LVDS Serial AER Link performance

L. Miró-Amarante, A. Jiménez-Fernández, A. Linares-Barranco, F. Gómez-Rodríguez, R. Paz, G. Jiménez, A. Civit.
Arquitectura y Tecnología de Computadores.
Universidad de Sevilla.
Av. Reina Mercedes s/n, 41012-Sevilla, SPAIN
lmiro@atc.us.es

R. Serrano-Gotarredona.
Instituto de Microelectrónica de Sevilla. CSIC.
Av. Reina Mercedes s/n. Edificio CICA. 41012-Sevilla. SPAIN

***Abstract*— Address-Event-Representation (AER) is a communication protocol for transferring asynchronous events between VLSI chips, originally developed for bio-inspired processing systems (for example, image processing). Such systems may consist of a complicated hierarchical structure with many chips that transmit data among them in real time, while performing some processing (for example, convolutions). The event information is transferred using a high speed digital parallel bus (typically 16 bits and 20ns-40ns per event). This paper presents a testing platform for AER systems that allows analysing a LVDS Serial AER link produced by a Spartan 3 FPGA, or by a commercial LVDS transceiver. The interface allows up to 0.728 Gbps (~40Mev/s, 16 bits/ev). The eye diagram ensures that the platform could support 1.2 Gbps.**

I. Introduction

Address-Event-Representation (AER) was proposed in 1991 by Sivilotti [1] for transferring the state of an array of analog time dependent values from one chip to another. It uses mixed analog and digital principles and exploits spikes for coding information. Figure 1. explains the principle behind the AER basics. The emitter chip contains an array of cells (like, for example, the pixels of a camera or an artificial retina chip) where each cell implements a continuously varying time dependent state that change with a slow time constant (in the order of *ms*). Each cell or pixel includes a local oscillator (VCO) that generates digital pulses of minimum width (a few nano-seconds). The rate of pulses is proportional to the state of the cell (or pixel intensity for a retina), assuming spike rate coding is used. Each time a pixel generates a pulse (which is called "event"), it communicates with the array periphery and a digital word representing a code or address for that pixel is placed on the external inter-chip digital bus (the AER bus). Additional handshaking lines (Acknowledge and Request) are used for completing the asynchronous communication. The inter-chip AER bus operates at the maximum possible speed. In the receiver chip the pulses are directed to the pixels whose code or address was on the bus. In this way, cells with the same address in the emitter and receiver chips are virtually connected with a stream of pulses. The receiver cell integrates the pulses and reconstructs the original low frequency continuous-time waveform. Cells that are more active access the bus more frequently than those less active.

Transmitting the cell addresses allows performing extra operations on the events while they travel from one chip to another. For example in a retina, the activity of the pixels in the array represents the input image. By translating the address of the events during transmission, the image can be shifted or rotated. This translation of the address can be achieved by inserting properly coded EEPROMs. Furthermore, the image transmitted by one chip can be received by many receiver chips in parallel, by properly handling the asynchronous communication protocol. The event-based nature of the AER protocol also allows for very efficient convolution operations within a receiver chip [2].

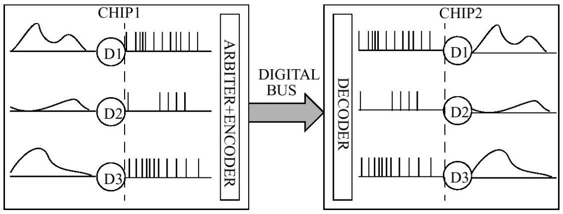


Figure 1. Rate-Coded AER inter-chip communication scheme.

There is a growing community of AER protocol users for bio-inspired applications in vision, audition systems and robot control, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [3]. The goal of this community is to build large multi-chip and multi-layer hierarchically structured systems capable of performing complex massively-parallel processing in real time [4][5].

These systems are limited by the number of bits of the AER bus and the bandwidth that parallel cables supports. There are several works [6] that uses ATA100 cables.

A previous Serial AER approach, proposed by Pouliquen and Andreou [7], reduced the number of lines of the AER bus (It uses eight wires). In this paper, we present a Serial AER that uses only one line.

In the following sections we present two Serial AER (SAER) communication scheme and their implementations into the CAVIAR USB2AER platform for testing and interfacing Serial AER systems. An experiment to test the SAER communications capabilities and an eye diagram are also presented.

II. LVDS Serial AER

Following the next-generation interconnect standards, we present a Serial AER scheme, based on the use of high-speed point-to-point serial connections.

