Large Developing Receptive Fields Using a Distributed and Locally Reprogrammable Address-Event Receiver

Simeon A. Bamford, Alan F. Murray, David J. Willshaw

***Abstract*—A distributed and locally reprogrammable address event receiver has been designed, in which incoming address-events are monitored simultaneously by all synapses, allowing for arbitrarily large axonal fan-out without reducing channel capacity. Synapses can change the address of their pre-synaptic neuron, allowing the distributed implementation of a biologically realistic learning rule, with both synapse formation and elimination (synaptic rewiring). Probabilistic synapse formation leads to topographic map development, made possible by a cross-chip current-mode calculation of Euclidean distance. As well as synaptic plasticity in rewiring, synapses change weights using a competitive Hebbian learning rule (spike-timing-dependent plasticity). The weight plasticity allows receptive fields to be modified based on spatiotemporal correlations in the inputs, and the rewiring plasticity allows these modifications to become embedded in the network topology.**

***Index Terms*—Neural network hardware, Neural network architecture, Neuromorphic VLSI, Topographic map, Synaptic rewiring, Synapse formation, Synapse elimination, Euclidean distance, Address Event Representation, AER.**

I. Introduction

Neuromorphic engineers create integrated electronic circuits which mimic neural computation in biological nervous systems, both to inform computational neuroscience and in pursuit of superior engineering solutions for classes of problems where biology currently outperforms artificial devices [1]. There is a need to form interconnects between many integrated neuron circuits to create neural networks. In many applications such as topographic map development [2], reconfigurability in the connections is essential to underpin map formation and maintenance. In a topographic map, one layer of neurons maps its connections to another layer such that neighbouring relationships between neurons in one layer are preserved in the other. In order for such a mapping to develop, neurons gradually change their patterns of connections according to both innate preferences and feedback induced by network input [3].

Time-division multiplexing facilitates massively-parallel connection between spiking neuron circuits across multiple chips. Spikes are commonly treated as address-events; the unique digital address of a neuron within a neural array is transmitted on a bus. This approach was first used in [4], [5], and has since been extended and improved. [6] gives a good summary of this still-evolving technique. Within this Address-Event Representation (AER) protocol, the number of wires required to connect N neurons scales as $log(N)$, such that the number of pins and wires necessary to interconnect chips is achievable. The development of word-serial AER reduces the number of wires required still further [7]. AER exploits the large difference in frequency between the spiking behaviour of biological neurons (on the order of 1-1000Hz) and the capability of digital electronic communication (many MHz). An AER bus can typically deliver spikes at around 10MHz, although recent publications have improved on this (see section VII-B for references). If biological spike rates are desired then, taking the conservatively high assumption of 100 spikes/second for the average output of a biological neuron [6], 100,000 neurons can share a single bus. (The lower assumptions of different authors about spike rate would raise the estimate of the number of neurons by 1 [8] or 2 [9] orders of magnitude).

AER was originally conceived as a point-to-point protocol. If each neuron in one neural layer has a unique connection to only one neuron in a corresponding neural layer in a topographic map arrangement, the outgoing bus can be decoded directly by a row-and-column decoder on a receiving chip, and spikes are delivered correctly to the same location on a corresponding chip. More commonly however neurons make connections to many other neurons (i.e. they have a large axonal "fan-out") and receive large numbers of incoming connections (dendritic "fan-in"). As two examples, [10] found an average fan-out of 167 for retinal ganglion cells in the tectum of the hamster, whilst [11] found an average fan-in of 85,000 onto the Purkinje cells of the cat. In order to implement arbitrary many-to-many network connectivity, address-events are commonly received not directly by a neural array chip but rather by a microcontroller, and are then compared to a look-up table in memory in order to find out the addresses of the neurons or synapses which should be targetted. These addresses are then sent as address-events, sequentially to one or more receiving neural arrays [12], [13]. This will be referred to hereafter as the "look-up table" approach. This reduces the capacity of the bus in the presence of large fan-out. If an average fan-out of 1000 is desired, the number of neurons which can then be supported is reduced by 3 orders of

Manuscript received January 03, 2009; revised May 10, 2009; accepted September 24, 2009. This work was supported by the Engineering and Physical Sciences Research Council. The authors are with the Neuroinformatics Doctoral Training Centre, Institute of Integrated Micro and Nano Systems, and Institute of Adaptive and Neural Computation respectively, University of Edinburgh, Edinburgh EH9 2AB, U.K. (e-mail: simeon.bamford@iss.infn.it). Digital Object Identifier 10.1109/TNN.2009.2036912

magnitude, e.g. from $\approx 100,000$ to ≈ 100.

The use of a microcontroller and a look-up table in memory has also been used to implement "synaptic rewiring", i.e. the formation and elimination of synapses such that connectivity between neurons changes [14]. In the scheme of [15], information from a synapse circuit was transmitted off-chip back to the microcontroller where it was used to modify the look-up table, implementing a simple model of topographic map formation. This is part of a trend of using a microcontroller to implement more of a neural network model. This trend has been extended by Vogelstein *et al* [16], where other synaptic variables relating to synaptic strength were also held in the look-up table, allowing each neuron to have a single synapse circuit which acted as a number of virtual synapses.

The system presented in this paper extends the preliminary work presented in [17]. In order to overcome the bottleneck on channel capacity as fan-out increases in the look-up table approach, whilst maintaining the possibility of adaptability by means of synaptic rewiring, a new approach has been taken. Address-events are broadcast across a neural array and may be simultaneously received by many synapses. This approach is described in section II and in section III its scalability is compared with the look-up table approach. In contrast to the aforementioned approach of [16], in this system, more information is stored locally at each synapse circuit. Specifically, the digital address of the pre-synaptic neuron is stored, alongside analogue synaptic variables such as weight. Given the local availability of information about incoming connectivity, the synapse circuits have been designed to take advantage of other information stored locally in order to change incoming connectivity, thus implementing synaptic rewiring. In order to constrain design choices and to demonstrate the utility of such an approach, a model of topographic map formation has been synthesised, in which the interplay between weight plasticity and synaptic rewiring allows probabilistically formed receptive fields to develop according to the statistics of spiking inputs. The model is outlined in section IV. The rationale for a distributed implementation of synaptic rewiring is explained in section V. Then the circuitry which achieves this is described in section VI. The functioning of the system is explored in section VII, with the results of a series of experiments. Finally, there is discussion of the performance of this system and possible alternatives.

II. The broadcast approach

For the reasons given in section I above, a new approach to spike delivery has been developed. Address-events from a sending chip are received directly by a receiving chip and broadcast across the receiving chip's neural array. Simultaneously, each synapse compares that address to an address which is stored locally to the synapse, to establish whether the address-event was intended for itself. Many synapses can store the same address and thus arbitrarily large axonal fan-out can be implemented without reducing bus capacity. This will be referred to as the "broadcast approach". This approach has been mooted before e.g. [12, page 170]:

> "Ideally each node should recognise its relevant source events, but our present multi-neuron chips use a DSP chip and lookup table to implement the fan-out from source address to the individual target synaptic addresses."

However, to date, no such system has been implemented based on real-time analog neurons and AER communication.

The look-up table approach allows the use of receiving circuitry as described by [6], which is shown in figure 1 (a). The receiving circuitry which implements the broadcast approach is shown in figure 1 (b). The chip-level address-event receiver is compatible with existing address-event transmitters. An incoming request is acknowledged immediately and triggers local latching of the address bus and a timed delay followed by a timed pulse to synapses. Synapses do not acknowledge receipt of an event, rather the chip-wide broadcast is timed to last long enough for all synapses to receive it. A minimum cycle time is imposed, sufficient to allow for the timed delays, before the acknowledge signal is dropped. An example timing diagram for the receipt of a single address-event is given in figure 1 (c).

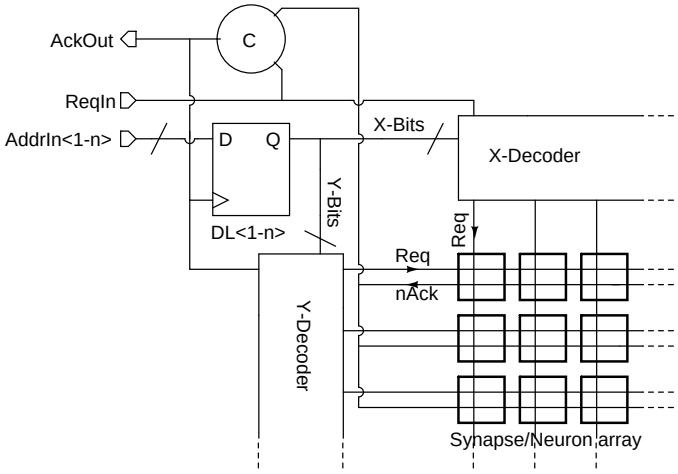
III. Scalability of broadcast approach

In this section the scalability of the broadcast approach is compared to that of the existing look-up table approach (examples of which are [12], [13]). It is not compared with systems such as [18] and [19]; these systems constrained each neuron in a layer to have the same receptive field shape. By so doing, [18] achieved a reduction in memory space at the level of the microcontroller and [19] additionally achieved a reduction in time to implement fan-out, but both did so at the expense of topological adaptability at the level of individual synapses. The approach of Vogelstein *et al* [16], however, is included in comparisons; this is included because it is a special case of the look-up table approach in which there is only one target synapse address per target neuron.

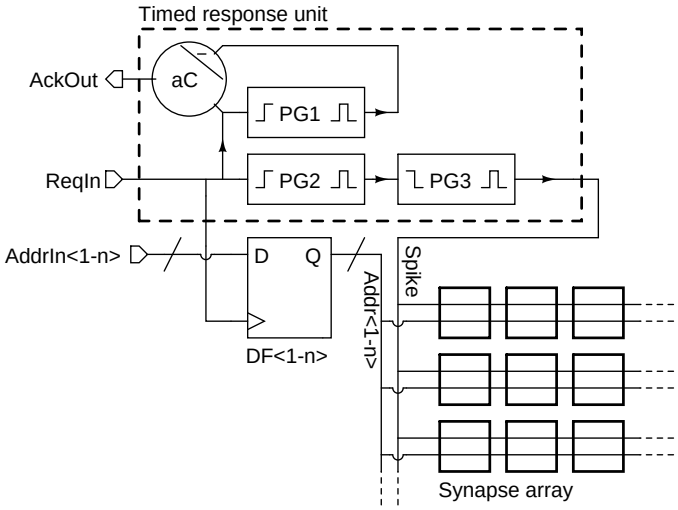
Scalability comparisons consider the amount of silicon area, transmission energy and transmission time required, as numbers of neurons and synapses in a system increase.

A. Silicon Area

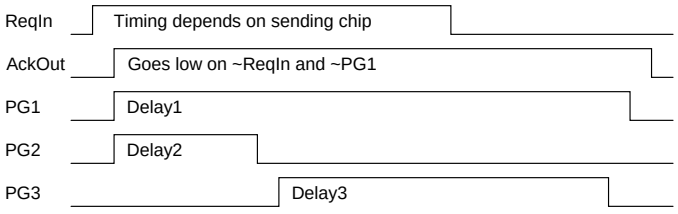
Each synapse, in order to implement its address-event receiver, must store as many bits in memory elements as the width of the incoming address bus. The total area of the receiving circuitry across the chip, or across the system, for a multi-chip system, then scales as $S_{max}Nlog_2(N)$, where N is the number of neurons in the system and S_{max} is the maximum fan-in, i.e. the number of dendritic (or incoming) synapses allowed per neuron. The $S_{max}N$ term represents the number of synapse circuits in the system and the $log_2(N)$ term represents the number of bits necessary to encode a neuron's address within each synapse. At first glance this scales poorly compared to the look-up table approach, which employs row and column



(a)



(b)



(c)

Figure 1. (a) Address-event receiver circuitry, functionally equivalent to that described in [6]. The incoming request *ReqIn* triggers the raising of the global acknowledge, *AckOut*, and the decoding of the incoming address (which is assumed to be valid upon arrival of *ReqIn*); a synapse (or neuron) is targeted; when this acknowledges, *AckOut* is lowered (once *ReqIn* has also been lowered), allowing the next event to be transmitted. (b) Proposed address-event receiver. Upon *ReqIn* going high, *AckOut* is immediately driven high, and a pulse generator (*PG1*) is also triggered, the output of which stays high for a precisely-timed (adjustable) period thereafter. *AckOut* stays high until *ReqIn* and *PG1* both drop. *ReqIn* also triggers the local latching of the incoming address bus. Once latched, the address is broadcast across the chip and all synaptic address-event receivers simultaneously compare this address to their own stored address to decide whether it is correct. From the rising of *ReqIn* there is a short delay (implemented by *PG2*) to allow the address data to propagate across the chip, before a pulse (*Spike*) is sent out across the chip (implemented by *PG3*) triggering those synapses with correct addresses to accept the event. The pulse generated by *PG1* is timed to be long enough to accommodate the joint delays of *PG2* and *PG3* before allowing *AckOut* to drop and the cycle to repeat. (c) Example timing diagram for timed response unit.

decoders allowing the area of the receiving circuitry to scale as $\sqrt{S_{max}N}log_2(S_{max}N)$, where the $\sqrt{S_{max}N}$ term represents the number of row or column decoder elements necessary to decode a target synaptic address and the $log_2(S_{max}N)$ term represents the number of bits necessary to encode a synaptic address (assuming a square grid of synapses; each decoder element must store one dimension, i.e. half the bits, of the synaptic addresses it encodes for). Importantly, however, the look-up table approach requires that additional memory external to the neural array is used to store the look-up table (typically on an external memory chip), in which area is required which scales as $S_{av}Nlog_2(S_{max}N)$, where S_{av} is average fan-out. The $S_{av}N$ term is the number of axonal (or outgoing) synapses in the system and the $log_2(S_{max}N)$ term is the number of bits necessary to encode a dendritic (or incoming) synaptic address. The costs of microcontrollers and memory are not normally considered, whether in terms of chip area or power consumption. This is acceptable for test systems, but if total power budget and space are considered it can be seen that in the broadcast approach, the chip space necessary to implement memory is simply being distributed throughout the neural array, rather than stored in a separate dedicated chip. The possibility of integrating a dedicated look-up table on each neural chip is currently under investigation by Shih-Chii Liu (Institute of Neuroinformatics, Zurich, personal communication); this approach may have some benefits over existing systems, but the scaling of area as described here would not be changed. The scaling expressions above are summarised in table I.

