

# Synthetic retina for AER systems development

R. Paz-Vicente, A. Linares-Barranco, A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Civit-Balcells

Dpt. Computer Architecture and Technology  
E.T.S. Ingeniería Informática. Universidad de Sevilla  
Sevilla, Spain  
[rpaz@atc.us.es](mailto:rpaz@atc.us.es)

**Abstract**—Neuromorphic engineering tries to mimic biology in information processing. Address-Event Representation (AER) is a neuromorphic communication protocol for spiking neurons between different layers. AER bio-inspired image sensor are called “retina”. This kind of sensors measure visual information not based on frames from real life and generates corresponding events. In this paper we provide an alternative, based on cheap FPGA, to this image sensors that takes images provided by an analog video source (video composite signal), digitalizes it and generates AER streams for testing purposes.

## I. INTRODUCTION

Biology provides efficient solutions for several problems. Trying to mimic biology, bio-inspired and neuro-inspired systems have been extended in the last years. High degree of parallelism and scalability makes this emerging technology of computing especially interesting for real time vision processing.

One of the neuro-inspired models which mimic the neuron layers in the brain are the spiking models. These models can use the AER protocol to communicate spikes between different layers. There is a growing community of AER system developers. We can find several sensors (cochleas, retinas...)[1][2][3], processing layers (filters, convolutions, WTA...)[4][5][6][7], robotic controlling (Central Pattern Generators, Eddie...)[8][9], computer interface for system monitoring and sequencing (PCI-AER, USB-AER, USBAERmini2...)[10][11], and computer software for real-time processing (jAER, MATLAB interface)[12][13]. The goal of this community is to build large multichip and multi-layer hierarchically structured systems, for real-time massively-parallel spike processing. One of the most extended fields is the vision processing using an AER chain for objects recognition, tracking, etc...[14][15][16] Figure 1 shows the biggest AER chain developed for vision processing under the EU CAVIAR project, composed by a temporal contrast retina, 2 convolution chips, 1 WTA filter chip and one learning chip, connected using mapper boards and inserting AER monitors for debugging purpose. Figure 3 shows synthetic retina used as first element in AER chains.

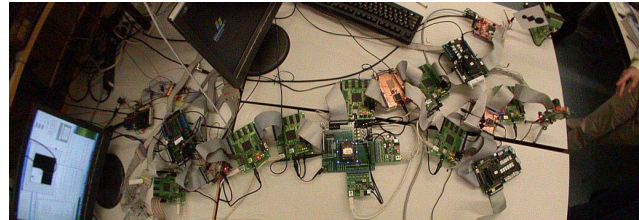


Figure 1. An AER chain for vision processing.

Spiking neurons codifies the information in a stream of pulses (spikes), whose frequency (or spike rate) is proportional to the neuron’s excitation, following a Pulse Frequency Modulation (PFM). AER was proposed by Mead lab in 1991[17], as we have said before, for communicating neuromorphic chips with spikes. If we have several layers with hundred or thousand neurons, became impossible to use a point to multi-point connection between every layer. AER tries to avoid this problem. The idea of the AER is to have only a common bus, the AER bus, giving a digital code address to every neuron. Every time a neuron fires a spike, it goes to an arbiter that manages collisions [15], and to an encoder, who writes the address in the AER bus (figure 2). Additional lines of request (REQ) and acknowledge (ACK) are required to communicate the address using a 4-step asynchronous hand-shake protocol. In the receiver, neurons will be listening the bus, looking for the spikes sent to them. Neurons are virtually connected by streams of spikes.

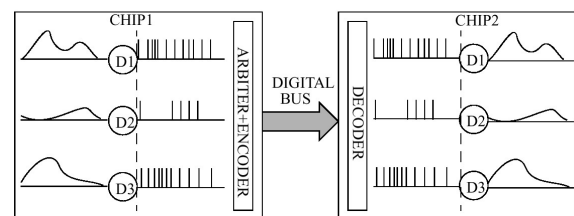


Figure 2. Spiking neurons communication between two layers using AER.

First layer in an AER image processing chain is an AER image sensor, also called retina that captures images from real life and generates AER event describing them. There are several working ASIC AER retinas [1][2] that can be used for this purpose. Some of these retinas provide events describing pixel luminosity, but others send only luminosity changes information. This last group is called derivative retinas. Independently of the kind of retina we are using, both of them are implement in ASIC CMOS chips, so high knowledge of chip designing is required, many different polarization voltages are used, they have very complex biases to adjust, and due to expensive of custom chip prototyping, this technology is out of the possibilities of some developing groups, or at least the access to this devices can be very limited.

