Tools for Address-Event-Representation Communication Systems and Debugging

M. Rivas, F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco,
S. Vicente, and D. Cascado

Departamento de Arquitectura y Tecnología de Computadores, Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
{mrivas, gomezroz, rpaz, alinares, satur, danic}@atc.us.es
http://www.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.

1 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.

Fig. 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.

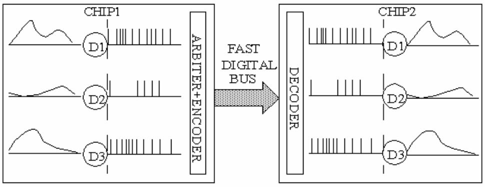

Fig. 1. AER inter-chip communication scheme

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

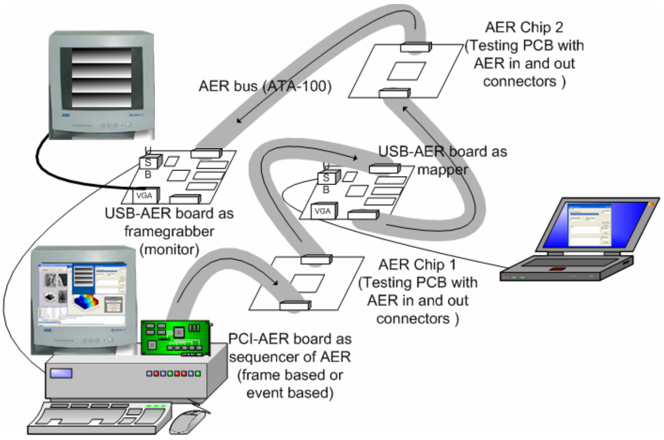

Fig. 2. AER tools usage scenario

For these purposes we have designed and implemented two different instruments: a PCI board capable of sequencing and monitoring events at a rate of over 15Mevents/s and a versatile 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 Fig. 2 where a computer with a PCI-AER board produces output for AER chip1. The output from this chip is remapped by a 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.

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 2 and 3 present the PCI and the USB solutions and their applications in AER testing. Section 4 presents a Switch-AER. Section 5 presents a small version of a USB board with lower capabilities and performance, but very simple to use. And finally in section 6 we conclude with two examples of connectivity.

2 PCI-AER Interface

Before the development of our tools the only available PCI-AER interface board was developed by Dante at ISS-Rome (See [3]). This board is very interesting as it embeds all the requirements mentioned above: AER generation, remapping and monitoring. Anyhow its performance is limited to 1Mevent/s approximately. In realistic experiments software overheads reduce this value even further. In many cases these values are acceptable but, currently many address event chips can produce (or accept) much higher spike rates.

As the Computer interfacing elements are mainly a monitoring and testing feature in many address event systems, the instruments used for these purposes should not delay the neuromorphic chips in the system. Thus, speed requirements are at least 10 times higher than those of the original PCI-AER board. Several alternatives are possible to meet these goals: extended PCI buses, bus mastering or hardware based Frame to AER and AER to Frame conversion.

The previously available PCI-AER board uses polled I/O to transfer data to and from the board. This is possibly the main limiting factor on its performance. To increase PCI bus mastering is the only alternative. The hardware and driver architecture of a bus mastering capable board is significantly different, and more complex, than a polling or interrupt based implementation.

The theoretical maximum PCI32/33 bandwidth is around 133Mbytes/s. This would allow for approximately 44Mevent/s considering 2 bytes per address and two bytes for timing information. Realistic figures in practice are closer to 20Mbyte/s. Thus, in those cases where the required throughput is higher a possible solution is to transmit the received information by hardware based conversion to/from a frame based representation. Although this solution is adequate in many cases, there are circumstances where the developers want to know precisely the timing of each event, thus both alternatives should be preserved.

The physical implementation of all the steps is equal. They differ in the VHDL FPGA code and in the operating system dependent driver. The first design was a VIRTEX based board which was completely redesigned after the first tests. It was established that most of the functionality, demanded by the users, could be supported by the smaller devices in the less expensive SPARTAN-II family. The Spartan Version of the board is shown in Fig. 3.

Currently a Windows driver that implements bus mastering is being tested. The Linux version with bus mastering is still under development. An API that is compatible, as much as permitted by the different functionality, with that used in the current PCI-AER board has been implemented. MEX files to control the board from MATLAB have also been developed.

Current performance of PCI-AER board is around 15 Mevents/second using PCI mastering capabilities.

