A Multichip Pulse-Based Neuromorphic Infrastructure and Its Application to a Model of Orientation Selectivity

Elisabetta Chicca, *Member, IEEE*, Adrian M. Whatley, Patrick Lichtsteiner, Vittorio Dante, Tobias Delbruck, *Senior Member, IEEE*, Paolo Del Giudice, Rodney J. Douglas, and Giacomo Indiveri, *Senior Member, IEEE*

Abstract—The growing interest in pulse-mode processing by neural networks is encouraging the development of hardware implementations of massively parallel networks of integrate-and-fire neurons distributed over multiple chips. Address-event representation (AER) has long been considered a convenient transmission protocol for spike based neuromorphic devices. One missing, long-needed feature of AER-based systems is the ability to acquire data from complex neuromorphic systems and to stimulate them using suitable data. We have implemented a general-purpose solution in the form of a peripheral component interconnect (PCI) board (the PCI-AER board) supported by software. We describe the main characteristics of the PCI-AER board, and of the related supporting software. To show the functionality of the PCI-AER infrastructure we demonstrate a reconfigurable multichip neuromorphic system for feature selectivity which models orientation tuning properties of cortical neurons.

Index Terms—Address event representation (AER), asynchronous, cooperative–competitive, neural chips, neural networks, neuromorphic, orientation tuning, peripheral component interconnect (PCI)-AER, VLSI, winner take all (WTA).

I. INTRODUCTION

NETWORKS of integrate-and-fire (I&F) neurons have been shown to exhibit a wide range of useful computational properties, including feature binding, segmentation, pattern recognition, onset detection, input prediction, etc. [1]. Implementing these functionality in VLSI circuits could lead to the construction of efficient hardware systems capable of solving complex sensory processing tasks in real-time. I&F neuron circuits are very well suited for VLSI implementation [2]–[8]. Large VLSI networks of I&F neurons can already be implemented on single chips, using today's technology. However implementations of pulse-based neural networks on multichip systems offer greater computational power and higher flexibility than single-chip systems. As inter-chip connectivity is limited by the small number of input-output connections available with standard chip packaging technologies, it is necessary to adopt time-multiplexing schemes for constructing large multichip networks.

A. Address Event Representation (AER)

In recent years, we have witnessed the emergence of new asynchronous communication protocols that allow aVLSI neurons to transmit their activity across chips using pulse-frequency modulated signals (in the form of events, or *spikes*). One of the most common asynchronous communication protocols used in these types of systems is the so-called address-event representation (AER) communication protocol [9]–[12]. In this representation, input and output signals are real-time digital events that carry analog information in their temporal relationships (inter-spike intervals). Each event is represented by a binary word encoding the address of the sending node.

The activity of biological neurons is sparse in time, with typical firing rates ranging from a few per second to a few hundred per second. The speed of digital buses (tens of megahertz) allows the outputs of many VLSI neurons firing at these biologically typical rates to be multiplexed over one AE bus. To further reduce the bandwidth required on the AE bus, local connectivity can be hardwired on-chip [4], [8]. To handle cases in which multiple sending nodes attempt to transmit their addresses at exactly the same time (event collisions) on-chip arbitration schemes can be used [9], [13]–[15].

Chips that communicate using the AER communication protocol can be divided into *senders* with AER output only (e.g., silicon retinas [16], [17], or silicon cochleas [18]), *receivers* with AER input only [19], and *transceiver* chips, which are both senders and receivers [2], [7], [8]. Systems containing more than one AER sender chips can be assembled using off-chip arbitration.

One of the earliest multichip systems using the AER communication protocol, a silicon model of stereoscopic vision, was implemented by Misha Mahowald [9]. The system, consisting of three silicon chips interconnected with asynchronous digital buses, was able to extract, in real-time, depth information from visual stimuli detected by two silicon retinas. At that time and since, logic analyzers were and are often used to monitor AE buses. While still useful for debugging problems with AE protocol communication they suffer from several disadvantages for

E. Chicca, A. M. Whatley, P. Lichtsteiner, T. Delbruck, R. J. Douglas, and G. Indiveri are with the Institute of Neuroinformatics (INI), University of Zurich, Zurich CH-8057, Switzerland and the Swiss Federal Institute of Technology (ETHZ), Zurich CH-8092, Switzerland.

V. Dante is with the Italian National Institute of Health, 00161 Rome, Italy.

P. Del Giudice is with the Italian National Institute of Health, 00161 Rome, Italy, and also with the National Institute of Nuclear Physics (INFN), 00185 Rome, Italy.

monitoring purposes. Good logic analyzers are typically bulky and heavy and therefore not very portable, and too expensive to equip every researchers' bench with one. They also do not usually permit on-line real-time monitoring, since data cannot be downloaded at the same time that it is being acquired into acquisition memory. This makes logic analyzers unsuitable for experiments in which it is desired to incorporate conventional software-based algorithms into the processing loop, e.g., for certain on-line learning experiments. General purpose data acquisition (digital I/O) boards are also not ideal, since most inexpensive boards are not designed for use with asynchronous buses and each individual event handshake must therefore be handled by software which makes such boards very slow. On the AE generation side, the counterpart of the logic analyzer is the pattern generator. These are also not designed with asynchronous buses in mind and cannot wait for a handshake to complete but can only emit a fixed programmed pattern. Like logic analyzers, pattern generators can also not take part in on-line software-in-the-loop experiments and are also very expensive. The joint consideration of the needs for AE input and output suggests that a single special-purpose device, natively suited to asynchronous communication, would provide a more convenient, cheaper solution.

B. AER Communication Infrastructures

Infrastructures for constructing multichip pulse-based neuromorphic systems based on AER have been further developed by several researchers [10], [13], [20]–[23]. A wide range of examples of multichip AER systems have been presented in the past. These examples have used AER infrastructures that range from very bulky and highly complex general purpose solutions [19], [20], [24], to custom solutions in the form of dedicated printed circuit boards (PCBs) with microcontrollers and/or look-up tables [6], [25], [26]. In addition, a new set of general purpose AER boards with USB and USB2 interfaces have been recently proposed [23], [27], [28]. These boards represent a good compromise between general-purpose functionality, and compactness. However, as they are typically placed between AER chips in the signal processing path, they often do not have access to the address events of all chips present in the system. Furthermore each individual board has often a limited set of functionalities (e.g., to monitor AEs from a sender, to generate and send synthetic AEs to a receiver, to merge AEs from two senders into a receiver input, to map AEs from one address space to another, etc.), or require reprogramming at the field-programmable gate array (FPGA)/VHDL level in order to assume one of these particular functions. In this paper, we present a general-purpose solution in the form of a peripheral component interconnect (PCI) board (the PCI-AER board) that has all of these functionalities, with the possibility of connecting up to four senders with up to four receivers, that has access to the global AER address space used by the system, and that has a well-defined software interface. The approach of using one single PCI-AER board rather than many smaller USB-AER boards has the disadvantages of requiring a PC workstation to be present, even when only mapping is required, and limiting to some extent the overall size of the AER system that can be constructed. But it has the advantage of allowing convenient and rapid prototyping (e.g., by

stimulating, monitoring and/or experimenting with different address-space mappings), and of allowing seamless integration of software algorithms [29] (e.g., that implement learning or that change the network topology based on the system's activity). The PCI-AER board is therefore an ideal tool for developing AER neuromorphic models of biological sensors and neocortical processing structures. Specifically, we propose to use the board to study a computational module based on a network of spiking neurons with cooperative-competitive interactions.

C. Computational System Overview

We are ultimately interested in developing neuromorphic systems that reproduce some characteristics of neocortical processing modules. Despite significant differences in function across the various cortical areas, the pattern of neuronal connections within each area is remarkably similar [30]. This regular structure suggests that the cortex may use a common core processing circuit, or *canonical microcircuit*, that can be tuned to perform specific tasks and used in a modular fashion for implementing different functionalities [30], [31]. The canonical microcircuit, and its later extensions, emphasize the role of first order recurrent connections between cortical neurons. These recurrent connections between neurons support soft winner-take-all (WTA) mechanisms, in which networks of neurons participate collectively in the generation of an appropriate interpretation of their input.