To popularize AER bus and neuro-inspired spiking systems, cheap image sensor that can be widely available, may be very interesting because that makes possible the developing of AER filters, testing and debugging them, makes this technology available for a more extensive community.

In this paper we present a synthetic retina implementation for FPGA using frame-based video source with a frame to AER conversion. Proposed synthetic retina try to provide this AER developing tool, working in a very similar way of real AER retinas, but at a lower cost. For that, most of the implementation of the synthetic retina consists in a digital circuit that is implemented in VHDL to be synthesized to fit on a CPLD or FPGA programmable digital device.

Image sensor consists in a cheap CMOS or CCD camera that output video composite signal. Alternatively, signal provided by a Video Recorder, DVD player, or computer output can be used. These video composite signals are captured using an ADC converter, and digitalized information is processed by digital logic to convert this video signal to AER events.

## II. USB-AER BOARD

Board used in retina implementation [10] is based on a Xilinx Spartan II-200 FPGA that can be reconfigured changing VHDL design loaded to this board using USB conector or SD card (figure 3).

The main features of this board are:

- Spartan 2 FPGA (XC2S200) clocked at 50MHz (up to 100Mhz clock using internal DLL)
- 24-MHz 8051 microcontroller from Sillicon Laboratories (C8051F320/342).
- 2 AER parallel ports (input and output one).
- 2Mbytes 10ns Static RAM configured as four bytes wide (512Kx32). Independent write signal for each RAM Bank.

- USB 2.0 interface provided by microcontroller.
- SD-Card slot for FPGA and RAM boot content loading.

Input AER port is used to connect to parallel ADC board. Output port is connected to next AER layer boards. External RAM chips are used in derivative retina, because internal FPGA RAM blocks (7Kbytes) is not enough for storing both frame buffers.

USB connection is not used in this experiment. Firmware of FPGA is loaded on SD card so this board works as a stand-alone board, with no PC connection required.

## III. SHYNTETIC RETINA (FRAME TO AER CONVERSION)

Due to the increasing of the AER community that develops AER devices to perform filters and transformation of AER images, some devices that generates AER events are needed in order to provide a stimulus to these filters.



Figure 3. Synthetic retina boards

In figure3, a USB-AER board (green board) is connected (input port) to an ADC board (copper) with video in connector, attached to a small B&W video camera.

### A. Video Composite Capture.

First step in frames to AER conversion rely on video capture sampling video composite signal provided by a camcorder or video camera. This camera should provide video composite signal. Like current AER filters works with monochrome images, and previously cited retinas are also monochromes, we are not going to process color signals.

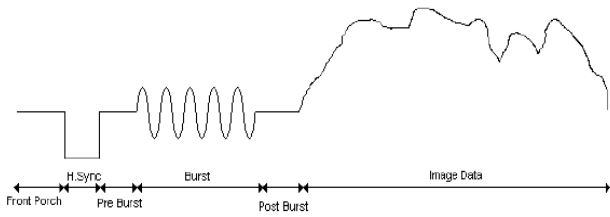


Figure 4. Video composite signal sample

Video composite signal modulates color in a subcarrier of 3.5795 Mhz for NTSC color system and 4.4297Mhz for PAL. Like synthetic retina proposed here is not going to use color information provided in this subcarrier (phase+amplitude modulation), only luminosity component is processed. Low pass filter below subcarrier frequency can be used to extract luminosity information (Y channel) and remove color information (U and V channels for PAL, I and Q for NTSC). Burst pulses, used for color subcarrier oscillator synchronization, can also be ignored. In B&W video signal, Burst is removed and there is a gap of about 7 us between horizontal sync section and image data.

For image capturing, a fast ADC manufactured by Analog Devices, ADC08100 is used. This ADC works with a clock signal between 20Mhz and 100Mhz providing up to 100MSPS (million of samples per second), delayed three clock cycles. It outputs digital data as an eight bits parallel word that is connected to input pins of FPGA. Clock signal provided to ADC is obtained from internal FPGA clock, dividing 50 Mhz clock signal by two to obtain an ADC clock of 25Mhz. This sample rate is enough to sample both, color and BW signal. A sample rate of about 4Mhz should be enough to sample BW signals, but ADC 08100 doesn't work below 20Mhz minimum frequency.

