the others. (c) Firing rate of neuron 15 versus the input rate on neuron 15. (d) Firing rates of the neurons other than 15 versus the input rate on neuron 15. The population average is plotted with a line.

multiple waves can propagate stably at different phases). This bistability that can be controlled by two different inputs is directly analogous to a digital latch circuit. Like the latch, this circuit's dynamical state stably stores a bit of information about the past inputs of the system. This system is an example of recurrent excitation giving rise to multiple stable states in a network's dynamics, which is the essence of the well-known Hopfield model for associative memory (Durstewitz, Seamans, & Sejnowski, 2000). In fact, a Hopfield network with the same connectivity captures the behavior demonstrated by this circuit.

The spiking activity recorded from this circuit is depicted in Fig. 13.

5. Power efficiency

If one is interested not only in the most expedient solution to a computational problem, but instead has a specific desire to see how the problem might be solved by a neural structure, then one is forced to somehow replicate the behavior of biological systems. In this case, perhaps the most common approach is to use a digital computer to numerically integrate model equations describing a network and its constituent neurons and synapses. From a computational efficiency standpoint, it is instructive to make a comparison between the power consumption of running such a model simulation on a neuromorphic IC and using numerical integration.

Based on the neuron biasing, the static power dissipated by the neurons can be estimated as 2 nW and the energy used by the neuron for spiking activity can be estimated at 170 pJ/spike. Based on the PSC amplitudes and durations, the energy per PSC is estimated at 10 pJ/spike. Based on the voltage supplies and load capacitances, the energy per AER event is estimated at 35 pJ/input event and 240 pJ/output event. Using these numbers, the energy consumed by the neurons, synapses, and AER communication for each of the network simulations presented in this article can be estimated, as shown in Table 2. Similarly, if the Hodgkin–Huxley model equations are simulated using a fixed time-step fourth-order Runge–Kutta integration with a step size of 50 μs, the number of multiply–accumulate (MAC) operations can be estimated as 20MACs/step for each synapse and 200MAC/step for each neuron. Assuming a generous computational efficiency of 100 pJ/MAC (Marr, Degnan, Hasler, & Anderson, 2013), the energy required to run the same simulations by numerical integration is also shown in Table 2. Clearly a significant advantage in power efficiency is obtained, with the improvement factor ranging from about 4000 for the bistable oscillator to nearly 300,000 for the WTA network. For a research group with limited access to supercomputing resources, the benefit of modeling networks on this neuromorphic system is that relatively large networks (up to a hundred neurons and thousands of synapses) can be simulated in real time, whereas the same network simulations would be very time consuming on a personal computer.

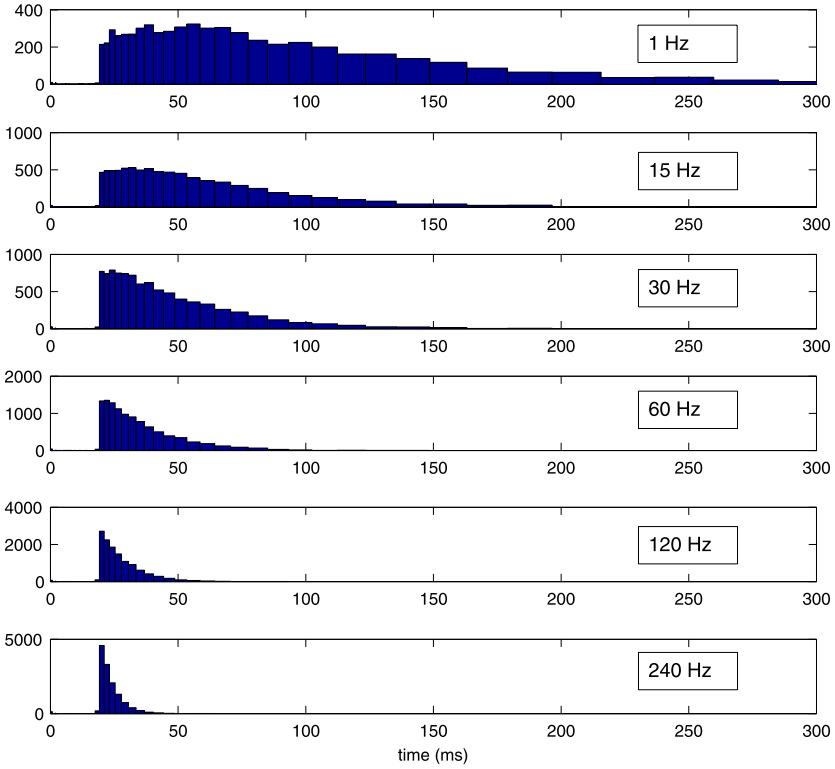

Fig. 12. Histogram of time since preceding inhibitory input for each excitatory action potential for 30-input WTA network. The average input rate on the "winner" neuron is annotated on each strip. The effect of the "winner" neuron's activity to reduce the time between consecutive inhibitory spikes is apparent. Also, the suppression of action potentials during the 20 ms after the inhibitory action potential is clearly visible.

