Exploiting Device Mismatch in Neuromorphic VLSI Systems to Implement Axonal Delays

Sadique Sheik*, Elisabetta Chicca*†, Giacomo Indiveri*
* Institute of Neuroinformatics
University of Zurich and ETH Zurich, Switzerland
Email: sadique@ini.phys.ethz.ch
† Cognitive Interaction Technology - Center of Excellence
Bielefeld University, Germany

***Abstract*—Axonal delays are used in neural computation to implement faithful models of biological neural systems, and in spiking neural networks models to solve computationally demanding tasks. While there is an increasing number of software simulations of spiking neural networks that make use of axonal delays, only a small fraction of currently existing hardware neuromorphic systems supports them. In this paper we demonstrate a strategy to implement temporal delays in hardware spiking neural networks distributed across multiple Very Large Scale Integration (VLSI) chips. This is achieved by exploiting the inherent device mismatch present in the analog circuits that implement silicon neurons and synapses inside the chips, and the digital communication infrastructure used to configure the network topology and transmit the spikes across chips. We present an example of a recurrent VLSI spiking neural network that employs axonal delays and demonstrate how the proposed strategy efficiently implements them in hardware.**

I. Introduction

Spiking neural networks are typically characterized by their network topology (e.g. multi-layer, feed-forward, recurrent, etc.) and by their distributions of synaptic weights, while they seldom make use of temporal delays to carry out information processing tasks. However, temporal delays can provide an extra degree of complexity for solving computationally demanding problems, and can be used to implement faithful models of real neural networks, as they account for the spike propagation delays that takes place along the neuron's axon. Indeed, axonal delays are often modeled to describe the temporal dynamics of biologically realistic spiking neural networks [1]–[3]. For example, it has been shown that transmission/conductance delays help enhance neural synchrony [4] and that axonal delays provide the anatomical and physiological basis for a neuronal map of inter-aural time differences in the nucleus laminaris of barn owls [5].

While there have been sporadic attempts at implementing axonal delays in hardware spiking neural networks [6]–[9], most VLSI neuromorphic setups do not support them, either at the single VLSI device level, or in *multi-chip setups*. Recent developments in the construction of VLSI spiking neural networks focus increasingly more on distributed, multi-chip setups [10]–[12]. These setups typically consist of several multi-neuron chips comprising hybrid analog/digital neuromorphic circuits, interfaced among each other using asynchronous event based digital communication modules. A common communication protocol used in these setups is the Address Event Representation (AER) [13], [14]. In this representation spikes (address events produced by neurons) are routed from one chip to the other using a specified addressing schemes via custom digital boards, typically comprising one or more Field Programmable Gate Arrays (FPGAs) [15], [16]. In principle, one could therefore exploit the digital domain used in the event-based communication across chips to emulate axonal delays, but this is not an optimal solution, as it requires additional dedicated hardware overhead. For example, in [17] axonal delays are implemented by accumulating address events in pulse packets, time-stamping them, and transmitting them to a dedicated digital network chip. Here the events are held, sorted, and buffered until a target delay is reached (after which they are sent to their target destination). While this approach is flexible and accurate, it requires specialized hardware for the computationally intensive real-time event sorting, and looses the efficient representation of time in the AER, where events are transmitted as they happen, and time represents itself.

An alternative approach that does not require to explicitly time-stamp each event and that can reduce these overhead costs, is to exploit both the digital domain used for the inter-chip communication and the analog one used with the silicon neurons and synapses inside the chips [9]. In this paper we follow this approach by making use of inherent device mismatch present in the analog neuromorphic circuits to implement axonal delays, and exploit the AER communication digital infrastructure to (re)configure the placement of these delays in the neural network.

