AER tools for Communications and Debugging

F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miro, S. Vicente, G. Jimenez, A. Civit.
Departamento de Arquitectura y Tecnología de Computadores. Universidad de Sevilla.
Seville, SPAIN
gomezroz@atc.us.es

Abstract— **Address-Event-Representation (AER) is a communications protocol for transferring spikes between bio-inspired chips. Such systems may consist of a hierarchical structure with several chips that transmit spikes among them in real time, while performing some processing. To develop and test AER based systems it is convenient to have a set of instruments that would allow to: generate AER streams, monitor the output produced by neural chips and modify the spike stream produced by an emitting chip to adapt it to the requirements of the receiving elements. In this paper we present a set of tools that implement these functions developed in the CAVIAR EU project.**

I. Introduction

Address-Event-Representation (AER) was proposed in 1991 by Sivilotti [1] for transferring the state of an array of neurons from one chip to another. It uses mixed analog and digital principles and exploits pulse density modulation for coding information. The state of the neurons is a continuous time varying analog signal.

Figure 1. explains the principle behind the AER. The emitter chip contains an array of cells (like, e.g., an imager or artificial retina chip) where each pixel shows a state that changes with a slow time constant (in the order of milliseconds). Each pixel includes an oscillator that generates pulses of minimum width (a few nanoseconds). Each time a pixel generates a pulse (called "event"), it communicates with the periphery and its address is placed on the external digital bus (the AER bus). Handshaking lines (Acknowledge and Request) are used for completing the communication.

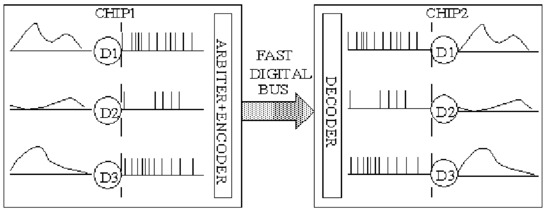


Figure 1. AER inter-chip communication scheme.

In the receiver chip the pulses are directed to the pixels or cells whose address was on the bus. This way, pixels with the same address in the emitter and receiver chips will "see" the same pulse stream. The receiver cell integrates the pulses and reconstructs the original low frequency continuous-time waveform.

Transmitting the pixel addresses allows performing extra operations on the images while they travel from one chip to another. For example, inserting memories (e.g. EEPROM) allows transformations of images.

There is a growing community of AER protocol users for bio-inspired applications in vision and audition systems, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [2]. The goal of this community is to build large multi-chip hierarchically structured systems capable of performing complicated array data processing in real time. The CAVIAR EU project has the objective to demonstrate this technology by targeting and following a moving ball. The planned AER system under CAVIAR uses the following AER chips: one Retina, four Convolutions, four Winner-Take-All (Object) and one Learning chip.

To make possible the right communication of these chips and for debugging purposes it is essential to have a set of instruments that would allow to:

- Sequence: Produce synthetic AER event streams that can be used as controlled inputs while testing and adjusting a chip or set of chips.
- Monitor: Observe the output of any element in the system.
- Map: Alter the stream produced by an emitter and send the modified stream to a receiver

For these purposes we have designed and implemented two different instruments: a PCI-based board capable of sequencing and monitoring events at a rate of over 15Mevents/s and a versatile USB-based board that can be used for sequencing, monitoring and mapping. This last board can be used either in a stand alone mode or connected to an external computer through a USB bus. A possible scenario for these tools is shown in Figure 2. where a computer with a PCI-AER [9] board produces output for AER chip1. The output from this chip is remapped by an USB-AER board and fetched to AER chip 2. The stream produced by chip 2 is monitored by another USB-AER board which can send its output directly to a VGA monitor or to a computer through USB bus.

0-7803-9390-2/06/$20.00 ©2006 IEEE

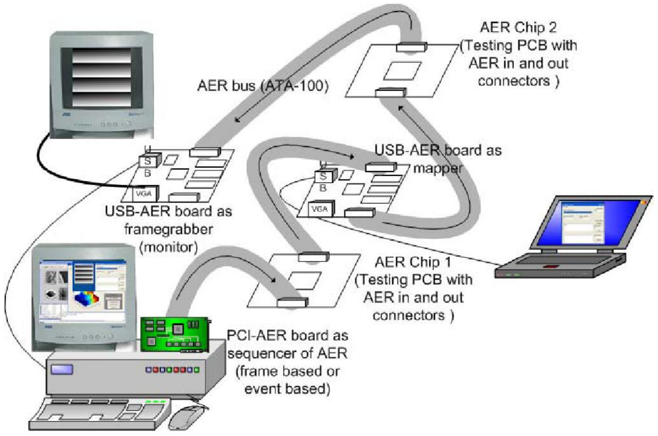


Figure 2. AER tools usage scenario

To be useful for debugging an AER tool should be able to receive and also send a long sequence of events interfering as little as possible with the system under test.