The computational abilities arising from soft WTA mechanisms are especially useful for feature extraction and pattern classification problems. In the second half of this paper, we describe an application example comprising an AE temporally differentiating vision sensor interfaced to a VLSI device with a cooperative-competitive network of spiking neurons via the PCI-AER board. We apply this AER-based vision system to the implementation and comparison of two models of *orientation selectivity*. The models of mechanisms responsible for orientation selectivity have been controversial since its discovery by Hubel and Wiesel [32]. Originally it was believed that the primary origin of the orientation selectivity of simple cells was due to feedforward convergence of thalamic input (*feedforward model*). Subsequent experimental studies suggested that this contribution alone is insufficient to account for all properties of orientation tuning observed in the visual cortex [33]–[35], leading to the proposal for the involvement of recurrent intracortical excitation and inhibition (*feedback model*) in orientation selectivity. The origin of orientation selectivity in primary visual cortex has been extensively studied as a means to understand cortical circuitry and cortical computation [34], [36]–[39], and several hardware models of orientation selectivity have been proposed in the past and fabricated with monolithic [40]–[42] or multichip configurations [4], [7], [43]–[47]. An advantage of multichip configurations is that the computational stage is decoupled from the sensing stage. In this way the orientation selectivity computational devices can be designed to be modular and expandable. Within the multichip configuration approaches there are two main streams: [4], [43], [45] and [46] implement specific architectures with local or hardwired connectivity for processing signals obtained from vision sensors, while [7] and [47] propose using general

(a) (b)
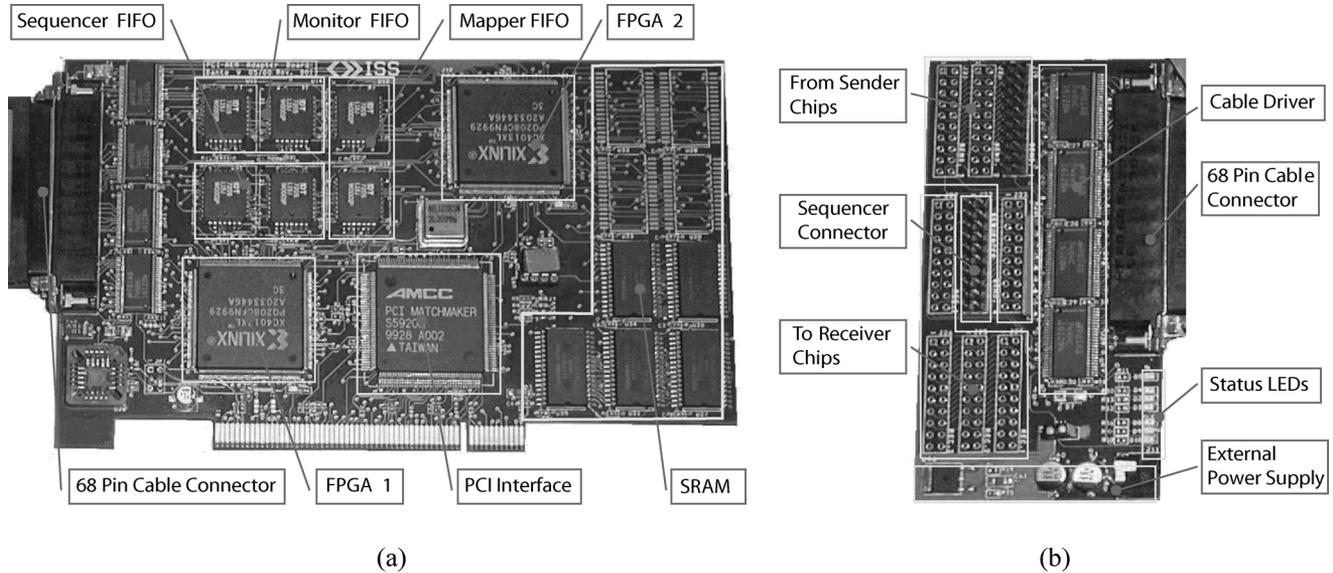
Fig. 1. PCI-AER board and header board. (a) PCI-AER PCI board. The devices involved in the implementation of the three major functional blocks (see Fig. 2) are highlighted: monitor, sequencer, and mapper FIFOs, the two FPGAs, and the SRAM used to hold the mapper's look-up table. A connector is used to connect to the header board [see Fig. (1b)] via a cable. The PCI interface chip controls the interface to the host PC. (b) PCI-AER header board. The header board connects to the PCI board via a ribbon cable. Cable drivers are used to ensure the integrity of the signals passing over this cable. The board can draw power either from the PCI board or from an external power supply. (The connector and associated components for this are highlighted.) The board provides connectors for up to four AER senders and four AER receivers plus a dedicated connector from the sequencer. The header board has five status LEDs: a "power on" LED, three LEDs which indicate whether each of the three functional blocks (monitor, sequencer and mapper) are enabled, and three FIFO full indicator LEDs (one each for the monitor, sequencer and mapper FIFOs).

purpose transceivers that rely on the AER communication infrastructure to construct receptive fields tuned to different orientations. The multichip system we designed is a hybrid of the approaches proposed in these two main streams, as it has local hardwired connections and supports arbitrary connectivity patterns via additional AER synapses. Specifically, in our system, we can map different types of sensory inputs (e.g., obtained from a silicon retina, a silicon cochlea, or other AER sensory systems) onto the network's AER synapses in a way to implement cooperation and competition across different types of feature maps. The computational part of the system is not explicitly designed for orientation selectivity. Instead, it models a more generic computational module (which represents a portion of a *cortical module* [30], [31]) that can be applied to the detection of other features, and to other sensory modalities. In our specific application example the receptive fields emerge both from the inter-chip feedforward connectivity [32] and the intra-chip recurrent cooperative-competitive connectivity.

In the next section, we describe the PCI-AER board and its supporting software. In Section III, we present the application example on orientation tuning, and in Section IV, we present a discussion and concluding remarks.

II. COMMUNICATION INFRASTRUCTURE

A. Hardware: PCI-AER Board

The PCI-AER board takes the form of a 33 MHz, 32-bit, 5-V PCI bus add-in card Fig. 1(a). It was designed by Dante and the PCBs were manufactured by Ermes Technology S.R.L. (Via Ivrea 18, 10080 San Benigno Canavese, Italy). Most of the two dozen or so boards in existence were assembled by SMTEC AG (Gewerbestrasse 5, 8451 Kleinandelfingen, Switzerland). When

the board is installed in a host PC, a ribbon cable is used to connect it to a small header board Fig. 1(b) which can be conveniently located on the bench-top and provides connectors for up to four AER receivers and four AER senders. The header board also electrically buffers the signals to and from the receivers and senders. If only one receiver is used, all 16 bits of the AER bus can be used by that receiver. If two receivers are used, the topmost bit is used to distinguish between them and only 15 bits of address are available to each receiver. If all four receiver channels are used then the two topmost bits are used to distinguish between them and only 14 bits of address are available to each receiver. The number of channels to be used (1, 2, or 4) can be configured by software. Similarly the senders may use 16, only 15 or only 14 bits of address according to whether the board is configured for 1, 2, or 4 sender chips. Senders must use the so-called "SCX" multisender AER protocol [48] in which request and acknowledge signals are active low and the bus may only be driven while the acknowledge signal is active. Receivers may use either this "SCX" protocol, or they may choose to use a point-to-point protocol [11] in which request and acknowledge are active high and the bus is driven while request is active. Which protocol is generated by the board may be selected under software control.

As illustrated in Fig. 2, the PCI-AER board can perform three functions which are executed by blocks we refer to as the *monitor*, *sequencer*, and *mapper*. These blocks are implemented in two FPGAs on the board. The division of the functionality between the two FPGAs is a consequence of the data flow. One FPGA deals with all incoming events, whether from external sources or from the PCI bus and optionally passes these events on to the mapper. Hence, this FPGA implements the monitor and sequencer functions, as well as handling the input to the
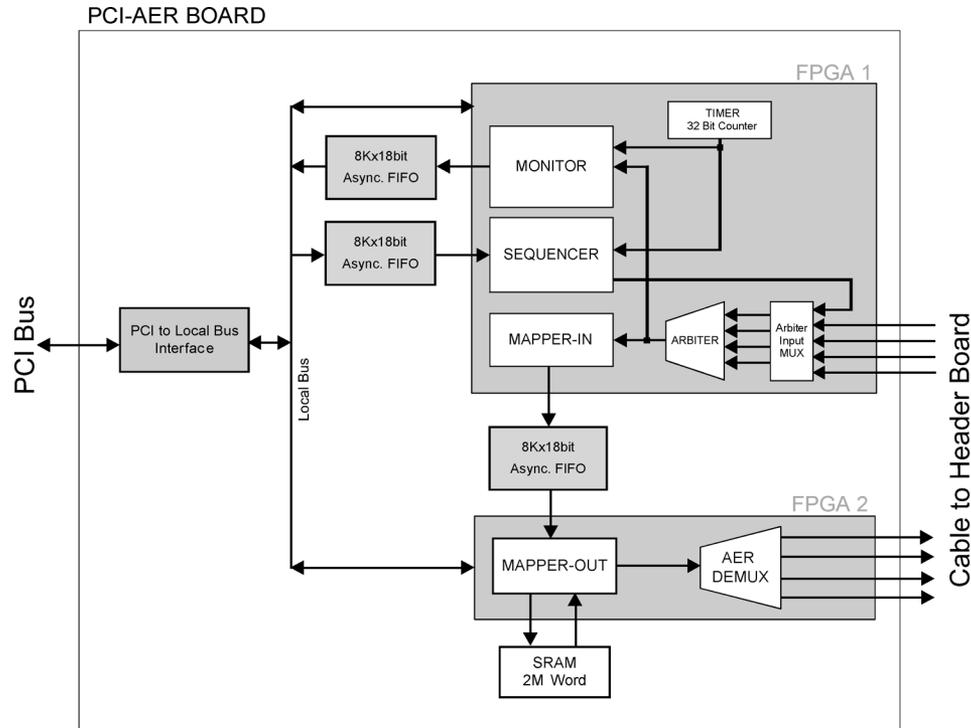
Fig. 2. Block diagram of the PCI-AER interface board showing its three major functional blocks, i.e., the MONITOR, the SEQUENCER, and the mapper (divided into MAPPER-IN and MAPPER-OUT). These (and other blocks) are implemented in two FPGAs. Also shown are the FIFOs, the interface from the PCI bus to the local bus provided by an AMCC S5920 chip [50], the SRAM used to hold the mapper's look-up table, and the interconnecting buses.

mapper. The other FPGA performs the mapper function proper (including managing the interface to the mapper's SRAM) and manages communication with the AER receivers.