The actual difference in area requirements between these approaches should not be overlooked. Memory on a dedicated memory chip takes up much less space than in the design presented in this chapter, for the following reasons. Firstly it is not integrated with address-bus receiving circuitry but rather optimised for its purpose. Static Random Access Memory (S-RAM) cells use just 6 transistors compared to 12 used in the receiver bit design which will be be presented in section VI-A, and Dynamic RAM (D-RAM) cells can be much smaller again, consisting of just one transistor and one capacitor; in section VIII, a word-serial receiver is proposed to overcome this limitation. Secondly, dedicated memory can arguably be smaller as it can be implemented in more recent processes with smaller geometry, whilst analogue neurons may need to be implemented with larger geometry to limit mismatch; advances in the implementation of homeostatic neural algorithms [20] may overcome this drawback, as suggested by [21]. Dedicated memory is also less costly simply because it is mass-produced; however, whilst chip area is much more expensive on trial application specific integrated circuits than on mass-produced memory, this may not always be the case if neuromorphic circuitry comes into mainstream demand. Notwithstanding the possible solutions to these issues, the broadcast approach currently yields synapses of significantly larger on-chip area, resulting in higher production costs for the foreseeable future.

Table I
SCALING OF AREA, ENERGY USAGE AND SPEED

System	On-chip receiver area	Off-chip memory area	Internal buffering energy per spike sent	Time per spike sent
Broadcast	$S_{max}Nlog_2(N)$	none required	$S_{max}Nlog_2(N)$	unity
Look-up table	$\sqrt{S_{max}N}log_2(S_{max}N)$	$S_{av}Nlog_2(S_{max}N)$	$S_{av}\sqrt{S_{max}N}$	S_{av}
Vogelstein *et al* [16]	$\sqrt{N}log_2(N)$	$S_{av}Nlog_2(N)$	$S_{av}\sqrt{N}$	S_{av}

N = number of neurons in system; S_{max} = maximum fan-in, i.e. number of dendritic synapses allowed per neuron; S_{av} = average fan-out.

B. Energy usage

In the broadcast approach, each incoming address event must be broadcast across the neural array to each synapse. Consequently each synapse contributes a capacitive load to the on-chip buffering and therefore energy consumption will scale linearly with $S_{max}N$. This term includes internal buffering to the neurons but does not include buffering from the pads to the peripheral latches shown in figure 1(b), because the look-up table approach has an equivalent cost. In the look-up table approach there is no need to broadcast the address across the chip; rather the spike signal can be targeted to the row and column of the correct synapse within the neural array. The energy cost of internal buffering should therefore be lower and should scale *per address-event received* as $\sqrt{S_{max}N}$. In the broadcast approach, however, energy usage remains constant *per spike sent*, whilst in the look-up table approach energy usage *per spike sent* increases linearly with axonal fan-out, as each axonal synapse requires a separate spike to be transmitted between chips and the correct synapse targeted. Bearing this in mind, scaling expressions for energy are given in table I. This suggests that if the choice of which approach to use is to be determined by energy usage then there will be a ratio of S_{av} / $S_{max}N$ above which the broadcast approach can be expected to outperform the look-up table approach. In other words, the broadcast approach may perform better in terms of energy for densely connected systems but it will not perform so well for sparsely connected systems. Here, the additional energy costs of the microcontroller and memory in the look-up table approach have not been considered.

C. Time

To ensure that communication succeeds in the broadcast approach, each communication cycle is deliberately slower than the average cycle time which could be achieved if the sender were allowed to proceed with the next event as soon as a synapse acknowledges, as in figure 1(a), though the difference need not be more than a small factor. However the time taken in the broadcast approach does not increase with S_{av} whereas in the look-up table approach it increases linearly. In fact the scaling of time in the broadcast approach may be above unity, because the time the address receiver takes to compare its stored address with the incoming address will scale as $log_2(N)$, due to the increased number of receiver bits which contribute to a NAND gate in the implementation used (see section VI-A). However this is likely to be inconsequential compared to the broadcast time, and so has not been included in table I. In any case, this effect would not significantly change the conclusion: the principal advantage of the broadcast approach is that as axonal fan-out increases, a speed advantage accumulates. Therefore, as the complexity of neural networks implemented expands, the broadcast approach can be expected to continue to support larger neural networks with large average fan-outs after the existing approaches run out of channel capacity.

D. Multi-chip scalability

It is also worth noting that the scaling expression in table I above for the on-chip area required by the look-up table approach only holds for a single-chip system. If the system is spread across multiple chips then the expression for the look-up table approach becomes $C\sqrt{S_{max}N_{chip}}log_2(S_{max}N_{chip})$ where C is the number of chips in the system and N_{chip} is the number of neurons per chip (so that $N = N_{chip}C$). Therefore as a neural network is scaled up by networking together more chips and the ratio of C/N_{chip} goes up, the on-chip area scaling advantage with respect to the broadcast approach due to row and column decoding ceases to accrue. This is also true of internal buffering energy, for the same reason. With this in mind, alternative figures are given in table II just for those terms in table I which are affected.

IV. MODEL

In order to explore the challenges and benefits of synaptic rewiring in neuromorphic VLSI, a model of topographic map formation was developed, which includes both synaptic weight plasticity and probabilistic synaptic rewiring. In addition, topographic map formation is assumed to involve a combination of activity-dependent and -independent processes. The model is intended to be general to the extent that it could apply equally to retinotectal, retinocollicular, retinogeniculate or geniculocortical projections. A full review of the bases of this model is beyond the scope of this paper; it is explained in greater detail in [22]. In brief, this model proposes the following:

1) Unspecified activity-independent processes impose a topographic mapping between a source and target layer (a layer is a 2D space in which neurons are located) and guide axons from the source layer towards their "ideal" location in the target layer, i.e. the location dictated by the topographic mapping.
2) Axon branching leads to formation of synapses over an area surrounding the ideal topographic location (broadly in line with e.g. the innervation of the tectum [23]).

Table II
MULTI-CHIP SCALABILITY

System	On-chip receiver area	Internal buffering energy per spike sent
Look-up table	$C\sqrt{S_{max}N_{chip}}log_2(S_{max}N_{chip})$	$S_{av}C\sqrt{S_{max}N_{chip}}$
Vogelstein *et al* [16]	$C\sqrt{N_{chip}}log_2(N_{chip})$	$S_{av}C\sqrt{N_{chip}}$

N_{chip} = number of neurons per chip; C = number of chips in system; N = number of neurons in system = $N_{chip}C$; S_{max} = maximum fan-in, i.e. number of dendritic synapses allowed per neuron; S_{av} = average fan-out.

3) Competitive Hebbian learning detects correlations in input patterns due to spatial proximity in the source layer, such that synapses from more spatially clustered afferent neurons are strengthened at the expense of synapses from neurons which are more distant from other afferents. The effective spread of the receptive fields of target neurons in the source layer is thereby reduced; this follows the model of [24]. If receptive fields contain input-specific features, such as ocular dominance segregation, these arise from this process.
4) Preferential elimination of weak synapses (as discussed in [25]) allows the reduction of spread to be embedded in the network topology.

The detail of the model used, including neuron and synapse dynamics, inputs and initial conditions, is given in algorithm 1; some explanatory notes follow here. Each cell in the target layer can receive a maximum number of afferent synapses. It can be said that each cell has a certain synaptic capacity, and this assumption is reflected in the design of the chip, where each neuron has the same fixed number of dedicated (potential) synapse circuits. The mechanism that yields the mapping between layers is unspecified; it could be thought of as a type I chemoaffinity mechanism (as defined by [26]) with fixed affinities, though other mechanisms could be inserted. There are two excitatory projections, a feed-forward projection from the input layer to the target layer and a lateral projection from the target layer back to itself. Axons within these projections compete for the synaptic capacity of the target neurons. Inhibitory lateral interactions are not implemented in this model, following the observation in [24, section on "Refinement of Cortical Maps"] that they are not necessary for topographic map formation providing that there is an initial bias towards a desired topology; in this model, bias towards a desired topology is not only initial but also ongoing. Although axons are guided towards their ideal locations, the formation of new synapses is with neurons which are randomly distributed around this ideal location. A Gaussian distribution is assumed, since a process which is initially directed towards a target site and then randomly branches on its way would yield a Gaussian distribution of terminations around the target site. A rejection sampling process is used to form the Gaussian distributions. The circuitry which has been created is capable of creating not just Gaussian distributions but any 2D iso-directional distribution with monotonically decreasing probability, as will be seen in section VII-D. In general, synapses implement some competitive Hebbian learning rule, such that correlations in inputs to a given target neuron result in preferential strengthening of those synapses at the expense of the strength of other synapses. For implementation, the synapses, neurons and type of input are based on the model of [24], i.e. with integrate-and-fire neurons with synaptic modulation governed by Spike-Timing-Dependent Plasticity (STDP).

V. RATIONALE FOR DISTRIBUTED IMPLEMENTATION OF REWIRING

The reason for implementing synaptic rewiring in circuitry local to each synapse is described here by incremental consideration of the functions of neurons and synapses.

Address-events which are broadcast across the chip are received by the receiver circuitry of each synapse. If an address-event is received by a synapse, the resulting pulse is used to create a synaptic conductance. This process is modified by the weight of the synapse, which is also stored locally in the synapse. The synaptic conductance affects the membrane potential, which is stored as a charge on a capacitor in the circuitry for the central functions of the neuron. If this potential passes a threshold then an outgoing spike is generated. Since analogue summation of currents (in this design representing increments of synaptic conductance) from the synapses into the neuron can be performed with a single wire, it is common for synapse circuits, together with storage of their weight values, to be implemented physically next to, or in a row leading to, the post-synaptic neuron which they serve. Thus, the neuron together with its dendritic synapses is formed of a contiguous block of circuitry; this block is located within a neural array and is not necessarily contiguous with the periphery of the chip.

The designer faces a choice at this point about where and how the synaptic learning rule should be implemented. STDP requires information about the pre-synaptic and post-synaptic spikes (specifically it requires their relative timings), and it affects the weight of the synapse. Consider that the pre-synaptic spike is necessarily available at the synapse as well as in a centralised mechanism for spike transmission at the periphery of the chip. Likewise the post-synaptic spike is available at the central neuron circuitry as well as in the periphery. The synaptic weight may be stored at the synapse for the reason given above. In deciding whether to implement STDP in an off-chip mechanism or locally to the synapse there is a trade-off between the respective costs, in energy and in design complexity, of transmitting weight information out of the neural array (or transmitting the implications of weight information into the neural array, see [16]) and the cost of transmitting post-synaptic spike information from the central neuron circuit to the locally-implemented synapses. In this project, an implementation of STDP local to the synapses has

Algorithm 1 Model summary

There are two 2D layers of the same size, the "input" and "target" layers; each is a square grid of neurons with periodic boundaries, and the "ideal location" of each neuron in the input layer is the location with the same coordinates in the target layer. Each target-layer neuron has the same number of "potential synapses"; these are dendritic locations in which actual synapses may form; synapses can be with a pre-synaptic neuron from either the input layer ("feed-forward connections") or the target layer ("lateral connections"), including the post-synaptic neuron itself ("recurrent connections").

Initial conditions: all potential synapses start formed, with conductance g_{max}.

Input: neurons are independent Poisson processes. A stimulus location s is randomly chosen and firing rates are set to $f_{base} + f_{peak} exp\left(-d/2\sigma^2_{stim}\right)$, where d is the distance from s. With a period t_{stim}, s moves and the process repeats.

Neuron dynamics (target-layer): the membrane voltage V_{mem} is described by:

$$\tau_{mem} \frac{\delta V_{mem}}{\delta t} = V_{rest} - V_{mem} + g_{ex}(t)\left(E_{ex} - V_{mem}\right)$$

E_{ex} = excitatory reversal potential; V_{rest} = resting potential; τ_{mem} = membrane time constant. Upon reaching a threshold V_{thr}, a spike occurs and V_{mem} is reset to V_{rest}. A pre-synaptic spike at time 0 causes a synaptic conductance at time $t \geq 0$ of $g_{ex}(t) = g.exp(\frac{-t}{\tau_{ex}})$ (τ_{ex} = synaptic time constant); this is cumulative for all pre-synaptic spikes.

