Spikes Monitors for FPGAs, an Experimental Comparative Study

Elena Cerezuela-Escudero, Manuel Jesus Dominguez-Morales,
Angel Jiménez-Fernández, Rafael Paz-Vicente,
Alejandro Linares-Barranco, and Gabriel Jiménez-Moreno

Departamento de Arquitectura y Tecnología de Computadores,
ETS Ingeniería Informática - Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
ecerezuela@atc.us.es

Abstract. In this paper we present and analyze two VHDL components for monitoring internal activity of spikes fired by silicon neurons inside FPGAs. These spikes monitors encode each spike according to the Address-Event Representation, sending them through a time multiplexed digital bus as discrete events, using different strategies. In order to study and analyze their behavior we have designed an experimental scenario, where diverse AER systems have been used to stimulate the spikes monitors and collect the output AER events, for later analysis. We have applied a battery of tests on both monitors in order to measure diverse features such as maximum spike load and AER event loss due to collisions.

Keywords: spiking neurons, monitoring spikes, Address-Event Representation, Field Programmable Gate Array, inter-chip communication.

1 Introduction

Neuromorphic systems provide a high level of parallelism, interconnectivity, and scalability; doing complex processing in real time, with a good relation between quality, speed and resource consumption. Neuromorphic engineers work in the study, design and development of neuro-inspired systems, like aVLSI (analog VLSI) chips for sensors [1][2], neuro-inspired processing, filtering or learning [3][4][5][6], neuro-inspired control pattern generators (CPG), neuro-inspired robotics [7][8][11] and so on. Spiking systems are neural models that mimic the neurons layers of the brain for processing purposes. Signals in spikes-domains are composed of short pulses in time, called spikes. Information is carried by spikes, and it is measured in spike frequency or rate [9], following a Pulse Frequency Modulation (PFM) scheme, and also from another point of view, in the inter-spike-time (ISI) [5]. If we have several layers with hundreds or thousands of neurons, it turns very difficult to use a point to multiple-point connection among neurons along the chips that implement different neuronal

* This work has been supported by the Spanish government grant project VULCANO (TEC2009-10639-C04-02) and BIOSENSE (TEC2012-37868-C04-02).

layers. This problem is solved thanks to the introduction of the Address-Event Representation (AER), proposed by Mead lab in 1991, facing this problem using a common digital bus multiplexed in time, the AER bus. The idea is to give a digital unique code (address) to each neuron. Whenever a neuron fires a spike a circuit should take note of it, manage the possible collisions with other simultaneously fired spikes and, finally, encode it as an event with its pre-assigned address. This event will be transferred through the AER bus, which uses additional control lines of request (REQ) and acknowledge (ACK), implementing a 4-phase asynchronous hand-shake protocol. In the receiver, neurons will be listening to the bus, looking for the spikes sent to them [10]. Using the AER codification, neurons are virtually connected by streams of spikes.

This work is focused on presenting in a detailed way two spikes monitors, written in VHDL for FPGAs, which encode each spike according to the Address-Event Representation, using different strategies in order to avoid spike collisions in time. Temporal spike collision is known as the situation where two or more spikes have been fired at the same time, and they should be sent using the AER bus. Fig. 1 shows the typical application of spikes monitors, where there are circuits that fire spikes (e.g. a set of spiking neurons), and there is a spikes monitor connected to them , which will encode the spikes and send them to another layer, using the AER bus.