The monitor can capture and timestamp events coming from the attached AER senders via an arbiter[1] and makes those events available to the PC for storage or further on-line processing. A timer is implemented in one of the FPGAs, and when an incoming AE is read, a timestamp is stored along with the address in a first-in first-out (FIFO) memory. This FIFO decouples the management of the incoming AEs from read operations on the PCI bus, the bandwidth of which must be shared with other peripherals in the PC such as the network card. The FIFOs fitted to the current boards are all 8-KWords deep (and 18-bit wide) and since the addresses occupy one word and the timestamps two words each, this is sufficient to hold 2730 complete events.[2] Interrupts to the host PC can be generated when the FIFO becomes half-full and/or full, and in the ideal case, the driver will read time stamped AEs from the monitor FIFO whenever the host CPU receives a FIFO half-full interrupt, at a rate sufficient that the FIFO never fills or overruns, given the rate of incoming AEs. If the CPU fails to empty the FIFO at a sufficient rate, the FIFO will fill up, the FIFO full light-emitting diode (LED) on the header board will light, and a FIFO full interrupt will be generated. At this point, incoming events will be lost until such time as the CPU can once again read from the FIFO. In the application example we present in Section III the monitor is used to record the activity of a sender and a transceiver chip.

[1]The arbiter is a binary-tree arbiter in which each binary cell is a priority based arbiter.

[2]The monitor can also be run without storing timestamps, in which case the FIFO can hold up to 8 K event addresses.

The sequencer allows events originated by the host PC to be sent out to the attached AER receivers. These events may for example represent a pre-computed, buffered stimulus pattern, but they might also be the result of a real-time computation. This allows for instance software simulations of VLSI devices to provide input to real VLSI hardware while the former VLSI devices are still under development. As soon as the real device is available, the software simulation can be seamlessly replaced. Like the monitor, the sequencer is decoupled from the PCI bus using an 8-KWord FIFO. The host writes a sequence of words representing addresses and time delays to the sequencer FIFO. The sequencer then reads these words one at a time from the FIFO and either emits an AE or waits the indicated number of microseconds. Since addresses and time delays are represented in the sequencer by one word each, and a stream of AEs usually consists of alternating addresses and time delays representing inter-spike intervals, the 8-KWord FIFO can typically hold up to 4096 events. FIFO half empty interrupts can be generated to signal the CPU to supply further data to the sequencer. If the CPU fails to supply data to the sequencer at a rate sufficient to prevent the sequencer FIFO becoming empty, this may indicate a failure of the system to generate the desired sequence of events with the desired timing. In this case a sequencer FIFO empty interrupt is raised to signal the underrun. The address events generated by the sequencer pass through the mapper and can therefore be transmitted on any of the four output channels.

The mapper implements programmable inter-chip synaptic connectivity. It maps incoming AEs from attached AER senders and/or the sequencer to one or more outgoing addresses for transmission to the attached AER receivers. It can operate in pass-through, one-to-one, or one-to-many modes. In

pass-through mode, the outgoing addresses are the same as the incoming addresses. In one-to-one mode, each incoming address is mapped to one outgoing address by using the incoming addresses as the index into a look-up table stored in on-board SRAM. This look-up table then holds the corresponding target output addresses. In one-to-many mode, each incoming address may be mapped to one or more outgoing addresses. This is achieved by using the contents of the look-up table as pointers to lists of output addresses, also stored in the 2 MWords of on board SRAM.[3] The mapper too has a FIFO which decouples the asynchronous reception of the incoming AEs from the generation of outgoing AEs. Should this FIFO become full, it is possible that events will be lost, and this eventuality can be signaled to the CPU by means of an interrupt. The mapper, once it has been configured and the look-up table filled with the required mappings, operates entirely independently of the host CPU, since all of the necessary operations, including table look-up are performed by one of the FPGAs. To implement the multichip system described in Section III we used the mapper in one-to-many mode.

A detailed description of the hardware (the Hardware User Manual) is available at [49].

B. Supporting Software

To enable the functionality of the board to be accessed robustly and conveniently, we have provided a Linux device driver and, on top of this, a C library. Both are fully documented and this documentation is available together with the source code at [49]. The open-source driver provides full integration of the PCI-AER board under Linux following the Unix "everything-is-a-file" model. This allows AE data streams to be accessed using standard Unix read and write calls and supports the use of standard shell redirection and command-line tools. The driver provides separate logical devices for each of the major functional blocks of the board: mapper; monitor; and sequencer, and supports multiple boards. It ensures that the AE data streams remain coherent by serializing accesses from multiple programs running simultaneously. It also forces word-multiple sized access to prevent corruption of the data streams due to misalignment. While read and write calls are used to read and write the AE data streams from and to the board's FIFOs, IOCTL (input/output control) calls are provided to set and get configuration states, and user programs are prevented from putting the board into an inconsistent state. The driver also manages the mapper look-up table memory to relieve users of the task of performing the necessary but onerous and error-prone table indexing and pointer arithmetic and so prevents the mapping table from becoming corrupted. Statistics (number of words read or written, number of interrupts, number of FIFO overruns or underruns etc.) are also maintained by the driver for each logical device and made available to user programs.

The library consists mainly of thin, fast wrapper functions around the driver open, close, read, write, flush and ioctl calls. Functions are also provided to convert from the PCI-AER hardware-specific format to a generic inter-spike interval/AE format for reading, and *vice versa* for writing. The conversion function

---

[3]In order to send the same address to more than one output channel, the appropriate target addresses must be listed serially in the output address lists.

for reading also attempts recovery when data are received out of order because of monitor FIFO overruns or other (hardware) errors.

C. Performance

The performance of the PCI-AER hardware and driver must be considered together since the board is unusable without a driver. Once the board has been initialised and configured, and desired mappings established in the mapper, mapping continues to operate without intervention from the driver, but a driver is necessary to perform those initialization and configuration steps, and to read and change mappings later, if so desired. For the sequencer, the driver can be used to fill the 4 K-events FIFO and may play no role until the FIFO empties. However it is not possible to know that the FIFO has emptied without using the driver, and such a one-shot, 4 K-events-only mode of use is not very interesting for many applications in which it is desirable to stimulate AER devices continuously. Similarly, events can be collected in the monitor until its FIFO fills up, but they cannot be read without the driver, nor again is it usually very useful to constrain data capture to 2 2/3 K events.

Some of the various aspects of the performance are highly dependent on the type of PC in which the board is installed and on which the driver is running, and on the version and configuration of the Linux kernel in use. Nonetheless the driver is instrumented for measuring the throughput of the monitor and sequencer and measurements have been performed using version 2.30 of the driver on a 1-GHz, 512-MB, AMD Athlon based machine running a SuSE 9.1 Linux distribution with kernel version 2.6.5. The PCI-AER board FPGA revisions used were 4202 (FPGA1) and 4203 (FPGA2). In order to eliminate as many outside influences as possible and obtain reproducible results, measurements were made with no graphical display system and no network stack running. The results of these measurements are shown in Table I. Data for the monitor was obtained by connecting one monitor input channel to a circuit which produces AE requests at a rate determined by a function generator input, while a test program runs a loop which does nothing but read from the driver into a static buffer. The AE rate was then gradually increased until FIFO overruns began to occur. The sequencer data was obtained by connecting together the AE request output and acknowledge input pins on one of the receiver chip connectors on the header board, thus providing the fastest possible acknowledgement of a request, and running a test program which configured the board for single channel output with the mapper in pass-through mode, and which then continuously writes the contents of a pre-prepared buffer to the driver. The inter-spike intervals were specified to be 0 s. The test software is available, along with the driver, at [49]. Minimum cycle times were measured by observing the request and acknowledge signals at the header board using an oscilloscope. Performance data obtained when using a more sophisticated software framework on top of a slightly earlier version of the driver on a different machine were presented in [29].

The limiting factors are various. The sequencer maximum sustainable rate lies in the region between using inter-spike intervals of 1 μs and inter-spike intervals of 0 s, but since the best available resolution of the timer controlling the emission of

TABLE I
PCI-AER BOARD AND DRIVER PERFORMANCE DATA

Monitor max sustainable rate (without FIFO overruns)	$\approx 420\ k\ events\ s^{-1}$
Sequencer max sustainable rate (without FIFO underruns)	$1.0\ M\ events\ s^{-1}$
Monitor min AE cycle time	$< 280\ ns$
Sequencer max instantaneous rate	$\approx 1.1\ M\ events\ s^{-1}$
Sequencer min AE cycle time	$\approx 240\ ns$

events from the sequencer is $1\ \mu s$, the timer resolution represents the limiting factor here. Were the resolution of the timer better, then the data transfer rate would become the limiting factor. The monitor maximum sustainable rate is determined principally by the speed at which the AE stream can be read from the board over the PCI bus and buffered in the PC's memory. Due to the lack of support for direct memory access (DMA) in the current design, this in turn depends on many factors—CPU speed, cache size and organization, PCI bus chipset, Linux kernel version etc.

The present board is unable to perform DMA to shift AE data across the PCI bus because the AMCC S5920 PCI interface chip [50] used does not support it, but later generations of hardware should certainly use DMA, not only to achieve greater throughput across the PCI bus, but also to offload the CPU, see for example [51]. Alternatively, with the present board, use might in future be made of the forthcoming Intel I/O Acceleration Technology (I/OAT)[4] in which the CPU includes a DMA subsystem.