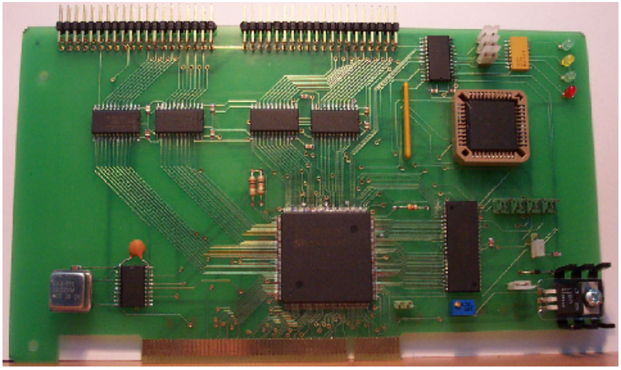

Fig. 3. CAVIAR PCI-AER board

3 USB-AER

The CAVIAR PCI-AER board can perform Address Event sequencing and monitoring functions but has no hardware mapping capabilities. Although software based mapping is feasible a specific device for this purpose is needed if we want to build AER systems that can operate without requiring any standard computer. 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 the 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 (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. 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 Modes are implemented:

- Mapper: 1 event to 1 event and 1 event to several events.
- Monitor (frame-grabber): using either USB or VGA as output. For the VGA output there are two possibilities: B/W VGA, using the VGA connector of the board. And Gray VGA, using a VGA-DAC board connected to the out-AER connector of the board.
- Sequencer: based on hardware frame to AER conversion using the Random or Exhaustive methods [4][5][6]. Can produce up to 25 Mevents/second. (40 ns per event).
- Datalogger: allows to capture sequences of up to 512K events with timestamps and send them to the PC offline through USB bus.
- Player (under development): to play up to 512Kevents with their timestamps.

These two modules are very interesting when a researcher wants to use the output stream produced by a chip from another researcher (probably in other country) 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. The USB-AER board is shown in Fig. 4.

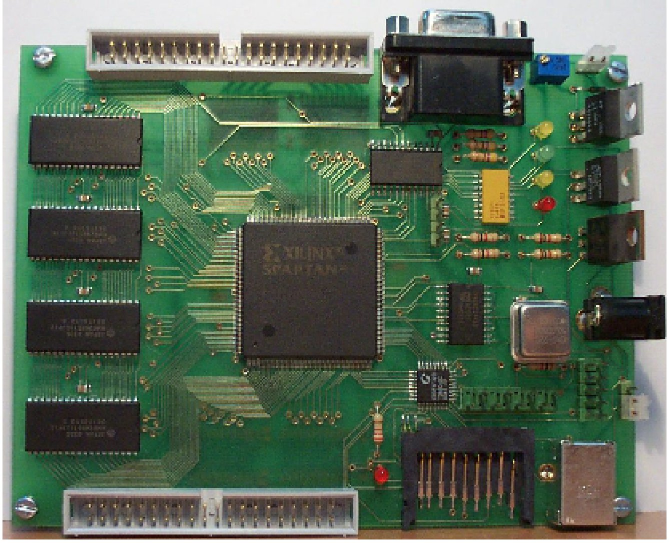

Fig. 4. USB-AER Board

A simple interface to control this board is available under windows. It allows loading modules into the FPGA, uploading or downloading data to the FPGA, and showing the received images when the board acts as a monitor. There is also available a MATLAB interface that support the same functionality.

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].

4 AER-Switch Board

A 4 to 1 and 1 to 4 AER-switch is presented in this paper. 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 asynchronously the protocol lines. It can work in 2 different modes: 4 input, 1 output mode and 1 input, 4 output mode, both in unicast mode (selecting one output) or broadcast mode. This functionality should be configured by jumpers. There are 5 different AER ports, where one of them works always as an output, and another as an input. The others three are bidirectional. Fig. 5 shows the current version of this board.

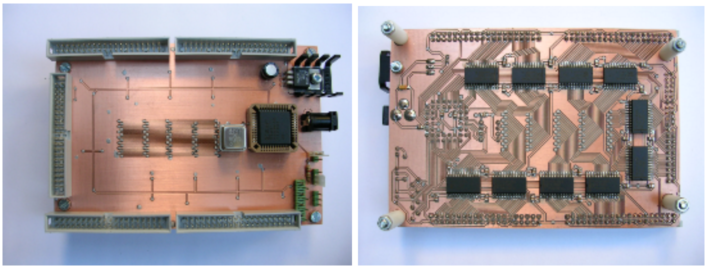

Fig. 5. AER-Switch Board

5 Mini-USB Board

For those tests or applications where it is not needed high speed performance, a small version of the USB board is available. This one doesn't have FPGA, nor MMC/SD card. This board can be connected to the PC through the USB bus, and all the functionality (Monitor or Sequencer) has to be programmed into the microcontroller under C code. Fig. 6 shows the current version of this board. The board has been developed also under CAVIAR project by INI partner and authors.