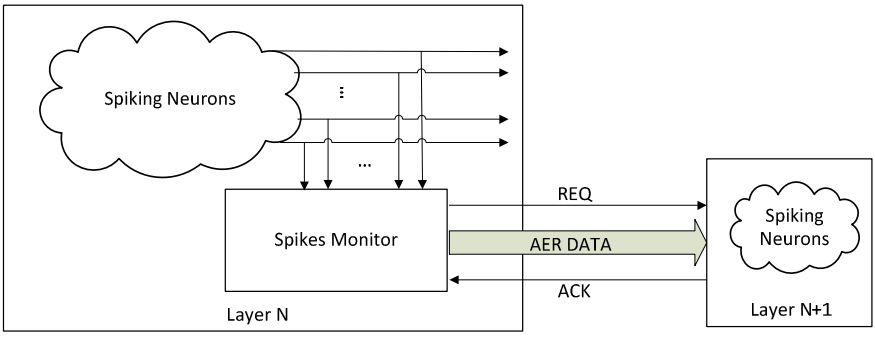

Fig. 1. Typical application of spikes monitor in a multilayer AER system

2 Spikes Monitors Description

We have implemented two spikes monitors, as mentioned before; these monitors implement different strategies to manage the spike collisions in time, being this an important concept, because it is the main difficulty in this kind of system. In an ideal scenario, where spikes are fired neuron by neuron, sequentially, without temporal collisions, spikes encoding as AER events will be automatic using a traditional digital encoder. However, when two or more spikes are fired simultaneously, two or more AER events should be transferred, but the AER bus is unique and multiplexed in time. In consequence, spikes that have been fired in parallel will be transmitted as AER events sequentially. Different strategies can be used to implement this functionality;

the results and monitor behavior will depend directly on the strategy adopted, and both monitors are generics and can be adapted to variable input spike number.

2.1 Massive Spikes Monitor(MSM)

MSM needs three blocks: the first block is used to avoid collisions taking a snapshot of the spikes activity every clock cycle; the second block to encode the spike with its address; and the third block sends the address with the hand-shake protocol. Fig. 2 shows the block diagram. In order to avoid collisions, MSM takes a snapshot of spikes and stores it in a FIFO (Spikes FIFO) if some spike has been fired (Fig. 2 top). If there are many '1's in a single word it means that more than one spike has been fired at the same time. Now we need to encode every spike with its address. We have designed a Finite State Machine (FSM) which, if the Spikes FIFO is not empty, loads a word into a register and looks for a spike bit by bit. If it finds one, it looks for its address in a ROM and writes the address in another FIFO (AER FIFO) to be transferred as an AER event (Fig. 2 middle). The AER FIFO contains the encoded spikes addresses, and they are ready to be sent through the AER port. Finally there is included another FSM for the 4-phase AER handshaking (Fig.2 right) [11].