To remove color subcarrier external LC filter can be used to filter it, or a simple FIR (Finite Impulse Response) or IIR (Infinite Impulse Response) digital filter can be implemented inside FPGA to digitally filter it. Oversampling the video signal at 25Mhz can be useful to perform mentioned digital filtering. For reducing FPGA usage, implementing FIR or IIR filter requires multiplication computation. Calculating multiplication need to implement this filter requires a big FPGA usage. Due to limited amount of resources contained in SPARTAN II FPGA series used in USB-AER board [10], external LC filter are preferred and extra samples are simply discarded.

*B. Intensity retina and derivative retina.*

Once the current frame is captured and stored in internal memory, the AER event generator uses this data stored in memory to scan it and generate events properly. In current AER state of the art, two different kind of retina has been developed: intensity and derivative one. Main difference

between them is that in "intensity" retina, AER events describe light intensity information, so event frequency for a given pixel is proportional to the amount of light that this pixel of the sensor receives.

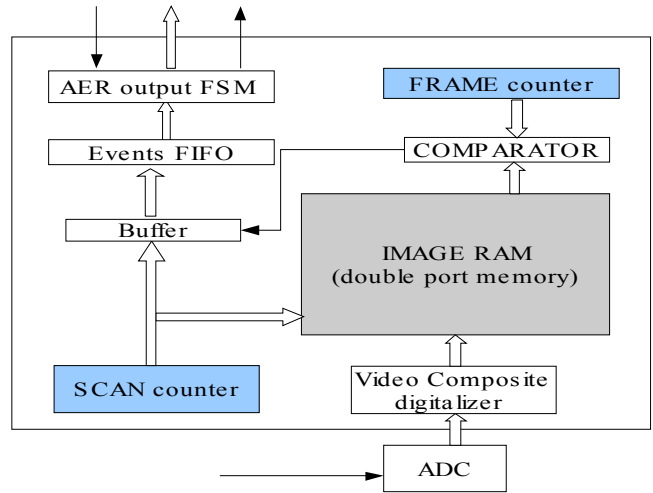


Figure 5. Synthetic retina block diagram

Figure 5 shows the block diagram for intensity synthetic retina. "Scan counter" is divided in two halves, row counter (less significant bits) and column counter (most significant bits). Content of scan counter is used to access image RAM and get pixel value. Data read from RAM is compared with frame counter to determine if event address has to be sent. If comparison results positive, it enables that the value of scan counter used for pixel addressing, is now used as event address and it is loaded to FIFO in order to be sent through AER output finite states machine.

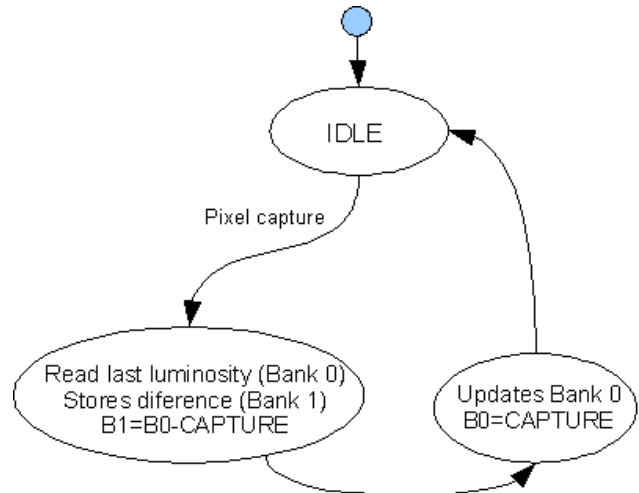


Figure 6. Synthetic retina hardware implementation state machine.

Frame counter is increased every time scan counter overflows, so they both can be implemented together, placing frame counter as the top most significant part in the combined counter. Frame counter bits are position reversed prior to comparison, as described later to enhance event time distribution.

Video composite capturer uses horizontal and vertical sync pulses to maintain capture line counter synchronized to current video line transmitted on video composite signal. Horizontal scan timer is synchronized with horizontal sync pulses to correctly temporize capture instants. It is implemented as another finite states machine. This states machine is shown in figure 6

In “derivative” retinas, event frequency is not proportional the amount of light received, but the difference of light between current captured frame and last one, so event frequency is proportional to the change of luminosity. Memory requirements for this kind of retina is twice than for intensity one because it is necessary to store two different frame information, current one and previous one so we can subtract them. Alternatively, it is also possible to implement derivative retina storing last frame and the difference between it and current. Difference frame is used to generate AER events, and last frame intensity information is used to calculate difference frame when next frame is going to be captured.