As neurons have the information coded in the frequency (or timing) of their spikes, the pixels that transmit their address through an AER bus also have their information coded in the frequency of appearance of those addresses in the bus. Therefore, inter-spike-intervals (ISIs) are critical for this communication mechanism. Thus, a well designed tool shouldn't modify the ISIs of the AER.

Sections II presents the USB solutions and their applications in AER testing. Section III presents a Switch-AER. Section IV present a reduced version of the USB-AER board with lower capabilities and performance, but very simple to use and better for low traffic applications. And finally in section V we conclude with one example of connectivity.

II. USB-AER INTERFACE

The CAVIAR PCI-AER board [9] can perform Address Event sequencing and monitoring functions but has no hardware mapping capabilities. Although software based mapping is feasible, if we want to build AER systems that can operate without requiring a computer, we need a specific device for this purpose. The USB-AER board allows standalone operation and has several functionalities, like hardware mapping and more.

This standalone operating mode requires to be able to load the FPGA and the mapping RAM from some type of non volatile storage that can be easily modified by the users. MMC/SD cards used in digital cameras are a very attractive possibility. However in the development stage users prefer to load the board directly from a computer and, for this purpose USB seems the most suitable solution.

Many AER researchers would like to demonstrate their systems using instruments that could be easily interfaced to a laptop computer. This requirement can also be supported with the USB-AER board as it includes a relatively large FPGA that can be loaded from MMC/SD or USB, a large SRAM bank and two AER ports. Thus the board can be used also as a sequencer or a monitor. Due to the bandwidth limitations of full speed USB 1.1 (12Mbit/s) hardware based event to frame conversion is essential in this board for high, or even moderate, event rates.

The USB-AER board is based around a Spartan-II 200 Xilinx FPGA, with a 512K*32 12ns SRAM memory bank. The board uses a Silicon Laboratories C8051F320 microcontroller to implement the USB and the MMC/SD interface. The USB interface allows up to 6 Mbit/s. A simple VGA monitor interface is also provided to allow the board to act as a monitor (frame-grabber).

The board will act as a different device according to the module that is loaded in the FPGA either through a MMC/SD card or from the USB bus. Currently the following modules are implemented:

A. *Mapper*

Mapping functionality consists in a change of the events addresses. Theses changes can be:

- One to one: This mapping is used in one to one communication, when the addresses of the sender device are different from the addresses of the receiver device.
- One to one replicated: This mapping replicates, for each input event, up to 8 times the same output event.
- One to several: This mapping is used in one to several communication. In this case one input event in sent to several output addressed.
- No event: this mapping no emits event.

Different mapping types can be specified for each input address.

B. *Monitor*

This functionality allows to record the traffic in an AER bus and to represent it in a VGA or in a computer. Monitor types are:

- Frame-grabber: The USB-AER board is inserted in the AER bus, and it captures the events during a time; then reconstructs and image and sends it to the computer using the USB port.
- Sniffer: The USB-AER board is connected to the AER bus without interfering in the traffic. In this case the sniffer must be faster than the other AER devices connected to the AER bus. The USB-AER board captures the events and stores it with a timestamps; when all internal RAM is full, it sends the information to the computer using the USB port.
- Logger: The USB-AER board is inserted in the AER bus, and captures the events with timestamps. In this case the logger interferes in the AER traffic.
- VGA: the USB-AER board captures the events and converts it in a B&W image and sends it using the VGA connector. A Gray VGA is possible using a VGA-DAC board connected to the out-AER connector of the board.

C. Sequencer

In this case the USB-AER board is used as events generator, there is two types of AER traffic generation:

- Frame-based generation: The USB-AER board generates events using a digital image stored in a computer. Computer sends to the board a digital image the USB-AER board converts it in an events stream that can be transmitted through AER bus. To convert the image to events stream Random or Exhaustive methods [4][5][6] are used.
- Player: The USB-AER board generates events using timestamp information. This information is sent to the board using the USB port, or this information can be the information captured using the board as a logger o as a sniffer.

The functionality sniffer, logger and player are implemented in the same module. The USB-AER can capture sequences of up to 512K events with timestamps and play these sequences back, without reprogramming it, only sending the correct command. These modules are very interesting when a researcher wants to use the output stream produced by a chip from another researcher as input to his or her chip.

