ine Institute of Neuromorphic Engineering

THE NEUROMORPHIC ENGINEER
A PUBLICATION OF INE-WEB.ORG

10.2417/1200605.0040

Address-event-representation tools

Alejandro Linares-Barranco and Antón Civit-Balcells

European-funded boards will support eight million events per second.

Complex systems developed by neuromorphic engineers require interfaces to interconnect them, and to connect them to PCs for development, debugging, or other purposes. This concept was the starting point for the development of a set of address-event-representation (AER) tools under the European CAVIAR (convolution address-event-representation vision architecture for real-time) project.

Our team is comprised of four partners working together in the design of a neuromorphic vision system based entirely on AER principles. CAVIAR connects the biggest AER chain constructed to date.[1] The front of the signal chain is composed of a 128×128 'retina' that spikes with temporal and contrast changes,[2] four convolution chips that can detect a ball at different distances from the retina,[3] and four object chips that filter the convolutional activity.[4] This is coupled to a two-chip learning stage comprised of a delay line and a learning element.[5] To make such a vision system usable, a set of AER-tools are not only useful, but also necessary, for developing interconnections and debugging.

These AER tools are divided into four different printed circuit boards (PCBs) which, depending on the firmware downloaded to them, can have eight different functionalities. Our group, the robotics and computer technology (RTC) team, has developed a PCI-bus-to-AER interface, that uses the Rome PCI-AER design (developed by Dante) as a starting point. It consists of two USB-to-AER interfaces and an AER-to-AER interface. All elements have their own Linux and XP drivers and Matlab interfaces.

A CAVIAR PCI-to-AER interface

The PCI bus's high bandwidth and wide availability of commercial PCI interface silicon makes it an excellent tool for injecting and reading events to an AER system, provided a suitable bridging interface to the AER bus can be developed. Our interface differs from the Rome PCI-AER[6] in several characteristics: The CAVIAR PCI-AER is not a communications center and does not include a mapper or a splitter/merger. Our PCI-AER interface simply serves as a very fast communication channel (up to 10M events/sec versus the 1M events/sec capacity of the Rome board) between the AER bus and the PC software.

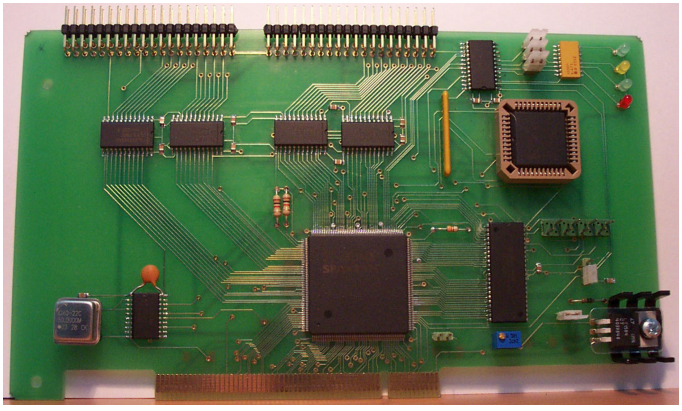

Figure 1. *A photograph of the CAVIAR PCI-AER interface board.*

This system, shown in Figure 1, is designed around a Xilinx Spartan II 200 field-programmable gate array (FPGA), with a PCI interface developed in very-high-speed integrated circuit (VHSIC) hardware description language (VHDL) by the RTC group. This makes it cheaper and faster. It has one AER output bus and one AER input bus. Both have their own first-in-first-out (FIFO) buffers that can save the event information and its timestamp for up to 128 events (output) and 256 events (input). The circuit uses a relative timestamp (indicating the distance between consecutive events), but the time controller is able to recover from protocol-induced delays. Therefore, if one event is delayed, the subsequent ones don't have to be. The timestamp function is configurable for resolutions ranging from 30ns to 480ns per timer tick. The CAVIAR PCI-AER interface supports PCI mastering.

A USB-to-AER interface

While a USB-based AER interface delivers less bandwidth than its PCI-based counterpart, its smaller size, ease of use, and versatility make it very valuable for many applications.

Continued on next page

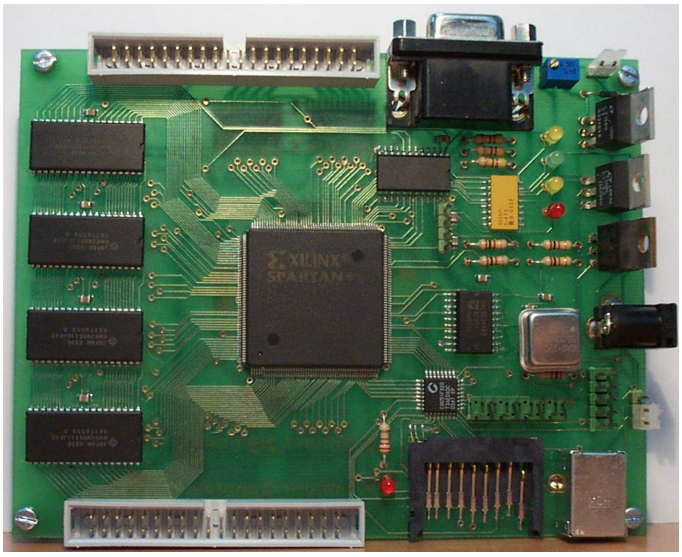

Figure 2. *A photograph of the USB-AER interface board.*

This interface is designed around a Spartan II 200 FPGA and a Cygnal 8051 USB microcontroller: elements that make it fully configurable. The USB interface's maximum throughput is 6Mbits/sec (∼187K events/sec), which limits the use of this interface for event based communications between the PC and the board. This allows the interface to receive the control information or frames (bitmaps) from the PC, whereupon it uses hardware-based frame-to-AER transformations to produce AER events.

