Connection-Centric Network for Spiking Neural Networks

Robin Emery, Alex Yakovlev, Graeme Chester
Newcastle University, UK
{r.a.emery, alex.yakovlev, graeme.chester}@ncl.ac.uk

Abstract

A reconfigurable network architecture applied to spiking neural networks is presented. For hardware platforms for neural networks that implement some degree of realism of interest to neuroscientists, connectivity between neurons can be a major limitation. Recent data indicates that neurons in the brain form clusters of connections. Through the combination of this data and a routing scheme that uses a hybrid of short-range direct connectivity and an AER (Address Event Representation) network, the presented architecture aims to provide a useful amount of inter-neuron connectivity. A connection-centric design can provide opportunities for NoCs such as optimising power, bandwidth or introducing redundancy. A method of mapping a network to the architecture is discussed, along with results of optimal hardware specifications for a given set of network parameters.

1. Introduction

The nature and purpose of connectivity between neurons in the brain is of great interest to neuroscientists as it may reveal how information is routed around the brain and is subsequently processed to produce higher level behaviours such as pattern recognition.

Connectivity in neural networks can be investigated using software and hardware models. However, limitations of the implementations of existing models - such as limited parallelism in software and limited neuron count, limited connectivity and limited learning capabilities in hardware - can frustrate such investigation[11]. These limitations can be overcome, but the solutions are non-trivial; for example, expensive massively parallel computer setups can be used to add parallelism to software simulations, or tightly constrained application specific chips can be designed and manufactured with limited versatility.

A new architecture for the implementation of a reconfigurable spiking neural network model in VLSI is formulated and presented. The architecture focuses on the connectivity between neurons, by implementing it as a hybrid of a relatively dense short-range spike routing resource and a relatively sparse long-range Address Event Representation (AER[17]) packet-based network-on-chip. This architecture aims demonstrate that the problem of limited scaling of interconnect in reconfigurable hardware can be alleviated through the use of a network-on-chip.

A method for mapping biologically-inspired networks to the architecture is detailed. Spiking neural networks are generated over a range of parameters and mapped to the architecture using a custom tool chain. The mapping aims to be faithful to the characteristics of the original network.

Following the background material, the architecture and mapping process are presented, followed by the results of the mapping process and a small demonstration circuit manufactured in 130nm CMOS.

2. Background

2.1. Neural Networks

Neurons in the brain communicate by short electrical pulses, known as "action potentials" or spikes[9], emitted when the potential across the cell membrane of the neuron exceeds a certain threshold. These spikes are generally stereotypical, of similar amplitude (about 100mV) and duration (about 1ms, with a refractory period of about 2ms). The information communicated between neurons is encoded in the timing and rate of spikes. The stereotypical nature of spikes has led to much effort in the area of spiking neuron models [8], as such models offer the potential to help to understand brain functions at a higher level than the neuron, possibly bridging the gap between neural networks and expressive behaviour such as sight or emotion.

A neuron forms connections with other neurons via chemical synaptic junctions. A spike propagates through the axon of the producer neuron to the synapse. This triggers the generation of a chemical neurotransmitter which moves over the small gap between the producer and consumer halves of the junction. At the consumer side of the synapse on the dendritic tree, the neurotransmitter produces

978-1-4244-4143-3/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

an offset of the membrane voltage of the consumer neuron. The size of this offset is known as the "synaptic efficacy", otherwise known as the strength of the synapse, and varies between 2mV and 3mV[13], although the effect is reduced by the time it reaches the soma of the neuron. The efficacy of the synapse is plastic, and varies according to a learning rule[19]. This plasticity is widely thought to be a major contributer to learning.

To make effective use of spiking models when attempting to understand the brain, the structure of real neural networks must be observed. The primary visual cortex is one of the most extensively studied areas of the brain as it is the earliest and simplest cortical area for visual processing, and shows a strong relationship to the topographic layout of the retina of the eye.

Experimental data on the morphology of neural networks in the brain provides varied results due to the difficulty of obtaining a reliable tissue sample and subsequent processing of a large amount of data. However, some conclusions can be drawn. Data on the density of neurons in the visual cortex of the rat indicate a range of $32,000 - 87,000\,mm^{-3}$ [10] (this paper makes an overestimate of $100,000\,mm^{-3}$ to compensate for possible losses during sample preparation), perhaps higher than $200,000\,mm^{-3}$ in primates [1].

