Inter-spike-intervals Analysis of Poisson Like Hardware Synthetic AER Generation

A. Linares-Barranco[1], M. Oster[2], D. Cascado[1], G. Jiménez[1],
A. Civit[1], and B. Linares-Barranco[3]

[1] Arquitectura y Tecnología de Computadores, Universidad de Sevilla,
Av. Reina Mercedes s/n, 41012-Sevilla, Spain
alinares@atc.us.es
[2] Institute of Neuroinformatics, UNI - ETH Zürich,
Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
[3] 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 images between chips, originally developed for bio-inspired image processing systems. Such systems may consist of a complicated hierarchical structure with many chips that transmit images among them in real time, while performing some processing (for example, convolutions). In developing AER based systems it is very convenient to have available some kind of means of generating AER streams from on-computer stored images. In this paper we present a hardware method for generating AER streams in real time from a sequence of images stored in a computer's memory. The Kolmogorov-Smirnov test has been applied to quantify that this method follows a Poisson distribution of the spikes. A USB-AER board and a PCI-AER board, developed by our RTCAR group, have been used.

1 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 pulse density modulation for coding information. Figure 1 explains the principle behind the AER basics. The Emitter chip contains an array of cells (like, for example, a camera or artificial retina chip) where each pixel shows 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-secconds). The density of pulses is proportional to the state of the pixel (or pixel intensity). Each time a pixel generates a pulse (which is called "Event"), it communicates to 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 also 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. This way, pixels with the same code or address in the emitter and receiver chips will "see" the same pulse stream. The receiver pixel integrates the pulses and reconstructs the original low frequency continuous-time waveform. Pixels that are more active access the bus more frequently than those less active.

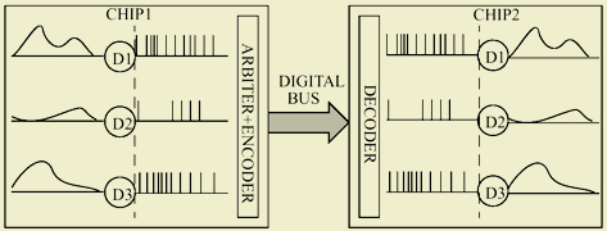

Fig. 1. Illustration of AER inter-chip communication scheme

Transmitting the pixel addresses allows performing extra operations on the images while they travel from one chip to another. For example, inserting properly coded EEPROMs allows shifting and rotation of images. Also, the image transmitted by one chip can be received by many receiver chips in parallel, by properly handling the asynchronous communication protocol. The peculiar nature of the AER protocol also allows for very efficient convolution operations within a receiver chip [2].

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 [3]. The goal of this community is to build large multi-chip and multi-layer hierarchically structured systems capable of performing complicated array data processing in real time. The success of such systems will strongly depend on the availability of robust and efficient development and debugging AER-tools. One such tool is a computer interface that allows not only reading an AER stream into a computer and displaying it on its screen in real-time, but also the opposite: from images available in the computer's memory, generate a synthetic AER stream in a similar manner as would do a dedicated VLSI AER emitter chip [4][6].

In the following sections we present the hardware implementation of one of the already existing methods for synthetic AER generation [9][10][11], and we evaluate the nature of the distribution of the events respect to the inter spike intervals (ISIs). For this analysis we have used a hardware USB-AER interface developed by RTCAR group.

2 Synthetic AER Generation by Hardware

One can think of many software algorithms that would transform a bitmap image into an AER stream of pixel addresses [5]. At the end, the frequency of appearance of the address of a given pixel must be proportional to the intensity of that pixel. Note that the precise location of the address pulses is not critical. The pulses can be slightly

shifted from their nominal positions because the AER receivers will integrate them to recover the original pixel waveform.

From the software methods already proposed, we present the hardware implementation in VHDL of the Random method. Some modifications have been made to reduce the resources needed in the FPGA.

This method sends a sequence of addresses to the AER bus without any buffer. Next event to be sent is selected in real-time from an image composed by *NxN* pixels, with *N* a power of *2*, and each pixel can have up to *k* grey levels, with *k=255* in this implementation. Each event needs a time to be sent. Let's call it time slot. The algorithm would implement a particular way of selecting the next address to be transmitted, through the AER bus, in the current time slot.

At each time slot an event could be placed in the AER bus, or not. This method uses a Linear Feedback Shift Register (LFSR) [7][8] for selecting the pixel of the image in charge of sending an event, and also to decide if the event is going to be sent or not. The LFSR has a resolution of *log(NxNxk)*, so the random number obtained for each time slot is divided into:

1. An address for selecting a pixel of the image and
2. A gray level value.

Thanks to the LFSR, each gray level value of each pixel is generated only once. If a pixel in the image has a value *p*, then the method will validate *p* events along time and will send those *p*, from the *k* possible, for this pixel. They will not be perfectly equidistant in time, but in average they will be reasonably well spaced. This method is very simple to be implemented in hardware. Next Sections explains in more details the implementation issues for this method.