The Serial AER scheme emerges to overcome a number of limitations of the Parallel AER scheme. The most significant limitations are the difficulty in increasing the data rate beyond 100 Mbytes/s and the fact that ATA100 cable uses a single-ended signaling system that is prone to induced noise. A signaling at gigabit speeds requires differential signaling. We have chosen a serial wire that uses the Low Voltage Differential Signaling (LVDS) standard, TIA/EIA-644-A-2001[7]. LVDS is designed for high-speed, low-power, and low-noise point-to-point communications. LVDS radiates less noise than single-ended signals due to the canceling of magnetic fields, and is more immune to noise because it is coupled onto the two wires as a common-mode signal.

On the other hand, Serial AER uses only 4 signal pins, improving pin efficiency over the Parallel AER interface which uses more than 18 signal pins going between devices.

So, this new point-to-point connection (Serial AER) provides greater dedicated bandwidth and smaller connectors.

Furthermore, Serial AER increases the flexibility and the scalability. Parallel AER is limited in scalability because of an increment of the number of neurons within a system requires an increment of the number of wires in parallel bus. Increasing bus width reduces the maximum achievable frequency due to skew between signals. More signals also requires more pins on a device, traces on boards and larger connectors.

Serial AER saves a significant large amount of space and does not limit on the physical length of the wire.

In addition, LVDS serializers and deserializers also contribute significant space and cost savings to designs. To transmit parallel Address Event, using a serial connection, it is necessary a parallel-to-serial conversion in the sender chip. Obviously, in the receiver chip, it is necessary a serial-to-parallel conversion. We compare two alternatives of implementing this: (a) with VHDL through a FPGA that support LVDS pin out with digital controlled impedance, and (b) using commercial serializers and deserializers to convert parallel data into LVDS serial.

A. LVDS with VHDL and FPGA:

A VHDL code has been implemented that convert the 16 parallel bits of the AER event into serial using clock forwarding technique (Double Data Rate, DDR signaling). The Spartan 3 of the USB2AER board has been used to test both the serializer and deserializer using a SATA cable to connect both components. In this case the clock is not embedded with the data, so two LVDS pairs are used in the PCB and in the SATA cable.

B. LVDS with commercial chips.

We have designed two small "plug-in" boards to convert the parallel data into serial LVDS. The Texas Instruments SN65LV1023ADB LVDS serialiser and the Texas Instruments SN65LV1224BDB LVDS deserialiser are used, in our design, to convert 11 TTL lines (10 data + 1 clock) into one high-speed LVDS pair. The serializer converts 10bit wide parallel LVTTL/LVCMOS data into an LVDS serial data stream with the clock embedded. A high-state start bit and a low-state stop bit are added internally and frame the 10bit parallel input data and ensure a transition in the serial data stream. Therefore, 12 serial bits are transmitted for each of the 10bit parallel input bits. The TI serializer accepts up to 66MHz as reference clock to produce a serial data rate of up to 792Mbps (12bits x 66MHz). However, since only 10 bits are from input data, the actual throughput is 10 times the reference clock frequency. The deserializer receives the serial output from the serializer and converts it back to 10bit-wide parallel data. Because the deserializer recovers both clock and data from the serial data stream, clock-to-data and data-to-data skew, that would be present with a parallel bus, are eliminated.

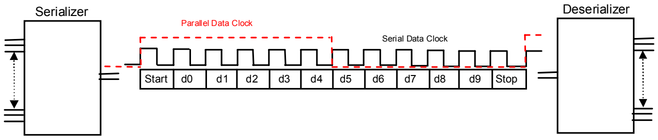


Figure 2. Clock embedded into serializer output.

In the previous Parallel AER scheme (Figure 1.), an asynchronous communication, data is sent without a timing clock using a handshake protocol. This involves two additional lines. However, our Serial AER scheme is a synchronous communication (data and timing information are sent together) Figure 2. This allows to eliminate the acknowledgment signal. In the handshake protocol, it is necessary to wait for acknowledgment before sending the next events. This could imply a delay in the next events. In Address-Event-Representation are important data (Address) and the time when this data appears on the bus (Event). So, the handshake protocol could generate errors in the timing information of the events. It is true that Serial AER scheme could generate events leak. But this error is not as important as receiving data without correct timing information.

In the biological neural communications do not exist any confirmation mechanism. Neurons send pulses to next layer and do not mind about if the pulses are received or not.

So, for these reasons, Serial synchronous AER scheme is more similar to biological neural communication and AER philosophy than Parallel handshake AER scheme.

III. AER Testing Platform

This section describes in detail the AER platform called CAVIAR USB2AER. This new interface is a second generation of the previous USB-AER board developed under the EU project CAVIAR [9]. This platform has been designed around a Spartan 3 400 FPGA with 2 parallel AER connectors (1 input and 1 output), 2 SAER connectors (Serial AER) based on SATA connectors, routed directly to the FPGA (1 input and 1 output). With these four buses a huge amount of possibilities are open for interconnecting AER based chips and for starting to test the serial version of the Address-Event-Representation bus for communicating neuro-inspired signals (SAER).