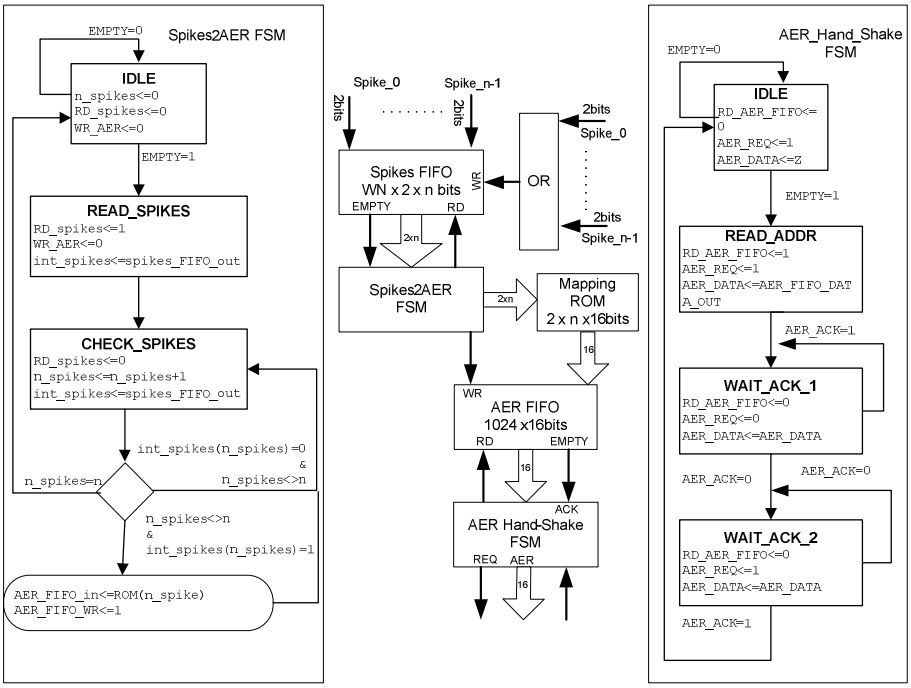

Fig. 2. MSM internal structure and FSM descriptions

The main problem of MSM resides in the fact that all spikes are stored in a single word, and when this word is relatively big MSM demands a high memory quantity for Spikes FIFO in synthesis time and a great number of clock cycles to search for spikes, introducing a high latency in spikes encoding, and consequently losing a high rate of spikes.

2.2 Distributed Spikes Monitor (DSM)

The DSM aims to avoid the problem of MSM, breaking the Spikes FIFO and encoding FSM into several identical sub-circuits, which distributes the task of spikes encoding in different FIFOs and a FSM that can now work in parallel. The DSM distributes spikes in four similar modules; therefore a quarter of the input spikes excite each module, which is shown in Fig. 3. Each module stores its spike portion in a register and looks for a spike bit by bit. If it finds one spike, it works out its partial AER address by the index on the register. Then, the module stores this partial address in a FIFO. Now we need to encode every spike with its complete address. We have designed a FSM which computes the full address from the partial address and empty signals. Finally, the monitor writes the address in the AER FIFO (Fig. 3 bottom). This contains the spikes addresses as events, and they are ready to be sent through the AER port.

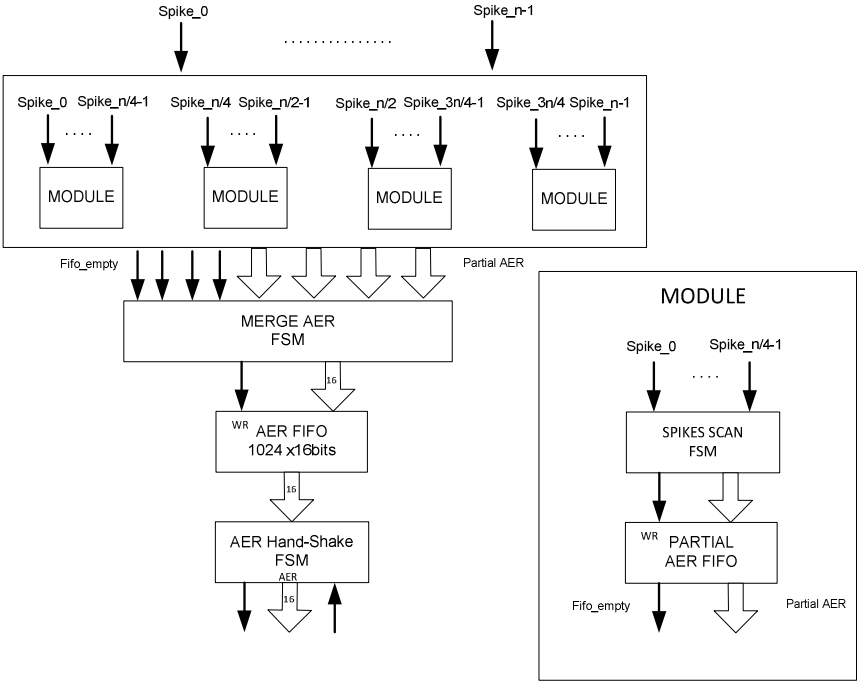

Fig. 3. Internal Structure of the DSM

3 Experimental Setup

In this work we want to study and analyze the monitors' behavior. In order to achieve this, we have designed an experimental scenario, where diverse AER systems have been used to stimulate the spikes monitors and to collect the output AER events, for later analysis. The experiment components are (Fig. 4):

1. First, a PC generates a test battery of spikes using MATLAB.
2. The PC sends the spikes information to an USB-AER board through USB interface. The USB-AER board stores the spikes in its RAM memory. We have implemented a component VHDL to manage RAM memory. These spikes are used to stimulate the spikes monitors.
3. At the end, the USB-AERmini2 receives the monitor outputs and sends them to the PC for later analysis.