Several factors affect the total number of connections made by a neuron. [1] reports that neurons in the monkey visual cortex receive an average of 3900 synapses. [2] indicates that the number of synapses of neurons in the visual cortex of the cat ranged between 975 and 9641. Results presented in [10] indicate that a neuron may receive about 30 synaptic inputs directly from similar neuron types and an order of magnitude more indirectly via small locally-projecting interneurons[1]. This paper also shows that while the probability of connection decreases as one moves away from the soma to about a tenth of what it was, the actual *number* of connections increases. This is due to the high density of neurons observed; however the volume examined in the paper was small, providing data for proximal connectivity only.

Work undertaken by Binzegger et al. [3] provides some results for distal connectivity. They find that axons of neocortical neurons form connections in multiple, separate clusters (see figure 1). Neurons formed between one and seven clusters, with almost all forming at least two and most forming about two or three. The cluster with the highest number of connections (primary cluster, average 2036 connections) typically formed near to the parent soma, and the distance to the next cluster was proportional to the size of the parent cluster. The second cluster typically contained four times fewer connections than the primary cluster, and the distance from the soma ranged between $0.5 - 2\,mm$. The work theorizes that the "spoke-like" arrangement of clusters may be a means of routing information.

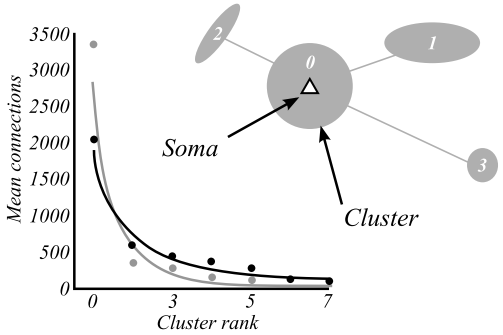

Figure 1. Neural connectivity in clusters (for a single neuron). The graph shows the relationship between the cluster rank and the number of connections formed (black: excitatory, grey: inhibitory). Adapted from [3].

2.2. Neural Networks in Hardware

Using digital or analogue circuits in silicon to model neural networks can offer several important advantages over software alone, notably massive parallelism and the associated improvement in simulation time.

Existing work on hardware for neural networks is mostly focused on application specific embedded hardware, rather than on accelerating general purpose computing. Artificial classifier networks tend to be the most common. Of neuromorphic devices, perhaps the Silicon Retina [12] is the most well known, but many other devices exist, generally affected by limited connectivity or neurons that are difficult to configure. The most successful reconfigurable system of recent years is the Silicon Cortex (SCX) [16], a board-level infrastructure for multiple chip neuromorphic systems in which analogue VLSI chips are connected by digital hardware. However, a limited neuron population and a complex infrastructure makes this system unsuitable for large-scale networks.

FPGAs would seem to offer an attractive compromise between software and custom hardware for neural network models. They offer relative low cost, versatility of function and a quicker design turnaround. However, preliminary work by the authors that based a reconfigurable neural network on a Xilinx FPGA determined that a hardware system must operate neurons asynchronously to be of interest to neuroscientists, and that a large area was required to implement a useful neuron due to the general nature of the FPGA logic. As a result of the large area required and the ample

[1] The neocortex is made up primarily of excitatory pyramidal neurons (80%) and inhibitory interneurons (20%).

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

routing resources available, much of the interconnect was unused and was wasted. A recent review of reconfigurable hardware for neural networks provided by [11] concludes that FPGAs do not efficiently implement biological neuron models (in part due to the size of multipliers), and that the interconnect structures do not scale with network density.

Address Event Representation (AER) is a means of asynchronously multiplexing stereotypical "events" (such as neural spikes) over a link shared between groups of processing blocks. It was first used in Mahowald's stereoscopic vision system [12], and a draft standard has subsequently been produced [17]. In AER, an event is represented by a packet consisting of at least a source or target address. If the address of the target is used in the packet, the packet can be routed over a complex network topology. Events are digital - abstract - so any degradation can be restored without affecting the information conveyed, reflecting the physiology of real neural networks. Where AER is used for communication all nodes share the same frame of reference, thus time can be seen as being the same throughout the system, with no skew. Information is then conveyed in the time and number of events, providing a robustness to process variation between chips. When used in an adaptive neural system, a small number of lost events will not massively affect the operation of the system, providing an inbuilt robustness to transient faults.

The Network-on-chip (NoC)[7] paradigm is particularly attractive for spiking neural networks as it offers scalability, parallelism and flexibility. In a NoC structure, functional blocks are connected to other blocks by high speed links and routers. Information is transmitted as packets, allowing the nodes on the network to implement advanced behaviours such as load balancing and prioritisation. Many network topologies can be used, but the most common is a mesh structure with a router at each crossing point. The blocks themselves are usually asynchronous or mesochronous relative to each other. For example, NoCs have been applied to hardware neural models as part of a large multiprocessor project[14].

