On-Chip and Inter-Chip Networks for Modelling Large-Scale Neural Systems

Steve Furber and Steve Temple
School of Computer Science, The University of Manchester
Oxford Road, Manchester M13 9PL, UK
Email: steve.furber@manchester.ac.uk;
temples@cs.man.ac.uk

Andrew Brown
Department of Electronics and Computer Science.
The University of Southampton,
Southampton, Hampshire SO17 1BJ, UK
adb@ecs.soton.ac.uk

Abstract—**The real-time modeling of large systems of spiking neurons is computationally very demanding in terms of processing power, synaptic weight memory requirements and communication throughput. We propose to build a high-performance computer for this purpose with a multicast communications infrastructure inspired by neurobiology. The core component will be a chip multiprocessor incorporating some tens of small embedded processors, interconnected by an NoC that carries spike events between processors on the same or different chips. The design emphasizes modeling flexibility, power-efficiency, and fault-tolerance, and is intended to yield a general-purpose platform for the real-time simulation of large-scale spiking neural systems.**

I. Introduction

The human brain remains as one of the great frontiers of science – how does this organ upon which we all depend so critically actually do its job? A great deal is known about the underlying technology – the neuron – and we can observe large-scale brain activity through techniques such as magnetic resonance imaging, but this knowledge barely starts to tell us how the brain works. Something is happening at the intermediate levels of processing that we have yet to begin to understand, but the essence of the brain's information processing function probably lies in these intermediate levels. To get at these middle layers requires that we build models of very large systems of spiking neurons, with structures inspired by the increasingly detailed findings of neuroscience, in order to investigate the emergent behaviours of those systems.

High-performance microprocessors have reached a brick wall in terms of improving single-thread performance. The technology advances that have delivered exponential performance gains over the last three decades will not deliver the same gains in the future, and a new approach is required. Industry giants, including Intel, are all agreed that the continuing increases in chip transistor count can no longer be turned into making one processor faster, but should instead be turned into putting more processors onto a chip. This delivers more total performance, but through parallelism, not single-thread performance.

The advent of chip multiprocessors (CMPs) raises an interesting question: should we maximize the performance of a single processor and then fit as many as we can on a chip, or should we employ a simpler processor and fit many more on a chip? Industry is following the former course, but on every measure apart from single-thread performance the latter course is more promising. Simpler processors are far more power-efficient (measured in MIPS/watt) and rather more area-efficient (measured in MIPS/mm^2) than high-end processors.

So, if an application can be broken into an arbitrary number of threads, the system built from larger numbers of simpler processors will win. Modeling large-scale neural networks is one such application. It is a computationally-challenging task due to the very high levels of concurrency and communication inherent in the system. The task is, however, highly parallelizable, and a new machine architecture based on a chip multiprocessor is a promising approach. We have developed such an architecture, based upon a novel multicast intra- and inter-chip communications infrastructure that is itself inspired by the connectivity patterns of the biological systems that we intend to use the system to model. Unlike conventional multiprocessor architectures there is no requirement for inter-processor memory coherency, which greatly simplifies the design of the machine.

II. Related work

A similar system was proposed (but not implemented) at UC Berkeley in the early 1990s. The Connectionist Network Supercomputer (CNS-1) architecture [1] was a design for a massively parallel computer capable of simulating one million neurons in real time. More recently, the MASPINN project at TU Berlin [2] produced a neuro-accelerator board with similar performance. Both of these projects were based on the use of custom VLSI devices. With the advent of very

This work is supported by the EPSRC Advanced Processor Technologies Portfolio Partnership Award. Steve Furber holds a Royal Society-Wolfson Research Merit Award. The support of ARM Ltd and Silistix Ltd in the proposed work is gratefully acknowledged.

