

Biologically Inspired Adaptive Address-Event Representation Imager for Particle Localization

Peng Xu Yiming Zhai Pamela Abshire

Department of Electrical and Computer Engineering, Institute for Systems Research
University of Maryland, College Park, MD 20742
{pxu, ymzhai, pabshire}@isr.umd.edu

Abstract

This paper describes a biologically inspired imager designed for particle localization in a lab-on-chip system. Lab-on-chip systems demand efficient sensing and communication. The imager serves as the front-end of the system and collects optical information from particles nearby the imager surface without any intervening optics. The imager uses address-event representation (AER), inspired from the behavior of spiking neurons, to encode and transmit locations of the particles, which can then be used to control a micro-actuator to manipulate the particles. Particles appear darker than the surrounding areas; therefore the imager detects dark regions instead of bright regions, in contrast to conventional imagers. Adaptation is implemented to adjust the rate of event generation according to the activity level and available communication bandwidth. In addition, hot electron injection is used to store local nonvolatile memory that compensates pixel mismatch. The imager is composed of a 64×64 array of pixels of size $36 \times 36 \mu\text{m}^2$ and fill factor 20.1%. The chip is currently in fabrication.

Keywords: Biologically inspired, AER, contact imager, adaptation, mismatch, lab-on-chip.

1. Introduction

There is fast-growing interest in developing lab-on-chip systems that promise to provide easy, fast, and low-cost biochemical monitoring and testing. Lab-on-chip systems demand high system integration and low power consumption: sensing, communication, control, and computation are integrated together in one microsystem; low power is necessary for portable and long-term operation. These requirements demand non-conventional approaches.

Neural systems are engineering marvels of heterogeneous integration, efficiency and effectiveness. The human brain is capable of performing tremendously complex computation in real time while consuming only 12 W with efficiency of computation

of about 3×10^{14} operations per joule. The power supply in biological systems is approximately 100 mV while the bandwidth of the constituent components at the macroscopic scale is only a few hundred hertz. Typical current levels in biological structures are in the pico- to nano-ampere range [1]. While biologists proceed to elucidate the principles of information processing in biological systems, engineers have sought to abstract such knowledge in human engineered microsystems. Carver Mead coined the term neuromorphic electronic systems over a decade ago to characterize electronic computing systems similar to neural systems [2]. Since then silicon retina and silicon cochlea have been successfully designed and manufactured [3], [4].

In this paper we show how information processing principles from neural systems can improve our lab-on-chip system [5]. In our system, a contact imager is used to monitor particles, where samples are placed on the surface of the imager and images are taken without any optics. The images are transmitted to a receiver chip and additional signal processing is performed there. Relevant information is extracted and passed to other components such as microfluidic actuators which manipulate the particles. In this scenario, what are important to the system are not high-resolution images, but easy and fast access to the locations of particles. The background is often much larger than the particles, but provides little information to the system and wastes communication and computation resources. If an imaging system can locate the interesting areas and transfer only information about these areas, it can save much power and bandwidth.

We designed an adaptive AER contact imager for the aforementioned application. The imager generates the addresses of pixels which correspond to particles on the surface. To obtain a high dynamic range, we implement an adaptation mechanism which matches the “spiking” rate to a desired operating frequency. The paper is organized as follows: in Section 2, we briefly introduce AER; in Section 3, we describe the imager design; in Section 4, we analyze the performance; in Section 5, we summarize the work.

2. Address Event Representation

Address event representation (AER) is an event-driven data representation and communication protocol inspired by biology. It has been applied to several neuromorphic systems [3], [6]. The basic idea of AER is that each unit in a system triggers a request for transmission when its internal state has crossed a threshold; its address is transmitted onto a common bus once this request is granted. The internal state depends on external physical stimuli or inputs from other units. The activity level of each unit is then represented by the frequency at which its address is transmitted. If the activity is caused by an external physical stimulus, the intensity of the stimulus is encoded by the address transmission frequency. This protocol was inspired by the mechanisms used by neural systems of biological organisms, where spikes are used to encode and transmit information.

In biological systems, massive interconnections implicitly identify the spikes' origins, so there is no need to transmit an address. However, in VLSI systems, it is not feasible to implement so many direct connections. Thus a common communication bus is shared between chips and addresses are transmitted explicitly. Each transmission of an address is referred to as a spike. Since data transmission is only triggered by activity, the communication bandwidth is not distributed evenly to all units but biased towards those units with high activities. By using AER in the lab-on-chip system mentioned above, the communication bandwidth is allocated more efficiently to locations of interest (LOIs).