Minimum AE cycle times on both the inputs from senders and on the outputs to receivers depend only on the frequency of the clock on the board (20 MHz on the present board) and the requirement to remain within the AER protocol specifications. They do not depend on the number of channels in use, although the latency before acknowledging a request on one arbiter input channel may of course be influenced by the presence of events on other channels if other input channels are in use.

Since the limiting factors are principally on the PC side and not on the AER side, it is clear that the overall bandwidth available for say monitoring remains constant irrespective of the number of channels in use, and that therefore if multiple channels are being monitored, they must share the available bandwidth. Note however that when not monitoring or sequencing, the bandwidth available on the PCI bus plays no role in the mapping performance, since the data-path is then from FPGA1 to Mapper FIFO to FPGA2 and does not even involve the Local Bus on the board (refer again to Fig. 2).

The throughput of the mapper depends on whether it is being operated in pass-through, one-to-one or one-to-many modes, and in the latter case, on the length of the target address lists being used. In measuring the sequencer rates given in Table I, the mapper is being used in pass-through mode, thus it is clear that at least in this mode it can sustain rates of $\approx 1\ M\ events\ s^{-1}$. This would allow 10^4 neurons to be actively firing at a rate of 100 Hz. Assuming no more than 10% of neurons are active at one time, a network of the order of 10^5 neurons could be supported by one PCI-AER card, but address space considerations
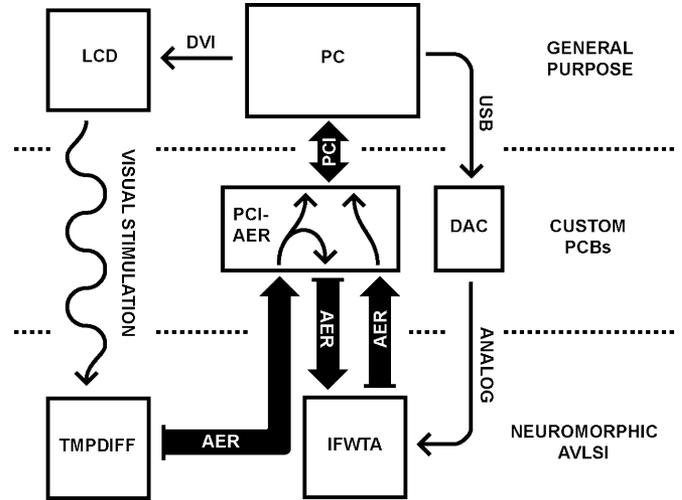
Fig. 3. AER orientation selectivity system setup. The PCI-AER board routes output events of the TMPDIFF chip in response to visual stimuli to the IFWTA chip and monitors the activity of both chips. The PC controls the LCD screen for stimulus presentation, the PCI-AER board and the DAC board.

restrict us to supporting a maximum of 65536 neurons on the sender side and a maximum of 65536 synapses on the receiver side. If the network produces more spikes than can be processed by the mapper in real-time, then when the mapper FIFO fills, spikes will eventually be lost, but this will have no influence on the AE protocol cycle times observed on the sender side.

Although the driver supports the use of more than one PCI-AER board in one PC, the user inevitably remains limited by the characteristics of that host PC and in particular the bandwidth available for monitoring and/or sequencing within the host system must then be shared between all of the boards which are fitted to that system. However as noted above, when boards are only being used for mapping, there is no impact on the host, so several boards fitted to the same PC might easily be used for mapping in larger AE systems while for instance only one board at a time performs a monitoring function. AEs do not need to be routed from one PCI-AER board to another via the PC but rather one of the output channels of one board could be connected to one of the input channels of another board, or perhaps to the input of a transceiver chip the output of which goes to the input of another board.

III. APPLICATION EXAMPLE: ORIENTATION SELECTIVITY USING A SILICON RETINA AND A WTA NETWORK

A. Orientation Selectivity System Components

The orientation selectivity system consists of two neuromorphic aVLSI AER chips, a PCI-AER board and supporting hardware (see Fig. 3). The neuromorphic chips are an AE temporally differentiating (TMPDIFF) vision sensor chip [52] and a recurrent competitive network of I&F neurons and short-term dynamic synapses (IFWTA chip) [53].

The AEs generated by the TMPDIFF chip and sent to the WTA chip are routed by the PCI-AER board mapper. The PCI-AER board monitor is used to read all AEs (generated by the two chips), timestamp them and log them on the host PC (see Fig. 3).

(a)   (b)
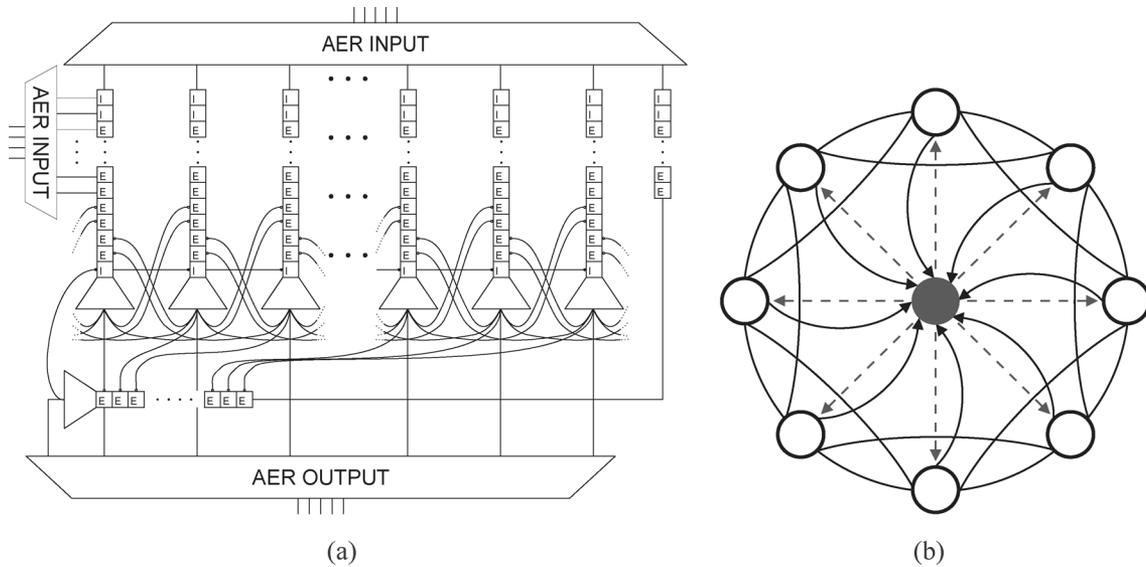
Fig. 4. (a) Chip architecture. Squares represent excitatory (E) and inhibitory (I) synapses, small unlabeled trapezoids represent I&F neurons. The I&F neurons can transmit their spikes off-chip and/or to locally connected synapses (see text for details). (b) Schematic representation of the connectivity pattern implemented by the internal hardwired connections (closed boundary condition). Empty circles represent excitatory neurons and the filled circle represents the global inhibitory neuron. Solid/dashed lines represent excitatory/inhibitory connections. Connections with arrowheads are monodirectional, all the others are bidirectional. Only eight excitatory neurons are shown for simplicity; the actual chip contains 31 excitatory neurons.

The supporting hardware comprises a custom digital–analog converter (DAC) board [54] for setting the analog biases of the neuromorphic chips, a liquid crystal display (LCD) screen for presenting visual stimuli, and a workstation for hosting and controlling the PCI-AER board, programming the DAC board and controlling the LCD screen.

The PCI-AER board mapper functionality was critical in this application example, as it allowed us to (re-)configure the mapping between the TMPDIFF pixels and the IFWTA synapses. Similarly, the board's monitoring function allowed us to store arbitrarily large numbers of AEs generated by the system for off-line analysis. The multichip system was not developed to process real images, rather it was designed to validate models of orientation selectivity and illustrate the functionalities of the PCI-AER board.

1) TMPDIFF Chip: The TMPDIFF chip implements the sensing stage of our system. The chip produces asynchronous AEs in response to temporal changes in logarithmic intensity. The stream of events encodes contrast change rather than absolute illumination change. The retinal computation is optimized to deliver relevant information and to discard redundancy using high temporal and low spatial resolution, similar to the biological magnocellular pathway. Because the TMPDIFF chip responds only to temporal changes in logarithmic intensity, static scenes produce no output. AEs represent relative changes in image intensity that usually are generated by viewpoint or object movement. The TMPDIFF pixel front end photoreceptor circuits independently compute the temporal derivative of the logarithm of the pixel illumination I in continuous time. The output of the photoreceptor circuit consists of an ON current for increasing intensities and an OFF current for decreasing intensities. The ON and the OFF currents are proportional to the temporal derivative of $\ln I$. The ON current is fed as an input current into an I&F neuron circuit that communicates quantized logarithmic changes

as ON-events. The OFF current is fed into another I&F neuron that produces OFF-events. As long as the temporal frequency of the visual stimulus is higher than the corner frequency of the input to the neuron circuits (2 Hz), each event means that the logarithmic intensity changed by a certain fixed amount since the last event. If the absolute pixel illumination is I, then each event represents a quantized change R in $\ln I$

$$R = d\ln I = dI/I. \qquad (1)$$

Thus, the temporal derivative is self-normalized. Pixel output consists of the stream of ON and OFF events. This vision sensor, more thoroughly described in [55] and [52], consists of an array of 32×32 pixels, a y-arbiter, an x-arbiter and a common address bus with two encoders [12]. An event occurring in a pixel is communicated to the outside of the chip as an 11-bit address that encodes the pixel X-Y location and the polarity (ON or OFF) of the event. Events are processed asynchronously in order of their arrival time. In case of colliding events the later events are queued. The vision sensor is a real-time device, as events are typically communicated within 100 ns of their occurrence. The AER communication system is particularly well suited for this application because it dedicates the full communication bandwidth to the active pixels of the vision sensor and preserves timing information. In response to a flashed bar, for example, within the first few milliseconds after presentation or removal of the bar a burst of a few hundred events is typically emitted; these bursts are preceded and followed by zero activity. The maximum event rate is about $2 \; \mathrm{M \, event \, s^{-1}}$. With sparse activation a very high temporal resolution is achievable, comparable with frame rates of several kilohertz.