Device mismatch in neuromorphic multi-neuron chips produces inhomogeneities in the response of the synapses and neurons present in the chip. An example of this effect is evident in Fig. 1, where we show a raster plot of spiking activity measured from a neuromorphic chip comprising 128 putatively identical silicon neurons [18]. In this example the neurons are stimulated with constant current injection, set by a common global bias. Ideally, all neurons should have the same firing rates, but given that the neuron circuits are analog and that the transistors operate in the weak-inversion regime [19], their response properties vary substantially. Device mismatch effects in these chips also affect several other neural network

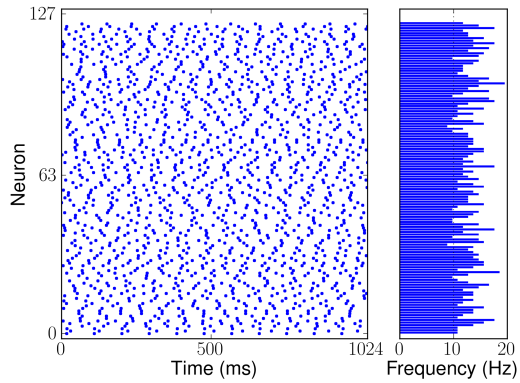

Fig. 1. Raster plot of spiking activity and output firing rate measured from a chip comprising 128 neurons stimulated by the same constant input current.

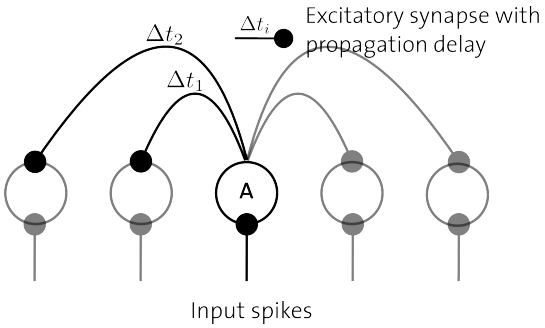

Fig. 2. A recurrent neural network with axonal delays. Spiking input is sent to the population of neurons from the bottom. The neurons are recurrently connected with excitatory synapses. The synaptic projections have transmission delays that vary with distance between the source and destination. Projections from only one of the neurons in the population are shown here, with their corresponding transmission delays.

properties, such as synaptic weights and time constants.

Device mismatch can be minimized using standard electrical engineering approaches and appropriate analog VLSI design techniques. But this leads to very large transistor sizes and large layout designs, which can significantly reduce the number of neurons and synapses that can be integrated onto a single chip. Rather than attempting to reduce mismatch using brute-force engineering approaches, neuromorphic approaches should try to exploit the adaptation mechanisms and learning strategies that they seek to model and implement in hardware. For example this has already been a successful strategy in neuromorphic vision sensors that employ adaptation (adaptive photoreceptor circuits), in each individual pixel rather than the single/global auto-gain mechanisms used in standard imagers [20]–[23]. Learning and plasticity are also very effective mechanisms for compensating the effects of device mismatch [24], [25], or homeostatic mechanisms [26]. Hardware neural networks can also employ population coding approaches and make use of redundancy, exploiting the large number of parallel elements present in these devices [27]. The use of these strategies would allow designers to implement large arrays of compact, redundant, possibly plastic synapses, that can carry out robust computation even if they are affected by mismatch. And mismatch can then be used as a feature, rather then being something to try to minimize.

In the next Section we show how we used mismatch to produce a range of variable response properties that can be exploited for efficiently modeling axonal delays.

II. Materials and Methods

Our methodology can be used to realize arbitrary connectivity patterns with axonal delays. To demonstrate our approach we chose a recurrent network architecture of the type shown in Fig. 2.

A. Network model

The network consists of a population A of 32 recurrently connected leaky Integrate and fire (I&F) neurons receiving spikes as input, and arranged as shown in Fig. 2. Each neuron projects to its nearest neighbors with excitatory connections, and each projection incurs a propagation time delay Δt_i proportional to the connection distance i. Equation (1) describes this relationship:

$$\Delta t_i = \Delta T + i/v \tag{1}$$

where v represents the propagation velocity and ΔT is the minimum possible propagation delay.