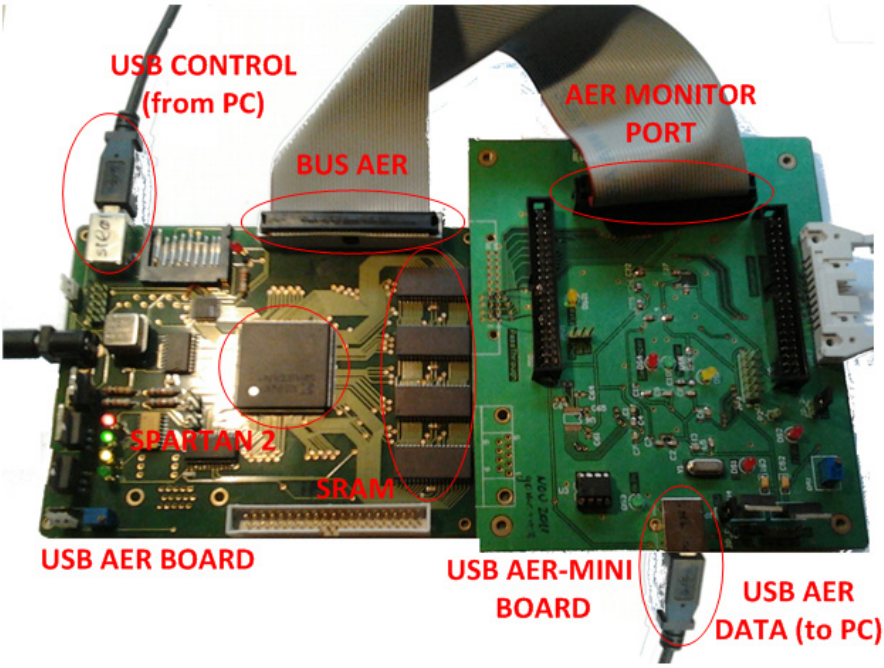

Fig. 4. Experiment Components

3.1 USB-AER and USB-AERmini2 Boards

We used an USB-AER board to load the MSM and DSM with the VHDL stimulus builder component. USB-AER board is based on a Xilinx Spartan II-200 FPGA that

can be reconfigured using the USB interface provided by SiliconLabs 8051 microcontroller or SD card. This board includes two AER parallel ports (input and output) and 2 Mbytes of static RAM (SRAM) [12].

We used the USB-AERmini2 board in order to monitor the AER traffic in a PC. This device allows monitoring and sequencing AER events with a time resolution of 200 nanoseconds. The device consists of a Cypress FX2LP microcontroller and a Xilinx Cool runner 2 CPLD. The CPLD is clocked with 30 MHz and achieves a peak monitor rate of 6 Megaevents per second and a sustained rate of 5 Megaevents per second, which is limited by the host computer. This board provides the captured AER events and the time instant at which they have been fired (time stamp) [13].

3.2 Stimulating the Monitors: Generating Spikes and Processing AER Events

The USB-AER board receives the spikes information from the USB interface (Fig. 5-1) and stores them in the SRAM memory, using a component which manages the communication between the USB and the SRAM (Fig.5-2). Then, the system reads SRAM (Fig.5-3) and uses these spikes to stimulate the MSM and DSM (Fig.5-4). The output AER events are sent to the USB-AERmini2 by the AER output parallel port.

We have implemented a MATLAB function which generates random spikes from set input parameters, such as the number of maximum active spikes in time instant, and the probability of this to happen. Being these spikes used to stimulate MSM/DSM inputs, the PC receives the MSM/DSM outputs by USB-AERmini2 and analyzes them.