Table 2
Estimated energy to perform the network simulations presented in this work on this neuromorphic platform versus a numerical integration approach.

Model description	Neuron energy (nJ)	Synapse energy (nJ)	AER energy (nJ)	Neuromorph. energy (nJ)	Num. Soln. energy (mJ)
Synfire chain	34	2.00	48	84	4.8
Pattern detect	27	0.08	39	66	2.0
Pattern gen.	8	0.47	11	20	0.35
WTA network	19	3.20	33	56	16
Bistable osc.	38	2.20	53	93	0.37

6. Conclusion

Having demonstrated modeling of multiple neural structures on this neuromorphic IC, we turn our attention to the topic of what computation the networks are performing, and how efficiently they are doing it. It is difficult to compare the computations done by a network of neurons to those performed by a traditional computing structure such as a microprocessor. The network of neurons is unable to find all of the eigenvalues or the inverse of a matrix, while this task is straightforward for a microprocessor. However, this does not mean that the network of neurons is not computing. In fact, the neural network can effectively solve problems such as generating intricately timed sequences of control commands, detecting specific patterns in input data, latching and storing input data, and implementing competitive comparisons among many input channels simultaneously. These are all useful functional blocks in sensory processing and motor control, and thus could form important building blocks for sophisticated autonomous robotic systems.

Given that the networks studied in this work can perform useful computations, it is interesting to ask what alternative approaches might yield the same computations. The synfire chain could be implemented simply using an oscillator, a binary counter, and a decoder. The pattern generation and pattern detection could then be implemented by adding some combinational logic gates to the synfire chain outputs. The volatile memory of the bistable oscillator could be performed simply with a latch. Making a similar statement for the WTA network requires specifying exactly what aspects of the WTA behavior must be achieved. In the case that the WTA network output is considered to be a "one-hot" encoding of the maximum of its inputs (replacing rate coding with static inputs representing rates), the WTA operation could be implemented using a tree of comparators. If one requires outputs with finer granularity than 0s and 1s, then matrix algebra would probably be required for any digital implementation. On the other hand, an analog circuit such as the one described in Lazzaro, Ryckebusch, Mahowald, and Mead (1988) is able to perform this function with few transistors and low power.

It is up to the system designer to decide whether he/she wishes to use one of these solutions or a neuromorphic approach. If he/she needs to solve the problem in a way that naturally interfaces with biology (perhaps in applications where neural activity is being recorded directly), then modeling biology directly is likely the best solution. And the most power-efficient method of doing this (other than growing actual neuronal tissue) is to use an analog neuromorphic system.

In summary, we have demonstrated several interesting and useful computations performed by a model neural structure simulated on neural hardware. These results are obtained with a computational efficiency that is orders of magnitude better than that offered by the approach of numerically integrating model equations. However, the computations performed by these neural models is, at best, only marginally competitive with the best custom (non-neural) solutions for performing the same computations. To the authors' knowledge, the same could be said of any applied computation that has been demonstrated on any other neuromorphic platforms that are intended for network simulations. This should