B. Hardware setup

The hardware setup used to implement the network described in Fig. 2 is outlined in Fig. 3. It consists of two multi-neuron chips connected in daisy-chain to an AER mapper [16]. A workstation is used to inject the input spikes in the network and log the network's output activity. The first multi-neuron chip (CHIP-1) houses 128 leaky I&F neurons, equipped with excitatory, inhibitory, and plastic synapses [18]. The second multi-neuron chip (CHIP-2) houses 2048 leaky I&F neurons, equipped with excitatory and inhibitory synapses. Both chips send and receive spikes using the AER. The chips were fabricated using a standard $0.35\,\mu m$ CMOS technology and occupy an area of $10\,mm^2$ and $15\,mm^2$ respectively. The AER mapper is a custom digital FPGA board that can route spikes from source neuron to the destination synapse, with a latency of $0.8\,\mu s$ and supports $66\,MHz$ peak event rates [16].

While this setup can efficiently support the implementation of fairly large networks of hardware neurons, there is no explicit mechanism dedicated to the implementation of axonal delays. In order to implement the network or Fig. 2 with the appropriate delays we have to resort to a second a population of neurons, and use them as intermediate delay elements. We call these neurons *delay neurons*.

C. Delay neurons

A common way of implementing temporal delays in electrical engineering is by using low-pass filters. We follow the same approach and use the low-pass filtering properties of the synapse circuits [28] present on the multi-chip labeled CHIP-2. We configure the synapse parameters such that the integration

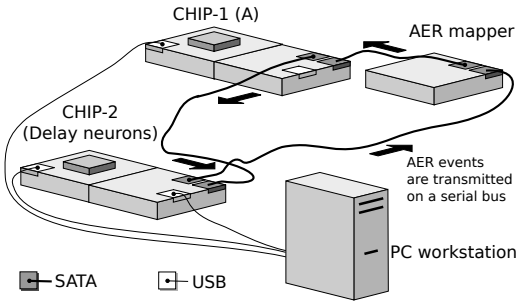

Fig. 3. The multi-chip setup used in this work. Two spiking multi-neuron chips transmit AER spikes to each other via a digital mapper board. A PC workstation is used to generate the input stimuli and log the network output spikes.

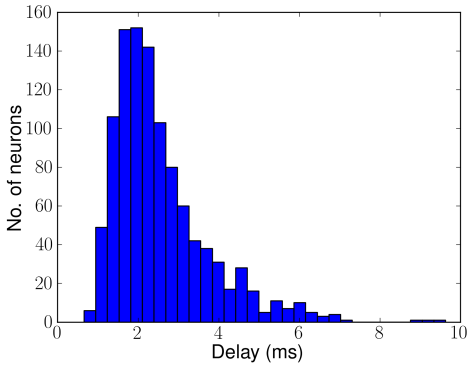

Fig. 4. Distribution of delays measured from the multi-neuron chip labeled CHIP-2.

of a single pulse (the input spike) produces an output spike, after a set delay Δt. The neurons connected to these synapses that have this behavior are the *delay neurons*.

Figure 4 shows the delays measured for all the neurons on CHIP-2. These delays depend on the synapse circuit time constant, on the synaptic weight, as well as the neuron's membrane time constant and firing threshold. These parameters shared among all delay neurons in the chip and ideally should produce a single common delay. As can be seen from the histogram, this is far from ideal: the neurons exhibit a broad range of delays, due to device mismatch. The distribution of delays can be modified by changing one or more of the four parameters mentioned above. For the set of biases used for this measurement, only a portion of the delay neurons produce usable delays. The rest of the neurons either have too strong or too weak synaptic efficacy leading to multiple or no spikes at their output.

D. Network Implementation

The population of neurons (population A in Fig. 2) was modeled using 32 silicon neurons of CHIP-1. The recurrent connectivity was implemented via the AER mapper and transmission delays were obtained by placing a delay neuron from CHIP-2 between each projection of the CHIP-1 I&F neurons. The routing of the address events is as follows:

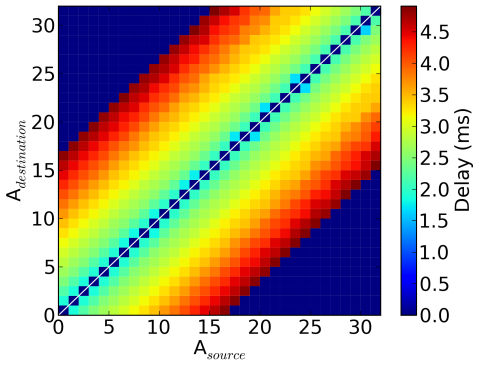

Fig. 5. Measured transmission delays in the hardware neural network. The delays increase linearly with distance between the neurons. A maximum distance of 16 was imposed on the network connectivity. Delays with the minimum value of 0 in this figure represents no connection.