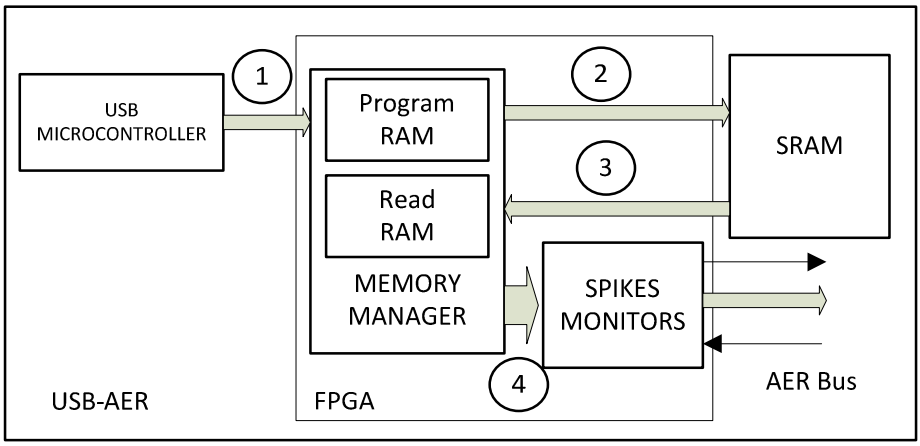

Fig. 5. Experiment Execution

We have designed the battery of tests from equation 1 which calculates the parameters to obtain a particular average spike rate.

4 Experimental Results

In order to characterize the monitors´ behavior we have excited both monitors with diverse stimulus inputs, creating a sweep of stimulus spike rate and changing the number of simultaneously fired spikes using equation 1. The spike rate generated changed from 2 to 20 MSpkes/Sec, and the number of simultaneous spikes from 8 to 16 spikes.

$$Avg.\,Events\ Rate = \frac{Number\ of\ Active\ Spikes * Probability}{Total\ Stimulus\ Time}\left(Events/_{Sec}\right) \quad (1)$$