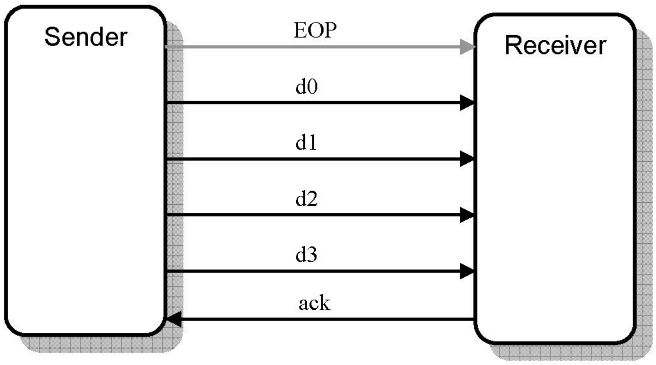

Figure 1. A CHAIN point-to-point self-timed link.

large FPGA devices in the last few years, interest has focused on the use of these new devices, often with external memory to provide weight storage. An example is the RAPTOR2000 system from the University of Paderborn [3].

III. Neuron models

A neuron is a relatively simple device, particularly if one accepts the leaky-integrate-and-fire (LIF) model as a sufficiently good approximation for large-scale systems. It could be argued that a general-purpose processor is therefore overkill for this application. While this may indeed turn out to be true, at this stage it is simply not possible to judge which of the biological details can safely be omitted and which are essential to the information processing functions of the neuron. Is the LIF model good enough? Is a more sophisticated basic model such as that proposed by Izhikevich [1] necessary to support various different dynamical modes such as resonant bursting in addition to leaky integration? On the other hand, would a more abstract neural model suffice for many purposes [5]? Can dendritic trees be modeled as simple summing processes, or are their non-linearities functionally important? Are axonal delays important? How can sparse connectivity be represented? What learning mechanisms are important?

All of these questions can be left open if the neural model is implemented in a fully-programmable medium such as a general-purpose processor or reconfigurable logic, and the former is more flexible than the latter. Once these questions have been answered then it is likely to be possible to optimize the hardware much more closely to its purpose, but it is too early to close down options at this stage. The answers to these questions will dominate the 'packing density' of the neuron-processor mapping.

IV. Scale

Biological brains employ neurons in very large numbers. The honey bee, for example, has 850,000 neurons. The human brain has 10^{11} or so. Each neuron receives inputs from, and connects to, thousands of other neurons. Each connection, or *synapse*, may adapt dynamically in response to local activity, and therefore requires many bits to represent its current state in a digital model.

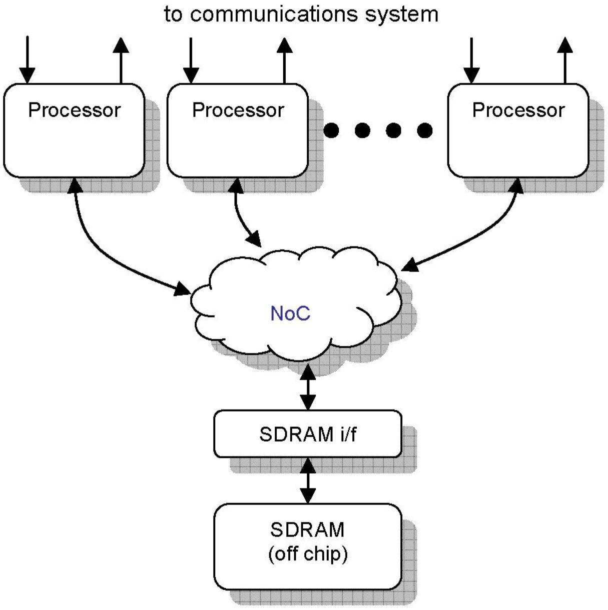

Figure 2. The CMP system NoC.

According to our current estimates a 200 MIPS integer embedded ARM9 processor should be able to model 1,000 LIF neurons, each with 1,000 inputs firing on average at 10 Hz, in real time. The synaptic connectivity information for these neurons requires around 4 Mbytes of memory and the neuron state requires around 100 Kbytes of memory. These figures have caused us to adopt a hybrid architecture where the synaptic data is held in an off-chip SDRAM while the neural state data is held in on-chip memory local to each embedded processor. A processing node in our system therefore comprises two ICs: a chip multiprocessor (CMP) with about twenty 200 MIPS embedded ARM9 processors, and an SDRAM chip. The synaptic data is accessed in large blocks by DMA and this enables an SDRAM bandwidth of around 1 GByte/s to provide this data at the required rate.