$(A_i \rightarrow Delayneuron(\Delta t_{ij}) \rightarrow A_j)$, thereby producing the desired transmission delays. The choice of the desired transmission delay is done by indexing the appropriate delay neuron, chosen from the histogram of Fig. 4. Every projection therefore passes through a dedicated delay neuron.

III. Results

We selected the appropriate set of delay neurons from CHIP-2 to implement the desired axonal delays in the network. We stimulated the neurons of the network with input spikes and measured the time they took to produce an output spike. The delays measured from the network are shown in Fig. 5. The axonal delay increases linearly with increasing distance between the source and destination neurons, as expected.

The strategy of choosing the delay neurons by programming the AER mapper appropriately, in order to set the desired axonal delays in the network can be used to implement arbitrary delay profiles. As an example, Fig. 6 shows the measurements from a recurrent network analogous to that of Fig. 2, but with random axonal delays on its recurrent connections.

In the networks described above we chose axonal delays comparable to the time constants of the delay neurons. Axonal delays longer than the typical time constant of the delay neurons can be achieved by stacking several delay neurons in a sequence. Specifically, by stacking N delay neurons together, it is possible to implement N different axonal delays with lower and upper bounds defined as:

$$min(\Delta t_i) \; \forall \; i \in (1 \ldots N) \leq \Delta t_i \leq \sum_{j=1}^{N} \Delta t_j \qquad (2)$$

where Δt_i is the axonal delay of the ith delay neuron.

This strategy is used to implement time delays that range up to $125\,ms$ (rather than the $8\,ms$ of the previous example), in the network of Fig. 2. Figure 6 shows the expected measured delays in this condition. The approach of stacking in sequence

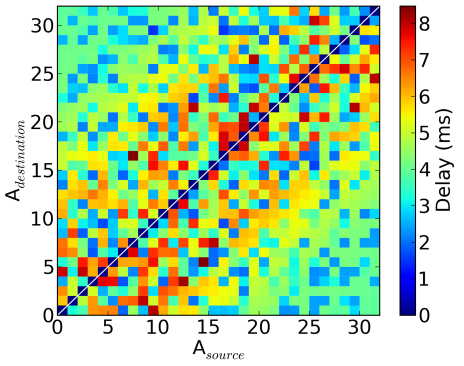

Fig. 6. The figure shows the resultant transmission delays for a connectivity as implemented and measured on the hardware setup. The transmission delays between the neurons are randomly chosen from the available pool of delay neurons.

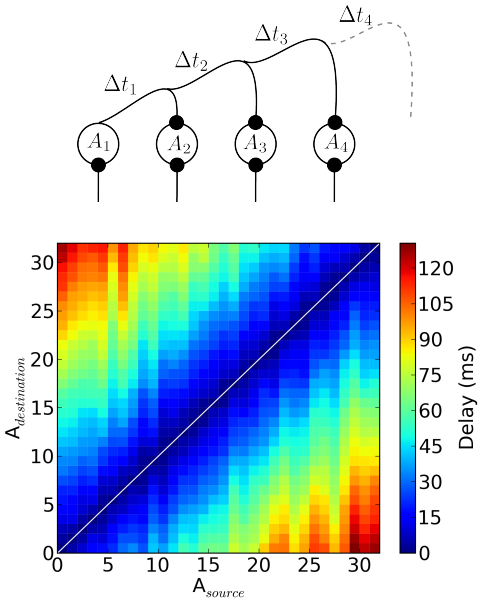

Fig. 7. Silicon neurons are stacked among each other to generate long delays (top): the first neuron A_1 projects to A_2 neuron with a delay of Δt_1 via a delay neuron. This delay neuron also projects to A_3 with a delay Δt_2 making the effective delay from A_1 to A_3 equal to $\Delta t_1 + \Delta t_2$. This process is repeated for every projection to gain incremental delays. Measurements of transmission delays from a hardware recurrent network that uses such a connectivity is shown on the bottom.

delay neurons would allows the implementation of arbitrary delays without necessarily requiring mismatch in the circuits.