3. Imager Design

The imager is an event-driven asynchronous system. When a spike is generated, it issues a request *Req* to the receiver, sends the address of the pixel on the bus, and raises line *Data_valid* to indicate that the address is valid. It resets the request once it receives an acknowledgement *Ack* from the receiver. Each pixel in the array connects to a column (X) request line and a row (Y) request line. All row requests go to a row (Y) arbiter and all column requests go to a column (X) arbiter. The arbiters determine which request is to be granted and activate the corresponding *X_select* and *Y_select* signals. The select signals reset the internal state of the selected pixel. The address encoders transmit the X and Y address of the selected pixel on the address bus. The request lines, arbiters, *Req*, and *Data_valid* are all reset by the *Ack* signal. Adaptation circuits monitor the total number of Y and X requests, and convert it to a voltage used to adjust the spiking frequency.

3.1. Pixel Design

In the contact imager, particles occlude the photodetectors and appear darker than the surrounding areas. Therefore our imager detects and reports dark regions, in contrast to conventional imagers [7], [8]. The voltage across a capacitor stores each pixel's internal state, so in order to detect dark regions we integrate a current that increases as the light intensity decreases. Thus from an initial low voltage, the capacitor in a dark pixel charges faster, its voltage reaches the threshold sooner, and spikes are generated faster and more frequently than for a bright pixel. Fig. 1 shows the implementation of the pixel circuit.

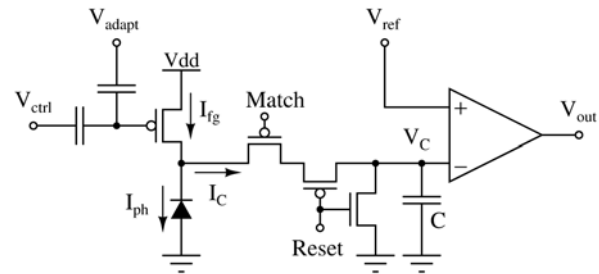


Fig. 1: Schematic of pixel.

A floating gate p type field effect transistor (pFET) is used to provide the total current I_{fg} . I_{ph} is the photocurrent generated by incident light on the photodiode. Their difference current I_C charges the capacitor C . The floating gate pFET allows for adaptation and mismatch compensation. A switch controlled by signal *Match* is off during mismatch compensation. The other switch between the photodiode and the capacitor controlled by signal *Reset* is off during reset so that V_C can be reset to ground. Once the reset period is over, V_C starts to increase as charges accumulate onto the capacitor from I_C . We call this photo-sensing period.

A differential pair with active load operates as a comparator to compare the capacitor voltage V_C to the reference voltage V_{ref} . The output of the comparator V_{out} is high when V_C is low after reset. When V_C increases above V_{ref} , V_{out} switches from high to low. This generates a spike request (described in Section 3.4). Fig. 2 shows a simulation of the pixel circuit.

3.2. Mismatch Compensation

The floating gate pFET sources a current from which photocurrent is subtracted. The difference current then charges the capacitor. Ideally pFET currents across the pixel array are identical so that the differences in charging currents reflect the differences in incident light. However the pFET currents are non-uniform because of threshold mismatch in the fabrication

process. We use hot-electron injection to compensate the mismatch [9].

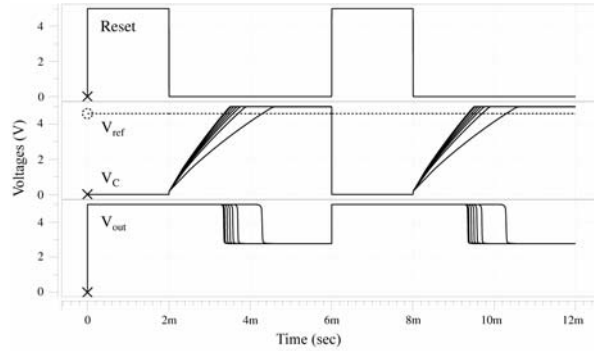


Fig. 2: Pixel simulation result. Top figure: reset signal, middle figure: capacitor voltage V_C , bottom figure: V_{out} . The photocurrents used in the simulation are 1, 10, 30, 50, 70, 100, 200 pA. V_C increases faster and V_{out} switches earlier for smaller photocurrent. ($V_{ctrl} = V_{adapt} = 4.02$ V, $V_{ref} = 4.6$ V)