In spiking neural networks all nodes are asynchronous and operate in parallel. The majority of links between neurons are relatively short range, but there is still a significant number of long range links. The interconnect of a NoC is scalable as the links of the network are shared by many signals, overcoming the problems associated with a large amount of dedicated interconnect such as excessive delay, wasted area and reduced reliability. As all links are active at the same time and the separate blocks are not synchronous, a high level of parallelism can be achieved. For a reconfigurable device, the flexibility of routed packets helps to reduce configuration effort, and for an adaptive device (one that supports real time learning) it can help offer an improved robustness to faults. As a spiking neural network is event based, an event based protocol such as AER is appropriate for the network links.

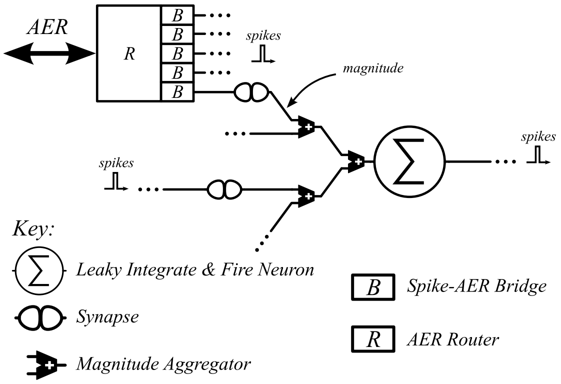

Figure 2. System elements

Combining a reconfigurable architecture with a NoC and AER provides the potential to implement a neural network that makes optimal use of the hardware resources. By using general traffic statistics or by back-propagating real traffic data, inter-tile links of the NoC can be fully utilised, partly utilised to allow for bursts of traffic, or disabled to conserve power.

In summary, a reconfigurable FPGA-like neural network device would be of interest to neuroscience, and observations of the structure of the brain support a hybrid of local and distal connectivity with a distinct transition from one to the other. A network-on-chip can provide a scalable interconnect structure for a hybrid reconfigurable platform.

3. Network Architecture

The structural elements of the proposed architecture are depicted in figure 2. The neuron is a simple leaky integrate-and-fire model, which functions as follows: 1. the neuron sums spikes as they arrive, with the magnitude of the increment forming part of the input. 2. The integration is subject to a small but constant decay. 3. If at any time the value of the integral exceeds a static threshold, the sum is set to zero and a spike is "fired". The focus of this system lies with the connectivity between neurons. While the leaky integrate-and-fire approximation of a neuron is simple, it captures enough of the behaviour of a real neuron to generate interesting network activity [4].

Neurons are connected to other neurons via synaptic junctions. In response to the arrival of a spike at the input, the synapse will emit a value that is a function of the strength of the synapse; this value then propagates to the input of the neuron. The strength of the synapse is plastic according to the rule of Spike Timing Dependant Plastic-

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

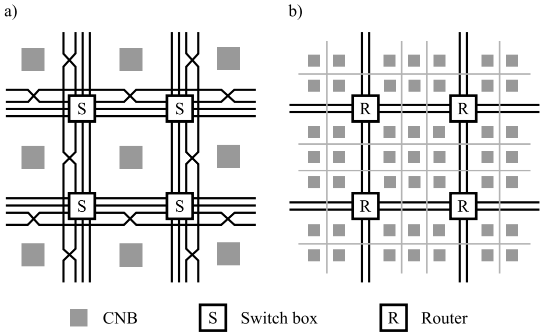

Figure 3. Network structure. a) example of local inter-CNB connectivity; b) example of superimposed network.

ity (STDP[19]), with a high output magnitude representing a strong synapse and a low magnitude representing a weak synapse. A positive magnitude represents an excitatory input, and a negative magnitude represents an inhibitory input.

A neuron, multiple synapses and a tree of magnitude aggregators are collected to form a Configurable Neural Block (CNB). These blocks are repeated many times to form a Cartesian grid, with spike links between blocks on a dedicated configurable routing resource. In case more synapses are required to feed a neuron, the neuron in each CNB can be bypassed and the aggregated post-synaptic magnitudes fed over short single-hop links into other CNBs.