3 Random Method

This method is an implementation of Linear Feedback Shift Register (LFSR) based random number generators. LFSR random number generators are based on a linear recurrence of the form:

$$x_n = (a_1 x_{n-1} + \ldots + a_k x_{n-k}) \bmod 2 \tag{1}$$

where k>1 is the order of the recurrence, a_k=1, and $a_j \in \{0,1\}$ for each j. This recurrence is always purely periodic and the period length of its longest cycle is 2^k-1 if and only if its characteristic polynomial

$$P(z) = -\sum_{i=0}^{k} a_i z^{k-i} \tag{2}$$

is a primitive polynomial over the Galois field with 2 elements.

With these premises and limiting the maximum number of address events necessary to transmit an image, we know the number of bits needed for the LFSR and the primitive polynomial. For this implementation, the limit corresponds with a 64x64 image of 256 gray levels, what implies a 20-bit LFSR.

The characteristics polynomial **P(z)** used for 20 bits is:

$$P(z) = z^{20} + z^{17} + 1 \tag{3}$$

which corresponds to the LFSR of Figure 2.

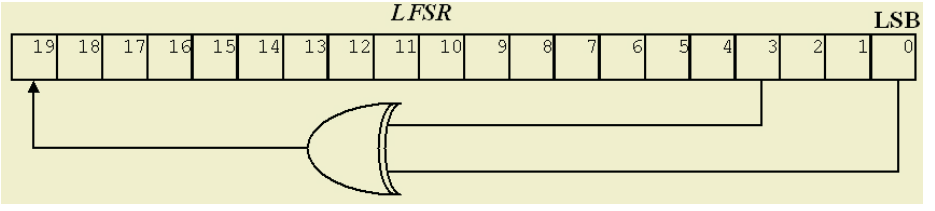

Fig. 2. Linear Feedback Shift Register for random synthetic AER generation

where all bits are '1' after a reset, what is the seed of the random numbers generator. This way, the 20-bit numbers are divided in two parts: the gray level correspond with the 8 more significant bits, and the address of the pixel in the image are the other 12 bits. The method works as follows:

- For each time slot, the LFSR generates a random 20-bit number.
- The 12 less significant bits are used to address the gray level of the image stored in memory.
- Once addressed that pixel, its gray level is compared with the 8 more significant bits of the LFSR.
- If the gray level of the pixel is greater or equal to the 8 MSB of the LFSR, an event is transmitted with the 12 LSB of the LFSR as the address.
- In the other case, no event is produced for this time slot.

The LFSR ensures that each possible event of each pixel is obtained from the LFSR only once per each $(2^{20}-1)$ time slots.

4 Inter-spikes-intervals Distribution Analysis

In this section we will compare the Inter-Spike-Intervals (ISIs) of this hardware synthetic AER generation method with the normalized distribution that it should have, using the Kolmogorov-Smirnov statistical test.

In neuro-inspired systems, signals can often be modelled by a Poisson distribution [12][13]. A Poisson distribution can be described by the following formula [14]:

$$P_n(T) = \frac{(\lambda T)^n}{n!} e^{-\lambda T} \tag{4}$$

where P is the probability of having n events the interval time T. The distribution of ISIs is the probability that no event occurs in the interval:

$$P_0(T) = e^{-\lambda T} \tag{5}$$

which is the exponential distribution.

An USB-AER board with a Spartan II 200 FPGA has been used for the Random method, configured with different gray values for the same pixel address. And a PCI-AER board that captures events and their timestamp, controlled through MATLAB, has been also used to capture the ISIs. Figure 3 shows the cumulative probabilistic distribution of ISIs: the expected exponential distribution versus the measured distribution generated by the Random method ISIs for gray levels 50 to 255, 10 by 10. For a high gray level, the distributions are close to each other, what implies that the spike train generated with the Random method is close to the Poisson distribution.

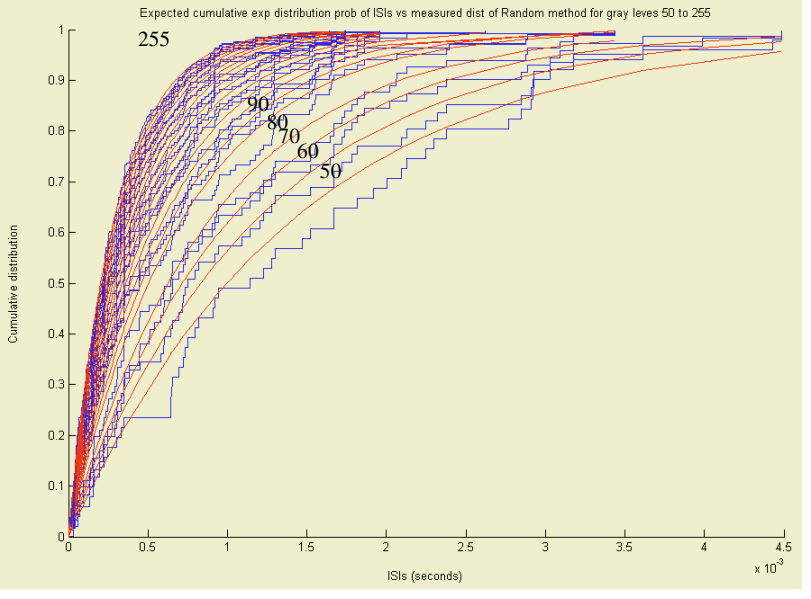

Fig. 3. Expected cumulative Exponential ISIs distribution versus measured ISIs distribution generated by the Random method for gray levels