During mismatch compensation, the pFET and photodiode are disconnected from the rest of the pixel. Then the pixel array is illuminated by a uniform strong light. Since photodiodes are inherently better matched during fabrication, we assume that there is no mismatch among them so that the photocurrents are identical. We set V_{ctrl} and V_{adapt} to values such that I_{fg} is close to I_{ph} , but smaller. Then the voltage at the drain of the pFET is pulled down, so there is a large voltage drop between the pFET's source and drain. This generates impact-ionized hot electrons that are injected onto the floating gate. The injected electrons cause the floating gate voltage to drop, which increases I_{fg} . This process continues until the floating gate current matches the photodiode current. Initially the pFET currents are non-uniform due to threshold mismatch across the pixel array. After compensation, all pFET currents are equal to the photocurrents. The charge stored on the floating node functions as a local memory and serves to compensate the mismatch.

3.3. Adaptation Circuits

The time from reset to spike is determined by

$$t = C(V_{ref} - V_{reset}) / (I_{fg} - I_{ph})$$

C , V_{ref} , and V_{reset} are fixed. I_{ph} is determined by the incident light. We can adjust t by varying I_{fg} . Two voltages V_{ctrl} and V_{adapt} control the floating gate voltage and therefore control I_{fg} . V_{ctrl} is set by an off-chip bias voltage, whereas V_{adapt} is set by an on-chip adaptation circuit which provides negative feedback from the pixel requests. Each row and column request controls a switch in series with a pFET current source, as shown in Fig. 3. When there are more requests than

desired, the current from the pFET sources is greater than the nFET sink. The voltage V_{adapt} increases, which reduces I_{fg} and consequently reduces the spike request rate. Together V_{ctrl} and the gate voltage of the nFET V_{ad_bias} determine the preferred operating frequency. Fig. 4 shows a simulation as the request number increases from 0 to 64. V_{adapt} changes from 3.255 to 3.375 V, providing sufficient voltage change for adaptation.

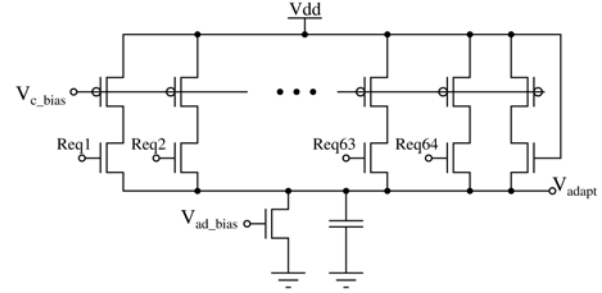


Fig. 3: Adaptation circuit.

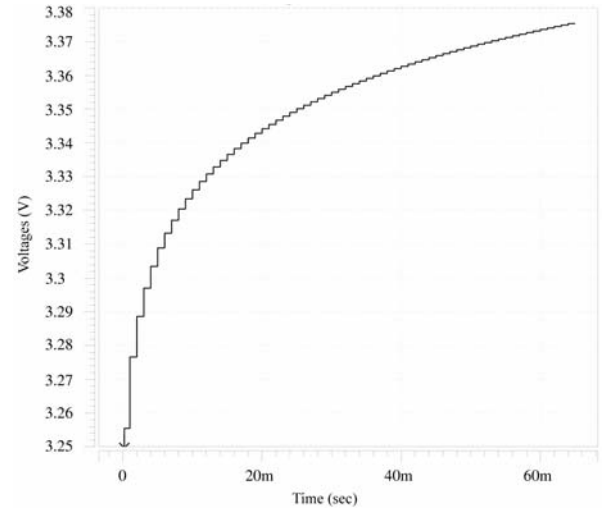


Fig. 4: Adaptation simulation result.

3.4. Readout Circuits

The readout circuits are similar to [3]. When V_{out} from a pixel drops from high to low, it initiates a request. First it raises the Y Request line that runs through the row. This request is processed by the Y arbiter. If selected, the Y Select line is activated and pulled low, and the Y address of this row is placed on the address bus by the Y address encoder. The active Y Select allows all requesting pixels in this row to raise X Request lines running through the column. One of the high X Requests is selected by the X arbiter. The X Select of the selected pixel is pulled low, and its X address is placed on the bus by the X address encoder. At the same time, $Data_valid$ is raised to notify the receiver. The activated X select and Y select together pull up $Reset$ to reset the pixel. The receiver can now

acquire the address from the bus and set *Ack* high. The pixel is reset before the rest of the readout circuitry to avoid duplicated requests. *Ack* resets all *X Request* lines, the *X* arbiter, the *X* address encoder, and *Data_valid*. Together withdrawal of the top level request from the *X* arbiter plus *Ack* reset *Y Request*, the *Y* arbiter and the *Y* address encoder. Withdrawal of the top level request from *Y* arbiter pulls down *Ack*. This ends the data transfer cycle. Now pixels can initiate requests again.