In natural neural networks, the distribution of connection lengths favours relatively short connections. For this architecture, the shorter inter-CNB "direct" spike connectivity consists of static wires of different lengths that form hops between configurable blocks (figure 3a). Configurable switch matrices placed between the CNBs enable routing of spike signals over the grid. Once configured, each route is a dedicated connection carrying spike information between a fixed spike source and a spike sink.

While the majority of connections in a neural network are relatively short, there are still many longer connections that must be made. The cost of making direct connections using long interconnects which require strong drivers and intermittent buffering is likely to be too high given the dedicated nature of the link. In order to minimise the cost of these connections, a network is *superimposed* over the shorter direct connectivity (figure 3b); the classical NoC method of imposing hard tile boundaries does not accurately reflect the structure of a neural network. The grid of CNBs is divided into rectilinear tiles of a suitably populous size, with one router per tile. Each router is connected to other routers via the serial AER inter-tile links in the north, south, west and east directions. The routers interface with the short-range interconnect (effectively the "core" of the tile) via simple bridges which provide conversion between the spike protocol and the AER network protocol. Tiles are addressed using a the simple *X,Y* scheme and individual targets inside the tile are addressed using an identification number that represents the appropriate AER-to-spike bridge.

All of the configurable elements of the network provide serial configuration ports. The system elements are connected together to form a serial configuration chain in a similar manner to an FPGA, through which a bit stream is clocked to set switches for connections and registers for system elements such as neural thresholds. By implementing a global pause signal, this chain can also be used to inspect the current state of the network by clocking out the values of registers, such as the current integral values of neurons.

This architecture will be demonstrated and evaluated by generating networks with similar characteristics to topographic structures seen in the brain, such as orientation maps in the visual cortex. These networks will be used to configure the system elements, then the architecture will be assessed for suitability for purpose by observing traffic loads, especially where bottlenecks could be affecting the quality of simulation, and by comparing outputs to real observed neural behaviours.

4. Mapping to a Field-Programmable Neural Array

The reconfigurable spiking neural network architecture discussed in the previous section was designed for biologically inspired networks. As an aid for exploration of the mapping and architecture, networks are generated by a software tool. The characteristics of these networks are based on data in [3]. The software tool takes in as parameters the neuron count N and the neuron density d as the number of neurons per mm^2. It then places the neurons uniformly at random on a 2-dimensional plane with square limits determined by d. Each neuron forms between 1 and 7 circular clusters of variable diameter, distance and angle from the soma of the neuron. Surrounding the soma is a circular catchment area of variable diameter (a rough approximation of a dendritic arbour); when a cluster overlaps the catchment area, a connection between the two neurons is formed (figure 4a). At the time of writing, the network generation tool creates networks by placing neurons and clusters uniformly at random. Work is currently underway to enhance these networks with groupings of clusters, producing uneven connectivity and networks that feed information towards fixed points.

The mapping flows along similar lines to placement and routing tools for IC design. These tools start with a gen-

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

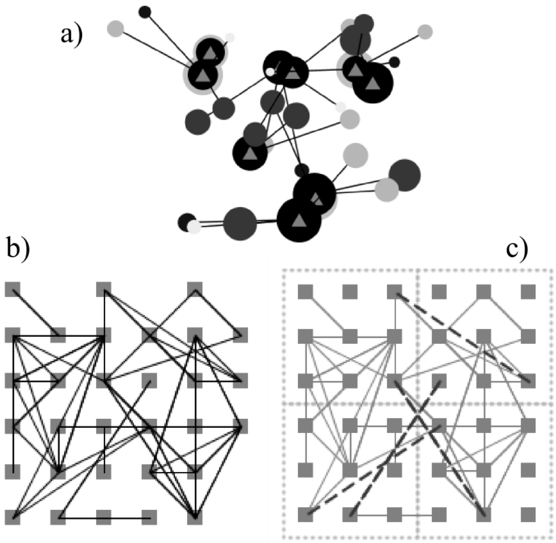

Figure 4. Mapping of a neural network to the CNB array. a) The original distribution of neurons; triangles: somas, clusters: circles. b) The network mapped to a grid; squares: neurons, lines: connections. c) The grid mapped to the reconfigurable array; squares: CNBs, solid lines: direct connections, dotted lines: tile boundaries, dashed lines: inter-tile connections.

eral floor plan, and then place cells according to this floor plan aiming to minimise area and interconnect cost for wire length, timing and congestion. The floor plan in this case is the layout of the generated network, however the placement and routing presented here is different in that it aims to optimise the mapping of the spiking neural network over the hybrid of multiple interconnect types of the architecture described in the previous section.