By choosing the right set of delay neurons, one can implement a wide range of neural network architectures with arbitrary axonal delay profiles, provided the availability of a large enough pool of inhomogeneous silicon neurons in an AER VLSI setup.

A. Limitations

There is a limitation to the approach of using delay neurons for generating axonal delays: as this approach relies on the integration time of the delay neuron input synapse (used as a low-pass filter), there is an upper-bound on the maximum input firing rate. If a spikes arrives at the delay neuron's input synapse before the delay neuron finished processing the first spike (i.e. before the delay neuron produces an output spike), the effective transmission delay is disrupted. In Fig. 8 we plot the measured the dependence of effective delays on the input Inter-Spike Intervals (ISIs). The effective delay decreases with decreasing ISI (increasing firing rate). This happens because of accumulation of residual synaptic currents after the spike generation of a delay neuron. Coincidentally, a similar relationship between transmission delays and firing rates was observed in physiological recordings of voluntary discharge properties of extensor motor units in humans [29]. Fast spiking neurons were reported to exhibit smaller axonal delays and slower ones longer.

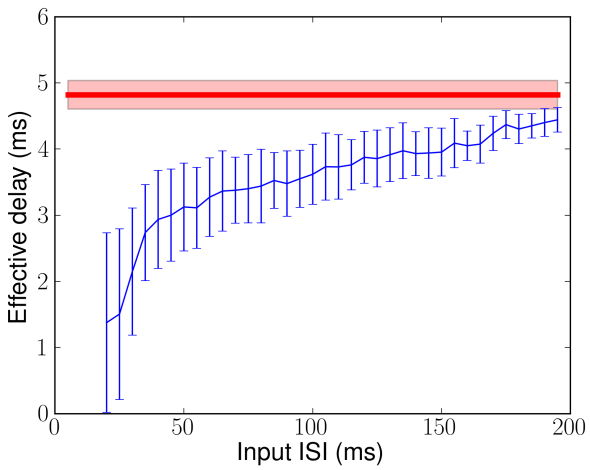

Fig. 8. Dependence of effective delay on the inter-spike-interval. With decreasing ISIs the effective delay of a delay neuron decreases. The top line shows the neuron's asymptotic delay value ($4.82 \pm 0.216\,ms$), i.e. the delay at very long ISIs.

Therefore we argue that this technique of generating delays is appropriate in biologically plausible conditions. But, with the constraint that the pre-synaptic ISI has to be greater than the set delay. This limitation can be overcome by stacking multiple delay neurons, each having a delay shorter than the minimum input ISI, also allowing the generation of longer propagation delays.

Another limitation is the variability of the delays, due to the noise present in the CMOS circuits and in the AER infrastructure. Figure 9 shows the delays generated by neurons on CHIP-2, sorted by delay value. The error bars show the standard deviation of generated delays over 20 trial measurements and is approximately 10% on average. This variability is very hard to overcome but is compatible with variability observed in biological systems.

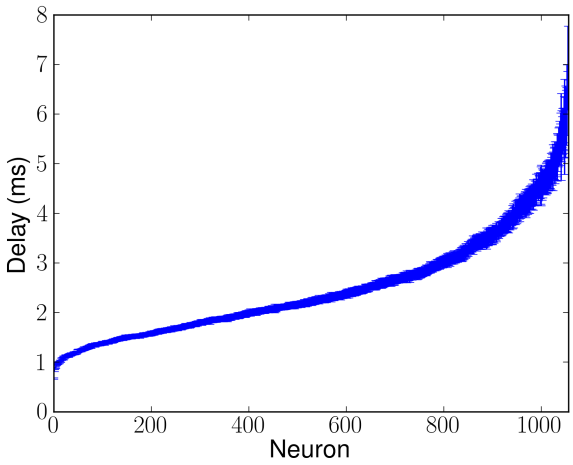

Fig. 9. The distribution of delays and variance across 1070 neurons on CHIP–2. Ordered according to the delays. The error bars show the variance in delay over 20 measurements. Longer delays are produced by smaller pre-synaptic current which are more sensitive to noise. This produces a higher variance for long delays.