The board is able to communicate with a PC through a high speed (400Mbps) digital bus, supported by a Cypress FX2 microcontroller, based on the 8051 core and a USB 2.0 full speed transceiver. A flash memory card, a MMC/SD card, (Multimedia Card/Secure Digital) make possible to have a stand alone system, totally independent from a PC or laptop. The MMC could have the FPGA core and the data required for the operation of the board.

The same amount of SRAM than the previous board is available in this generation.

The LVDS communication is managed directly through the FPGA, so the serializer and deserializer circuits have to be implemented inside. This is going to limit the speed of the LVDS communication to the speed supported by the device. For this Spartan 3 400, the maximum clock supported is 280 MHz and up to 622Mbps per I/O [10].

Figure 4. shows the block diagram of the board and Figure 3. shows the prototype board.

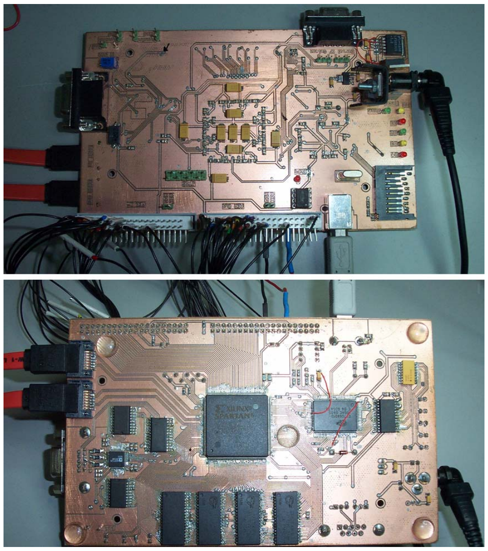


Figure 3. USB2AER board photograph.

IV. Experiments

We propose a simple experiment to analyze these two implementations of the SAER: using commercial ser-deser chips, and using a custom ser-deser design for a commercial FPGA. We demonstrate that SAER is feasible and it has a lot of possibilities from the point of view of speed, scalability, and easiness to implement. Figure 4. shows the experiment block diagram.

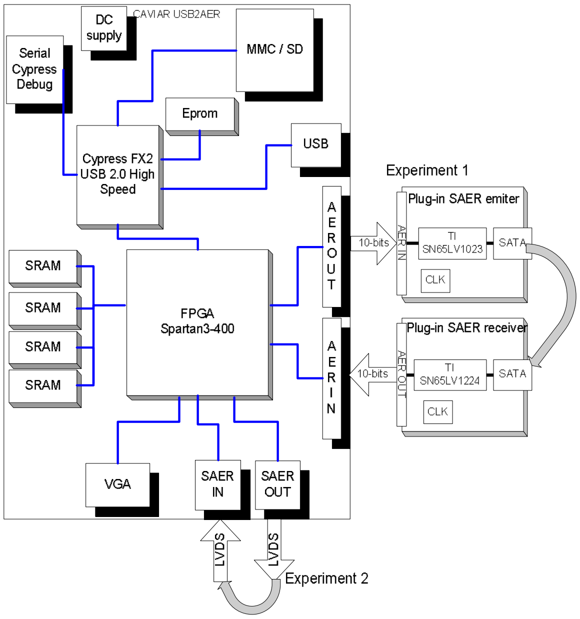


Figure 4. Two SAER link experiments block diagram

The first part of this experiment consists in programming a simple VHDL code, for the Spartan 3 400 FPGA of the USB2AER platform, for testing the maximum speed supported by LVDS serializer and deserializer transceivers. The VHDL code describes a simple state machine that generates continuously 10-bit word to transmit them through the external serializer plug-in board. The external serializer can work with a clock frequency of up to 66 MHz. We have used a 50MHz clock. The serial output of the plug-in board, a LVDS signal, is sent using a 100cm SATA cable connected to another plug-in board with a deserializer manufacturer interface. This second plug-in is connected to the AER IN connector of the USB2AER interface. The VHDL code is receiving the 10 bits data continuously and is comparing the transmitted word with the received one. In this experiment, the two Texas Instrument chips transmit both data and clock into the same differential signal.

The second part of the experiment consists in using an FPGA to serialize and to deserialize parallel AER Event. In this case, we have chosen to use a clock forwarding technique where the data and clock signal are forwarded together. The clock frequency used is half the data transmission frequency, which means that data changes on each edge of the clock (DDR signaling). The advantage of this method is that the harmonic content of the clock and data lines are the same.