The generated network is mapped to a grid (figure 4b), where each grid point represents a CNB and every neuron is mapped to a single CNB. This is accomplished by dividing the network plane into a grid of equally sized squares with an area calculated by dividing the plane area by N, giving the mean area per neuron. This process provides a starting point, as there will often be several neurons in a grid square when only one is allowed. To correct this, the neuron count per grid point is treated as another dimension and the grid is repeatedly "squashed", spreading out the excess neurons until no grid points with multiple neurons remain. This method retains the largely proximal nature of the connectivity between neurons by minimising the disruption of the original layout as much as possible.

When the grid is complete, the AER network infrastructure is superimposed (figure 4c). An additional parameter C, the number of CNB units that form a tile, is provided. C is used to calculate the number of tiles that are superimposed by dividing N by C. The majority of tiles will be the same size (and usually square) but can vary a little if required by the dimensions of the grid.

Layer	*Direct Connection*	*Indirect Connection*
Network (OSI 3)	*Not required* (point-to-point only, no routing)	AER. All nodes have unique address. Full-duplex mesh topology. Unicast
Data-link (OSI 2)	Spikes are buffered at the the receiver; new spikes are dropped if there is contention for the link. Source and destination are implicit, no access control required. No error detection service on physical layer.	Spikes can be ingress and egress queued if link is busy. Error detection service on the physical layer (malformed frames).
Physical (OSI 1)	Single wire per link; RZI; spike is falling edge. Simplex, fixed at configuration time. Pulse has set minimum width, glitches will be filtered.	High-speed asynchronous serial on-chip link; simplex. Simple error detection on frames; acknowledge signal.

Table 1. Summarised direct and indirect spike protocols

As the network is superimposed the connections between CNBs are not simply routed over the AER network if they cross a tile boundary. A simple rule is used: if the connection reaches further than the longest possible route within a tile (corner-to-corner), and crosses at least one tile boundary, it is deemed to be a long connection and is routed over the AER network. This method maintains the hybrid nature of the connectivity.

Spike information is conveyed through the system in two forms: abstract point-to-point spikes, and AER packets. To define the interaction between these forms, a series of protocols have been defined according to the OSI layer model[18] and summarised in table 1. The protocol

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

stack facilitates the translation between spikes and packets by defining the services that must be offered by the neurons, synapses and bridges. The stack does not make general provision for tolerance of error or saturation, in keeping with the biological influence. However, a greater level of error detection is required in areas that are more greatly abstracted from the biology. Specifically, the loss or delay of many AER packets - representing many links - would affect many more elements than the loss of a similar number links between neurons in the brain[15].

5. Implementation & Results

5.1. Model

To facilitate the exploration of the hybrid spiking network architecture and make the analysis more efficient, an abstract model was implemented at the transaction level using VHDL. The following tool chain was developed around this using a mixture of Java, VHDL and scripting languages:

1. Network Generator
2. FPNA Mapper
3. Abstract network simulation (VHDL)
4. Traffic Analysis

The network generator and FPNA mapper have already been described. The VHDL simulation model was implemented at the transaction level and consists of the structural elements of the system along with spike traffic generators and traffic monitors. The delays of each element were arbitrarily small, as the purpose of the model was to give an indication of relative traffic levels. When the model is simulated, the traffic monitors embedded in the model output a report to a file. When the simulation is complete, the traffic analysis tool is used to extract metrics from the data.

The tool chain is completely automatable as the output of each stage forms part of the input to the next without any human intervention. Currently, the tool chain can generate and map networks of up to $N = 10000$ neurons and a density of $d = 600mm^{-2}$ within an acceptable time (less than an hour). The amount of work the tools must complete increases exponentially with N and d. Work is underway to improve the efficiency of each stage to attempt to produce networks of up to $N = 100000$ neurons and a density of $d = 100000mm^{-2}$ on a usable time scale (hours - 1 day at the most).

The effect of the input parameters N, d, and C upon the result of the mapping process are shown in figures 5, 6 and 7. A sample of 60 mapped networks was made, with N varying between 10 and 10000, d varying between $200mm^{-2}$ and $600mm^{-2}$, and C varying between 0 and 1000. These parameter ranges were chosen to expose trends that may apply to all network sizes, provide results for the sort of neuron numbers that would be implemented on a small- to medium-size FPNA device, and produce a result in a reasonable period of time.