STDP: a pre-synaptic spike at time t_{pre} and post-synaptic spike at t_{post} modify the peak synaptic conductance by $g \rightarrow g + g_{max}F(\Delta t)$, where $\Delta t = t_{pre} - t_{post}$ and $F(\Delta t) = A_+ exp\left(\Delta t/\tau_+\right)$ if $\Delta t < 0$, otherwise $F(\Delta t) = -A_- exp\left(-\Delta t/\tau_-\right)$, where $A_{+/-}$ are magnitudes and $\tau_{+/-}$ are time constants for potentiation and depression respectively. This is cumulative for all pre- and post-synaptic spike pairs; g is bounded in $(0, g_{max})$.

Synaptic rewiring: At a fixed rate f_{rew}, a potential synapse is randomly chosen. If it is an actual synapse, the elimination rule is used, otherwise the formation rule is used.

Formation rule: A possible pre-synaptic neuron is randomly selected from either the input or target layer, and synapse formation occurs if:

$$r < p_{form} e^{-\frac{\delta^2}{2\sigma^2_{form}}} \quad (1)$$

r = uniform random number in $(0, 1)$; p_{form} = peak formation probability; δ = distance of possible pre-synaptic neuron from ideal location of post-synaptic neuron; σ_{form} = standard deviation of the receptive field. p_{form} and σ_{form} may differ based on which layer the possible pre-synaptic neuron is from.

Elimination rule: If the synapse's conductance is below $0.5g_{max}$ it is eliminated with probability $p_{elim-dep}$, otherwise probability $p_{elim-pot}$ is used.

been chosen, following the intuition that the latter costs are likely to be lower, due to the reduced need for communication into and out of the neural array (area usage may be higher, however).

The approach of implementing functions locally to the synapse and neuron is then pursued towards its logical conclusion with the implementation of the synaptic rewiring rules presented in section IV. By additionally storing a binary variable at each synapse indicating whether or not the synapse actually exists, the synapse circuit becomes a circuit representing a potential synapse, (part of the neuron's total synaptic capacity), and can be connected or disconnected by flipping this bit. The synapse elimination rule requires weight information, as this affects the probability of elimination. As it is probabilistic, it also needs a stochastic process as input. If elimination occurs, the effect is to reset the bit which encodes the connectedness of the synapse. Inputs and outputs to the elimination rule are therefore available locally, with the possible exception of a stochastic process. The synapse formation process also requires a stochastic input. Additionally it requires a randomly chosen potential pre-synaptic partner; this must be transmitted to the synapse in at least some cases, since it is used to change the address stored in the synapse if formation is successful, therefore the means to transmit this address to the synapse must exist. (The possibility of using incoming address-events as an available source of potential pre-synaptic partners is discussed in section VI-C). The probability of formation is influenced by the proximity of the ideal location of the pre-synaptic neuron to the post-synaptic neuron's location. Location is a property of the neuron, as neurons are modelled as points and dendritic morphology is not considered; therefore circuitry for calculating proximity is localised in the circuitry for the neuron's central processes. In fact, rather than explicitly encoding the location of each neuron, the location is implied by the neuron circuit's physical location on the chip, as will be seen in section VI-D.

VI. Implementation

A. Synaptic address-event receiver circuitry

The total area of the synapse scales as the number of bits necessary to encode a neuron's address in the system. It is therefore necessary to make the storage of each bit and its associated circuitry as compact as possible. This has been achieved using a static memory element, with a transmission-gate implementation of an XNOR gate for comparison with the incoming address bit. The result of the comparison contributes to a NAND gate for the whole receiver, the output of which (*nAeCorrect*) indicates whether or not the incoming address is a match. Additional circuits allow for overwriting. The synaptic address receiver circuitry is shown in figure 2.

B. Broadcast

The signal labelled *Spike* in figure 1(b) is a pulse which is used to generate an increase in synaptic conductance. In this

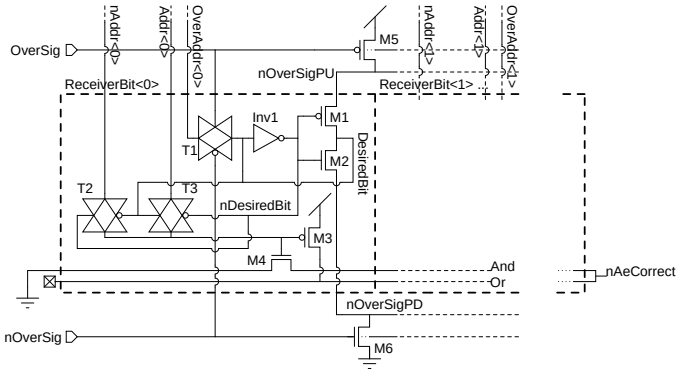


Figure 2. Address-event receiver circuitry. The receiver is composed of a chain of blocks each of which receives a single bit of the incoming address bus; one of these "receiver bits" is shown here (the zeroth bit). A bit of the address (*DesiredBit*) is stored in a memory element composed of Inv1 and M1-M2. An XNOR is continuously performed between *DesiredBit* and the incoming address bit (*Addr<0>*) by means of T2-T3 (the incoming bit's complement *nAddr<0>* is also required). The result of the XNOR contributes to a NAND gate implemented throughout the receiver array by transistors M3-M4. The result is *nAeCorrect*, indicating whether the full incoming address matches the full stored address. When *OverSig* goes high (and its complement *nOverSig* goes low), this is the signal for the receiver's address to be overwritten with the address on the *OverAddr* bus, a separate bus broadcasting the address of the potential pre-synaptic partner. *OverSig* chokes off transistors M1-M2 using transistors M5-M6 (these are common for all the receiver bits) while T1 opens, allowing *DesiredBit* to take the value of *OverAddr<0>*.

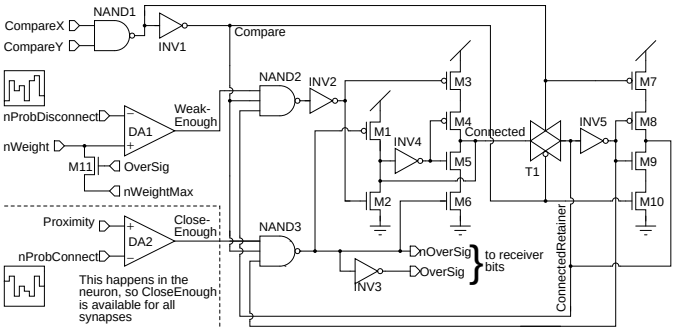


Figure 3. Circuitry for synaptic rewiring. The synapse's *Connected* state is stored in a memory element composed of INV4 and M4-M5. This state can be overridden by a disconnection signal from NAND2 and INV2, using M2-3, or by a connection signal from NAND3 using M1 and M6. *Compare* is driven high by the conjunction of *CompareX* and *CompareY* from row and column decoders, to indicate that rewiring is under consideration. While *Compare* is high, the value of *Connected* is latched in a separate memory element *ConnectedRetainer* (INV5 and M8-M9). This ensures that only connection or disconnection can occur, avoiding oscillations during the *Compare* signal. The complete condition for connection is that (a) *Proximity* is above the random value *nProbConnect* such that the neuron is judged to be *CloseEnough* (the output of differential amplifier DA2) to the potential pre-synaptic partner; (b) the *Compare* signal is high; and (c) the synapse is not currently connected i.e. *ConnectedRetainer* is low. On connection, the override signal *Oversig* and its complement are sent to the address-event receiver, allowing the address under consideration to override the address stored in the receiver bits; *nWeight* is also set to its strongest value, by M11. Complementary conditions apply to disconnection, the first being that *nWeight* is above *nProbDisconnect*. Amplifiers DA1-2 are capable of operating between *Vdd* and *Gnd*.

implementation, the length of the pulse is used, together with the weight of the synapse, to control the magnitude of the increase in synaptic conductance. It is therefore important for this pulse to last the same length of time at each synapse, bringing in issues of clock distribution. In fact, two other signals are broadcast across the chip at the same time, whose durations are used to parameterise the STDP circuit. Within each synapse, the result of the address-event receiver (i.e. the *nAeCorrect* signal from figure 2) is used to decide (with straight-forward digital circuitry) whether these pulses should be applied to the synapse.

C. Synaptic rewiring circuitry

Synapses can be individually targeted for rewiring by a chip-wide mechanism, which employs row and column decoders in the periphery. This allows for both the explicit setting of synaptic variables from an off-chip control mechanism, and for ongoing probabilistic rewiring.

The circuitry which implements the connection and disconnection algorithm is shown in figure 3. When a synapse is selected as a candidate for rewiring, its behaviour depends on its state of connectedness, stored in a static memory element. If it is connected then it is considered for disconnection. Its analogue weight voltage (*nWeight* – this is negatively defined, with a low voltage representing a strong synapse) is compared to a voltage *nProbDisconnect*, randomly chosen from a probabilistic distribution. If the weight is below the random value then the synapse is disconnected. The random value is common for all the synapses on the chip but is only used at one synapse at a time and changes between each usage, avoiding correlations between synapses. In this implementation the voltage is produced off-chip, but it could be produced on-chip in a mature implementation. By changing the Probability Density Function (PDF) of *nProbDisconnect*, different relationships of weight to probability of elimination can be implemented, for example, a thresholded rule, as used here, or alternatively a linear interpolation between high and low probabilities, etc.

If the synapse is disconnected and it is then selected as a candidate for rewiring, the possibility of it taking a new pre-synaptic partner is considered. The pre-synaptic partner considered is randomly chosen. The method of choosing the last neuron to have fired would be attractive as it could further reduce the amount of communication required, since it is an existing source of random addresses which is already available at the synapse. However, in practice this was not used, both due to practical difficulties in implementation, and to allow rewiring to be explicitly controlled where necessary.

The randomly chosen synapse addresses come from off-chip in the test implementation but they could come from an on-chip pseudo-random-number generator in a mature implementation. The potential pre-synaptic partner is latched separately by each chip and is broadcast across all chips at the point that a rewiring consideration takes place, using a different address bus to the one used to transmit spikes. It is also used to allow a cross-chip calculation to take place, providing a value, which can be made available at any one neuron, of the geometric proximity of that neuron to the incoming address. The synapse under consideration then compares this proximity value to a random value (*nProbConnect*), similar to the random value for

disconnection but separate, created according to a probabilistic distribution for synapse formation. If the proximity value is higher than the random value then the synapse becomes connected and it adopts the broadcast address of the potential pre-synaptic partner in consideration as its new stored address.

By changing the PDF of *nProbConnect*, differently profiled receptive fields can be created, for example, Gaussian receptive fields as used in the model in section IV, or any other radially symmetric profile where the probability of connection decreases monotonically with distance from the centre; the example of an isodensitic bounded (i.e. cylindrical) receptive field PDF is given in section VII-D.

According to the model in section IV, the standard deviation σ_{form} and peak formation probability P_{form} can change depending on the layer from which the potential pre-synaptic partner is chosen (i.e. whether the projection is feed-forward or lateral). Therefore it must be possible to choose *nProbConnect* from a different PDF depending on the layer of the potential partner. For this test system, values for *nProbConnect* are generated off-chip alongside the potential pre-synaptic partner addresses. However, to demonstrate one way in which this circuitry may be generalised, values of *nProbConnect* are generated separately for each of two distributions; peripheral circuitry on the chip then selects the correct value to broadcast to the neurons based on the part of the address of the potential pre-synaptic partner which indicates its layer (one bit in this case, since the test system requires only two incoming projections).

In the model under consideration there is a strong topographic mapping between successive neural layers, which applies to both the feed-forward projection and to the lateral projection, but this assumption is not essential to the system described here. The effect of proximity on the probability of rewiring can be eliminated altogether if it is not required, by reducing the distribution of *nProbConnect* to a binary choice between an extremely high value, i.e. *Vdd*, where the synapse will not connect no matter how high the proximity, and an extremely low value, i.e. *Gnd*, where the synapse will definitely connect regardless of proximity. This fact can be used for directly controlling rewiring where necessary, for example in setting up an initial network topology.

D. Euclidean Distance Circuit

In this section, circuitry is described which generates the proximity value required by the synapse formation rule.

1) Basic circuit: Euclidean distance calculation is based on the known principle of using the squared V-I relationship of saturated MOSFETs in strong inversion [27]. The circuit which calculates Euclidean distance is presented in figure 4 and its functioning is described here.