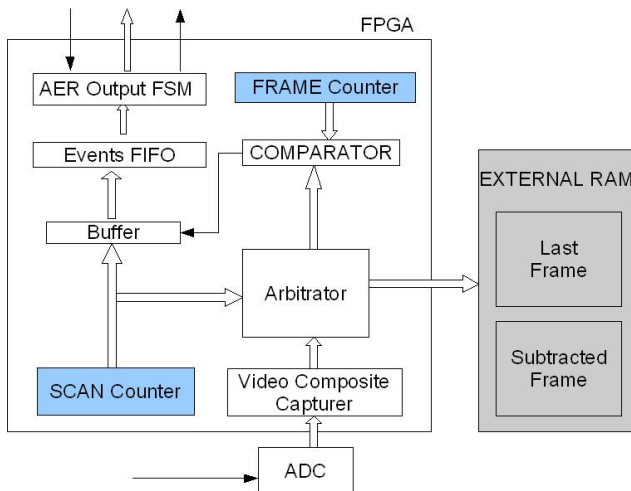


Figure 7. Derivative retina block diagram.

In both case, as mentioned, memory requirements doubles requirements of intensity retina. Other difference that is interesting to take in consideration is that the result of subtracting two frame can have positive or negative sign, depending on pixel luminosity change: from darker to lighter, or opposite. In order to correctly represent sign of subtraction when events are generated, event frequency is proportional to absolute value of subtraction, but event address is different if sign is positive or negative.

Derivative retinas can be useful because the AER traffic generated is lower than brightness retina. Images with no variation or small variations between frames generate low event activity. Only those parts of the images that changes generates AER events. This can be used to easily track moving objects in a fixed background.

In figure 7 is shown the block diagram for synthetic derivative retina. As derivative retina needs more RAM memory, internal FPGA RAM blocks is not enough, so external memory is used. For external RAM, only one access bus is available (are implemented using fast static one port RAM chips), so an arbiter is needed to manage access to RAM from video composite capturer and AER event generator. The number of RAM access from event generator are several times bigger than access from video capturer, so usually RAM chips is assigned to it, except when a new pixel is captured, that momentary interrupts the generator.

### C. AER event generation

Once we have digitalized a frame, AER event generation consists in reading this frame information from memory and to generate events, in a manner that event frequency for each pixel is proportional to stored value, and the time distance between events of a fixed pixel tries to be homogenously distributed.

As shown in figure 7, external RAM is divided in two different frame buffers, one storing last frame information, and the other that stores image difference between last frame and previous one. Choosing which frame is used to feed the AER generation, we obtain a Brightness or Derivative Retina.

To generate AER events, exhaustive method has been implemented [19]. A binary counter is used to scan captured frames and generate events for each pixel if comparison between pixel value and slice counter gives that slice counter is below pixel value. Slice counter divides a frame time in slices. To transmit pixel information of 255 (maximum allowed value), 255 events needs to be transferred, and so all slots are occupied. A pixel with a value of 100 only needs to transfer 100 events of the 255 given slots. Bit order of the slice counter is reversed to avoid event concentration in the first slices of each frame. Table 1 illustrates event distribution for different pixel values of a maximum of 8 gray levels allowed.

	Slice counter							
Gray	0	1	2	3	4	5	6	7
0	0	4	2	6	1	5	3	7
1	0	4	2	6	1	5	3	7
2	0	4	2	6	1	5	3	7
3	0	4	2	6	1	5	3	7
4	0	4	2	6	1	5	3	7
5	0	4	2	6	1	5	3	7
6	0	4	2	6	1	5	3	7
7	0	4	2	6	1	5	3	7

Figure 8. Table 1 Slice counter.



AER generation method can be modified in order to test other method of Frame-to-AER conversion [24][26]. For example, scan counter and slice counter can be substituted by a LFSR to implement a Random method.

Finally, a FIFO memory is used to store events that has to be sent in order to make independent frame scan (event generation) and event transmission. Event transmission can be delayed because of REQ/ACK protocol when a slow receiver is used. This way one module implemented as a finite state machine manage AER protocol between retina and receiver, meanwhile other finite state machine scans frame and generate AER events, that are stored in the FIFO. When the FIFO is full, frame scan stalls, to avoid missing events.