The design is composed by a transmitter module and a receiver module. Both modules implement AER protocol to interface with Address Event Systems. The transmitter module processes a 16-bit Parallel Address Event at a frequency of 48MHz, and sends it serially on a differential pair (LVDS) using DDR, at a frequency of two times the system clock, 100MHz aprox.. The clock is also sent through another LVDS link (using DDR signaling too). The receiver module accepts input data and clock serial signals. These are used to deserialize the data to 16-bit Parallel Address Event.

Figure 5. shows the oscilloscope LVDS signal and the clock extracted by the oscilloscope. The transmission works at 600 Mbps. Figure 6. shows the eye diagram of the differential pair when the experiment is running. It can be seen that for this cable length (100 cm) the speed transmission can be almost double faster. Table 1 shows the parameters of both implementations.

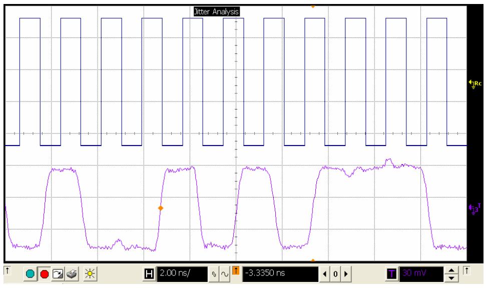
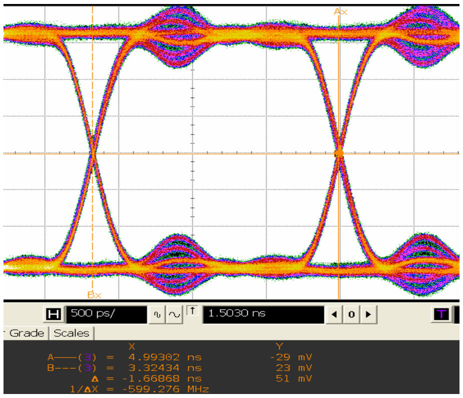


Figure 5. SAER link oscilloscope with clock extraction.



```
                X                     Y
A——(3) =  4.99302 ns            -29 mV
B---(3) =  3.32434 ns             23 mV
      Δ = -1.66868 ns             51 mV
   1/ΔX = -599.276 MHz
```

Figure 6. SAER link eye diagram.

Parameter	Experiment 1	Experiment 2
Total Jitter	150 ps	
Symbol duration	1,66 ns	
Bit rate	~600 Mbps	~100 Mbps
Reference clock	50 MHz	48 MHz
Event rate (16-bits)	31,25 Mev/sec	6,25 Mev/sec

Table 1. Two SAER link eye diagram parameters.

V. Conclusions

This paper presents a Serial Address-Event-Representation (SAER) synchronous proposal for high speed inter-chip neuro-inspired communications. A simple serializer and deserializer plug-in boards, based on commercial chips, and another implemented by FPGA are presented to convert a conventional parallel AER bus into serial and vice-versa.

An AER testing platform is also presented and used to test the LVDS transmission characteristics through SATA cable. The eye diagram ensures that the communication could work up to two times higher, what implies 1,2 Gbps. If you suppose 2 bits for command and 8 for x or y event information for this 10-bit interface, you will need two 10-bit word to transmit an AER 16-bit word. Therefore, the SAER bandwidth could be up to 66Mev/sec.

The future work is focussed on increasing the bandwidth of the FPGA implementation, and on characterizing the LVDS signal parameters for different cable lengths and higher bandwidths using faster reference clocks (60MHz). We will characterize also the BER and power consumption for the different versions.

Acknowledgments

This work was in part supported by EU grant IST-2001-34124 (CAVIAR), and spanish grant TIC-2003-08164-C03-02 (SAMANTA). Authors thank to Raphael Berner for his effort on the development of the USB2AER board.

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] 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.

[3] 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 2004. [www.ini.unizh.ch/telluride]

[4] Kwabena A. Boahen. “Communicating Neuronal Ensembles between Neuromorphic Chips”. Neuromorphic Systems. Kluwer Academic Publishers, Boston 1998.

[5] Misha Mahowald. VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function. PhD. Thesis, California Institute of Technology Pasadena, California, 1992.

[6] 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. AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems. Vancouver.

[7] P.O. Pouliquen, A.G. Andreou, “Bit-Serial Address-Event Representation”. Proceedings of the 33rd Annual Conference on Information Sciences and Systems, Baltimore, March 1999.

[8] ANSI/TIA/EIA-644-A Telecommunications Industry Association, “Electrical Characteristics of Low Voltage Differential Signaling (LVDS)”, 2001.

[9] F.Gomez-Rodriguez, R.Paz, A.Linares-Barranco, M.Rivas, L. Miro, G. Jimenez, A. Civit. “AER tools for Communications and Debugging”. ISCAS-2006. Kos, Greece.

[10] Digital Clock Managers (DCMs) in Spartan-3 FPGAs. Xilinx Application note. XAPP462(v1.1). January 2006.