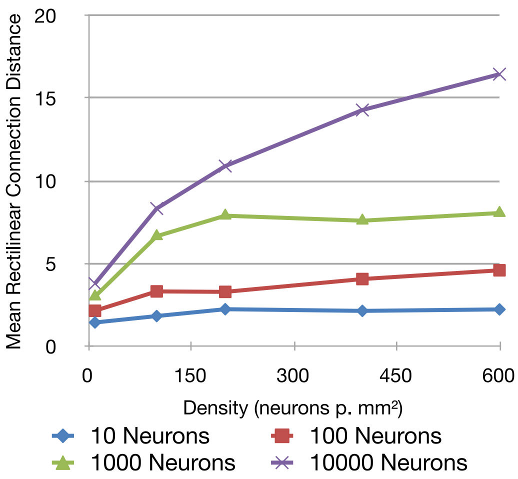

Figure 5. CNB output connection lengths by neuron count

The average rectilinear distance traversed by the outputs of CNBs is dependant upon N and d, as shown in figure 5 (mean: 5.7, range: 1-16.7). At larger neuron counts the distance increases with neuron density due to the “squashing” behaviour of the mapping process, as neurons are spread out to accommodate each other. It is expected that this distance continues to increase with density, increasing the demand for routing resources.

The average number of inputs to each CNB increases with d as shown in figure 6 (mean: 30.5, range: 0.5-125.7). Denser networks form more connections, provided there is a large enough number of neurons in the network. It is expected that this trend will continue as d increases. This response is directly related to the number of synapses required either in a single CNB, or a group of CNBs if some neurons are not used. Thus, for a given N, d affects the amount of connectivity available in the architecture.

The average number of inputs a tile receives is dependant upon d and C as shown in figure 7 (mean: 1634, range: 0-6147). In this figure, N is fixed at 10000 (CNB grid dimensions: 101x100). This result shows that for a given neuron count and clustering parameters, there is a maximum number of inputs to a tile as the size of a tile, C, is varied. When the tile size is small, the input count is low as there are not many CNBs to connect to. When the tile size is large, larger than the typical range of the clusters of a neuron, fewer connection must be routed over the network. The largest

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

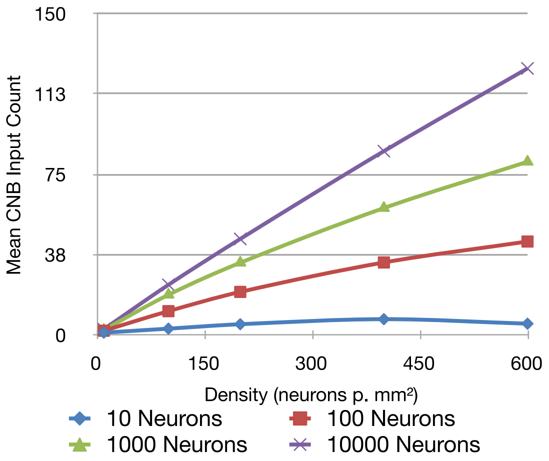

Figure 6. CNB input count by neuron count

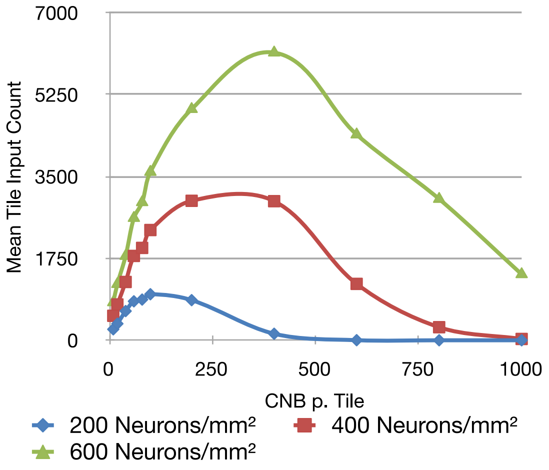

Figure 7. Tile input count by density

number of inputs is seen between these two limits. An important related result is the proportion of total connections that are routed over the AER network, shown in figure 8 (mean: 0.25, range: 0-0.75). This proportion is not strongly affected by d but is closely related to C. As the tiles grow larger, fewer connections are routed over the network.

These trends indicate that, for fixed clustering characteristics and a given N and d, a suitable trade-off can be reached between the amount of direct and indirect routing, and the number of synapses in the CNB.

These results demonstrate trends that are expected to apply to networks with a greater number of neurons and a greater density as well as those contained in the sample. It is clear that by applying timing and area costs to the responses of interconnect requirements for direct connectivity and network connectivity, and to the number of synapses per CNB, these trends can be used to derive an optimal mapping of the neural network.