#### IV. EXPERIMENTAL CAPTURES

In this experiment two moving object (black circles) glued to a bigger white circle (used as background) attached to a controllable speed electric motor is used in order to compare reconstructed image using a real retina[1] and synthesized one. Following picture (figure 9) shows experimental setup.

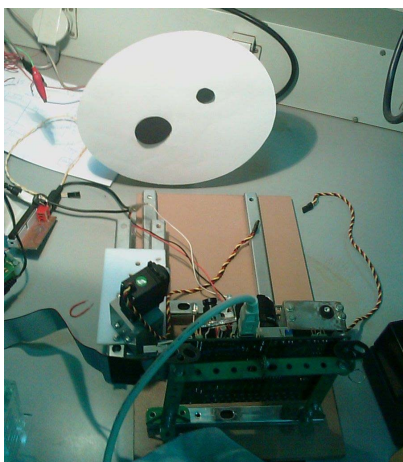


Figure 9. Synthetic retina capture scenery

It is difficult to quantitatively compare two different image sensors because there are many variables that can influence the results. First one is sensor sensibility. Sensibility of sensor and lighting condition affects observed object contrast and scene noise. In video composite cameras, logic controls shutter speed, so sensor tries to automatically accommodate to ambient light. Slow shutter speeds affect maximum moving speed of observed objects. Depending on sensor quality and ambient light, quality of reconstructed image may vary, and some added noise can appear.



Figure 10. Synthetic retina capture

Optic configuration used has influence in observed object size and field of view. Making two different captures from a fixed place, at a fixed distance and relative position, we can obtain different size and distance between observed objects. Object size affect the number of affected neurons, and therefore bus traffic. Relative position differences affect event's address appearing on the bus.

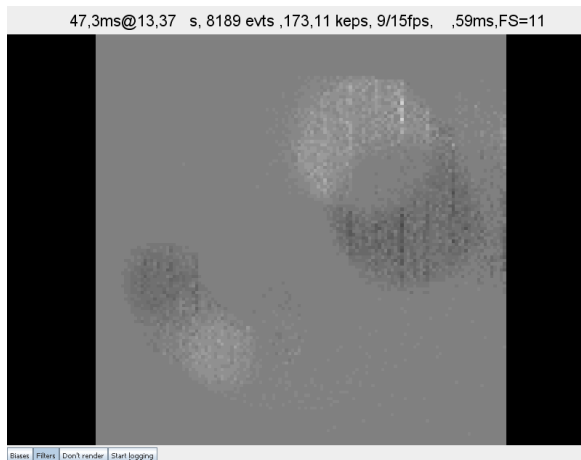


Figure 11. Real derivative retina capture

As observed in figures 10 and 11, synthesized retina has more contrast (light compensation of the sensor, and bigger relative events frequency (generates more events, for the same luminosity change). Due to this, noise contrast also increases.

It is interesting to mention than when motor speed is increased, in synthetic retina, due to frame rate limitation, aliasing problem appears and circles seems to rotate opposite to real movement.

## V. CONCLUSIONS

In this paper we have presented a digital VHDL design that implements an AER synthetic retina that tries to mimic ASIC retinas. Image captures are taken using both, a synthetic and a real one to compare them. Synthetic retina allows the development and testing of AER filters, allowing to replace a real one, in sceneries where this kind of retinas are not available. Choosing a previously pre-recorded video sequence as video source for this video-frame to AER events converter provides a fixed and know stimulus to a multi layer AER filters, so different configuration can be tested using the same fixed input.

## ACKNOWLEDGMENT

This work has been supported by the Andalucía Council with the BrainSystem project (P06-TIC-01417), and by the Spanish project SAMANTA II (TEC2006-11730-C03-02).