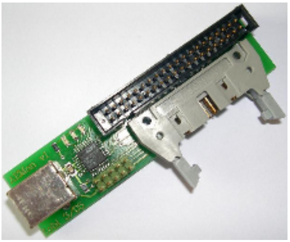

Fig. 6. AER-Switch Board

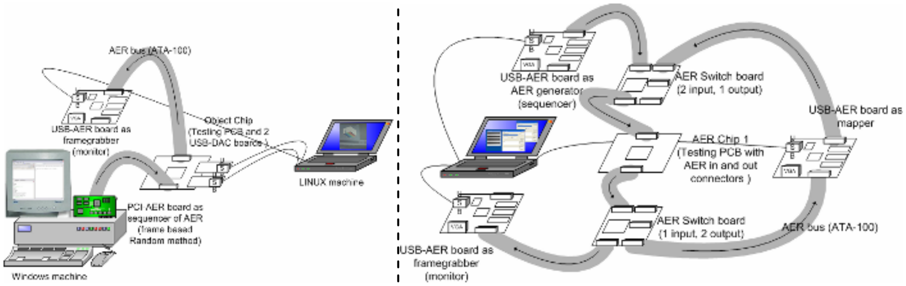

Fig. 7. Two demonstration Scenarios

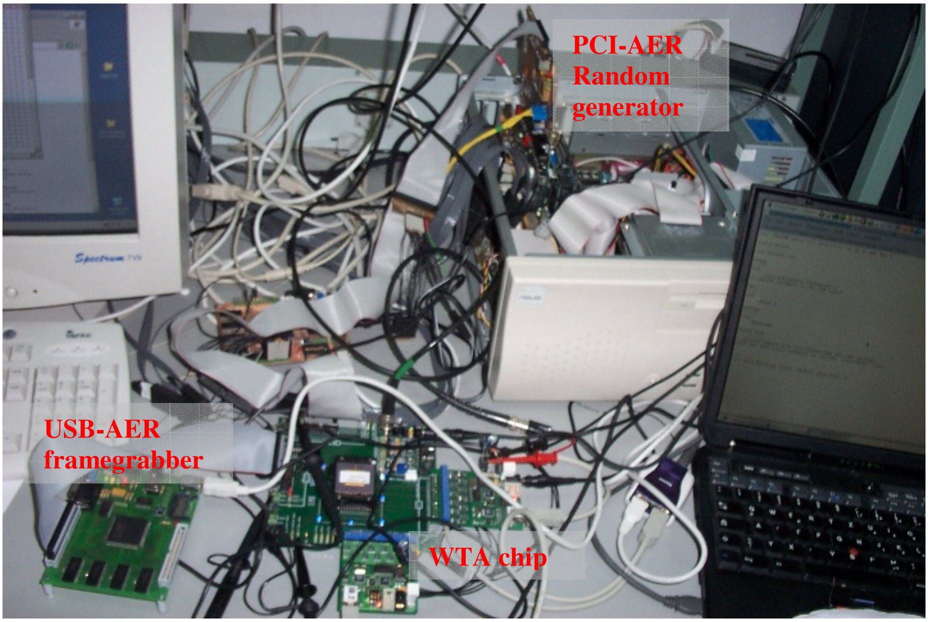

Fig. 8. Scenario Photograph

6 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 Fig. 7. In the left case a PCI-AER board is generating a stream of events from a digital frame, using a hardware synthetic method. This sequence is used to feed a WTA filter chip, developed at INI. The output of the WTA chip is captured using a USB-AER board configured as a frame-grabber. A photograph of this scenario is shown in fig. 8.

On the right a 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 in the laptop 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 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.

Acknowledgements

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

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. V. Dante. “PCI AER Adapter board”, http://neural.iss.infn.it/Board/draft.html.
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, R. Senhadji-Navarro, I. García-Vargas, F. Gómez-Rodríguez, G. Jimenez and A. Civit. Synthetic Generation of Address-Event for Real-Time Image Processing. ETFA 2003, Lisbon, September. Proceedings, Vol. 2, pp. 462-467.
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]