2) The IFWTA Chip: The architecture of the IFWTA chip is shown in Fig. 4(a). It is a two-dimensional array containing a row of 32 I&F neurons, each connected to a column of afferent synaptic circuits. Each column contains 14 AE excita-

tory synapses, 2 AE inhibitory synapses and 6 locally connected (hardwired) synapses. When an AE is received, the synapse with the corresponding row and column address is stimulated. If the synaptic current resulting from the AEs routed to the neuron integrates to the neuron's voltage threshold for spiking, then that neuron generates an AE which is transmitted off-chip. The AE input synapses can be used to implement arbitrary network architectures, by (re)mapping AEs via the PCI-AER board.

Synapses with local hardwired connectivity are used to realize a cooperative-competitive network with recurrent interactions [see Fig. 4(b)]: 31 neurons of the array send their spikes to 31 local excitatory synapses on the global inhibitory neuron; the inhibitory neuron, in turn, stimulates the local inhibitory synapses of the 31 excitatory neurons; each excitatory neuron stimulates its first and second neighbors on both sides using two sets of locally connected excitatory synapses. The first and second neighbor connections of the neurons at the edges of the array are connected to pads. This allows us to leave the network open, or implement closed boundary conditions (to form a ring of neurons [56]), using off-chip jumpers. The local synapses are nonlinear integrators which produce analog currents in response to digital input spikes. The local hardwired connectivity was implemented as described above in order to reduce AER bandwidth usage, while trying to keep the additional area occupied small with respect to the overall network size. Furthermore, it provides the flexibility to use the chip as a standalone module for single-chip experiments in which there is no need for mapping.

All of the synapses on the chip can be switched off by appropriately setting the external bias voltages that control their synaptic weights; the local and AER synapses are controlled by independent bias voltages. This allows us to inactivate either the local or the AE synaptic connections, or to use them in some arbitrary combination. A detailed description of the IFWTA chip was presented in [53], [57].

B. Orientation Selectivity Experiments

In our application example, broad orientation selectivity is achieved by appropriately mapping feedforward connections from the TMPDIFF pixels to the IFWTA chip neurons (via the PCI-AER board), and it is sharpened by activating the local recurrent connections on the IFWTA chip. The feedforward mapping is set so that each IFWTA neuron collects all the TMPDIFF ON and OFF events that belong to a bar with a specific orientation and position (discarding polarity), as shown in Fig. 5. We mapped 31 different groups of TMPDIFF pixels onto 31 neurons of the IFWTA chip so as to form 31 differently oriented receptive fields. The orientations of these receptive fields are indicated by the bars shown as insets in Fig. 6(a).

In our experiments we displayed to the TMPDIFF chip oriented white bars on a dark background, flashing on an LCD screen. The activity of the TMPDIFF chip was monitored by the PCI-AER board and transmitted (via the PCI-AER board mapping tables) to the IFWTA chip. Using the PCI-AER board, we time-stamped and logged both the TMPDIFF and IFWTA AEs for data analysis. To characterize the system we collected the system's activity in response to bars of 30 different orientations (6 degrees apart from each other) chosen independently of the
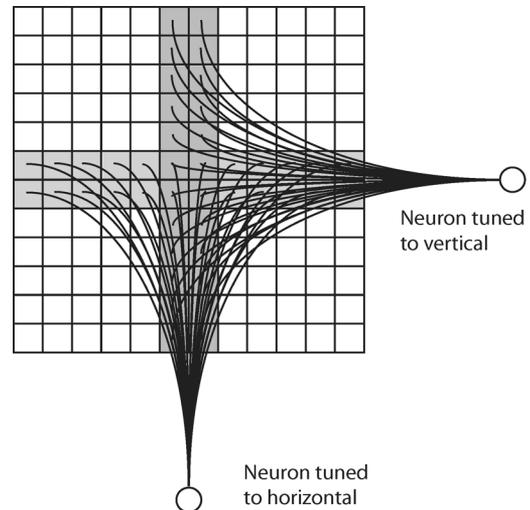
Fig. 5. Sketch representative of the mapping from the TMPDIFF chip to the IFWTA chip. The TMPDIFF retina is represented by a twelve by twelve array of pixels, lines represents excitatory connections from the TMPDIFF chip to neurons of the IFWTA chip (represented by circles).

set of pre-wired preferred orientations. Each oriented bar was flashed at a rate of about 2.5 Hz, producing one ON and one OFF transition per cycle and the AE data was monitored for 25 seconds. Because it was not easy to synchronize the stimulus onset with the start of the monitoring, we decided to start the monitoring 5 s after executing the command to start the stimulus. In this way, we were sure that the stimulus was already present when we started monitoring.

We repeated the same experiment for two different conditions in terms of the local connectivity of the WTA chip. In the first condition the biases of the WTA chip were set to implement a purely feedforward model: local recurrent synapses were inactive and the neurons' inputs were completely determined by the activity of the retinal pixels. Subsequently, we activated the recurrent connectivity to implement the feedback model maintaining all other parameters unchanged. Three sets of local synapses were used to implement the feedback model: 1) first neighbor excitatory to excitatory synapses to simulate the mutually excitatory connections among cells with similar preferred orientations; 2) inhibitory and 3) excitatory synapses connecting the global inhibitory neuron to the excitatory neurons and *vice versa* to simulate the mutual inhibition among cells with different preferred orientations [see Fig. 4(b)]. The effect of competition alone is described in [57].

Orientation tuning curves (i.e., graphs of neural response versus stimulus orientation) are typically measured in experiments related to the characterization of orientation selectivity in visual cortical neurons. We applied the same analysis to our data: the recorded activity of the WTA neurons was used to compute the mean firing rate of each neuron in response to the stimuli and tuning curves were obtained by plotting these data for each neuron as a function of stimulus orientation. Fig. 6(a) shows the computed tuning curves for each neuron of the IFWTA. Each sub-figure represents the mean response of the neuron to different orientations. The inset in each sub-figure represents the retinal pixels mapped to that particular neuron.
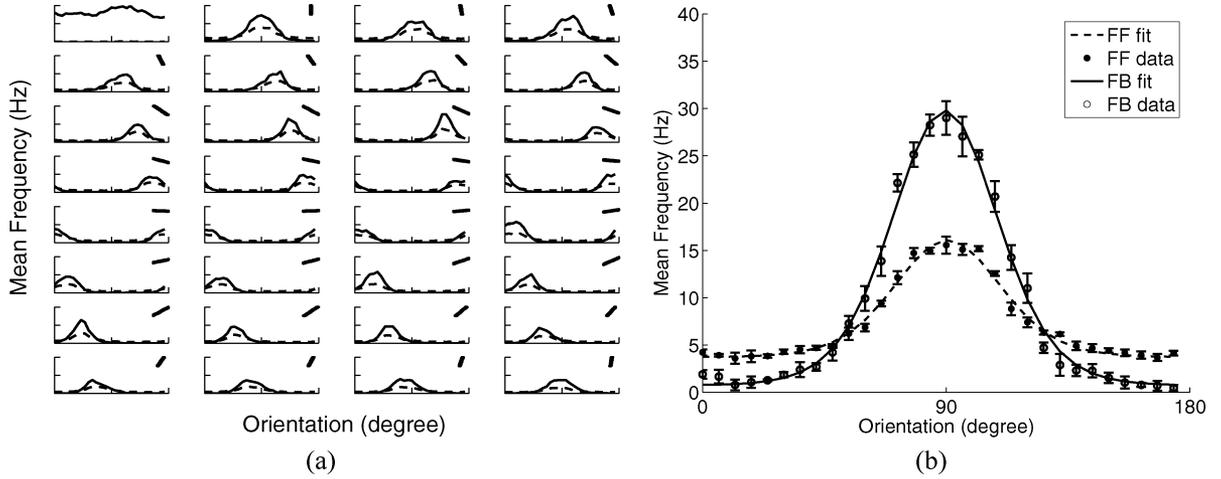
Fig. 6. Tuning curves for the feedforward (dashed line) and the feedback (solid line) model of orientation selectivity. (a) The mean frequency (hertz) of each neuron is plotted as a function of stimulus orientation [the scales are the same for all plots and can be seen in (b)]. The top left graph shows the activity of the inhibitory neuron, the other graphs show the activity of the excitatory neurons (a bar representing the retinal pixels mapped to the neuron, i.e., its preferred orientation, is shown in each plot). The tuning curves of the feedforward model have a larger amplitude and a smaller half-width at half-height compared to the tuning curves of the feedback model. (b) Tuning curves for the feedforward (dashed line and filled circles) and the feedback (solid line and empty circles) model of orientation selectivity for the neuron with vertical preferred orientation [enlargement of the second from left panel in the first row of (a)]. The lines represent the von Mises functions fitted to the data, represented by circles and error bars (standard deviation over the measured mean frequency.)