This new board was interfaced in Telluride 04 [7] to the current version of the CAVIAR retina and to an imager developed at JHU. Later in the CAVIAR meeting in September 04 it was interfaced to the remaining project chips. Later in May 05 it was used successfully during the CAVIAR project workshop.
The USB-AER board is shown in Figure 3.

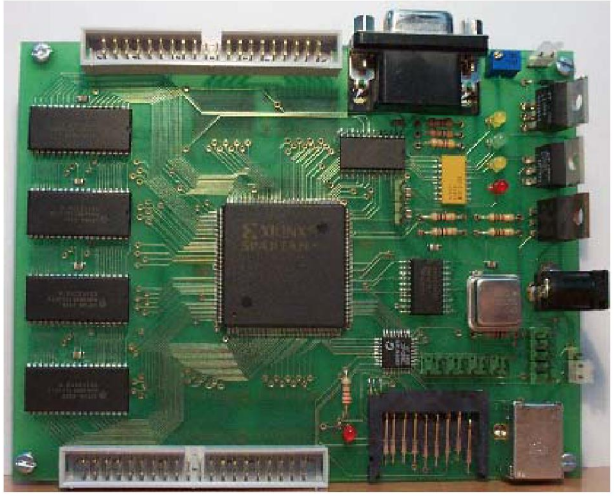


Figure 3. USB-AER Board.

A simple interface to control this board is available under windows. It allows to load modules into the FPGA, uploading or downloading data to or from the board, and showing the received images when the board acts as a monitor. Furthermore an API for C++ and for MATLAB is also available.

A Linux driver for the USB-AER is currently under test. With this driver the USB-AER board can be easily integrated with several MATLAB applications developed at INI [8].

III. AER-SWITCH

This board allows to connect several AER devices to the same AER BUS. The connection possibilities are: 4 to 1 or 1 to 4. This board allows:

- The connection of more complex AER systems.
- An easier debugging by inserting PCI-AER or USB-AER board without modifying the structure of the global system to be tested.

This board has a CPLD as a communication centre that manages the different modes and controls the protocol lines. It can work in two modes:

- Merger: 4 input and 1 output
- Splitter: 1 input and 4 output mode.

Both can be unicast (selecting one output) or broadcast. This functionality should be configured by jumpers. There are 5 different AER ports; one of them works always as an output, and another as an input. The others three are bidirectional and work as input or output depending on the mode of operation. Figure 4 shows the current version of this board.

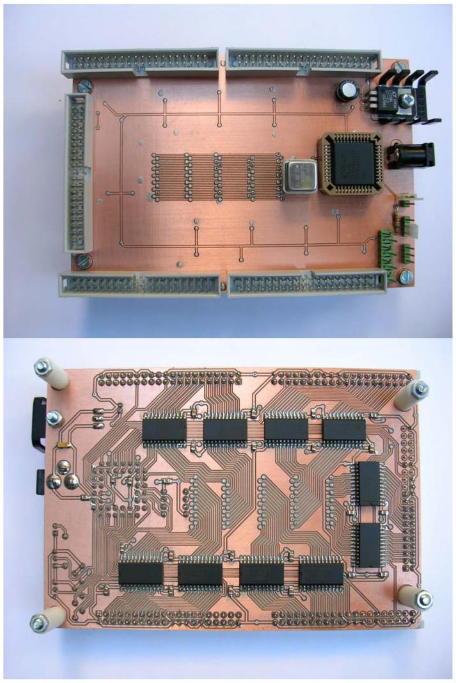


Figure 4. AER-Switch Board.

IV. MINI USB-AER

For those tests or applications where it is not needed high speed performance, a reduced version of the USB-AER board is available. This one doesn't have FPGA and MMC/SD card. This board can be connected to the PC through the USB bus, the functionality (Monitor or Sequencer) has to be programmed into the microcontroller under C code. Figure 5. shows the current version of this board.

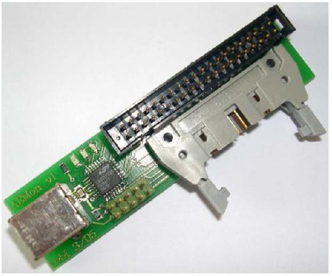


Figure 5. Mini AER-USB Board.