We use the Kolmogorov-Smirnov (KS) test to quantify how good the observed distribution of ISIs follows the theoretical exponential distribution. Figure 4 shows the result of applying the KS test to the Random distribution obtained with different gray levels. The test is passed if the result is bellow 5%. It is shown that for small gray levels, behaviour is not much Poisson like. This is due to the LFSR because all the possible numbers obtained from the LFSR are used to produce a sequence of events for just one frame, resulting in only few different ISIs for a pixel of low intensity. We will address this problem by increasing the number of bit in the LFSR, so the shift register has a longer period than a frame and has a different initialization at the start of each frame.

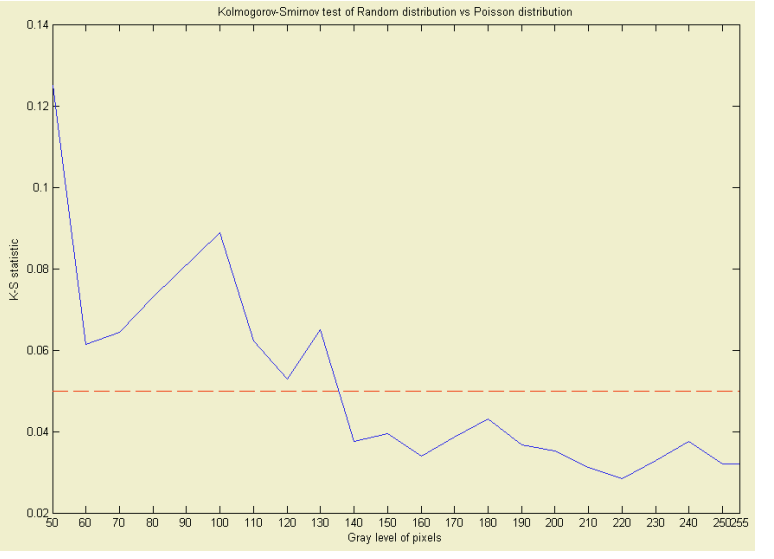

Fig. 4. Kolmogorov-Smirnov test results for Random method distribution

5 Conclusions

We have presented a hardware implementation of the Random method. We have demonstrated that it has a really well similar behaviour to the Poisson distribution. Therefore, it could be a very realistic method to be used for neuro-inspired systems, as AER systems. But for low frequency of events, this method becomes worse and worse for this kind of systems. So it could be improved in this way.

In contrast to existing AER generators like [15], this USB-AER board directly generates Poisson spike trains instead of sequencing raw spike trains.

The USB-AER board running with the Random method for AER Poisson synthetic generation, and the PCI-AER board with the MATLAB interface, are a useful tool for testing and debugging AER system. These two boards have been presented in the paper title "Test Infrastructure for Address-Event-Representation Communications" of this proceeding.

Acknowledgements

This work was in part supported by EU grant IST-2001-34124 (CAVIAR), and spanish grant 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] 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 2001. [www.ini.unizh.ch/telluride]

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

[5] A. Mortara, E.A. Vittoz and P. Venier. "A Communication Scheme for Analog VLSI Perceptive Systems", IEEE Journal of Solid-State Circuits. Vol. 30, No. 6, pp. 660-669, 1995.

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

[7] Pierre L'Ecuyer, François Panneton. "A New Class of Linear Feedback Shift Register Generators". Proceedings of the 2000 Winter Simulation Conference.

[8] Linear Feedback Shift Register V2.0. Xilinx Inc. October 4, 2001. http://www.xilinx.com/ipcenter.

[9] 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. http://www.atc.us.es/alinares

[10] 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.

[11] Alejandro Linares-Barranco, Gabriel Jimenez-Moreno, Antón Civit-Ballcels, and Bernabé Linares-Barranco. On Synthetic AER Generation. ISCAS 2004. Vancouver, Canada, May, 2004.

[12] P. Dayan and L. Abbot, Theoretical Neuroscience (MIT Press, Cambridge, MA, 2001)

[13] F. Rieke, D. Worland, R. de Ruyter van Steveninck, W. Bialek. "Spikes: Exploring the Neural Code". The MIT Press, 1999.

[14] J.R. Cogdell. "Modeling Random Systems". Pearson Prentice Hall, 2004.

[15] Dante, V. and Del Giudice, P. and Whatley, A. M. "PCI-AER Hardware and Software for Interfacing to Address-Event Based Neuromorphic Systems". The Neuromorphic Engineer, 2:(1) 5-6, 2005, (In Press)