(A) shows the layout of chip. A 2D array of neurons contains a cell, marked *T* for target, which contains a synapse which is not currently connected and has been randomly selected to carry out its rewiring rule. A pre-synaptic neuron is also

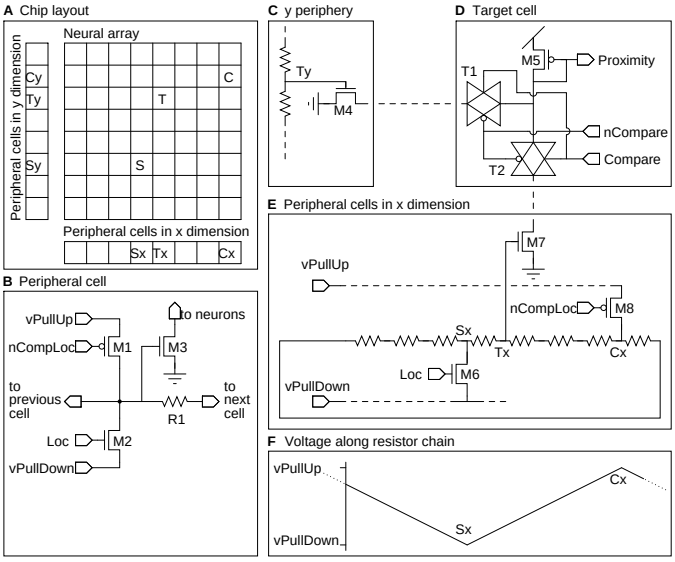


Figure 4. Euclidean distance calculation circuit (single chip, toroidal topology). (A) Layout of the chip; (B) circuit for each peripheral cell; (C) a section of the circuit along the periphery in the Y dimension; (D) circuit inside each neuron; (E) circuit along the periphery in the X dimension; (F) idealised voltage along the chain of resistors in (E).

randomly selected, whose ideal location in this neural layer is marked *S* for source. These locations are transmitted to the chip (or simultaneously to all the chips, in a multi-chip system) and decoded in peripheral row and column decoders (not shown). The complementary location, marked *C*, is the cell which is the furthest away from the ideal location (assuming a wrap-around topology). Along two edges of the chip are a row and column of identical peripheral cells. Peripheral cells corresponding to the target, ideal and complementary locations are marked *Tx/y*, *Sx/y* and *Cx/y* respectively.

(B) shows the circuit within each peripheral cell. A central node in each cell is connected to the central node of its two neighbours via a resistor R1. In Sx and Sy, the *Loc* signal is raised by the row and column decoders, switching on M2 so that the central node is pulled down to a voltage reference *vPullDown*. Likewise, in Cx and Cy, the *~CompLoc* signal is lowered, switching on M1 so that the central node is pulled up to the voltage *vPullUp*. The central node gates M3, whose drain is at the end of a wire which spans across the chip and is available to all neurons in the row or column corresponding to the peripheral cell.

(E) shows the circuit along one edge of the chip. All of the central nodes for each peripheral cell connect via resistors, to form a chain of resistors which travels along the edge of the chip and then wraps around at the edge. For simplicity, the only transistors shown are those which would be active in the case shown in (A). (F) gives a graph of voltage along the chain of resistors (as the number of cells tends to infinity). At the ideal location *Sx*, the voltage is at *vPullDown*. It then rises linearly in each direction, reaching a maximum of *vPullUp* at the complementary location *Cx*. The voltage on the resistor chain therefore represents the distance (in one dimension) from the ideal location, on a linear scale from *vPullDown* to

vPullUp. *vPullDown* is set to be approximately the threshold voltage of the nMOSFETs gated by the nodes of the chain of resistors, i.e. M3 in (B). If these nMOSFETs are saturated, the currents through them will be proportional (to first order approximation) to the square of the distance of their node away from the ideal location. There is also a circuit, which is equivalent to that shown in (E), along the vertical edge of the chip, as indicated by (C) (for simplicity, only the transistor which is active in this case is shown).

(D) shows the circuitry inside the target cell where the synapse is to carry out its rewiring rule. Two transmission gates T1-T2 are opened and the currents are allowed to flow through the nMOSFETs in the two corresponding peripheral cells, M4 and M7. These currents sum together to travel through a single diode-connected pMOSFET M5. Providing that M4, M7 and M5 all stay in saturation, and providing that only one cell has its transmission gates opened at one time, the voltage created at the gate of M5 is proportional to the square-root of the sum of the square of the distance from the ideal location of the potential pre-synaptic neuron to the targeted post-synaptic neuron in each dimension, which is the Euclidean distance.

To summarise then, an analogue current-mode circuit uses the squared V-I relationship of saturated transistors in strong inversion to directly implement a calculation of Euclidean distance. As transistors are operated in strong inversion the currents required are on the order of tens of microamps, but these currents need only flow for a few μs while a calculation is being carried out; given the rate at which synapses rewire in biology, the duty cycle of this circuit *per synapse* can therefore be extremely small. During a calculation, three currents flow, one through each resistor chain and one through the pMOSFET. Suitable sizing of the resistors and the nMOSFETS respectively can limit the magnitudes of these currents. The circuitry inside each neuron is rather compact, just one transistor connected by two transmission gates. (It would be possible to further refine the circuit to use single transistors in place of each of the transmission gates if strict assumptions were made about the possible range of *vPullDown* to *vPullUp*). The circuitry in the peripheral cells is rather less compact, partly because of the need to integrate resistor components with suitably large resistances and partly because longer nMOSFETs result in smaller currents. However, the area required by the peripheral circuitry scales as $C\sqrt{N_{chip}}$ (according to the definitions in section III) therefore the area required becomes increasingly irrelevant as the number of neurons integrated on each chip increases.

2) Circuit for non-toroidal topology: Periodic boundary conditions (i.e. a toroidal topology) have been used in the model presented in section IV for mathematical convenience, in line with models such as [28]. Biological topographic maps, however, typically do not have a toroidal topology. To implement non-toroidal topologies (as has been done in most models of topographic map formation e.g. [29]) an alternative version of this circuit is required. The circuit is explained in figure 5.

The fabricated chips contain circuits for both toroidal and non-toroidal topologies and these circuits share the same wires across the neural array and transistors within the neurons. That is, there is only duplication of circuitry in the peripheral cells.

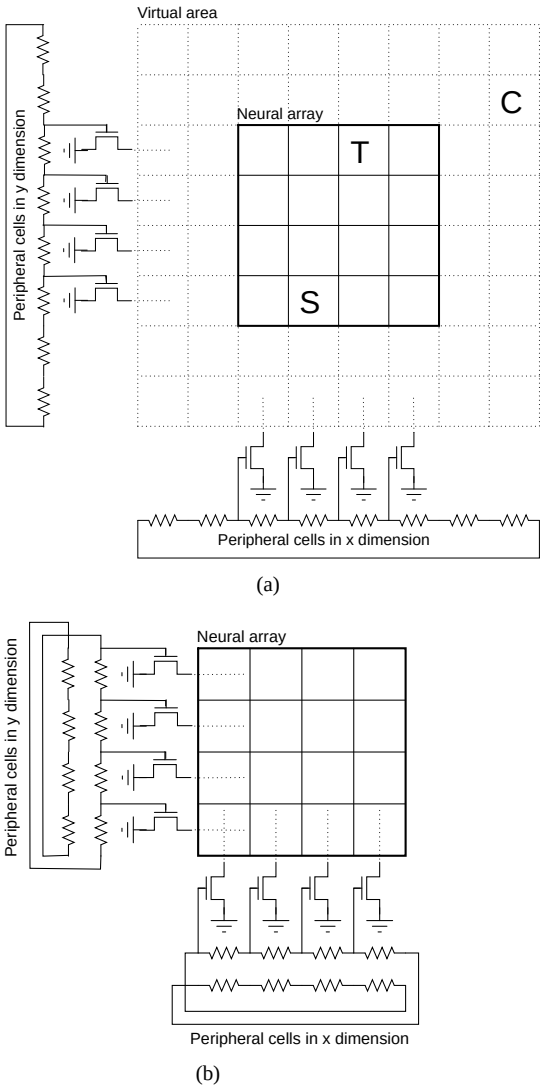


Figure 5. Proximity circuit for non-toroidal topology. The circuit is shown for a chip with a 4×4 neural array. (a) The actual neural array is extended into a virtual space twice as large in each dimension, by extension of the peripheral chain of resistors in each dimension. The source *S* and target *T* locations are always in the actual neural array, whereas the complement *C* of the source location is always in the virtual area; thus voltage on the resistor chain always increases away from S towards the edges of the chip in each dimension. Calculation then proceeds as for the circuit presented in figure 4. (b) For compact implementation, each peripheral cell contains the nodes and resistors for the corresponding source location and its virtual complementary location.

3) Multi-chip circuit: If larger neural areas are required than can be integrated on a single chip, the circuit which has been presented above can be easily extended to multi-chip systems. The wiring scheme is explained in figure 6 (for simplicity, only the circuit for toroidal topology is shown). Wires pass between chips in order to implement the chains of resistors, and there is one chain of resistors for each row and each column of chips. As the circuit is implemented over different dies, process variation between the dies may affect the performance of the circuit more than would be expected within a single die.

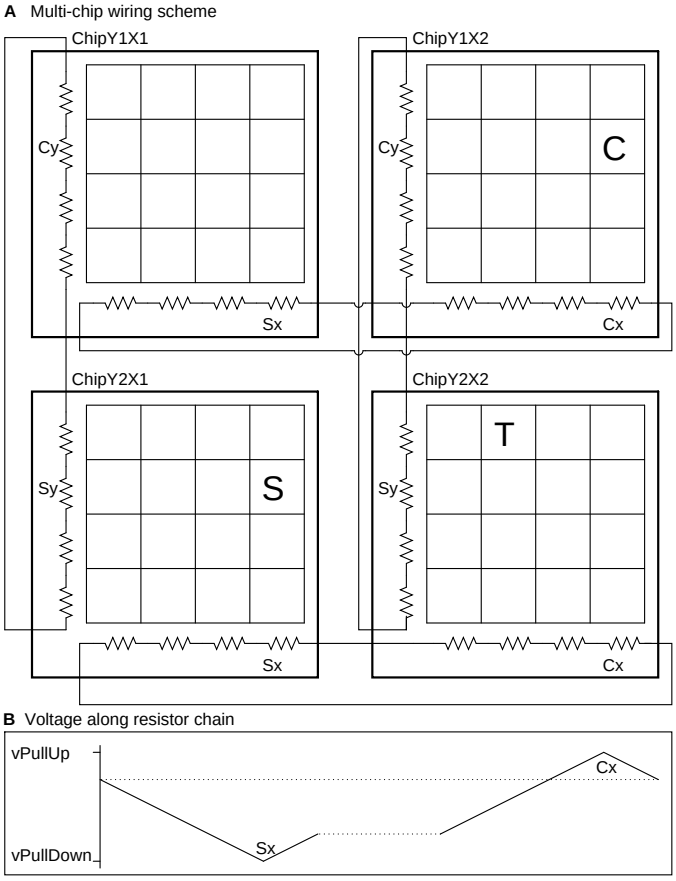


Figure 6. Proximity circuit for multi-chip system, shown for a 2×2 array of chips each with a 4×4 array of neurons. (A) layout of chips. The source *S* and target *T* locations can be anywhere on any of the chips, whereas the complement *C* to the source location is guaranteed to be on a different chip than *S*. There is one complete row of resistors for each row of chips and likewise for columns. In all other respects, the circuit functions as with the circuit presented in figure 4. Decoders, and transistors which implement the shorting of resistor chain nodes to *vPullDown* in Sx/y and *vPullUp* in Cx/y are not shown, for simplicity. (B) The idealised voltage profile along the chains of the resistors in the X dimension is shown for the case illustrated in (A). Voltage rises across across the neural arrays from *vPullDown* at *Sx* on the left chips up to *vPullUp* at *Cx* on the right chips. Dotted lines represent the voltage of resistor chains at links between chips.

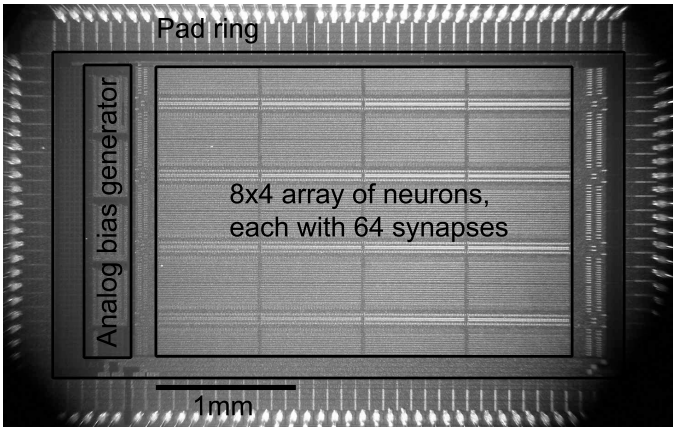


Figure 7. Micrograph of chip.

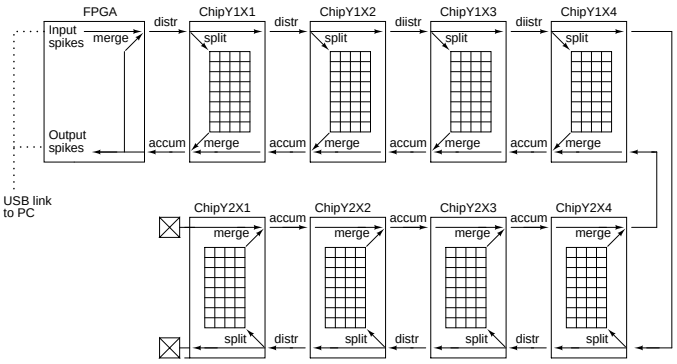


Figure 8. Grid system. Timestamped input spikes from the PC are sent at the correct time (or as soon as possible thereafter if the bus is not free) to the first chip (chip Y1X1). The first chip broadcasts this across its neural array and simultaneously (again depending on bus availability) transmits it to the second chip, and so on. The spikes are thus distributed throughout the grid using the chain of buses labelled *distr*. Spikes generated by neurons in the final chip in the chain, chip Y2X1, are transmitted to the next chip (chip Y2X2). This chip receives those spikes and merges them sequentially, using an arbiter, with any spikes from its own neural array, before transmitting them on. Thus spikes generated from the chips' neurons accumulate along the chain of buses labelled *accum*, until they arrive at the FPGA. Here they are timestamped and sent to the PC. They are also, optionally, merged with input spikes and redistributed across the chips, allowing lateral or recurrent synapses to be implemented.