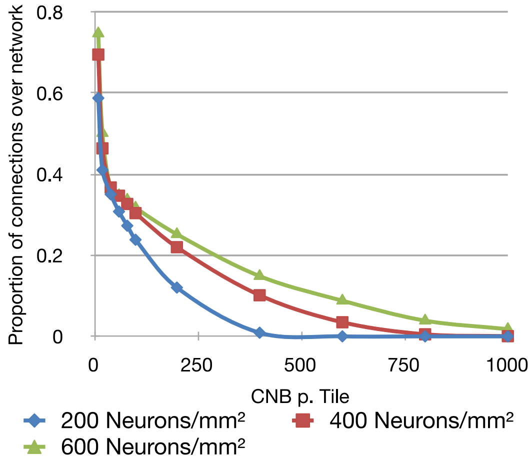

Figure 8. Proportion of total connections routed over the AER network, by density

Further, generalised predictions of traffic volume will enhance the quality of the mapping. Modifying the tool chain to accommodate back-propagated traffic data from simulation can provide a significantly more accurate prediction of traffic flow and can be used to guide placement of neurons to balance traffic load on links to optimise bandwidth usage for a neural network application. The traffic data can also be combined with layout data and used to analyse the power consumption of the network, for example to adjust the layout to reduce the effect of hot spots, or switch off unused links.

The range of parameters used to generate the sample of mapped networks did not reach the levels seen in networks in the brain (see section 2.1), but was chosen to show important trends and produce the sample in a reasonable amount of computing time. Future work will include improving the efficiency of the tool chain, adding back-propagation of simulation data, and implementing routing, power and timing cost minimisation. To improve the accuracy of costings, the system elements will be modelled at the gate level and assessed for area and speed. This will also improve the routing stage of the mapping by providing accurate distances to and from network bridges.

5.2. VLSI Neuron

A leaky integrate and fire model of a neuron has been designed using standard cells, simulated, and manufactured at 130nm using the Europractice mini@sic service. The

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

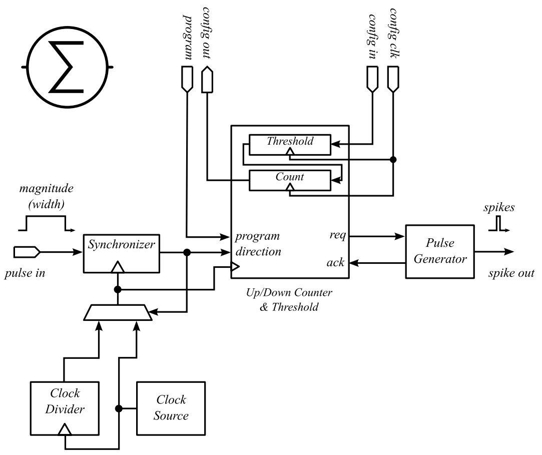

Figure 9. Leaky integrate & fire model

Area	$1145.6\mu m^2$ (90nm: $700\mu m^2$)
Gates	390
Density	$873\,p.\,mm^2$ (90nm: $1429\,p.\,mm^2$)
Spike Period	$4.5ns$
Generated clock frequency	$160MHz$
Max. Spike Rate (threshold=100)	2.35 million p. second

Table 2. 130nm LI&F neuron statistics

schematic is shown in figure 9, and table 2 has some area and timing statistics. The neuron is self-contained and is asynchronous relative to the rest of the system, including other neurons. At the heart of the neuron is a 7-bit up/down counter, driven by a generated clock (as some measure of time is required). A clock divider provides a slower clock that is used for decay. The direction of the counter is controlled by the input pulse which also chooses between the original clock and the divided form, producing a quick increment when an input pulse is present and a slow decay when it is not.

The resolution of the counter, and the range of threshold values, is derived from real data on post-synaptic potentials and neural thresholds. For this model, a typical threshold value would be about 100, and each incoming pulse would add between 1 and 3 to the count value.

When the counter reaches the configured threshold value, the counter halts and requests that a spike be emitted by the pulse generator. The asynchronous pulse generator, synthesized using the Petrify tool [5], produces a single stereotypical spike in response to a request. Once the spike is emitted, the counter resets and the neuron starts the cycle again. The width of the pulse is determined by the delay element; the size of this element ($3.4ns$) was chosen to safely produce the shortest possible pulse while still being visible through the IO.

The neuron model is simple, but it will enable a network to show interesting behaviour and demonstrate the interconnect architecture. The neuron occupies a small area such that enough neurons will fit on even a small mini@sic chip, when the anticipated size of the other system elements is accounted for, such that a network built with these should be able to produce an interesting behaviour.