## REFERENCES

- [1] P. Lichtsteiner, C. Posch, T. Delbruck., "A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor"IEEE Journal on Solid-State Circuits, vol. 43, No 2, pp. 566-576, Feb-2008.
- [2] E. Culurciello, R. Etienne-Cumming, and K.A. Boahen, " A biomorphic digital image sensor" IEEE journal of Solid-State Circuits, vol. 38, p 281-204, 2003
- [3] V. Chan, S.C. Liu, A.van Schaik, "AER EAR: A Matched Silicon Cochlea Pair with Address-Event-Representation Interface". IEEE Transactions on Circuits and Systems-I. Vol. 54, No 1. pp. 48-59. Jan-2007.
- [4] Teresa Serrano-Gotarredona, Andreas G. Andreou, Bernabé Linares-Barranco. "AER Image Filtering Architecture for Vision-Processing Systems". IEEE Transactions on Circuits and Systems. Fundamental Theory and Applications, Vol. 46, N0. 9, September 1999.
- [5] Oster, M.; Douglas, R.; Shih-Chii Liu, "Quantifying Input and Output Spike Statistics of a Winner-Take-All Network in a Vision System" Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on 27-30 May 2007 Page(s):853 – 856
- [6] P. Häfliger. "Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip". IEEE Transactions on Neural Networks, vol. 18, No 2, pp. 551-572. March-2007.
- [7] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, C. Serrano-Gotarredona, J.A. Perez-Carrasco, B. Linares-Barranco, A. Linares-Barranco, G. Jimenez, A. Civit. "On Real-Time AER 2-D Convolutions Hardware for Neuromorphic Spike-Based Cortical Processing. IEEE Transactions on Neural Networks, Vol. 19, No 7, pp. 1196-1219. July-2008.
- [8] F. Gomez-Rodríguez, A. Linares-Barranco, L. Miró, S.C. Liu, A. van Schaik, R. Etienne-Cummings, M.A. Lewis, "AER Auditory Filtering and CPG for Robot Control". ISCAS 2007.
- [9] R. J. Vogelstein, R. Etienne-Cummings, N.V. Thakor, A. H. Cohen. "Dynamic Control of Spinal Locomotion Circuits". ISCAS 2007.
- [10] Gomez-Rodriguez, F.; Paz, R.; Linares-Barranco, A.; Rivas, M.; Miro, L.; Vicente, S.; Jimenez, G.; Civit, A.; "AER tools for communications and debugging" Circuits and Systems, 2006. Proceedings. 2006 IEEE International Symposium on 21-24 May 2006. ISCAS.2006.
- [11] R. Paz-Vicente, A. Linares-Barranco, D. Cascado, S. Vicente, G. Jimenez, A. Civit. "PCI-AER interface for Neuro-inspired Spiking Systems". ISCAS 2006. Kos, Greece.
- [12] Perez-Carrasco, J.A.; Serrano-Gotarredona, T.; Serrano-Gotarredona, C.; Acha, B.; Linares-Barranco, B.; "High-speed character recognition system based on a complex hierarchical AER architecture" Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on 18-21 May 2008 Page(s):2150 – 2153. ISCAS.2008
- [13] R. Berner, Tobi Delbrück, Antón Civit Balcells, Alejandro Linares-Barranco. "A 5 Meps \$100 USB2.0 Address-Event Monitor-Sequencer Interface". ISCAS 2007.
- [14] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz, F. Gomez-Rodriguez, H. Kolle Riis, T. Delbrück, S.C. Liu, S. Zahnd, A.M. Whatley, R. Douglas, P. Häfliger, G. Jimenez, A. Civit, T. Serrano-Gotarredona, A. Acosta, B. Linares-Barranco et al. "AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems". NIPS 2005.
- [15] Kwabena A. Boahen. "Communicating Neuronal Ensembles between Neuromorphic Chips". Neuromorphic Systems. Kluwer Academic Publishers, Boston 1998.
- [16] A. Jiménez-Fernández, A. Linares-Barranco, R. Paz-Vicente, C.D. Luján-Martínez, G. Jiménez, A. Civit. "AER and dynamic systems co-simulation over Simulink with Xilinx System Generator". ICECS 2008
- [17] M. Sivilotti, Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks, Ph.D. Thesis, California Institute of Technology, Pasadena CA, 1991.
- [18] A. Linares-Barranco, R. Paz-Vicente, G. Jimenez, J.L. Pedreño-Molina, J. Molina-Vilaplana, J.L. Coronado et al.. "AER Neuro-Inspired interface to Anthropomorphic Robotic Hand". IEEE World Conference on Computational Intelligence. IJCNN. Vancouver, July-2006.
- [19] A. Jiménez-Fernández, R. Paz-Vicente, M. Rivas, A. Linares-Barranco, G. Jiménez, A. Civil. "AER-based robotic closed-loop control system". ISCAS 2008.
- [20] F. Gomez-Rodriguez, R. Paz, L. Miro-Amarante, Alejandro Linares-Barranco, Gabriel Jiménez, "Two Hardware Implementation of the Exhaustive Synthetic Aer Generation Method". LNCS 2005.
- [21] jAER software. <http://jaer.wiki.sourceforge.net/>