The processors on a CMP share access to the SDRAM using a self-timed packet-switched Network-on-Chip (NoC). This fabric will use the CHAIN technology [6] developed at the University of Manchester and commercialized by Silistix Ltd [7]. CHAIN is based upon 6-wire links that encode data using a 1-of-5 return-to-zero (RTZ) code – 4 wires are used to encode 2 bits of data and the 5th to encode end-of-packet (EOP), as illustrated in Fig. 1. The 6th wire carries a return-to-zero acknowledge signal to complete the self-timed handshake. Multiple 6-wire links are deployed in parallel to deliver the throughput required by the application in each part of the fabric, and long interconnects can be pipelined by inserting delay-insensitive repeater stages as required. An unoptimized implementation of CHAIN in a 180nm smart card chip yielded a throughput of 1 Gbit/s per 6-wire link [8]. The organization of the system NoC that connects the processor subsystems to the SDRAM is shown in Fig. 2. Each processor subsystem comprises a processor, instruction and data memory, timers, interrupt and DMA controllers and a communications NoC interface (Fig. 3).

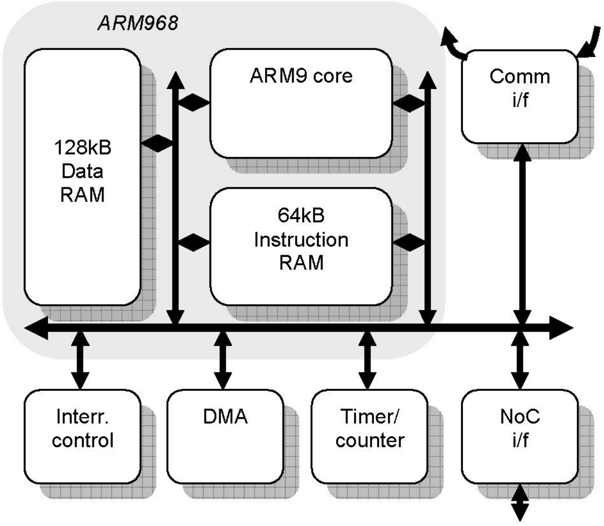

Figure 3. Processor subsystem organization.

V. THE COMMUNICATIONS SYSTEM

The major challenge in designing a scalable multi-chip neural modeling system is to emulate the very high connectivity of the biological system. The high fan-in and fan-out of neurons suggests that an efficient multicast communication system is required. We propose a communication NoC fabric based upon address-event signaling, but carried over a second self-timed packet-switched fabric rather than the usual bus-based fabric. The self-timed fabric decouples the many different clock domains within and across the CMPs.

The CHAIN protocol is efficient for on-chip communication, but we wish to extend the communication system to include inter-chip links. Here the trade-off between simplicity and power-efficiency leads to the choice of a different protocol: self-timed RTZ signaling incurs four chip-to-chip delays per symbol (the rising data transition, the rising acknowledge response, the falling data transition and the falling acknowledge response) whereas a non-return-to-zero (NRZ) protocol incurs only two chip-to-chip delays per symbol. In addition, I/O pins are at a premium, and the energy costs of an off-chip transition are high. We have therefore chosen an 8-wire inter-chip link that employs a self-timed 2-of-7 NRZ code [9] with an NRZ acknowledge. 16 of the 21 possible 2-of-7 codes are used to carry four bits of data, and a 17th code carries EOP. Each 8-wire link has a capacity of around 1 Gbit/s when connecting two CMPs on the same circuit board, and the self-timed protocol guarantees correct operation (albeit at a lower data rate) when the CMPs are on different circuit boards, automatically adapting to the addition delays incurred by any signal buffering that may be required.

The complete communications subsystem on a CMP is illustrated in Fig. 4. The inter-chip links are accessed via input protocol converters ('Rx i/f' in Fig. 4) that translate the off-chip 2-of-7 NRZ codes to the on-chip 1-of-5 RTZ codes, and output protocol converters ('Tx i/f') that perform the inverse translation. Each of the on-chip processing subsystems ('fascicle processor') is also a source of network traffic and a potential destination. All of the on- and off-chip sources are merged through an asynchronous arbiter into a single stream of packets that passes through the multicast router which will, in turn, propagate the packet to a subset of its on- and off-chip outputs. The monitor processor is identical to a fascicle processor but is dedicated to system management functions rather than neural modeling. It is chosen from among the fascicle processors at boot time; the flexibility in its selection removes another possible single point of failure on the CMP, improving fault tolerance.