The TMPDIFF central pixels are mapped to all neurons, therefore each WTA neuron is also receiving input events when its nonpreferred orientation is presented to the retina. The effect of this "base line" input is clearly visible in the feedforward model, where the activity of the WTA neurons simply reflects the input from the retina. In this case, the frequencies in the tuning curves are greater than zero for all orientations and a maximum is observed at the preferred orientation. In the feedback model the "base line" activity is suppressed and the activity in response to the preferred orientation is amplified.

We fitted the tuning curves to quantitatively estimate the effect of recurrent connectivity on the response of the orientation selective neurons. We used a von Mises function as the fitting function [58], defined as

$$M(\theta) = Ae^{k[\cos 2(\theta - \phi) - 1]} \tag{2}$$

where A is the value of the function at the preferred orientation ϕ, and k is a width parameter, from which the half-width at half-height $\theta_{0.5}$ may be calculated (in radians) as:

$$\theta_{0.5} = 0.5 \arccos\left[(\ln 0.5 + k)/k\right]; \quad k > -0.5 \ln 0.5. \tag{3}$$

The von Mises function approximates a Gaussian in shape over a biologically likely range of values of k. A least-squares fit of the data to the von Mises function was used to estimate the parameters of the tuning curve of each selective oriented neuron. Fig. 6(b) shows the tuning curve of the neuron tuned to vertical orientation: the data and the von Mises fitted function are plotted for both the feedforward and feedback model. The data points used to perform the fits are the mean frequencies of the neurons computed over the 25 s of data acquisition. The IFWTA chip is stimulated only during and shortly after the appearance and disappearance of the bar, when the ON and OFF pixels of the TM-PDIFF chip are activated by the visual stimulation. High variability is then induced in the pattern of activity of the TMPDIFF
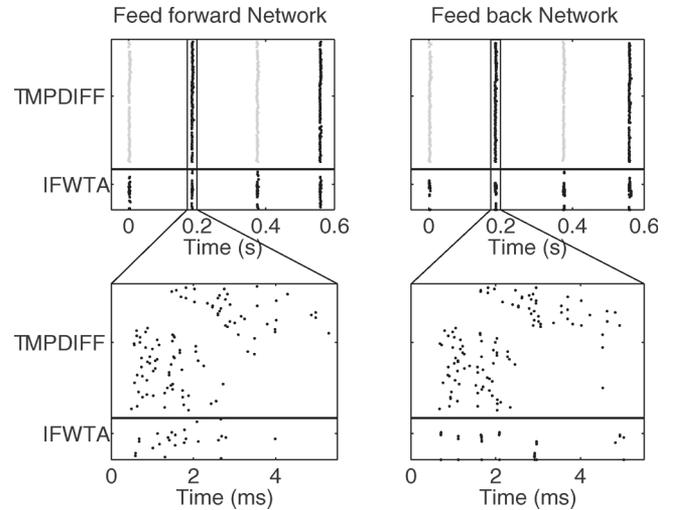
Fig. 7. Raster plot of the response of the TMPDIFF pixels and IFWTA neurons to a vertical bar. The graphs in the top row show the ON (black dots) and OFF (grey dots) response of the TMPDIFF chip (top) and the response of the IFWTA chip (bottom) to two cycles of the flashing stimulus. The graphs in the bottom row show a magnified versions of one of the bursts of the top row graphs. In the left column the IFWTA is configured to implement the purely feedforward model. In the right column the IFWTA chip implements the recurrent network described in the text. This graphs show how the orientation selectivity system produces bursts of activity in response to the appearing/disappearing of the flashing bars and no activity when the stimulus is static. As shown by the two graphs in the second row, the response delays of the feedforward network and feedback network are comparable.

and IFWTA chips (see Fig. 7), with bursts of events during the appearance and disappearance of the flashing bar and gaps of no activity in between. Ideally, the spike rate during each single burst should be measured and considered as a single measurement. The mean and standard deviation over many repetitions of this measurement would provide a good estimation of the mean frequency and its variation. To allow a simpler manipulation of the data and start from a more reliable "single" measurement
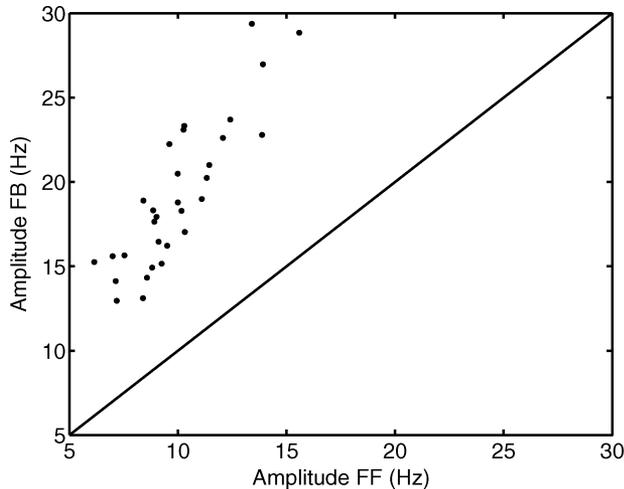
Fig. 8. Population data for the amplitude of the tuning curve at the preferred orientation (feedback versus feedforward model).
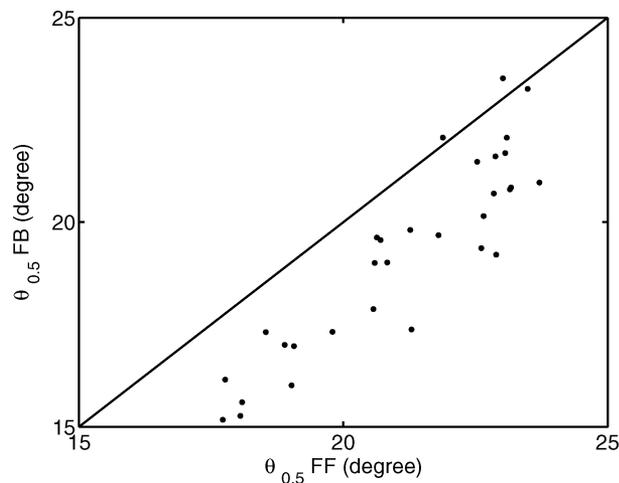
Fig. 9. Population data for the half-width at half-height of the tuning curve (feedback versus feedforward model).

we decided to divide our 25-s acquisition time into five 5-s intervals, and consider the means over these intervals as single measurements of the neurons' mean frequencies in response to the stimulus. The variability of our data [shown as error bars in Fig. 6(b)] is then computed as the standard deviation over the five measured frequencies.

To evaluate the goodness of the fits we used the R-square value (the square of the correlation between the measured values and the values predicted by the fit). It can take on any value between 0 and 1, with a value closer to 1 indicating a better fit. We calculated R-square for all the fits: the mean of all the computed values is 0.982 with a standard deviation of 9×10^{-3}, which indicates that on average the fits can explain 98% of the total variation in the data.

Figs. 8 and 9 show the estimated amplitude and half-width at half-height respectively, for all the neurons in the network in the feedback versus the feedforward configuration. All neurons lie above the diagonal in Fig. 8, showing that the response to the preferred orientation is amplified in the feedback network with respect to the response in the purely feedforward network. Sharpening of the tuning is shown in Fig. 9, where neurons tend

TABLE II
PARAMETERS OBTAINED BY LEAST-SQUARES FITTING OF THE DATA TO THE VON MISES DISTRIBUTION. THE MEAN AND STANDARD DEVIATION OVER THE POPULATION OF 31 ORIENTATION SELECTIVE NEURONS

	Feed-forward Model		Feed-back Model	
	Mean	Standard deviation	Mean	Standard deviation
Amplitude (Hz)	10	2	19	4
$\theta_{0.5}$ (°)	21	2	19	2
Baseline activity (Hz)	1.7	0.6	0.07	0.11
Preferred orientation error (°)	3	2	3	2

to lie below the diagonal. The population mean values of these parameters plus the baseline activity and the preferred orientation error are listed in Table II. On average the peak activity in the feedback network is twice the peak activity in the feedforward network and the ratio between the half-width at half-height for the two configurations is 0.9 (feedback over feedforward).

IV. CONCLUSION AND OUTLOOK

We have presented a flexible hardware/software infrastructure for building complex neuromorphic systems using the AER. It provides monitoring, sequencing and mapping functions easily accessible through the software interface and it allows convenient and rapid prototyping (e.g., by stimulating, monitoring and/or experimenting with different address-space mappings). The PCI-AER board is therefore an ideal tool for developing single and multichip AER systems. Additional application examples that rely on this PCI-AER board have been recently presented in [2], [29], [59]–[61]. This infrastructure and its documentation has reached the point at which it can be easily used by researchers and labs which were not involved in its development. At the time of writing, five such labs have acquired one or more boards each. Some users have written small C or C++ applications for spike-train generation and data logging directly using the library Application Programming Interface (API). A Matlab toolbox [62] has been developed for the off-line generation of spike trains to be sent to the PCI-AER board via the library and driver. A client-server architecture [29] has also been developed on top of the library to enable the use of the board on-line from within Matlab, including real-time data display.