E. Multi-chip system

The chips were fabricated using the AMS 0.35μ 4-metal 2-poly process. Each chip contains an array of $8 \times 4 = 32$ neurons. Figure 7 gives a micrograph of the chip. The chip was $\approx 14mm^2$, of which $\approx 6mm^2$ was dedicated to the neural array. Each neuron has 64 synapses (there are therefore 2048 synapses per chip). Each synapse has a 9-bit reprogrammable address-event receiver. The area of the synapse is $2841\mu m^2$, of which 56% is dedicated to the address-event receiver. The remaining area is dedicated to: storing the additional synaptic variables; implementing the connection and disconnection circuitry; creating an increase in the neuron's level of synaptic current when a spike arrives; and implementing STDP. 98.6% of the area of each neuron is dedicated to its synapses. 8 chips were linked together according to the scheme shown in section VI-D3, to create a 16×16 grid of neurons for the target layer. These chips were also linked in a grid arrangement [30] so that any neuron on any chip could send or receive address-events with negligible delay. Address-events from the (simulated) input layer could be sequenced from a PC and streamed with time stamps to an FPGA (Xilinx Spartan 3 on an Opal Kelly XEM3010 integration module). The FPGA would then transmit the address-events at the correct times. Spikes sent by the neurons on the chip were received by the FPGA; these were time-stamped and sent to the PC; they were also optionally merged into the stream of input spikes and sent back to the chips, in order to implement lateral or recurrent connections. The grid system is shown in figure 8; it differs from that used by [30] in that the addresses used are absolute, not relative. Whilst there are alternative schemes which would minimise the number of transmissions necessary for spike delivery, this scheme allowed for intervention in recurrent spike delivery by the FPGA, for testing purposes.

In normal operation (see the experiments in section VII-E) the

chips each consume $\approx 8.6mW$ when carrying out rewiring and $\approx 6.3mW$ without rewiring. The contribution of the system of spike delivery to these totals is too small to be accurately measured; simulations including capacitances extracted from layout showed that to broadcast a single spike internally across the neural array (not including transmission from the pads to the peripheral buffers) should consume $\approx 1.8nJ$, so in normal operation each chip should consume $\approx 18\mu W$ for internal spike delivery.

F. The rest of the circuitry

Space prohibits full discussion of the circuitry needed to implement the rest of the model presented in section IV (the interested reader is directed to [31]). The threshold, fire and reset and spike generation circuitry used is similar to that [32], except with no short-term depression, and with a controllable threshold implemented with a comparator instead of a source follower, decoupling the choice of threshold from the current used. Synaptic weight storage is volatile, using a capacitor. The implementation of STDP bears comparison to the circuits described in both [33] and [34], except that: (a) weight dependence of plasticity is not explicitly modelled (although some naturally arises from the imperfect ability of transistors to act as current sources); (b) the weight node is protected from subthreshold leakage by using negative gate-source voltage between events, in a manner suggested by [35], allowing a modestly sized capacitor (approximately 0.5pF) to retain some trace of a learnt memory over a period of minutes; (c) exponential decays of potential for depression and potentiation are implemented with switched-capacitor resistors. The excitatory synaptic conductance of all synapses is integrated and decayed, and used to implement a synaptic current onto the membrane, which is itself decayed, all using a switched-capacitor implementation.

VII. Results

Where neuron locations are described they are given as zero-based Y and X coordinates, e.g. neuron Y15X15 is the bottom-most right-most neuron in a given 16×16 layer.

A. Simultaneous receipt of a spike by many synapses

Figure 9 gives results which demonstrate the ability of synapses to simultaneously receive the same spike, by showing the action of incoming spikes on a variable voltage *SynCond*, internal to the neuron, which represents the total excitatory conductance due to synapses.

B. Channel capacity

The ability of the distribution chain of buses to distribute spikes is demonstrated in figure 10. A burst of spikes was distributed from the FPGA starting at 0s. Spikes were passed through each chip in the chain with a latency of $\approx 13ns$.

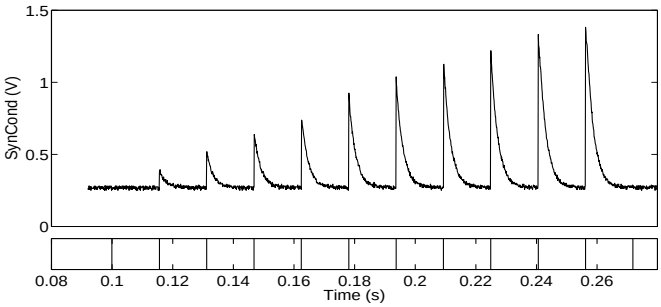


Figure 9. Simultaneous receipt of a spike by multiple synapses. Synapses of a single target neuron were programmed so that one synapse was receiving from neuron Y0X1 in the input layer, two synapses were receiving from neuron Y0X2, three were receiving from Y0X3, and so on, with one more synapse receiving from each incrementally higher neuron address, up to neuron Y0X10, which had 10 synapses. A sequence of spikes was sent in, starting at time 0.1s with frequency of 64Hz. The first spike in the sequence was from neuron Y0X0, the next from Y0X1, and so on. The time constant for the decay of *SynCond* was set to $\approx 2.5ms$. The upper plot shows *SynCond* for the neuron whilst the lower plot shows incoming spike times. The first spike did not elicit any response from the neuron since no synapse was programmed to receive from that address. Thereafter each spike caused a progressively (and approximately linearly) larger instantaneous increase in *SynCond*, as more synapses simultaneously received each subsequent spike.

The rate at which spikes can be distributed is limited by the total broadcast cycle time as defined by the (programmable) length of the pulse generated by pulse generator PG1 as shown in figure 1(c) plus various latencies imposed by the system. The delivery of a spike to a chip and the broadcast within it in the test system took upwards of 60ns; in section VIII there are suggestions on how to improve this. In fact in the test system created, the FPGA contributed the greatest delay; thus the speed achieved was far short of address-event delivery speeds achieved in recent publications (e.g. 41.66MHz [36]; 78.125MHz [37] - two systems for delivery of address-events, which would in fact be compatible with the system presented here, although the rates reported are for network links rather than delivery to end points). Nevertheless, even with spikes being *sent* at only $\approx 4.7MHz$, as the network was configured with an average fan-out of 64, spikes were being *received* at a rate of $\approx 300MHz$; increasing the fan-out would increase the spike delivery rate by the same degree. Although it is an unfair comparison, this spike delivery rate can only be matched in AER-based systems by those which also implement fan-out simultaneously by some means (e.g. [19]).

C. Proximity values

In order to parameterise the circuit with values for *vPullDown* and *vPullUp*, an experiment was carried out, the results of which are shown in figure 11(a). For values for the gate voltage of transistor M3 in figure 4 in the range $\approx 0.4V$ up to $\approx 2V$, *Proximity* reduces approximately linearly from its maximum level. With the gate voltage above $\approx 2V$, the rate of change of *Proximity* reduces as the nMOSFETs go out of saturation. 0.4V is therefore a good value for *vPullDown*. For linear performance across the entire range, 2V would be a good value for vPullUp; however, by extending *vPullUp* further into the non-linear region for *Proximity*, the total range of

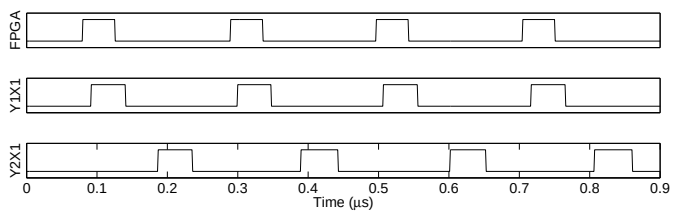


Figure 10. Address-events being distributed at maximum speed (with a minimal delay imposed, i.e. delay 1 in figure 1(c)). A set of address-events all with time stamp 0 were distributed from the FPGA as fast as possible. Graphs show the distribution buses' request signals output from: Top - the FPGA; Middle - the first chip in the chain (Y1X1); Bottom - the final chip in the chain (Y2X1). A rate of 4.74MHz was achieved. This includes all delays in the receiving and sending chain in the FPGA, PCB and chips. The largest delay is in the FPGA in this implementation (approx 150ns per cycle). The address-events took 107ns to pass through the grid, about $\approx 13ns$ latency per chip, whereas the receive and broadcast time of a chip was $\approx 60ns$.

Proximity can be extended, allowing greater accuracy in the comparison with *nProbConnect* for high proximities, whilst incorrect Euclidean distance calculations will only occur for pre-synaptic neurons whose ideal location is far from the post-synaptic neuron.

Figure 11(b-c) shows the results of the cross-chip *Proximity* calculation. (b) gives mean results from neurons on each of 8 different chips whilst (c) shows these results separately for the two most extremely mismatched chips, to give some indication of the effects of mismatch. In (c), each data set individually achieves good linearity and the main effect of mismatch is a shift in the output voltage, suggesting that the main cause of mismatch is threshold variation in transistor M5 in figure 4. This variation is likely to cause different neurons to have incoming connection fields with different amounts of spread. As stated in the figure caption, the proximity value of only one neuron on each chip was sampled. This was due to a design limitation; thus, mismatch comparisons are between neurons on different chips, not between neurons on a single chip.

D. Ability to form receptive fields probabilistically

The functioning of the synapse formation rule is demonstrated here in experiments where receptive fields are formed. The elimination rule is demonstrated later in section VII-E.

1) Creating differently shaped receptive fields : The example of Gaussian distributions is used to demonstrate how to generate the signal *nProbConnect*. Equation 1 is re-arranged for distance:

$$distance < Re\left(\sqrt{-2\sigma^2 ln\left(\frac{r}{p_{form}}\right)}\right)$$

This condition is then put in terms of *Proximity*:

$$Proximity > peak - \left|\frac{\delta Prox}{\delta dist}\right| .Re\left(\sqrt{-2\sigma^2 ln\left(\frac{r}{p_{form}}\right)}\right)$$

where *peak* is the peak *Proximity* voltage when distance = 0, and $\delta Prox/\delta dist$ is its gradient with respect to distance,

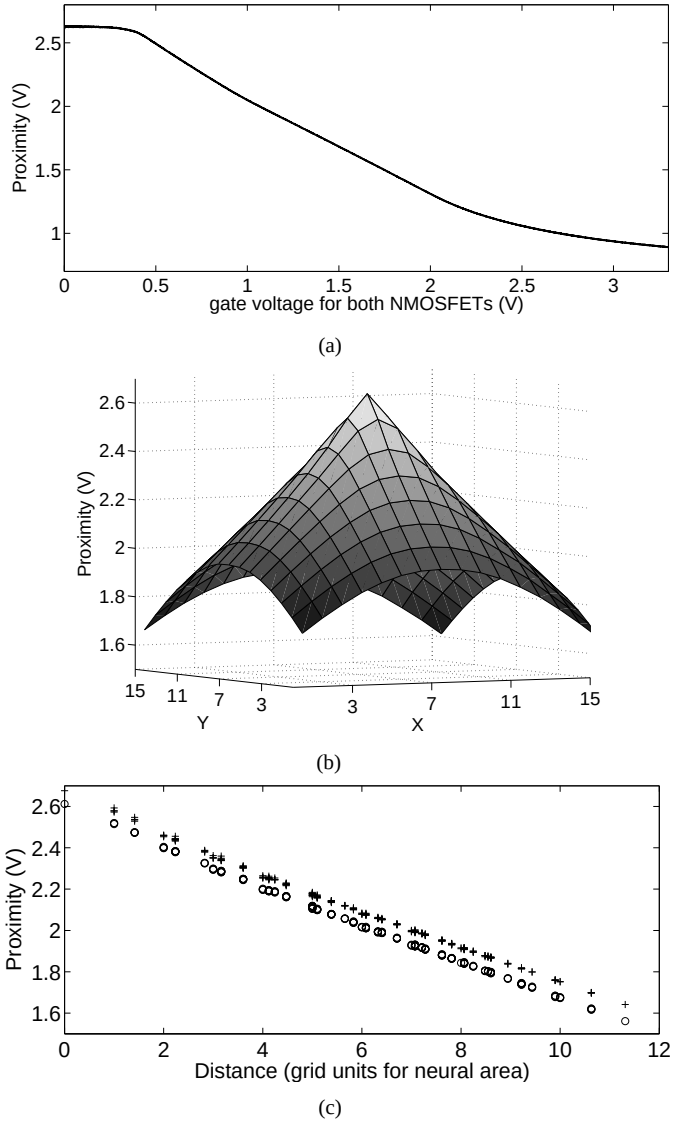


Figure 11. Proximity. (a) A single chip was configured so that the voltage along both of its resistor chains was linked and externally controlled. This value was swept in the range *0-3.3V (i.e. Gnd-Vdd)* and the *Proximity* value calculated by the bottom-right neuron (Y7X3) was recorded. (b)-(c) For one neuron in the same position on each of 8 chips, a synapse was considered for connection with a neuron in each possible source location and *Proximity* was recorded. $vPullDown = 0.4V$; $vPullUp = 2V$. (b) The results were shifted so that the highest *Proximity* in each case occurred at Y7X7, and the mean of the results from eight chips was taken for each location. The lowest *Proximity* occurred at the complementary position i.e. Y15X15. (c) *Proximity* is plotted against the distance which it is intended to represent. A different symbol is used for the data points from each of the two target neurons which gave the most outlying results; for clarity, neurons which gave intermediate results are not shown.

both taken from results such as those given in figure 11(a). The term on the right of the inequality was used to generate values for *nProbConnect*, based on the uniformly distributed random number *r*. *nProbConnect* was constrained to a minimum value of 0V, which guarantees connection, since *Proximity* cannot go so low. On the other hand, for $r > p_{form}$, *nProbConnect* was raised to Vdd, to ensure that connection did not take place. *nProbConnect* can be generated for any required PDF for connection by similar considerations, as can *nProbDisconnect* for any required disconnection rule.