IV. Conclusions

We implemented propagation delays using a population of silicon neurons whose time constants were comparable to the desired time delay. We were able to select delay neurons with different time constants by exploiting the mismatch effect in their analog circuit implementations.

Our methodology allows to implement arbitrary recurrently connected networks of I&F neurons with axonal delays. The results described in this paper show that this approach is suitable for implementing architectures used to demonstrate polychronization by Izhykevich [30].

This is a promising approach that can allow the construction of complex multi-chip neural processing systems, and is currently being used to implement a hardware model of an auditory processing system that can learn spectro-temporal correlations in its input stimuli [10].

Acknowledgment

This work was supported by the European FP7 grant "SCANDLE" (231168), the EU ERC Grant "neuroP" (257219), and the Cluster of Excellence 277 (CITEC, Bielefeld University). The authors would like to thank the NCS group (http://ncs.ethz.ch/) for contributing to the development of the AER and multi-chip experimental setups and Dr. Martin Coath for helpful discussions.

References

[1] G. B. Ermentrout and N. Kopell, "Fine structure of neural spiking and synchronization in the presence of conduction delays." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 3, pp. 1259–1264, Feb. 1998. [Online]. Available: http://www.pnas.org/content/95/3/1259.full.abstract

[2] E. M. Izhikevich, J. A. Gally, and G. M. Edelman, "Spike-timing Dynamics of Neuronal Groups," *Cereb. Cortex*, vol. 14, no. 8, pp. 933–944, Aug. 2004. [Online]. Available: http://dx.doi.org/10.1093/cercor/bhh053

[3] M. Rubinov, O. Sporns, J.-P. Thivierge, and M. Breakspear, "Neurobiologically Realistic Determinants of Self-Organized Criticality in Networks of Spiking Neurons," *PLoS Comput Biol*, vol. 7, no. 6, pp. e1 002 038+, June 2011. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1002038

[4] M. Dhamala, V. K. Jirsa, and M. Ding, "Enhancement of neural synchrony by time delay," *Phys. Rev. Lett.*, vol. 92, p. 074104, Feb 2004. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.92.074104

[5] C. E. Carr and M. Konishi, "Axonal delay lines for time measurement in the owl's brainstem," *Proceedings of the National Academy of Sciences*, vol. 85, no. 21, pp. 8311–8315, Nov. 1988. [Online]. Available: http://www.pnas.org/content/85/21/8311.abstract

[6] J. Elias, "Artificial dendritic trees," *Neural Computation*, vol. 5, pp. 648–664, 1993.

[7] D. Northmore and J. Elias, "Building silicon nervous systems with dendritic tree neuromorphs," in *Pulsed Neural Networks*, W. Maass and C. Bishop, Eds. MIT Press, 1998, ch. 5, pp. 135–156.

[8] Y. Wang and S.-C. Liu, "Multilayer processing of spatiotemporal spike patterns in a neuron with active dendrites," *Neural Computation*, vol. 8, pp. 2086–2112, 2010.

[9] R. Wang, C. T. Jin, A. McEwan, and A. van Schaik, "A programmable axonal propagation delay circuit for time-delay spiking neural networks," in *International Symposium on Circuits and Systems (ISCAS 2011), May 15-19 2011, Rio de Janeiro, Brazil*. IEEE, 2011, pp. 869–872.

[10] S. Sheik, M. Coath, G. Indiveri, S. Denham, T. Wennekers, and E. Chicca, "Emergent auditory feature tuning in a real-time neuromorphic vlsi system," *Frontiers in Neuroscience*, vol. 6, no. 17, 2012. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Sheik_etal12.pdf

[11] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, H. Kolle Riis, T. Delbruck, S.-C. Liu, S. Zahnd, A. Whatley, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civit, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "AER building blocks for multi-layer multi-chip neuromorphic vision systems," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press, Dec 2005.