The arbiter is a binary tree of two-input arbitration cells. Each cell receives two requests from below and a select from above. It transmits a request to the next higher level of the tree to indicate that there is at least one request, and passes down two select signals to indicate which request has been chosen. The cell is implemented by an OR gate, a selection circuit, and a steering circuit [3]. Six-bit binary encoders are used for both *X* and *Y* address encoders.

4. Performance analysis

Given the layout of our pixel, we estimate that the photocurrent is between 0 and 500pA. With a total current of 500pA from the floating gate, we estimate that the interval from reset to spike is

$$t_{\min} \approx CV_{\text{ref}} / (I_{\text{fg}} - I_{\text{dark}}) \approx CV_{\text{ref}} / I_{\text{fg}} \approx 920 \mu\text{s}$$

For a difference in photocurrents of 0.1pA, the relative difference in the interval from reset to spike is

$$\Delta t_{\min} \approx V_{\text{ref}} C \Delta I_{\text{ph}} / I_{\text{fg}}^2 = t_{\min} \Delta I_{\text{ph}} / I_{\text{fg}} = 0.18 \mu\text{s}$$

To accurately distinguish two pixels reset simultaneously with a 0.1pA difference in photocurrent, the communication bus needs to run at a minimum speed of 5.5MHz.

When running at high enough speed, the receiver can accurately recover the spike rate of each pixel and reconstruct the light intensity at each pixel. However, that is not the main goal for this class of imager. We use the imager in a setting where the pixels have sparse activity and the LOI needs to be identified. The receiver must be able to tolerate the spike delay caused by collisions of pixel requests and use the approximate inter-spike time to determine which regions are most relevant to its operation. If we limit the system load such that at most α percent of the $n \times n$ pixels generate spikes during the interval t_{\min} computed above, all spikes can be serviced when the readout runs at $\alpha\% \times n \times n / t_{\min}$ Hz. For example, a 64 by 64 imager with 50% load requires 2MHz. In fact, the LOI is much smaller than the surrounding area; therefore the load can be very small, reducing the requirements of the communication channel substantially. We can also determine the maximal load given the available

bandwidth, and use adaptation to adjust the operating frequency of the imager for full use of the channel.

5. Summary

We described an adaptive AER contact imager for lab-on-chip systems. The imager locates and transmits the LOIs which correspond to particles. Spiking rate can be adjusted according to lighting conditions and available communication bandwidth. Mismatch compensation is also implemented at the pixel level. In future work, we will continue to identify and implement efficient computations needed for lab-on-chip systems using information processing principles inspired by biology.

6. References

- [1] P. A. Abshire and A. G. Andreou, "Capacity and energy cost of information in biological and silicon photoreceptors," *Proceedings of the IEEE*, vol. 89, pp. 1052-1064, July 2001.
- [2] C. A. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.
- [3] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Kluwer Academic Publishers, 1994.
- [4] L. Watts, D. A. Kerns, R. F. Lyon, and C. A. Mead, "Improved implementation of the silicon cochlea," *IEEE J. Solid-State Circuits*, vol. 21, no. 5, pp. 692-700, May 1992.
- [5] N. Reeves, Y. Liu, N. M. Nelson, S. Malhotra, M. Loganathan, J. M. Lauenstein, J. Chaityupatumpa, E. Smela, and P. A. Abshire, "Integrated MEMS structures and CMOS circuits for bioelectronic interface with single cells," in *Proc. of ISCAS*, Vol. 3, pp. 673-676, 2004.
- [6] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, "A biomorphic digital image sensor," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, Feb. 2003.
- [7] T. A. Holz and J. Harris, "An integrate and fire pixel with contrast outputs for a CMOS imager," in *Proc. of ICCSS*, 2003.
- [8] R. A. Blum, C. S. Wilson, P. E. Hasler, and S. P. DeWeerth, "A CMOS imager with real-time frame differencing and centroid computation," in *Proc. of ISCAS*, Vol. 3, pp. 329-332, 2002.
- [9] P. Hasler, B. A. Minch, C. Diorio, and C. Mead, "An autozeroing amplifier using pFET hot electron injection," in *Proc. of ISCAS*, Vol. 1, pp. 325-328, 1996.