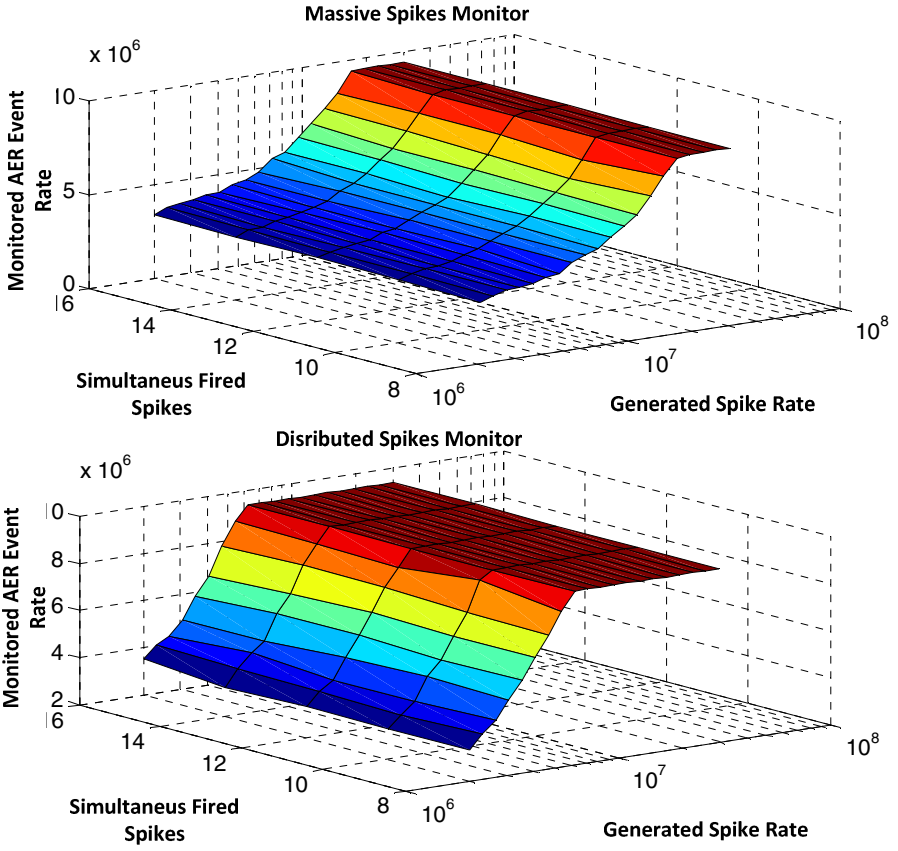

Fig. 6. MSM and DSM output AER events rate

The first measurement done was the average AER events rate monitored for every test case; Fig. 6 shows the results for both MSM (top) and DSM (bottom). Due to the

structure of the MSM, many spikes are lost, and for example, when it is excited with 10MSpikes/Sec, it only provides about 6 MEvents/Sec. However, it needs about 8 MSpikes/Sec to reach an AER event rate of 9.8 MEvents/Sec, being this the maximum capacity of AER events monitoring for the USB-AERmini2 board, saturating in consequence the AER bus. Opposite to the MSM, the DSM shows a better behavior, providing at its output an AER event rate very similar to the input spikes, saturating the AER bus when it is excited with 10 MEvents. In both cases, the number of simultaneously fired spikes does not affect significantly the spikes monitors performance.

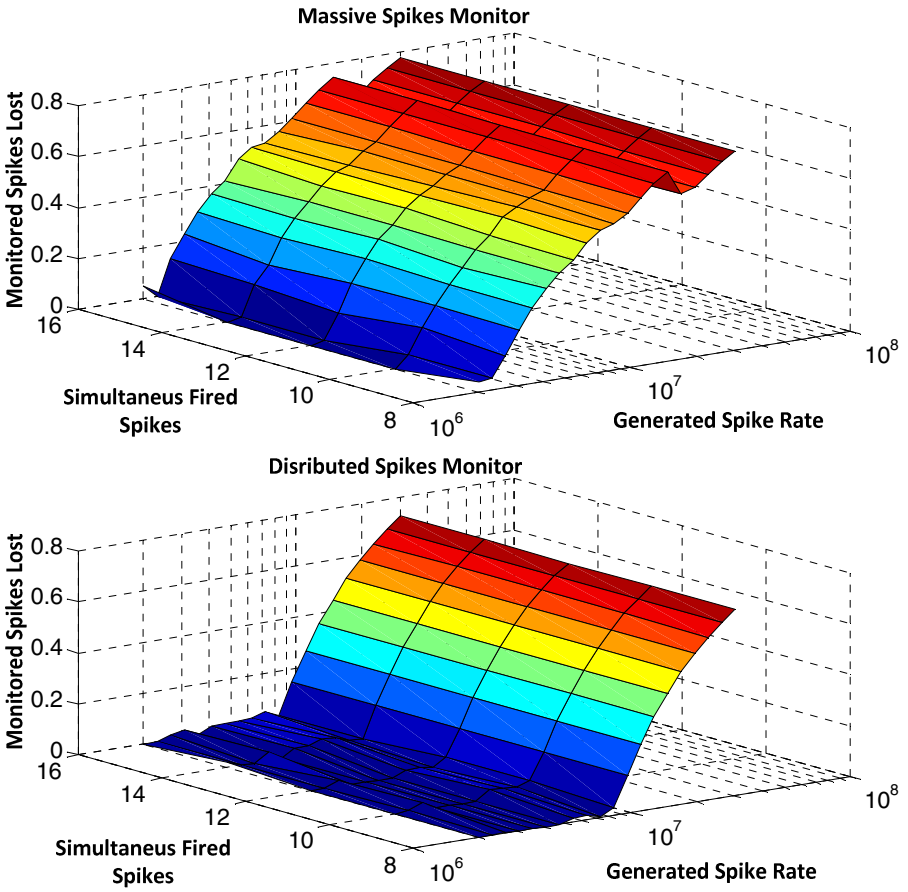

Fig. 7. MSM and DSM spikes loss

Next, we have measured the ratio of spikes lost in the same conditions as in the previous measurement. Fig. 7 contains the spikes loss ratio for the MSM (top) and DSM (bottom). For the MSM, there are a low number of spikes lost for the lower spike rate values, only being discarded a few of them at 2-3MSpikes/Sec. However,