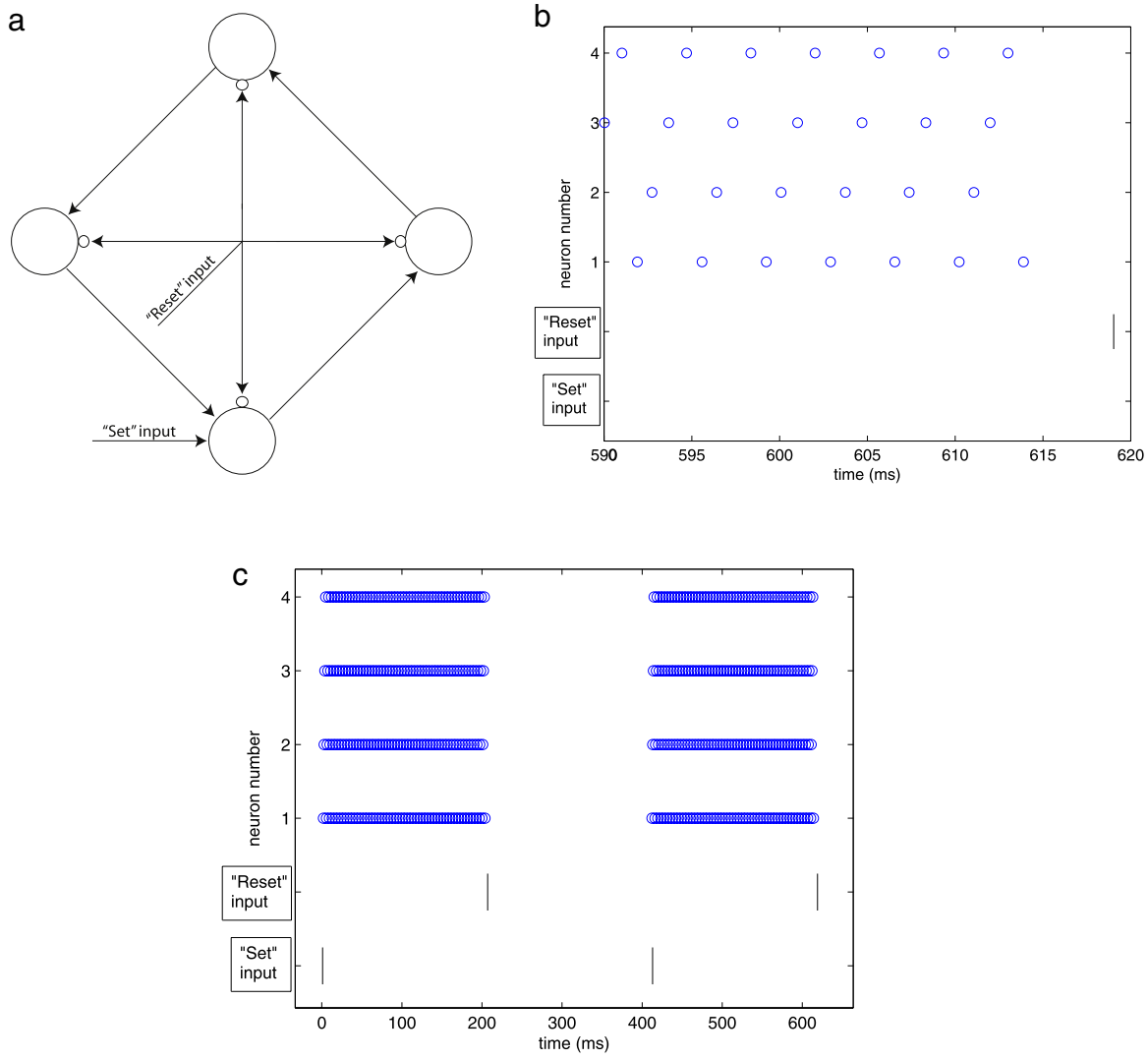

Fig. 13. Simulation results for bistable oscillator network. (a) Schematic depiction of network connectivity. (b) Zoomed-in depiction of the end of the data shown in (c), to illustrate the timing of the cyclically propagating wave of activity. (c) Raster plot showing activity of the network as the inputs drive it back and forth between the stable oscillations and the stable fixed point. In (b) and (c), input spikes are denoted by vertical tick marks, while outputs are denoted by open circles.

not be seen as an indication that models of neural systems on configurable neuromorphic platforms cannot outperform other approaches. To the contrary, it is our belief that such systems have enormous potential to compete in applications, a potential that will increasingly be realized as neuromorphic platforms continue to improve. The Heidelberg group shows one good example of network studies implemented on a neuromorphic hardware platform (Capocaccia, 2010). That study provided significant inspiration for the present work, and the fact that both of these systems use PyNN should make it easier for results to be compared across these two platforms.

References

Abeles, M., Bergman, H., Margalit, E., & Vaadia, E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *Journal of Neurophysiology*, *70*(4), 1629–1638.

Abeles, M., Hayon, G., & Lehmann, D. (2004). Modeling compositionality by dynamic binding of synfire chains. *Journal of Computational Neuroscience*, *17*(2), 179–201.

Arnoldi, H., Englmeier, K., & Brauer, W. (1999). Translation-invariant pattern recognition based on synfire chains. *Biological Cybernetics*, *80*(6), 433–447.

Arthur, J., & Boahen, K. (2006). Learning in silicon: timing is everything. *Advances in Neural Information Processing Systems*, *18*, 75.

Basu, A., Petre, C., & Hasler, P. (2008). Bifurcations in a silicon neuron. In *IEEE international symposium on circuits and systems, 2008*, ISCAS 2008. (pp. 428–431). IEEE.

Brink, S., Ramakrishnan, S., Hasler, P., Wunderlich, R., Basu, A., & Degnan, B. (2013). A learning-enabled neuron array ic based upon transistor models of biological phenomena. *IEEE Transactions on Biomedical Circuits and Systems*, *7*(1), 71–81.