[12] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. Olsen, "Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools," *Journal of Neuroscience*, vol. 27, no. 44, p. 11807, 2007.

[13] S. Deiss, T. Delbruck, R. Douglas, M. Fischer, M. Mahowald, T. Matthews, and A. Whatley, "Address-event asynchronous local broadcast protocol," World Wide Web page, 1994, http://www.ini.uzh.ch/~amw/scx/aeprotocol.html.

[14] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Transactions on Circuits and Systems II*, vol. 47, no. 5, pp. 416–34, 2000.

[15] ——, "A burst-mode word-serial address-event link – I: Transmitter design," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 7, pp. 1269–80, 2004.

[16] D. Fasnacht and G. Indiveri, "A PCI based high-fanout AER mapper with 2 GiB RAM look-up table, 0.8 μs latency and 66 mhz output event-rate," in *Conference on Information Sciences and Systems, CISS 2011*, Johns Hopkins University, March 2011, pp. 1–6. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Fasnacht_Indiveri11.pdf

[17] S. Scholze, S. Schiefer, J. Hartmann, C. Mayr, S. Höppner, H. Eisenreich, S. Henker, B. Vogginger, and R. Schüffny, "VLSI implementation of a 2.8 gevent/s packet based AER interface with routing and event sorting functionality," *Frontiers in Neuroscience*, vol. 5, 2011.

[18] G. Indiveri and E. Chicca, "A VLSI neuromorphic device for implementing spike-based neural networks," in *Neural Nets WIRN11 - Proceedings of the 21st Italian Workshop on Neural Nets*, Jun 2011, pp. 305–316. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Indiveri_Chicca12.pdf

[19] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI:Circuits and Principles*. MIT Press, 2002. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Liu_etal02b.pdf

[20] T. Delbruck and C. Mead, "Analog VLSI phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits," in *Vision Chips: Implementing vision algorithms with analog VLSI circuits*,

C. Koch and H. Li, Eds. Los Alamitos, CA: IEEE Computer Society Press, 1995, pp. 139–161.
[21] G. Indiveri, "Neuromorphic analog VLSI sensor for visual tracking: Circuits and application examples," *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 11, pp. 1337–1347, 1999. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Indiveri99.pdf
[22] R. Harrison and C. Koch, "A robust analog VLSI motion sensor based on the visual system of the fly," *Autonomous Robots*, vol. 7, pp. 211–224, 1999.
[23] S.-C. Liu, "Silicon retina with adaptive filtering properties," *Analog Integrated Circuits and Signal Processing*, vol. 18, no. 2/3, pp. 243–254, February 1999.
[24] K. Cameron and A. Murray, "Minimizing the effect of process mismatch in a neuromorphic system using spike-timing-dependent adaptation," *Neural Networks, IEEE Transactions on*, vol. 19, no. 5, pp. 899–913, May 2008.
[25] "Silicon synapses self-correct for both mismatch and design inhomogenities."
[26] C. Bartolozzi and G. Indiveri, "Global scaling of synaptic efficacy: Homeostasis in silicon synapses," *Neurocomputing*, vol. 72, no. 4–6, pp. 726–731, Jan 2009. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Bartolozzi_Indiveri09.pdf
[27] E. Neftci and G. Indiveri, "A device mismatch compensation method for VLSI spiking neural networks," in *Biomedical Circuits and Systems Conference BIOCAS 2010*. IEEE, 2010, pp. 262–265. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Neftci_Indiveri10.pdf
[28] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Computation*, vol. 19, no. 10, pp. 2581–2603, Oct 2007. [Online]. Available: http://ncs.ethz.ch/pubs/pdf/Bartolozzi_Indiveri07b.pdf
[29] J. Borg, L. Grimby, and J. Hannerz, "Axonal conduction velocity and voluntary discharge properties of individual short toe extensor motor units in man." *The Journal of Physiology*, vol. 277, no. 1, pp. 143–152, 1978. [Online]. Available: http://jp.physoc.org/content/277/1/143.abstract
[30] E. M. Izhikevich, "Polychronization: Computation with spikes," *Neural Comput.*, vol. 18, pp. 245–282, February 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1117652.1117653