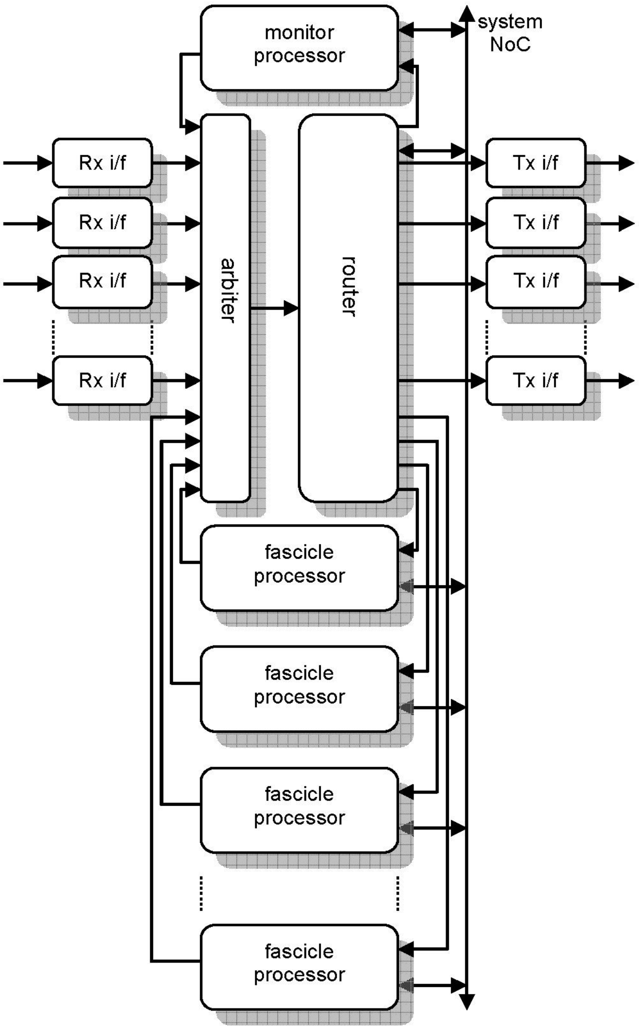

Figure 4. The communications NoC.

The heart of the communication subsystem is the associative multicast router which directs every incoming packet to one or more of the local processors and output links using a routing key based on the source ID and a route look-up table.

VI. System architecture

The full system comprises a 2D array of nodes interconnected through bi-directional links (so each link is 16 wires). The 2D configuration was chosen for convenience. Nothing in the communications architecture precludes the use of more complex topologies, but the 2D mesh is very straightforward to implement on a circuit board and also provides many alternative routes between any pair of nodes which is useful for reconfiguration to isolate faults.

Each node comprises a CMP and an SDRAM chip, giving it the integer processing power of a typical PC but at much lower power and in a compact physical form. The six bidirectional links support a total of 6 Gbit/s of bandwidth into the node and the same out of the node. A system of 100 x 100 nodes will deliver a total of 40 teraIPS, sufficient to simulate perhaps 200 million spiking neurons in real time, and will have a bisection bandwidth of 200 Gbit/s.

VII. Power-efficiency

In any system of the proposed scale power-efficiency must be engineered in from the outset. We have estimated that, based on published figures for the power-efficiency of an ARM968 [9] of 0.12 to 0.23 mW/MHz on a 130 nm CMOS process, a neuron with 1,000 inputs firing at a mean rate of 10 Hz will consume 23 to 36 µW. Of this, 3 to 6 µW is used to compute the neuron dynamics every millisecond, 10 to 20 µW is used to compute the 10,000 inputs received by the neuron every second, and 10 µW is expended in accessing the synaptic weights in the SDRAM. The communication power is negligible, with each spike consuming 1 nJ for each Router and 1 nJ for each inter-chip link it passes through.

Each CMP chip will consume 250 to 500 mW, enabling it to be deployed in low-cost packaging. A large-scale system capable of modeling a billion spiking neurons in real time will require 50,000 nodes and consume 23 to 36 kW.