Future developments should include a refinement of this client-server architecture to enable multiple data-sinks to read the monitored AE stream concurrently in a coordinated way. Other possible future developments include Java support, and a stimulation tool for the on-line generation of AE patterns to drive the sequencer. The instrumentation of the driver and consequent availability of performance data will aid the assessment of the present communication infrastructure, and can be used to guide future driver optimization work. The library could also be ported to other AER monitoring, mapping and sequencing hardware providing cross-platform compatibility for higher level software.

We presented the implemented orientation selectivity system composed of a sensing stage (the TMPDIFF chip) and a computational module (the IFWTA) not explicitly designed for this purpose by using a specific mapping between the two chips. We showed how the recurrent connectivity in the computational module has an effect on the response to oriented stimuli

similar to those described in theoretical models of orientation selectivity.

This experiment demonstrates the feasibility of real-time AER-based inter-chip communication through the PCI-AER interface. The orientation selectivity system was assembled exploiting the monitoring and mapping functionality of the PCI-AER board, thereby demonstrating the capabilities of the board. Other experiments exploiting the monitor and sequencing functionality have been described elsewhere (e.g., [2], [63]–[65]).

REFERENCES

[1] W. Maass and C. M. Bishop, *Pulsed Neural Networks*. Cambridge, MA: MIT Press, 1998.

[2] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 211–221, Jan. 2006.

[3] F. Tenore, J. Vogelstein, R. Etienne-Cummings, G. Cauwenberghs, and P. Hasler, "A floating-gate programmable array of silicon neurons for central pattern generating networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3157–3160.

[4] P. Merolla and K. Boahen, "A recurrent model of orientation maps with simple and complex cells," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, Dec. 2004, vol. 16, pp. 995–1002.

[5] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, S. Fusi, and P. Del Giudice, "A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long term memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1297–1307, Sep. 2003.

[6] R. J. Vogelstein, F. Tenore, R. Philipp, M. S. Adlerstein, D. H. Goldberg, and Cauwenberghs, "Spike timing-dependent plasticity in the address domain," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003.

[7] D. Goldberg, G. Cauwenberghs, and A. Andreou, "Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons," *Neural Netw.*, vol. 14, no. 6–7, pp. 781–793, Sep. 2001.

[8] G. Indiveri, T. Horiuchi, E. Niebur, and R. Douglas, "A competitive network of spiking VLSI neurons," in *World Congress on Neuroinformatics*, ser. ARGESIM Rep. no. 20, F. Rattay, Ed. Vienna, Austria: ARGESIM/ASIM-Verlag, 2001, pp. 443–455.

[9] M. Mahowald, "VLSI analogs of neuronal visual processing: a synthesis of form and function," Ph.D. dissertation, Dep. Comput. Neur. Syst., California Institute of Technology, Pasadena, CA, 1992.

[10] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 523–528, May 1993.

[11] The address-event representation communication protocol AER 0.02, Caltech, Pasadena, CA, Internal Memo, Feb. 1993 [Online]. Available: http://www.ini.uzh.ch/~amw/scx/std002.pdf

[12] K. Boahen, "Communicating neuronal ensembles between neuromorphic chips," in *Neuromorphic Systems Engineering*, T. S. Lande, Ed. Norwell, MA: Kluwer Academic, 1998, pp. 229–259.

[13] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[14] K. A. Boahen, "A burst-mode word-serial address-event link—I: Transmitter design," *IEEE Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.

[15] Z. Kalayjian and A. Andreou, "Asynchronous communication of 2-D motion information using winner-takes-all arbitration," in *Neuromorphic Systems Engineering*, T. S. Lande, Ed. Norwell, MA: Kluwer Academic, 1998, pp. 217–227.

[16] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "Arbitrated address-event representation digital image sensor," *Electron. Lett.*, vol. 37, no. 24, pp. 1443–1445, Nov. 2001.

[17] P. Lichtsteiner, C. Posch, and T. Delbrück, "A 128 × 128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change," in *Dig. Tech. Papers IEEE ISSCC*, 2006, pp. 508–509.

[18] A. van Schaik and S.-C. Liu, "AER EAR: A matched silicon cochlea pair with address event representation interface," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, vol. V, pp. 4213–4216.

[19] G. Indiveri, A. Whatley, and J. Kramer, "A reconfigurable neuromorphic VLSI multichip system applied to visual motion computation," in *Proc. 7th Int. Conf. Microelectronics for Neural, Fuzzy and Bio-Inspired Systems; Microneuro'99*, Los Alamitos, CA, Apr. 1999, pp. 37–44.

[20] S. R. Deiss, R. J. Douglas, and A. M. Whatley, "A pulse-coded communications infrastructure for neuromorphic systems," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. Cambridge, MA: MIT Press, 1998, ch. 6, pp. 157–178.

[21] A. Mortara, E. Vittoz, and P. Venier, "A communication scheme for analog VLSI perceptive systems," *IEEE J. Solid-State Circuits*, vol. 30, no. 6, pp. 660–669, Jun. 1995.

[22] V. Dante, P. Del Giudice, and A. M. Whatley, "PCI-AER—hardware and software for interfacing to address-event based neuromorphic systems," *The Neuromorphic Engineer* vol. 2, no. 1, pp. 5–6, 2005 [Online]. Available: http://ine-web.org/research/newsletters/index.html

[23] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, H. K. Riis, T. Delbrück, S. C. Liu, S. Zahnd, A. M. Whatley, R. J. 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. Cambridge, MA: MIT Press, Dec. 2005, vol. 15.

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

[25] C. M. Higgins and C. Koch, "A modular multi-chip neuromorphic architecture for real-time visual motion processing," *Anal. Integr. Circuits Signal Process.*, vol. 24, pp. 195–211, 2000.

[26] P. Merolla and K. Boahen, "Dynamic computation in a recurrent network of heterogeneous silicon neurons," in *Proc. IEEE Int. Symp. Circuits Syst*, May 2006, pp. 4539–4542.

[27] F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miro, S. Vicente, G. Jimenez, and A. Civit, "AER tools for communications and debugging," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3253–3256.

[28] R. Berner, "High-speed USB2.0 AER interfaces," Diploma thesis, Dept. Elect. Eng., University of Züurich, ETH Züurich, and Universidad de Sevilla, Zurich, Switzerland, 2006.

[29] M. Oster, A. M. Whatley, S.-C. Liu, and R. J. Douglas, "A hardware/software framework for real-time spiking systems," in *Proc. 15th International Conf. Artificial Neural Netw.: Biological Inspirations (ICANN'05)*, , W. Duch, J. Kacprzyk, and E. Oja, Eds. *et al.*, Warsaw, Poland, Sep. 11–14, 2005, vol. 3696, pp. 161–166.

[30] R. J. Douglas and K. A. C. Martin, "Neural circuits of the neocortex," *Annu. Rev. Neurosci.*, vol. 27, pp. 419–451, 2004.

[31] R. Douglas, K. Martin, and D. Whitteridge, "A canonical microcircuit for neocortex," *Neural Comput.*, vol. 1, pp. 480–488, 1989.

[32] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.

[33] D. Ferster and K. D. Miller, "Neural mechanisms of orientation selectivity in the visual cortex," *Annu. Rev. Neurosci.*, vol. 23, pp. 441–471, 2000.

[34] H. Sompolinsky and R. Shapley, "New perspective on the mechanisms for orientation selectivity," *Current Opinion Neurobiol.*, vol. 7, pp. 514–522, 1997.

[35] T. W. Troyer, A. E. Krukowski, N. J. Priebe, and K. D. Miller, "Contrast-invariant orientation tuning in cat visual cortex: Thalamocortical input tuning and correlation-based intracortical connectivity," *J. Neurosci.*, vol. 18, no. 15, pp. 5908–5927, 1998.

[36] R. Shapley, M. Hawken, and D. L. Ringach, "Dynamics of orientation selectivity in the primary visual cortex and the importance of cortical inhibition," *Neuron*, vol. 38, pp. 689–699, 2003.

[37] D. C. Somers, S. B. Nelson, and M. Sur, "An emergent model of orientation selectivity in cat visual cortical simple cells," *J. Neurosci.*, vol. 15, pp. 5448–5465, 1995.

[38] R. Ben-Yishai, R. Lev Bar-Or, and H. Sompolinsky, "Theory of orientation tuning in visual cortex," *Proc. Nat. Acad. Sci.*, vol. 92, no. 9, pp. 3844–3848, Apr. 1995.

[39] R. J. Douglas, M. A. Mahowald, and K. A. C. Martin, "Hybrid analog-digital architectures for neuromorphic systems," in *Proc. IEEE World Congr. Comput. Intell.*, 1994, vol. 3, pp. 1848–1853.

[40] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER imager filtering architecture for vision-processing systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 9, pp. 1064–1071, Sep. 1999.

[41] G. Cauwenberghs and J. Waskiewicz, "Focal-plane analog VLSI cellular implementation of the boundary contour system," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 2, pp. 1064–1071, Feb. 1999.

[42] B. E. Shi, "A low-power orientation-selective vision sensor," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 435–440, May 2000.

[43] P. Venier, A. Mortara, X. Arreguit, and E. A. Vittoz, "An integrated cortical layer for orientation enhancement," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 177–186, Feb. 1997.