The reconfigurable architecture achieves a focus on connectivity by using simple models of neurons and synapses. By using simple models, the implementation of the network is made more straightforward, and by grouping them into CNBs, the grid is formed through simple repetition of identical units. There is currently no support for axonal delay which is important to spiking networks a mechanism for neural oscillation and synchrony[6], which may be important to interpretation of sensory input and as a potential solution to the binding problem (how different neuronal interpretations of stimuli are united). A simple mechanism for intentionally delaying spikes will be added to the architecture as part of future work. The representation of post-synaptic magnitudes is intended for short range communication (intra-CNB, or single-hop inter-CNB) only, but makes the size of the CNB very sensitive to the encoding used. Future work will also include examining an optimal representation of magnitude as part of an implementation of synaptic plasticity and a critical analysis of the level of abstraction of the model.

6. Conclusions

A connection-centric reconfigurable hardware architecture for spiking neural networks was presented. The architecture is a hybrid of classic on-chip interconnect and a Network-on-Chip, alleviating the connectivity scaling problem in spiking neural networks. A method of generating and then mapping a neural network to the architecture was presented and discussed. The results of this mapping indicate trends that can be used to minimise the cost of the mapping, optimising for time and area and also power if simulated traffic data is back-propagated. A simple neuron model in VLSI was also presented.

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.

References

[1] C. Beaulieu, Z. Kisvarday, P. Somogyi, M. Cynader, and A. Cowey. Quantitative distribution of gaba-immunopositive and-immunonegative neurons and synapses in the monkey striate cortex (area 17). *Cerebral Cortex*, 2(4):295–309, 1992.

[2] T. Binzegger, R. J. Douglas, and K. A. C. Martin. A Quantitative Map of the Circuit of Cat Primary Visual Cortex. *J. Neurosci.*, 24(39):8441–8453, 2004.

[3] T. Binzegger, R. J. Douglas, and K. A. C. Martin. Stereotypical Bouton Clustering of Individual Neurons in Cat Primary Visual Cortex. *J. Neurosci.*, 27(45):12242–12254, 2007.

[4] A. Burkitt. A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input. *Biological Cybernetics*, 95(1):1–19, 2006.

[5] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavango, and A. Yakovlev. Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. In *XI Conference on Design of Integrated Circuits and Systems*, 1996.

[6] S. M. Crook, G. B. Ermentrout, M. C. Vanier, and J. M. Bower. The role of axonal delay in the synchronization of networks of coupled cortical oscillators. *Journal of Computational Neuroscience*, 4(2):161–172, 04 1997.

[7] W. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. *Design Automation Conference, 2001. Proceedings*, pages 684–689, 2001.

[8] W. Gerstner and W. M. Kistler. *Spiking neuron models : single neurons, populations, plasticity*. Cambridge University Press, 2002.

[9] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

[10] C. Holmgren, T. Harkany, B. Svennenfors, and Y. Zilberter. Pyramidal cell communication within local networks in layer 2/3 of rat neocortex. *J Physiol*, 551(1):139–153, 2003.

[11] L. Maguire, T. McGinnity, B. Glackin, A. Ghani, A. Belatreche, and J. Harkin. Challenges for large-scale implementations of spiking neural networks on FPGAs. *Neurocomputing*, 71(1-3):13–29, 2007.

[12] M. Mahowald. *An analog VLSI system for stereoscopic vision*. Kluwer Academic Publishers, 1994.

[13] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.

[14] A. Rast, S. Yang, M. Khan, and S. Furber. Virtual synaptic interconnect using an asynchronous network-on-chip. *Neural Networks, (IJCNN) 2008. IEEE International Joint Conference on*, pages 2727–2734, June 2008.

[15] P. STERLING and M. FREED. How robust is a neural circuit? *Visual Neuroscience*, 24(04):563–571, 2007.

[16] `http://www.ini.unizh.ch/~amw/scx/scx.html` (checked: 27/01/09). Silicon Cortex (SCX) Project Page, Institute of Neuroinformatics, Zurich.

[17] `http://www.stanford.edu/group/brainsinsilicon` (checked: 27/01/2009). Extended Address Event Representation Draft Standard v0.4.

[18] I.-T. R. X.200. Information technology - open systems interconnection - basic reference model: The basic model, 1994.

[19] L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and M.-m. Poo. A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395(6697):37–44, 1998.

Authorized licensed use limited to: Newcastle University. Downloaded on May 24,2010 at 12:24:12 UTC from IEEE Xplore. Restrictions apply.