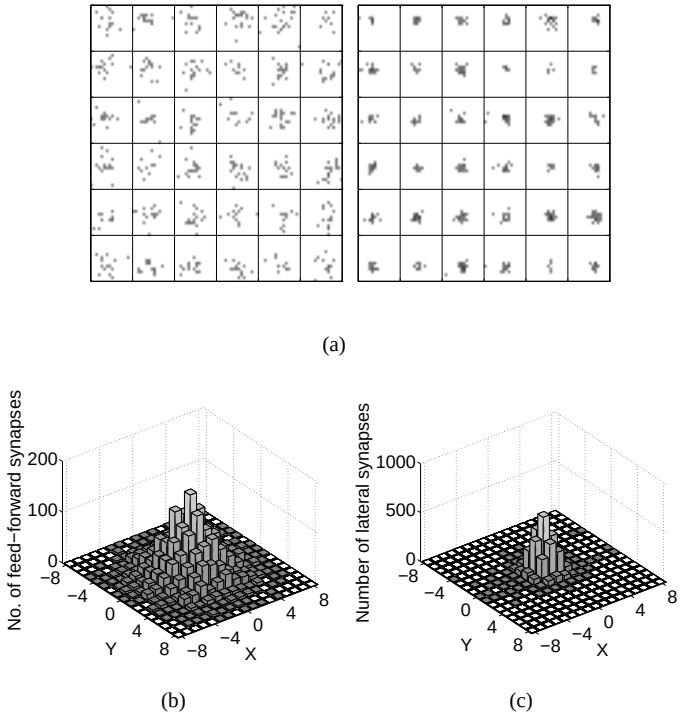


Figure 12. Gaussian receptive field formation. *nProbConnect* signals for feed-forward and lateral projections were created based on the respective pairs of parameters: $\sigma_{form-feedforward} = 2.5$; $p_{form-feedforward} = 0.16$; and $\sigma_{form-lateral} = 1$; $p_{form-lateral} = 1$. With no synapses initially connected, rewiring was run for 50 seconds with 10,000 rewiring iterations per second ($\approx$ 30 rewiring opportunities in total per synapse); *nProbDisconnect* was maximised, i.e. no synapse elimination. Afterwards there were an average of 16.5 feed-forward and 15.7 lateral synapses per target neuron. $\sigma_{measured-ff} = 2.51$, $\sigma_{measured-lat} = 1.22$. left: feed-forward receptive fields; right: lateral receptive fields. (a) Receptive fields for a subset of target neurons (for clarity, only target neurons in the range Y5-10, X5-10, are shown). Within each receptive field, white space indicates no synapses formed with neurons in that part of the afferent layer; pixels of increasingly darker grey shades indicate higher numbers of synapses with the neuron in that position; (b-c) combined receptive fields for all target neurons, with the afferent neuron whose ideal location matches the location of the target neuron centred at (0,0).

Figure 12 demonstrates the ability of the system to form receptive fields with a Gaussian profile. Note that lateral connections are formed by the same means as feed-forward connections, though σ_{form} is different for each projection. p_{form} was set to compensate for this difference, allowing the same overall probability of formation for each projection; this resulted in similar numbers of feed-forward and lateral synapses.

For a given desired standard deviation (σ_{form}) and a given number of synapses, each physical neuron has a different resulting standard deviation, measured relative to its ideal location, ($\sigma_{measured}$), due to mismatch. The distributions which result have mean $\sigma_{measured}$ which tends to be higher than the intended σ_{form}. The inputs could be fine-tuned to yield good approximations to the desired distributions, by altering the PDF of *nProbConnect* to compensate for any error; some manual adjustment of *peak* and $\delta Prox/\delta dist$ was used to achieve the results in figure 12, but such optimisation was not used for the experiment presented in section VII-E.

Figure 13 demonstrates another possible receptive field distribution, in this case a bounded isodensitic receptive field. Ideally the boundary of the receptive field would be sharp. However, the uncertainty at the boundary due to mismatch is apparent. Additionally, a small number of outliers can be seen, most of which result from a design flaw, described in the following paragraph.

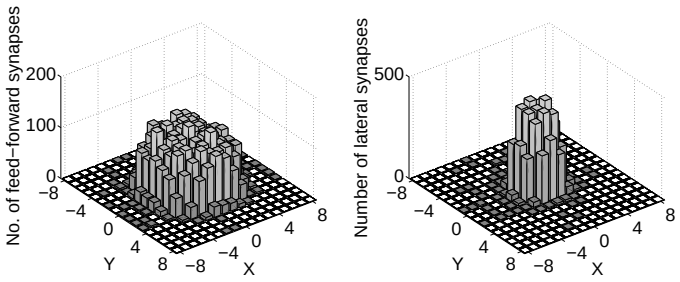
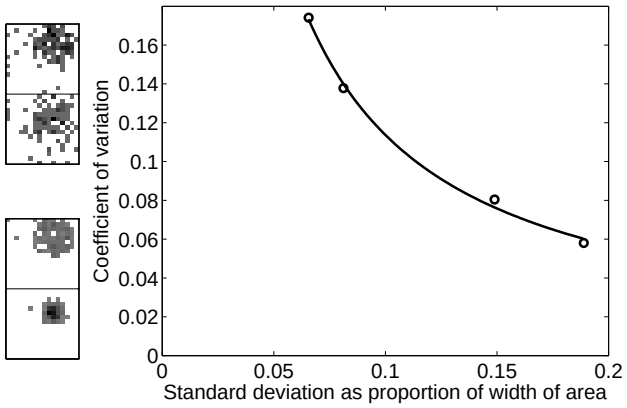


Figure 13. Formation of receptive fields with isodensitic bounded PDF. The experiment was as in figure 12 but with *nProbConnect* created based on a formation probability density which was constant up to a certain boundary distance and zero thereafter. Combined receptive fields for all target neurons are shown, with the afferent neuron whose ideal location matches the location of the target neuron centred at position (0,0); Left: feed-forward projection (*boundary distance*=5; $p_{form-feedforward} = 0.25$); right: lateral projection (*boundary distance*= 2.5; $p_{form-lateral} = 1$).

2) Insufficient open-loop amplification: In figure 3, the amplifier *DA2* which compares *Proximity* to *nProbConnect* is in open loop configuration and typically outputs a voltage close to Vdd or Gnd. Only when *Proximity* and *nProbConnect* are very close will it output an intermediate value, and the following gate *NAND3* applies further amplification to this signal to yield *nOverSig*. Nevertheless in a system where there are a large number of comparisons made, a proportion of these result in an intermediate value for *nOverSig*. Since this is applied separately to T1 in figure 2 for each receiver bit, mismatch in the transistors which make up the transmission gate provide an overriding signal of varying strength to each of the receiver bits, such that some bits take their new value whilst others do not. Thus in some cases, where the decision that the pre- and post-synaptic neurons are close enough to connect is marginal, the pre-synaptic address can be stored wrongly in the synapse by the failure to latch some bits. The small extent of this problem in practice can be seen in figure 13. To correct the design, a memory element could be inserted between DA2 and NAND3, with a similar design to INV5 and M7-M10, which would be activated shortly before the compare signal, ensuring that a stable all-or-nothing output state was achieved. To evaluate the performance of the model, significance tests were designed so that the small number of errors introduced do not skew results, as described in section VII-E.

3) Variation in variance: For a desired σ_{form}, each neuron develops a different $\sigma_{measured}$. This is partly expected due to the random nature of the input stream, and partly due to mismatch between neurons in the circuitry which generates the *CloseEnough* signal. This mismatch is partly due to the MOSFETs that generate the *Proximity* voltage, as suggested



(a) (b)

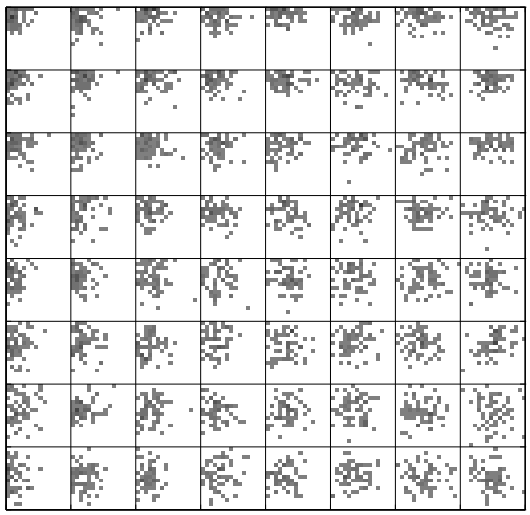
Figure 14. Variation of $\sigma_{measured}$ vs σ_{form}. (a) The results of several attempts to form Gaussian distributions were pooled, until the set of incoming connections from each layer for each target neuron had more than 64 afferent synapses (though different synapses could be between the same pair of neurons). All synapses in the pool are shown, for two exemplary neurons only, vertically adjacent to each other on one chip (neurons Y5-6, X11); upper pair: feed-forward synapses, $\sigma_{form-ff} = 2.5$; lower pair: lateral synapses, $\sigma_{form-lat} = 1$. The receptive field of the upper neuron is more diffuse than that of the lower neuron, though this is only apparent for the lateral synapses, with smaller σ_{form}. (b) For four initial pools of connectivity formed based on different σ_{form}, the coefficient of variation of $\sigma_{measured}$ is plotted. For generality, σ_{form} is expressed as a proportion of the width of the layer (i.e. $\sigma_{form}/16$). The best fit line is shown for a $1/x$ curve.

by figure 11(c). There are also likely to be differing offsets between neurons for amplifier DA2 in figure 3. The effect of this mismatch is more apparent when σ_{form} is small, as is shown in figure 14. (a) demonstrates visually that outliers can be seen more easily for a small σ_{form}. (b) shows this effect as an inversely proportional relationship between σ_{form} and the coefficient of variation of $\sigma_{measured}$.

4) Non-toroidal topology : Figure 15 demonstrates the ability of the chips to calculate proximity based on a non-toroidal topology by showing the effect of this on receptive field formation. In the receptive fields of neurons towards the edge of the chip, more connections form with pre-synaptic partners whose ideal locations are in the area which is available, resulting in denser sampling.

E. Effects of rewiring

As an indication of the utility of the system, an experiment was performed which establishes the difference in behaviour depending on whether synaptic rewiring is carried out. The network was initialised with all potential synapses of target neurons connected to topologically appropriate neurons, half of which were in the input layer (feed-forward connections) and half of which were in the target layer (lateral connections); all synaptic weights were maximised. Then input containing spatiotemporal correlations in firing rate were applied in two experiments, one in which synaptic rewiring was applied and one in which it was not. The extent of the resulting change in the receptive fields is compared.



(a)

Figure 15. Gaussian receptive field formation - Non-toroidal topology. *nProbConnectFF* was created based on $\sigma_{form-feedforward} = 2.5$, and $p_{form-feedforward} = 1$. *nProbDisconnect* allowed no synapse elimination and only feed-forward connections were considered. With no synapses initially connected, rewiring was run for 100 seconds with 10,000 rewiring iterations per second. Receptive fields for the feed-forward projection are shown for a example set of target neurons (neurons in the range Y0-7, X0-7). White space indicates that no synapses formed with neurons in that part of the afferent layer; squares of increasingly darker grey shades indicate higher numbers of synapses with the pre-synaptic neuron in that position.

Table III
EXPERIMENTAL PARAMETERS

Wiring	Inputs	Membrane & STDP
$N_{layer} = 16 \times 16$	$f_{mean} = 20Hz$	$V_{rest} = -70mV$
$S_{max} = 64$	$f_{base} = 5Hz$	$E_{ext} = 0V$
$\sigma_{form-ff} = 2.5$	$f_{peak} = 152.8Hz$	$V_{thr} = -54mV$
$\sigma_{form-lat} = 1$	$\sigma_{stim} = 2$	$\tau_m = 20ms$
$p_{form-ff} = 0.16$	$t_{stim} = 0.02s$	$\tau_{ex} = 5ms$
$p_{form-lat} = 1$		$\tau_+ = 20ms$
$p_{elim-dep} = 0.0245$		$\tau_- = 64ms$
$P_{elim-pot} = 1.36e^{-4}$		$g_{max} \approx 0.24$
$f_{rew} = 10^4 Hz$		

The experimental set up was as described in section VI-E. Parameters used were as in table III; the rationale for these choices of parameters is described in more detail in [22]. The values for V_{rest}, E_{ext} and V_{thr} are given as physiologically appropriate voltages rather than values in the arbitrary voltage range into which they were linearly mapped. The parameter g_{max} is difficult to quantify precisely due to slight non-linearity in the implementation, and A_+ and A_- are not given, as weight dependence in the synapse circuit means that these change depending on the state of the system. Rather, the biases which work together to create the parameters g_{max}, A_+ and A_- were treated as free parameters in order to achieve mid-range weight distributions and output spike rates close to the input spike rates. This parameterisation was nevertheless difficult to achieve due to interdependencies between parameters.