The board's integrated 32-bit wide 2MB SRAM, memory card slot, USB connector and dual AER connectors (as shown in Figure 2), make it a very attractive platform for developing an interesting set of AER tools. The elements of the tool collection can be broken into two categories; PC dependent, and independent. In PC-dependent tools, the USB bus is used both to download FPGA firmware and pass commands and data to and from the interface. When used as a PC-independent tool, the interface's embedded memory card has the firmware stored and the microcontroller downloads it to the FPGA without the need of a host PC.

This board currently has several available firmware options that provide different functionalities. The AER generator downloads a frame (bitmap) from a PC and, through a method for synthetic AER generation,[7] uses it to create and transmit a sequence of events. One of the generation methods uses a Poisson distribution of events.[8] The board's AER mapper function can be firmware-configured in either a 1-1 or 1-N (with N from 0 to 8) arrangement. Firmware is also available to instantiate a probabilistic version that assigns a probability to each of the possible output events associated to an input event.

The board's firmware can also be used to invoke either a USB-based or a VGA-based frame grabber. The USB frame-grabber supports both 32×32 and 64×64 image sizes, and the VGA version supports 64×64 and 256×256 image sizes that use an additional AER-VGA daughter board. The firmware also supports a data-logger and player. It uses the 2Mb SRAM to capture up to 512K events with 16 bits of relative time-stamp resolution. It can also play back a sequence of events stored in the SRAM, received from the PC via a USB link.

The AER switch

The AER switch interface is able to support both one-to-many and many-to-one connections between chips or PCs. It does this using two different operations. The switch's AER splitter operation routes an AER input to up to four AER outputs. These outputs can have the input traffic replicated, or be assigned to different ranges. In the AER merger function, up to four inputs can be joined to one output. The merger function can also append tagging bits to the data that identify the input channel if necessary.

The AER switch is based on a Xilinx 9500 complex programmable logic device (CPLD). It has five AER ports: one input, one output, and three bidirectional ports.

The mini-USB-AER

A reduced version of the USB-AER that is appropriate for small event rates and simple operations has been developed together with Toby Delbruck from the Institute of Neuroinformatics. This interface allows an AER bus to connect to a PC in both directions (sequencer or monitor). The mini-USB-AER PCB is based around a Cygnal 8051 microcontroller, with no FPGA.

A software kit is available that supports this board as a monitor for speeds of around 100K events/sec. A new version that supports high speed USB2.0 and is capable of event rates up to 8M events/sec is currently under test.

Author Information

Alejandro Linares-Barranco and Antón Civit-Balcells
Robotic and Computer Technology Applied to Rehabilitation Group
University of Seville
Spain
http://www.atc.us.es/

Continued on next page

References

1. R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, R. Gómez -Rodríguez, H. Kolle Riis, T. Delbruck, S. C. Liu, P. Hafliger, 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**, **NIPS '05**, Vancouver, December 2005.
2. P. Lichtsteiner and T. Delbruck, **64×64 Event-driven logarithmic temporal derivative silicon retina**, **2005 IEEE Workshop on Charge Coupled Devices and Advanced Image Sensors**, Nagano, Japan, June 2005.
3. R. Serrano, T. Serrano, A. José Acosta, and B. Linares-Barranco, **An arbitrary kernel convolution AER-transceiver**, **ISCAS '06**, Kos, Greece, May 2006.
4. M. Oster and S. C. Liu, *Awinner-take-all spiking network with spiking inputs*, **ICECS 2004**, 2004. Tel Aviv
5. H. Kolle Riis and P. Haefliger, *Spike based learning with weak multi-levelstatic memory*, **ISCAS '04 5**, pp. 393–395, May 2004. Vancouver, Canada
6. V. Dante, P. Del Giudice, and A. M. Whatley, *Hardware and software for interfacing to address-event based neuromorphic systems*, **The Neuromorphic Engineer 2** (1), March 2005.
7. A. Linares-Barranco, G. Jiménez-Moreno, B. Linares-Barranco, and A. Civit-Ballcels, *On algorithmic rate-coded AER generation*, **IEEE Trans. Neural Networks**, May 2006.
8. A. Linares-Barranco, M. Oster, D. Cascado, G. Jiménez, A. Civit, and B. Linares-Barranco, *Inter-spike-intervals analysis of poisson like hardware synthetic AER generation*, June 2005.