V. CONCLUSIONS

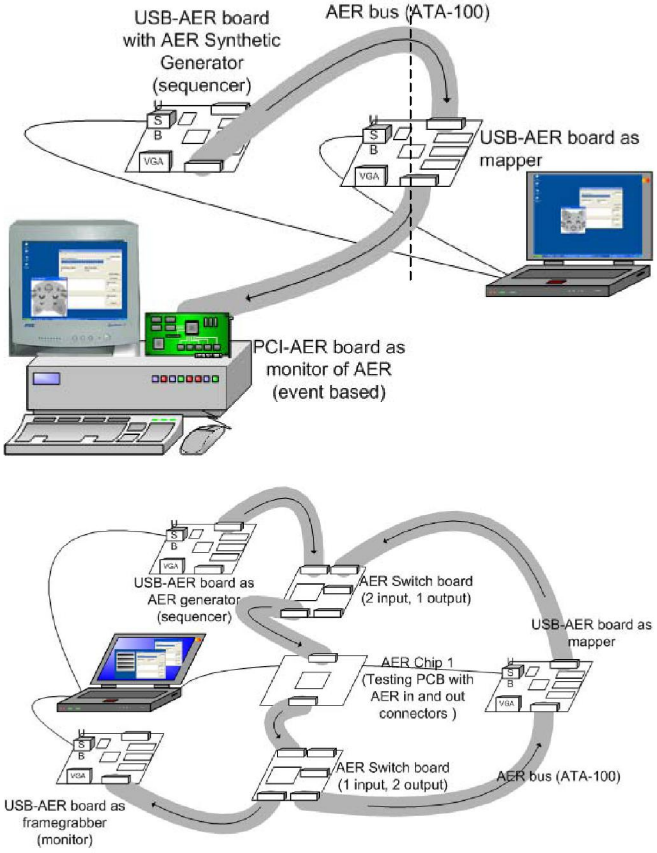
A set of tools has been developed that allow efficient testing and demonstration of address event based systems. Two demonstration scenarios are shown in Figure 6. In the first case a USB-AER board is generating a stream of events, using a hardware synthetic method, that correspond to the image shown in Figure 6. on the laptop's screen, which is rotated by a second USB-AER board used as a mapper to produce an AER events stream, which is captured by a PCI-AER board used as a monitor to produce the output shown in the desktop PC's screen.

In the second scenario an USB-AER is working as a frame to AER sequencer to feed a AER chip. This chip receives also the transformed output of another AER chip using the AER-Switch. The output of the second chip can be viewed on the laptop's screen, using another USB-AER as a monitor.

In this scenario only the presented tools are shown. In real world cases the tools are used to evaluate or to tune neural chips. In the CAVIAR project the chips have been interfaced to two different retinas, a convolution chip, a winner take-all (object) chip and a learning chip.

ACKNOWLEDGMENT

This work was in part supported by EU project IST-2001-34124 (CAVIAR), and Spanish project TIC-2003-08164-C03-02 (SAMANTA).



USB-AER board with AER Synthetic Generator (sequencer)
AER bus (ATA-100)
USB-AER board as mapper
PCI-AER board as monitor of AER (event based)
VGA
USB-AER board as AER generator (sequencer)
AER Switch board (2 input, 1 output)
USB-AER board as mapper
AER Chip 1 (Testing PCB with AER in and out connectors)
AER Switch board (1 input, 2 output)
AER bus (ATA-100)
USB-AER board as framegrabber (monitor)

Figure 6. Two demonstration Scenarios.

REFERENCES

[1] M. Sivilotti, Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks, Ph.D. Thesis, California Institute of Technology, Pasadena CA, 1991.

[2] A. Cohen, R. Douglas, C. Koch, T. Sejnowski, S. Shamma, T. Horiuchi, and G. Indiveri, Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA, June-July 2001. [www.ini.unizh.ch/telluride]

[3] A. Wilen, J.Schade, R. Thornburg. "Introduction to PCI Express", Intel Press, 2003.

[4] A. Linares-Barranco. Estudio y evaluación de interfaces para la conexión de sistemas neuromórficos mediante Address- Event-Representation. Ph.D. Thesis, University of Seville, Spain, 2003

[5] A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels, and B. Linares-Barranco. "On Algorithmic Rate-Coded AER Generation". Accepted for publication on IEEE Transaction on Neural Networks.

[6] Linares-Barranco, A.; Jimenez-Moreno, G.; Civit-Ballcels, A.; Linares-Barranco, B.; On synthetic AER generation. ISCAS '04. Proceedings of the IEEE 2004 May 2004 Pages:V-784 - V-787 Vol.5

[7] Avis Cohen, et.al., Report on the 2004 Workshop On Neuromorphic Engineering, Telluride, CO. June - July , 2004 [www.ini.unizh.ch/telluride/previous/report04.pdf]

[8] M. Oster, Serverbased Software Architecture for AER systems [http://www.ini.unizh.ch/~mao/AerSoftware/SoftwareOverview.pdf]

[9] R. Paz-Vicente, A. Linares-Barranco, D. Cascado, S. Vicente, G. Jimenez, A. Civit. PCI-AER interface for Neuro-inspired Spiking Systems. Submitted to ISCAS 2006. Kos, Greece.