In order to create an initial network topology with which to programme the synapses at start up, the results of several attempts to form distributions with a given set of parameters (as in section VII-D1) were pooled, with synapse formations recorded separately for each neuron on each physical chip. Then a subset of incoming (dendritic) synapses was chosen from this pool for each neuron, up to a required number of feed-forward and lateral synapses (32 each).

The experiments ran for 5 minutes (based on the observation that there was little change in results after 3-5 minutes); rewiring rates were set to be much higher than in biology in order to observe the effects within brief experiments. Results are given in table IV. During the rewiring experiment, the number of connected synapses dropped from its maximum and the post-synaptic spike rate lowered.

For each neuron, the neural layer was searched for the location around which the afferent synapses had the lowest weighted variance (σ^2_{aff}), i.e.:

$$\sigma^2_{aff} = \frac{\sum_i w_i |\vec{p}_{x^* i}|^2}{\sum_i w_i} \; where \; x^* = \arg\min_{\vec{x}} \frac{\sum_i w_i |\vec{p}_{xi}|^2}{\sum_i w_i} \tag{2}$$

where i is a sum over synapses, $\vec{x}$ is a candidate receptive field centre, $|\vec{p}_{xi}|$ is the minimum distance from that location of the afferent for synapse i and w_i is the weight of the synapse; if connectivity is evaluated without reference to weights, synapses have unitary weight (this location is used rather than the more obvious centre of mass measure, in order to avoid a bias which is introduced when used in a toroidal space - see [22] for details). Then, mean σ_{aff} was calculated for all target neurons, only for the feed-forward projection.

It is now necessary to compare the neurons' receptive fields at the end of the experiment to the receptive fields which the neurons tend to develop in the absence of spiking input, for example, the receptive fields in the initial condition. However, σ_{aff} is dependent on the numbers and strengths of synapses and these can change during development (i.e. during the experiment); therefore to observe the effect of the activity-dependent development mechanism irrespective of changes in synapse number and strength, comparison was made in two ways. Firstly, for evaluating change in feed-forward mapping quality based only on changes in connectivity without considering the weights of synapses, a new mapping was created from the same pool used to create the initial mapping, using the final number of feed-forward synapses for each target neuron. σ_{aff} was then calculated for each neuron in each of the mappings and the means of these (i.e. mean σ_{aff}) were compared, applying significance tests between the values for the population of neurons for the two mappings, i.e. all the neurons for the final mapping *vs* for the reconstructed mapping. Having established what effect there was on connectivity, the additional contribution of weight changes was considered, by creating a new mapping with the same topology, taking the final weights of synapses for each target neuron and randomly reassigning these weights amongst the existing synapses for that neuron. The two mappings were then compared as described above. The key of table IV explains the abbreviations used hereafter.

Table IV
SUMMARY OF RESULTS

Case	Rewiring	No rewiring
Target neuron mean spike rate	11.2	15.4
Final mean no. feed-forward synapses per target neuron	25.4	32 (as initially)
Weight as proportion of max for the initial no. of synapses	0.50	0.41
Mean $\sigma_{aff-init}$	2.94	2.94
Mean $\sigma_{aff-fin-con-shuf}$	2.94	NA
Mean $\sigma_{aff-fin-con}$	2.51	2.94
p (WSR; $\sigma_{aff-fin-con}$ vs $\sigma_{aff-fin-con-shuf}$)	6.8×10^{-29}	NA
Mean $\sigma_{aff-fin-weight-shuf}$	2.45	2.91
Mean $\sigma_{aff-fin-weight}$	2.16	2.48
p (WSR; $\sigma_{aff-fin-weight}$ vs $\sigma_{aff-fin-weight-shuf}$)	2.3×10^{-22}	7.3×10^{-33}

For comparisons, mean σ_{aff} was calculated for the feed-forward connections of the following networks: (a) the initial state with weights not considered i.e. mean $\sigma_{aff-init}$; (b) the final network with weights not considered but only connectivity with all synapses weighted equally, i.e. mean $\sigma_{aff-fin-con}$; (c) for comparison with mean $\sigma_{aff-fin-con}$, the final number of synapses for each target neuron, "shuffled", that is to say, randomly placed in the same way as the initial synapses (not applicable in the case without rewiring), i.e. mean $\sigma_{aff-fin-con-shuf}$; (d) the final network including weights, i.e. mean $\sigma_{aff-fin-weight}$; (e) for comparison with mean $\sigma_{aff-fin-weight}$, the final connectivity for each target neuron with the actual weights of the final synapses for each target neuron randomly reassigned amongst the existing synapses, i.e. mean $\sigma_{aff-fin-weight-shuf}$. Results were compared using Wilcoxon Signed-Rank (WSR) tests on σ_{aff} for incoming connections for each target neuron over the whole target layer for a single experiment for each of the two conditions under consideration.

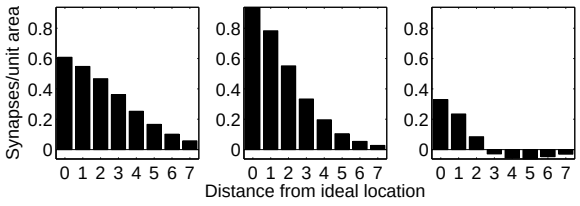


Figure 16. Mean feed-forward synapse density as a function of distance between the post-synaptic neuron and the ideal location of the pre-synaptic neuron, radially sampled at each whole unit of distance. Left: initial condition; centre: final condition; right: difference between initial and final conditions; Synapse density increased close to the centre but decreased further away.

The change in receptive field shape in the experiment with rewiring is shown in figure 16. The effect of rewiring is considered by comparing the case with rewiring to the case with no rewiring. Considering topology change, in the rewiring case, mean $\sigma_{aff-fin-con}$ drops to 2.51, c.f. 2.94 for mean $\sigma_{aff-fin-con-shuf}$; this drop is significant. In the case of no rewiring, mean $\sigma_{aff-fin-con}$ is constrained to remain at mean $\sigma_{aff-init} = 2.94$. Considering weight change, in the rewiring case, mean $\sigma_{aff-fin-weight}$ drops to 2.16, c.f. 2.45 for mean $\sigma_{aff-fin-weight-shuf}$. In the case with no rewiring, mean $\sigma_{aff-fin-weight}$ drops to 2.48, c.f. 2.91 for mean $\sigma_{aff-fin-weight-shuf}$. Both drops are significant.

Mean $\sigma_{aff-fin-weight}$ appears to be lower with rewiring than without it. It is not possible to say for sure that this superior reduction of variance is due to the effect of the rewiring

mechanism because the different numbers and weights of final synapses in each case make a comparison impossible. However, there is a good reason to believe that this is so: the drop in mean $\sigma_{aff-fin-con}$. This drop on its own indicates that the rewiring mechanism has helped to reduce variance and would also lay the groundwork for different final measures of σ_{aff} when weights are considered.

It can be seen then that (a) the weight-changing learning rule causes some reduction in the variance of the receptive fields, and (b) when the rewiring mechanism is applied, the network topology develops such that a variance reduction can be observed in the placement of the synapses, irrespective of their weight. Since the rewiring mechanism on its own can only generate synapse distributions according to the variance used by the formation rule it has no means to reduce this variance except by the influence from the effect of the weight change mechanism, whereby outlying synapses are weakened and become subject to preferential elimination. Thus, the variance reduction is caused by the weight-change mechanism and becomes embedded in the network topology as a result of the rewiring mechanism. In so doing, a developed trait which is initially stored in volatile memory on the capacitors which store synaptic weight is transferred to storage in the stable memory elements which encode network topology.

VIII. Discussion

A. Distributed receiver

The address-event receiver which has been implemented redefines synapse circuits as potential synapses and, in a straightforward manner, shifts the burden of decoding and receiving spike events into them. Alternative designs are possible and may prove beneficial. For example, if word-serial AER were adopted the memory elements in synapses could be decoupled from the circuitry which compares them to incoming address-events. Thus, while there would be one memory element for each bit of the address, there would only need to be enough comparison elements for one digital word of the address. This would allow for a more standard choice for a repeating memory element, and for better area scaling as the address space expands. If standard 6-transistor S-RAM elements were used then the size of the repeating memory element could be much reduced compared to the present implementation. As a further alternative, whilst floating gate technology is not best suited to storing synaptic weights because the high frequency of changes usually required by synaptic learning rules would lead to eventual dielectric breakdown, the low rates of synaptic rewiring in natural systems make storage of pre-synaptic addresses on floating gates an attractive option. Analogue storage of many address bits on a single gate (a form of multi-valued logic) could be explored for a possible space saving. The similarity in the function of the distributed address-event receiver to that of standard Content Addressable Memory is notable [38]; such a design could be used instead for a possible space saving, at the expense of a more complex comparison cycle, since it requires that lines which indicate a match are pre-charged before a comparison takes place. The speed performance of the system could be optimised by decoupling the delivery of an address event from the production of currents which drive the synapse's processes, since the pulse lengths necessary are longer than the minimum time necessary to simply register a pulse. Such optimisation would require pulse generators at each synapse, or alternative mechanisms for implementing synaptic processes. A simple improvement to speed performance would be to replace PG1 in figure 1(b) with a state machine driven by PG2 and PG3, eliminating one source of delay from the present design, since PG1 must extend the delays of PG2 and PG3 to allow a margin for error.

B. Rewiring circuitry

The synapse design which has been presented allows synapses to be rewired during operation. Whilst it is possible to impose an arbitrary network topology by external programming, it is also possible to allow a topology to form probabilistically and, if desired, to continue to develop within the system according to the biologically inspired model presented in section IV, without any details of the topology being made available off-chip. This system therefore allows a black-box approach to network wiring at the level of individual synapses. Rewiring probabilities can be made arbitrarily low, even achieving biologically-realistic rates of synapse formation and elimination, i.e. hours, days or months between events. Although supporting stochastic processes were generated off chip for this test system, these could be integrated on-chip.

A demonstration has been given of the development of receptive fields to achieve a reduction in spatial variance. Space prohibits a full exposition of the capabilities of the system but it is worth mentioning that it extends to less trivial forms of receptive field development, such as patterns of ocular dominance, in which receptive fields may become asymmetric or discontinuous when driven by appropriate forms of input (the interested reader is directed to [31]).

The proximity calculation circuits presented deliver a measure of distance which is iso-directional, resulting in fully radially symmetric receptive fields. This sets it apart from the systems of [19] and [39] which could achieve receptive fields with radial symmetry of limited order with angular phase linked to the axes of the chips. The measure of distance is also linear. Linearity is not in fact necessary for the system described, as supporting PDFs could be profiled to compensate for any monotonically decreasing measure of proximity, but the calculations necessary to profile the PDFs are simplified with a linear solution. This may prove advantageous if random value generation was moved on-chip.

The cross-chip proximity calculation circuit represents an advance in multi-chip neuromorphic systems. Whilst multiple chips have previously been used together in single systems, each chip has either represented a separate neural layer [40] or else a separate set of cells or function within a layer [39]. The system presented here, however, consolidates the

use of multiple chips to create a single expansive layer, by implementing a 2D cross-chip proximity calculation, which requires that all chips have a dedicated place within a 2D lattice of chips which form a neural layer. It remains to be seen how expansive a neural layer can be created. As this system scales, speed of operation may eventually become an issue, since the maximum speed at which rewiring can take place is inversely proportional to the number of synapses in the neural layer. Assuming, arbitrarily, that each synapse should be given one opportunity to rewire each hour, then at the speed of 10KHz used in this test system, 3.6×10^7 synapses can be accommodated.

The inverse relationship of σ_{form} to the coefficient of variation of $\sigma_{measured}$ suggests that this effect will become more problematic if the size of neural layers were scaled up and σ_{form} were not scaled up accordingly. That is, if receptive fields over small regions of a large neural layer were desired, the resulting receptive fields would be poorly matched in size. There are a few possible solutions to this. Firstly, there is the aforementioned possibility of stretching the range of the *Proximity* measure which represents close values for greater accuracy at the expense of accuracy at a distance. Secondly, with more complex decoding circuitry it would be possible to set two complementary locations on each resistor chain, a certain distance away from the source location in each direction; this would set a boundary for the range over which the proximity calculation would work, but would give better definition within that boundary.

The approach of distributing the circuitry which achieves rewiring throughout the synaptic array has been pursued on the grounds of conceptual neatness, and was achieved to some extent, although the rewiring rule does not run autonomously at each synapse but requires central control and central generation of supporting stochastic processes. As such, this design serves to highlight a pole on a spectrum of design choices regarding the amount of functionality implemented within synapses and indeed on-chip as opposed to in external digital processing. However, hybrid approaches are possible and may prove beneficial. Rewiring functions could be centralised to a single circuit on the periphery of each chip. This would remove about 20% of the area of the synapse design in the present system, with the expense that the synapse would have to buffer its analogue weight value out to the periphery and some additional signal rails would be required. A hybrid approach might be to implement disconnection in the synapse but connection at the level of the chip or the network. The decision to disconnect could then be easily based on the weight and would result in a simple digital message to peripheral circuitry. This could trigger the reassignment of that potential synapse circuit by external circuitry with the benefit that the implications for higher level network routing tables such as those proposed by [9] could be resolved at the same time.