when the stimulus spike rate is increased, the MSM Spikes FIFO is full very soon, and many spikes are discarded.

One more time, the DSM presents a better response, losing a very small quantity of spikes, thanks to its FIFOs and FSM distribution, and it only stars losing a considerable amount of spikes when it is excited with more than 10 MSpikes/Sec. However, this is the AER bus maximum reachable event rate using the USB-AERmini2 board, which starts discarding AER events.

After these experiments, the DSM denotes a better behavior than the MSM, being very adequate for this kind of system since it provides a higher bandwidth, in terms of AER events, which can be transferred using the AER bus, being this now the communication bottleneck.

5 Conclusions

In this work we want to study and analyze the behavior of two different spikes monitors' with spiking neurons. We have designed an experimental scenario, where diverse AER systems have been connected together, building a test infrastructure for stimulating MSM and DSM, and collect their AER information for later analysis. Finally, we have analyzed MSM and DSM responses in terms of output AER events rate, and the ratio of spike loss.

The DSM has shown better behavior, providing a higher AER events rate than the MSM, having a low spike loss and reaching a higher AER events rate than the AER bus capacity.

References

1. Lichtsteiner, P., et al.: A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. IEEE Journal on Solid-State Circuits 43(2) (2008)
2. Chan, V., et al.: AER EAR: A Matched Silicon Cochlea Pair With Address Event Representation Interface. IEEE T. Circuits and Systems I 54(1) (2007)
3. Serrano-Gotarredona, R., et al.: On Real-Time AER 2-D Convolutions Hardware for Neuromorphic Spike-Based Cortical Processing. IEEE T. Neural Network 19(7) (2008)
4. Oster, M., et al.: Quantifying Input and Output Spike Statistics of a Winner-Take-All Network in a Vision System. In: IEEE International Symposium on Circuits and Systems, ISCAS 2007 (2007)
5. Hafliger, P.: Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip. IEEE T. Neural Networks 18(2) (2007)
6. Indiveri, G., et al.: A VLSI Array of Low-Power Spiking Neurons and Bistables Synapses with Spike-Timig Dependant Plasticity. IEEE T. Neural Networks 17(1) (2006)
7. Linares-Barranco, A., et al.: Using FPGA for visuo-motor control with a silicon retina and a humanoid robot. In: IEEE International Symposium on Circuits and Systemsm, ISCAS 2007 (2007)
8. Jiménez-Fernández, A., et al.: AER and dynamic systems co-simulation over Simulink with Xilinx System Generator. In: IEEE I. Conference on Electronic, Circuits and Systems, ICECS 2008 (2008)

9. Shepherd, G.: The Synaptic Organization of the Brain. Oxford University Press (1990)
10. Boahen, K.: Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events. IEEE T. Circuits and Systems II 47(5) (2000)
11. Jiménez-Fernández, A., et al.: Spike-based control monitoring and analysis with Address Event Representation. In: IEEE Int. Conference on Computer Systems and Applications, AICCSA 2009 (2009)
12. Gómez-Rodríguez, F., et al.: AER tools for communications and debugging. In: Proceedings of IEEE Int. Sym. on Circuits and Systems, ISCAS 2006 (2006)
13. Berner, R., et al.: A 5 Meps $100 USB2.0 Address-Event Monitor-Sequencer Interface. In: IEEE International Symposium on Circuits and Systems, ISCAS 2007 (2007)