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

[45] T. Y. W. Choi, P. A. Merolla, J. V. Arthur, K. A. Boahen, and B. E. Shi, "Neuromorphic implementation of orientation hypercolumns," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1049–1060, Jun. 2005.

[46] K. Shimonomura and T. Yagi, "An orientation-selective multi-chip aVLSI applicable to texture analysis," in *Proc. Int. Joint Conf. Neural Netw.*, 2005, pp. 3267–3271.

[47] U. Mallik, R. J. Vogelstein, E. Culurciello, R. Etienne-Cummings, and G. Cauwenberghs, "A real-time spike-domain sensory information processing system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 3, pp. 1919–1922.

[48] S. R. Deiss, T. Delbrück, R. J. Douglas, M. Fischer, M. Mahowald, T. Matthews, and A. M. Whatley, Address-event asynchronous local broadcast protocol, Inst. Neuroinform., Zurich, Switzerland, 1994 [Online]. Available: http://www.ini.uzh.ch/~amw/scx/aeprotocol.html,

[49] A. M. Whatley, PCI-AER Board Driver, Library & Documentation, Inst. Neuroinform., Zurich, Switzerland, 2007 [Online]. Available: http://www.ini.uzh.ch/~amw/pciaer/

[50] AMCC, PCI Products data book, Section 2: S5920 PCI target interface, , Applied Micro Circuits Corporation, San Diego, CA, 1998 [Online]. Available: http://www.amcc.com

[51] R. Paz-Vicente, A. Linares-Barranco, D. Cascado, M. Rodriguez, G. Jimenez, A. Civit, and J. Sevillano, "PCI-AER interface for neuro-inspired spiking systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3253–3256.

[52] P. Lichtsteiner, T. Delbruck, and J. Kramer, "Improved ON/OFF temporally differentiating address-event imager," in *Proc. 11th IEEE Int. Conf. Electronics, Circuits Syst.*, Dec. 2004, pp. 211–214.

[53] E. Chicca, G. Indiveri, and R. J. Douglas, "An event based VLSI network of integrate-and-fire neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. V-357–V-360.

[54] M. Oster, "Tuning aVLSI chips with a mouse click," *Neuromorph. Eng.* vol. 2, no. 1, p. 9, 2005 [Online]. Available: http://ine-web.org/research/newsletters/index.html

[55] J. Kramer, "An integrated optical transient sensor," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 9, pp. 612–628, Sep. 2002.

[56] R. Hahnloser, R. Sarpeshkar, M. Mahowald, R. J. Douglas, and S. Seung, "Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

[57] E. Chicca, "A neuromorphic VLSI system for modeling spike-based cooperative competitive neural networks," Ph.D. dissertation, Dept. Phys., ETH Zürich, Zürich, Switzerland, Apr. 2006.

[58] N. V. Swindale, "Orientation tuning curves: Empirical description and estimation of parameters," *Biolog. Cybern.*, vol. 78, pp. 45–56, 1998.

[59] S. Mitra, S. Fusi, and G. Indiveri, "A VLSI spike-driven dynamic synapse which learns," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 2777–2780.

[60] C. Bartolozzi and G. Indiveri, "Silicon synaptic homeostasis," in *Proc. Brain Inspired Cogn. Syst.* , 2006, pp. 1–6.

[61] E. Chicca, P. Lichtsteiner, T. Delbruck, G. Indiveri, and R. J. Douglas, "Modeling orientation selectivity using a neuromorphic multichip system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 1235–1238.

[62] D. Muir, Spike Toolbox [Online]. Available: http://www.ini.uzh.ch/~dylan/spike_toolbox/ 2005

[63] G. Indiveri, "VLSI reconfigurable networks of integrate-and-fire neurons with spike-timing dependent plasticity," *Neuromorph. Eng.* vol. 2, no. 1, pp. 4–7, 2005 [Online]. Available: http://ine-web.org/research/newsletters/index.html

[64] C. Bartolozzi and G. Indiveri, "Selective attention implemented with dynamic synapses and integrate-and-fire neurons," *Neurocomput.*, vol. 69, pp. 1971–1976, 2005.

[65] M. Oster and S.-C. Liu, "Spiking inputs to a winner-take-all network," in *Advances in Neural Information Processing Systems (NIPS)*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, Dec. 2005, vol. 18, Neural Information Processing Systems Foundation, pp. 1051–1058.

Elisabetta Chicca (M'06) received the Laurea (equivalent to M.S.) degree in physics from the University of Rome La Sapienza, Rome, Italy, and the Ph.D. degree from the Swiss Federal Institute of Technology (ETHZ), Zurich, Switzerland, in 1999 and 2006, respectively.

She is currently a Postdoctoral Fellow at the Institute of Neuroinformatics, University of Zurich, Zurich, Switzerland, and UZH-ETHZ. Her research interests include neuromorphic analog VLSI models of cortical circuits and learning.

Dr. Chicca is currently a member of the IEEE BioCAS technical committee.

Adrian M. Whatley received the B.Sc. degree in chemistry from the University of Bristol, Bristol, U.K., in 1986.

After working for almost ten years in the British computer industry, he took up his current software engineering position at the Institute of Neuroinformatics, University of Zurich, Zurich, Switzerland, where he works primarily on address-event communication systems.

Patrick Lichtsteiner received the Diploma (equivalent to M.S.) degree in physics and the Ph.D. degree from the Swiss Federal Institute of Technology (UZH-ETHZ), Zurich, Switzerland, in 2002 and 2006, respectively.

He is currently working for the Institute of Neuroinformatics, University of Zurich, Zurich, Switzerland, and ETHZ. His research interests include CMOS imaging, neuromorphic vision sensors and high-speed vision.

Vittorio Dante received the electronics diploma from ITIS Enrico Fermi, Rome, Italy, in 1983.

He has been an Electronics Designer working on microcontrollers and embedded systems (from 1980 to 1990). Since 1990, he has been with the Physics Laboratory, Italian National Institutes of Health, Rome Italy. In the framework of Italian National Institute for Nuclear Physics (INFN) projects, he worked on systems based on VME, PCI, and FB+ buses. Since 1997, he has been involved in research on efficient communication in neuromorphic multichip systems.

Tobias Delbruck (M'99–SM'06) received the B.S. degree in physics and applied mathematics and the Ph.D. degree in computation and neural systems from California Institute of Technology (Caltech), Pasadena, CA, in 1993.

He worked for several years for Arithmos, Synaptics, National Semiconductor, and Foveon, where he was one of the Founding Employees. In 1998 he moved to Switzerland to join the Institute of Neuroinformatics (INI), University of Zurich, Zurich, Switzerland, where he is currently a Group Leader. In 2002, he was a Lead Developer of the tactile luminous floor used in INI's exhibit "Ada: Playful Intelligent Space." He is currently also a Visiting Scientist at Caltech. His main interest is in developing neuromorphic electronics, particularly vision sensor chips. Prior to working for Arithmos, he co-invented the standard neuromorphic adaptive photoreceptor circuit and bump circuit. He has been awarded eight patents, and has over 30 refereed papers in journals and conferences, 4 book chapters, and 1 book.

Dr. Delbruck and his students have been awarded 4 prizes for IEEE conference papers, including the 2006 ISSCC Best European Paper Award.

Rodney J. Douglas received degrees in science and medicine and the Ph.D. degree in neuroscience from the University of Cape Town, Cape Town, South Africa.

He joined the Anatomical Neuropharmacology Unit, Oxford, U.K., where he continued his research on the anatomy and biophysics of neocortical circuits together with K. Martin. He is currently a Professor of neuroinformatics and Co-Director of the Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland, and the University of Zurich (UZH-ETHZ), Zurich, Switzerland. He was a Visiting Associate, and then a Visiting Professor, at the California Institute of Technology, Pasadena, where he extended his research interests to neural computation to the modeling of cortical circuits using digital methods (together with C. Koch), and also by the fabrication of analog VLSI circuits (together with M. Mahowald). In 1996, he and K. Martin joined the UZH-ETHZ, to establish the Institute of Neuroinformatics.

Dr. Douglas was awarded the Korber Foundation Prize for European Science in 2000.

Paolo Del Giudice received the graduate degree in physics from the Rome University "La Sapienza," Rome, Italy, in 1985.

Since 1991, he has been with at the Complex Systems Unit, Technologies and Health Department, Italian National Institutes of Health, Rome Italy. He is also with the he National Institute of Nuclear Physics, Rome, Italy. He has been mostly active in the theory, simulation and electronic implementation of neural network models, recently focusing in particular of the collective stochastic dynamics of spiking neurons and neuromorphic multichip systems. He also worked on computational problems in radiotherapy, and the statistical analysis of DNA sequences.

Giacomo Indiveri (SM'06) received the graduate and Ph.D. degrees in electronic engineering from the University of Genova, Genova, Italy.

He is an Assistant Professor at the Institute of Neuroinformatics (INI), University of Zurich, Zurich, Switzerland. He was awarded a Fellowship for the Italian "National Research Program on Bioelectronic technologies" (completed *cum laude* in 1995), and worked as a Post-Doctoral Fellowship at the California Institute of Technology, Pasadena, from 1994 to 1996. His main interests are in the design and fabrication of neuromorphic electronic circuits that implement cortical processing and in the construction of spike-based learning, winner-take-all architectures, and hardware attentional systems.

Dr. Indiveri is member of numerous IEEE technical committees.