The model in section IV assumes a fixed relationship between neural layers. Additionally, the neural layers are of the same size and shape and there is no transformation in the mapping between the layers. This allows for the address of the pre-synaptic neuron in the source layer to be directly decoded and used to specify a cell in the target layer as the ideal location of the pre-synaptic neuron. However, the model is not incompatible with transformations between layers. To apply transformations between layers in this system, some transformation must be applied to the source address. For simple transformations such as rotation through right-angles, mirroring, or expansion or compression by multiples of 2, simple bit-wise or bit-shifting operations on the source address prior to decoding would be sufficient, as in [39]. Any linear transformation could be achieved by matrix multiplication, and for ultimate generality, there could be a piecewise arbitrary mapping between addresses implemented by a look-up table; note that this proposal differs from the use of the look-up table identified in section I since source neuron addresses would not be converted to target synapse addresses but rather to source neuron addresses in a transformed topology, and axonal fan-out would still be implemented by the distributed address-event receiver.

IX. Conclusions

A design has been presented for an address-event receiver, which is composed of elements which are distributed through the synaptic array and act simultaneously on broadcast address-events. This allows a spike to be received simultaneously by all the synapses on the axonal arbor, allowing for arbitrarily large axonal arbors to be implemented without reducing channel capacity. This receiver is compatible with existing address-event senders. The receiver is reprogrammable during run-time, allowing synaptic rewiring to be implemented. The scalability of this system has been analysed and compared against existing systems with respect to the silicon area, transmission energy and transmission time required, as numbers of neurons and synapses in a system increase. This system scales particularly well in terms of speed as synaptic fan-out increases. Results have been presented from fabricated chips. In particular, spike sending rates have been shown which, when multiplied by the axonal fan-out being implemented, can be interpreted as spike delivery rates which are in excess of those achieved by any published AER-based neuromorphic system to date, except those which also implement simultaneous fan-out, demonstrating the potential speed advantage.

Circuitry has been developed for implementing synaptic rewiring within each synapse and results have been presented. The circuit is capable of connecting and disconnecting a synapse in response to either explicit external programming or a probabilistic learning rule. The connection rule requires a calculation of the distance between a neuron and the ideal location of a potential pre-synaptic partner; consequently, circuitry for Euclidean distance calculation has been presented. This circuitry is based on established principles for calculating Euclidean distance; its notable features include current mode operation across multiple chips and the capability of implementing both toroidal and non-toroidal topologies. The ability of the rewiring circuitry, together with the distance calculation circuitry, to allow the formation of receptive fields

based on radially symmetric PDFs with arbitrary relationships of connection probability to distance from the centre, has been demonstrated. Finally a demonstration has been given of how receptive field changes which develop from the influence of spiking input on synaptic weights, can become embedded in the topology of the network.

Acknowledgements

Although not described here, the fabricated chip contains sections of circuitry acquired from Giacomo Indiveri, Tobi Delbrück and Elisabetta Chicca at INI Zurich. AER Sender schematics were reworked for Cadence by Vasin Boonsobhak. We are grateful to Katherine Cameron and others at the Institute of Integrated Micro and Nano Systems for their help and to many people for helpful discussions at the Telluride Neuromorphic Engineering Workshop.

References

[1] R. Sarpeshkar, "Borrowing from biology makes for low-power computing," *IEEE Spectrum*, vol. 43, pp. 24–29, May 2006.

[2] D. Willshaw and D. Price, *Modelling Neural Development*, ch. Models for topographic map formation, pp. 213–244. MIT Press, 2003.

[3] H. Cline, "Sperry and Hebb: oil and vinegar?," *Trends in Neurosciences*, vol. 26, pp. 655–661, Dec 2003.

[4] M. Sivilotti, *Wiring considerations in analog VLSI systems, with application to field-programmable networks*. PhD thesis, Computer Science Dept. California Institute of Technology, 1991.

[5] M. Mahowald, *VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function*. PhD thesis, Computer Science Dept. California Institute of Technology, 1992.

[6] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address- events," *Circuits and Systems, IEEE Transactions on*, vol. 47, pp. 416–434, 2000.

[7] K. Boahen, "A burst-mode word-serial address-event link i: Transmitter design," *Circuits and Systems, IEEE Transactions on*, vol. 51, pp. 1269–1280, 2004.

[8] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2008.

[9] M. Khan, D. Lester, L. Plana, A. Rast, X. Jin, E. Painkras, and S. Furber, "Spinnaker: Mapping neural networks onto a massively-parallel chip multiprocessor," in *Neural Networks, IEEE International Joint Conference on*, pp. 2849–2856, 2008.

[10] M. Xiong, S. Pallas, S. Lim, and B. Finlay, "Regulation of retinal ganglion cell axon arbor size by target availability: Mechanisms of compression and expansion of the retinotectal projection," *Journal of Comparative Neurology*, vol. 344, pp. 581–597, Oct 1994.

[11] M. Palkovits, P. Magyar, and J. Szentagothai, "Quantitative histological analysis of the cerebellar cortex in the cat," *Brain Res.*, vol. 34, pp. 1–18, 1971.

[12] S. Deiss, R. Douglas, and A. Whatley, *in Maas W, Bishop CM. eds. Pulsed Neural Networks*, ch. 6: A pulse-coded communications infrastructure for neuromorphic systems, pp. 157–178. Cambridge, MA: MIT Press, 1999.

[13] S. Mitra, S. Fusi, and G. Indiveri, "Real–time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 1, 2009.

[14] D. Chklovskii, B. Mel, and K. Svoboda, "Cortical rewiring and information storage," *Nature*, vol. 431, pp. 782–788, 2004.

[15] B. Taba and K. Boahen, "Topographic map formation by silicon growth cones," in *Neural Information Processing Systems, Proceedings of,*, 2002.

[16] R. Vogelstein, U. Mallik, J. Vogelstein, and G. Cauwenberghs, "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses," *IEEE Transactions on Neural Networks*, vol. 18, pp. 253–265, 2007.

[17] S. Bamford, A. Murray, and D. Willshaw, "Large developing axonal arbors using a distributed and locally-reprogrammable address-event receiver," *International Joint Conference on Neural Networks (IJCNN)*, 2008.

[18] S. Liu, J. Kramer, G. Indiveri, T. Delbrück, T. Burg, and R. Douglas, "Orientation-selective aVLSI spiking neurons," *Neural Networks*, vol. 14, no. 6-7, pp. 629–643, 2001.

[19] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike-based event processing vision systems," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 53, no. 12, pp. 2548–2566, 2006.

[20] G. Turrigiano, "Homeostatic signaling: the positive side of negative feedback," *Current Opinion in Neurobiology*, vol. 17, pp. 318–324, 2007.

[21] K. Cameron, V. Boonsobhak, A. Murray, and D. Renshaw, "Spike Timing Dependent Plasticity (Stdp) can ameliorate process variations in neuromorphic VLSI," *IEEE Transactions on Neural Networks*, vol. 16, pp. 1626–1637, 2005.

[22] S. Bamford, A. Murray, and D. Willshaw, "Synaptic rewiring for topographic map formation," *International Conference on Artificial Neural Networks (ICANN)*, pp. 218–227, 2008.

[23] T. McLaughlin, R. Hindges, and D. O'Leary, "Regulation of axial patterning of the retina and its topographic mapping in the brain," *Current Opinion in Neurobiology*, vol. 13, no. 1, pp. 57–69, 2003.

[24] S. Song and L. Abbott, "Cortical development and remapping through spike timing- dependent plasticity," *Neuron*, vol. 32, pp. 339–350, Oct 2001.

[25] K. Miller, "Equivalence of a sprouting-and-retraction model and correlation-based plasticity models of neural development," *Neural Computation*, vol. 10, pp. 529–547, 1998.

[26] M. Prestige and D. Willshaw, "On a role for competition in the formation of patterned neural connexions," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 190, pp. 77–98, 1975.

[27] U. Cilingiroglu and D. Aksin, "A 4-transistor euclidean distance cell for analog classifiers," in *IEEE International Symposium on Circuits and Systems*, pp. 84–87, 1998.

[28] T. Elliott and N. Shadbolt, "A neurotrophic model of the development of the retinogeniculocortical pathway induced by spontaneous retinal waves," *Journal of Neuroscience*, vol. 19, pp. 7951–7970, 1999.

[29] G. Goodhill, "Topography and ocular dominance: a model exploring positive correlations," *Biological Cybernetics*, vol. 69, pp. 109–118, 1993.

[30] P. Merolla, J. Arthur, B. Shi, and K. Boahen, "Expandable Networks for Neuromorphic Chips," *IEEE Transactions on Circuits and Systems - 1: Regular Papers*, vol. 54, no. 2, pp. 301–311, 2007.

[31] S. Bamford, *Synaptic Rewiring in Neuromorphic VLSI for Topographic Map Formation*. PhD thesis, School of Informatics, University of Edinburgh, 2009.

[32] G. Indiveri, "A low power adaptive integrate-and-fire neuron circuit," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. 820–823, 2003.

[33] A. Bofill-i Petit and A. Murray, "Synchrony detection and amplification by silicon neurons with stdp synapses," *Neural Networks, IEEE Transactions on*, vol. 15, pp. 1296–1304, 2004.

[34] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, pp. 211–221, 2006.

[35] B. Linares-Barranco and T. Serrano-Gotarredona, "On the Design and Characterization of Femtoampere Current-Mode Circuits," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 8, pp. 1353–1363, 2003.

[36] H. Berge and P. Hafliger, "High-Speed Serial AER on FPGA," *IEEE International Symposium on Circuits and Systems*, 2007.

[37] D. Fasnacht, A. Whatley, and G. Indiveri, "A Serial Communication Infrastructure for Multi-Chip Address Event Systems," in *IEEE International Symposium on Circuits and Systems*, pp. 648–651, 2008.

[38] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 3, pp. 712–727, 2006.

[39] T. Choi, P. Merolla, J. Arthur, K. Boahen, and B. Shi, "Neuromorphic Implementation of Orientation Hypercolumns," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 52, no. 6, pp. 1049–1060, 2005.

[40] E. Chicca, A. Whatley, P. Lichtsteiner, T. Delbruck, P. Del Giudice, R. Douglas, and G. Indiveri, "A Multichip Pulse-Based Neuromorphic Infrastructure and Its Application of Orientation Selectivity," *IEEE*

Transactions on Circuits and Systems - I: Regular Papers, vol. 54, no. 5, pp. 981–993, 2007.

Simeon A. Bamford In 1995 he received a BA Hons in Artificial Intelligence from the School of Cognitive and Computing Sciences at the University of Sussex. After an entrepreneurial career he returned to study and in 2009 received a PhD in Neuromorphic Engineering from the Neuroinformatics Doctoral Training Centre at the University of Edinburgh. He currently works with the Modelling Complex Systems Group at Istituto Superiore di Sanità, Rome, and is also associated with the Laboratory for Synthetic, Perceptive, Emotive and Cognitive Systems at Universitat Pompeu Fabra, Barcelona.

Alan F. Murray was born in 1953 in Edinburgh, where he also went to school. In 1975 he received a BSc Hons in Physics at the University of Edinburgh, and a Ph.D. in Solid State Physics in 1978. He worked for 3 years as a Research Physicist (2 in Canada), and for 3 years as an Integrated Circuit Design Engineer. In 1984 he was appointed a lecturer in Electrical Engineering at Edinburgh University, became a Reader in 1991 and Professor of Neural Electronics in 1994. He is interested in all aspects of neural computation and hardware issues and applications have been his primary research interest since 1985. In 1986, he developed the "pulse stream" method for neural integration. His interests have since widened to include all aspects of neural computation, particularly hardware-compatible learning schemes, probabilistic neural computation and neural forms that utilise the temporal- and noisy characteristics of analogue VLSI - as well as applications of hardware neural networks. He is also developing a new interest in the interface between silicon and neurobiology, along with colleagues in Biomedical Sciences and Glasgow University. Alan Murray has over 200 publications, including an undergraduate textbook and research texts on neural VLSI, applications of neural networks and noise in neural training. He is a member of INNS and a Fellow of IEE, IEEE, HEA and the Royal Society of Edinburgh.

David J. Willshaw is Professor of Computational Neurobiology at the University of Edinburgh, UK. He has a long career in neural networks and computational neuroscience stretching back over 30 years with over 100 scientific papers. In 1992 he was the recipient of the IEEE Neural Networks Council 1992 Pioneer Award. He has worked in a variety of research areas including associative memory, novel algorithms for combinatorial optimisation (where with Richard Durbin he developed the Elastic Net algorithm for the TSP) and the development of patterned nerve connections in the visual and neuromuscular systems, which is his current research interest. He is past Editor-in-Chief of the computational neuroscience journal Network: computation in neural systems. Since 1984 he has held long term research support from the UK Medical Research Council and the Wellcome Trust. In addition he is the grantholder of the Edinburgh Doctoral Training Centre in Neuroinformatics and Computational Neuroscience funded by UK Research Councils. Since 2002, this Centre has trained over 50 PhD students from the physical and informational sciences who are applying quantitative approaches to neuroscience and to neurally inspired computing.