VIII. Fault-tolerance

Any system designed on this scale must incorporate some level of fault-tolerance. The approach taken here will be based upon a combination of redundancy, reconfigurability, and exploitation of the intrinsic fault-tolerance of biological neural systems.

The machine architecture provides a good starting point – it is highly regular and symmetric. If one processor node on a particular chip fails the chip is still highly functional provided that the failed node does not draw excessive power or interfere with its peers in any way. There will be some single points of failure on a chip, but so long as these represent a small proportion of the silicon area their failure will be relatively rare. A chip failure, if it does occur, will cause that chip to be mapped out of the system, requiring cooperation between the run-time system (which will detect the fault) and the configuration software (which will map around the fault).

Localized faults due, for example, to memory errors, will generally manifest themselves as misbehaviour by an individual neuron at the application level. Biological plasticity will allow adaptation to correct for such failures. A speculative aspect of the design for fault-tolerance is that we will explore structural run-time self-repair along the lines of biological recovery from a stroke.

IX. Conclusions

The design of a high-performance computer for simulating large-scale systems of spiking neurons requires a new approach to on-chip and inter-chip communications. We have proposed such a system, basing the machine on a highly-parallel configuration of small, power-efficient embedded processors. The implementation employs nodes configured in a mesh network where each node comprises a chip multiprocessor and an SDRAM chip.

The communications fabric is based upon a multicast router on each CMP that handles both on-chip and inter-chip traffic in a symmetric way. Self-timed NoCs are used both to implement the links in the communications fabric and to connect the processor nodes to the shared SDRAM interface on the CMPs. The NoC protocol employs a delay-insensitive 1-of-5 RTZ protocol for on-chip links, but inter-chip links employ a delay-insensitive 2-of-7 NRZ protocol. The protocol conversion integrates seamlessly into the communications NoC fabric to enable the NoC to extend across a scalable fault-tolerant multi-chip system with very large numbers of small embedded processors.

References

[1] K. Asanovic, J. Beck, T. Callahan, J. Feldman, B.S. Irissou, B. Kingsbury, P. Kohn, J. Lazzaro, N. Morgan, D. Stoutamire & J. Wawrzynek, *CNS-1 Architecture Specification*, EECS Department, UC Berkeley, Technical Report No. UCB/CSD-93-747, 1993.

[2] T. Schoenauer, N. Mehrtash, A. Jahnke & H. Klar, "MASPINN: Novel Concepts for a NeuroAccelerator for Spiking Neural Networks", *Proc. VIDYNN'98*, Stockholm, June 22-26, 1998.

[3] M. Porrmann, M. Franzmeier, H. Kalte, U. Witkowski & U. Rückert, "A Reconfigurable SOM Hardware Accelerator", *Proc. ESANN'2002 - European Symposium on Artificial Neural Networks*, Bruges, Belgium, 24-26 April 2002, pp. 337-342.

[4] E.M. Izhikevich, "Which Model to Use for Cortical Spiking Neurons?", *IEEE Trans. Neural Networks* **15**, 2004, pp. 1063-1070.

[5] S.B. Furber, W.J. Bainbridge, J.M. Cumpstey and S. Temple, "A Sparse Distributed Memory based upon N-of-M Codes", *Neural Networks* **17**(10), December 2004, pp. 1437-1451.

[6] W.J. Bainbridge and S.B. Furber, "CHAIN: A Delay-Insensitive Chip Area Interconnect", *IEEE Micro*, special issue on the Design and Test of System-on-Chip **22**(5), September/October 2002, pp. 16-23.

[7] http://www.silistix.com

[8] W.J. Bainbridge, L.A. Plana & S.B. Furber, "The Design and Test of a Smartcard Chip Using a CHAIN Self-timed Network-on-Chip", *Proc. DATE'04*, Vol. 3, Paris, Feb 2004, p. 274.

[9] W.J. Bainbridge, W.B. Toms, D.A. Edwards, S.B. Furber, "Delay-Insensitive, Point-to-Point Interconnect using m-of-n codes", *Proc. Async'03*, Vancouver, May 2003, pp. 132-140.

[10] http://www.arm.com/products/CPUs/ARM968E-S.html