Camilleri, P., Giulioni, M., Dante, V., Badoni, D., Indiveri, G., Michaelis, B., et al. (2007). A neuromorphic Avlsi network chip with configurable plastic synapses. In *7th international conference on hybrid intelligent systems, 2007*, HIS 2007. (pp. 296–301). IEEE.

Capocaccia, (2010). Exploring network architectures with the facets hardware and PyNN. http://capocaccia.ethz.ch/capo/wiki/2010/facetshw10 (accessed: 5/31/12).

Choudhary, S., Sloan, S., Fok, S., Neckar, A., Trautmann, E., & Gao, P. et al. (2012). Silicon neurons that compute. In *Artificial neural networks and machine learning–ICANN 2012* (pp. 121–128).

Davison, A., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., et al. (2008). Pynn: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, *2*.

Douglas, R., Mahowald, M., & Mead, C. (1995). Neuromorphic analogue VLSI. *Annual Review of Neuroscience*, *18*, 255–281.

Durstewitz, D., Seamans, J., & Sejnowski, T. (2000). Neurocomputational models of working memory. *Nature Neuroscience*, *3*, 1184–1191.

Farquhar, E., & Hasler, P. (2005). A bio-physically inspired silicon neuron. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *52*(3), 477–488.

Gao, P., Benjamin, B. V., & Boahen, K. (2012). Dynamical system guided mapping of quantitative neuronal models onto neuromorphic hardware. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *59*(10), 2383–2394.

Goldberg, D. H., Cauwenberghs, G., & Andreou, A. G. (2001). Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons. *Neural Networks*, *14*(6–7), 781–794.

Hasler, P., Diorio, C., Minch, B., & Mead, C. (1995). Single transistor learning synapses. *Advances in Neural Information Processing Systems*, *7*. Citeseer.

Indiveri, G., Chicca, E., & Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, *17*(1), 211–221.

Knudsen, E. (1981). The hearing of the barn owl. *Scientific American*.

Koziol, S., Schlottmann, C., Basu, A., Brink, S., Petre, C., Degnan, B., et al. (2010). Hardware and software infrastructure for a family of floating-gate based FPAAs. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, ISCAS. (pp. 2794–2797). IEEE.

Lazzaro, J., Ryckebusch, S., Mahowald, M., & Mead, C. (1988). Winner-take-all networks of o (n) complexity. Tech. rep., DTIC Document.

Liu, S.-C., & Douglas, R. (2004). Temporal coding in a silicon network of integrate-and-fire neurons. *IEEE Transactions on Neural Networks*, *15*(5), 1305–1314.

Marr, B., Degnan, B., Hasler, P., & Anderson, D. (2013). Scaling energy per operation via an asynchronous pipeline. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *21*(1), 147–151.

Niebur, E., & Koch, C. (1998). Computational architectures for attention. In *The attentive brain* (pp. 163–186).

Oster, M., & Liu, S.-C. (2006). Spiking inputs to a winner-take-all network. *Advances in Neural Information Processing Systems*, *18*, 1051.

Ramakrishnan, S., Hasler, P., & Gordon, C. (2011). Floating gate synapses with spike-time-dependent plasticity. *IEEE Transactions on Biomedical Circuits and Systems*, *5*(3), 244–252.

Robinson, M., Yoneda, H., & Sanchez-Sinencio, E. (1992). A modular cmos design of a hamming network. *IEEE Transactions on Neural Networks*, *3*(3), 444–456.

Schemmel, J., Fieres, J., & Meier, K. (2008). Wafer-scale integration of analog neural networks. In *IEEE International joint conference on neural networks, 2008. IJCNN 2008. (IEEE world congress on computational intelligence)* (pp. 431–438). IEEE.

Schemmel, J., Grubl, A., Meier, K., & Mueller, E. (2006). Implementing synaptic plasticity in a VLSI spiking neural network model. In *International joint conference on neural networks, 2006*, IJCNN'06. (pp. 1–6). IEEE.

Thorpe, S. (1990). Spike arrival times: a highly efficient coding scheme for neural networks. In *Parallel processing in neural systems* (pp. 91–94).

Yu, T., & Cauwenberghs, G. (2010). Log-domain time-multiplexed realization of dynamical conductance-based synapses. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, ISCAS. (pp. 2558–2561). IEEE.

Yu, T., Park, J., Joshi, S., Maier, C., & Cauwenberghs, G. (2012). 65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing. In *2012 IEEE biomedical circuits and systems conference*, BioCAS. (pp. 21–24). IEEE.