EFFICIENT EVENT-DRIVEN FRAME CAPTURE FOR CMOS IMAGERS

[1]Mohamad Susli, [1]Farid Boussaid, [2]Chen Shoushun, and [2]Amine Bermak

[1]School of Electrical, Electronic and Computer Engineering
The University of Western Australia, Perth Australia
[2]Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, SAR.

ABSTRACT

In this paper, we present an efficient event-driven frame capture technique for CMOS image sensors. In the proposed scheme, read-out is not carried out systematically and sequentially for all pixels. Instead it is initiated by each individual if its value has changed since the last frame capture. This approach is shown to lead to an efficient use of the transmission bandwidth, low power operation and improved imaging quality. Results from a 0.35µm CMOS implementation are presented.

1. INTRODUCTION

Rapid advances in semiconductor industry fabrication CMOS process have enabled the concept of a camera-on-a-chip or CMOS imager [1]. The fully integrated single-chip camera features image capture and built-in image processing capabilities such as noise reduction or motion detection [1]. Fabricated in a standard CMOS process, it exhibits significantly reduced development and fabrication costs, while its high miniaturization level makes it ideal for integration in a wide range of consumer imaging products like mobile phones and PDAs to name a few.

In a conventional digital CMOS camera, images are formed by scanning sequentially a photosensitive array [1]. Each pixel value is read-out after a fixed integration period, during which photo-generated electron-holes pairs are collected. The amount of signal generated by each pixel will depend on the amount of light that falls on the photosensitive cell as well as on the duration of the integration period. This conventional read-out technique is not adapted to high resolution imagers, which comprise a large number of pixels; this, because a systematic sequential scan of all pixels would lead to excessive power dissipation since the scanner is always active. Furthermore, this serial scanning of large arrays would dramatically reduce the frame rate, making real-time processing difficult to achieve.

To overcome these limitations, a promising solution is to adopt a pixel-driven read-out strategy, in which pixels are only read-out if their values have changed since the last frame capture. In this scheme, active pixels (i.e., pixels with relatively large inter-frame intensity variation) will be favored and granted access to the bus more frequently than less active pixels, which will in turn consume much less communication bandwidth. In terms of power consumption, this frame capture strategy is also more efficient than the conventional fixed time-slot (synchronous) allocation of resources; this because not all pixels are likely to require computation/communication resources at the same time, hence there is no waste of resources. In the next section, we will review the adopted event-driven frame capture technique. Section 3 describes its proposed VLSI implementation in a 0.35µm CMOS process. Results are discussed in Section 4 and conclusions are drawn in Section 5.

2. EVENT-DRIVEN FRAME CAPTURE

The adopted read-out strategy is based on a biologically inspired data representation referred to as Address-Event-Representation or AER. It is an event driven communication protocol, modeled after the transmission of neural information in biological systems [2][3][4][5]. AER is used to transmit "events" through a single communication channel. Events are generally in the form of a spike or a pulse. They are characterized by a location (address) and the time of occurrence. In the case of an image sensor, the communication channel will be an asynchronous digital bus (Figure 1). The address will identify one particular pixel of the array whereas the time of the event will here be defined as the time at which a pixel has reached a given threshold voltage. Each time an event occurs, a pixel generates a spike and communicates asynchronously with a peripheral arbiter, which will take the pixel address and place it on the bus (Figure 1). As a result, the asynchronous bus will carry a flow of pixel addresses. Access to the bus is allocated according to pixel demand. At the receiver end of the bus, address and time information are combined to retrieve the original data (e.g. pixel brightness value).

The row arbiter may receive several bus requests at the same time. After arbitration, a single active row is selected. Next, column arbitration grants bus access to those pixels that are active within the selected row. The arbiters prevent collisions on the output bus but at the cost of larger delays in output bus access for queuing pixels. To reduce the number of collisions and hence timing errors, we propose to allow a pixel to request read-out only if the inter-frame pixel value difference exceeds a predefined threshold. As a result, frame capture can be speed up and the number of pixel requests can be reduced.

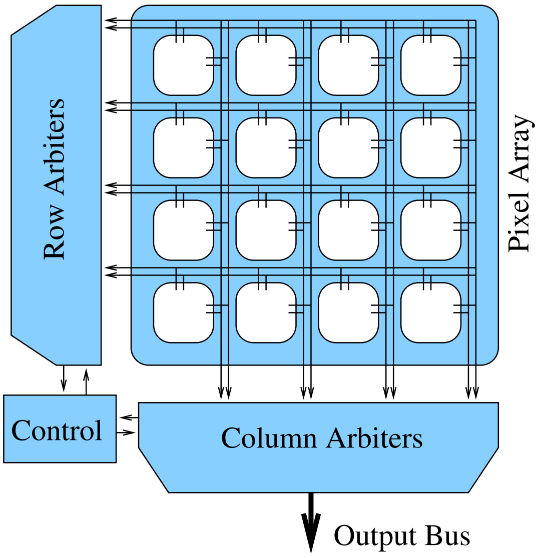

Figure 1. Imager architecture

3. VLSI IMPLENTATION

The key element in the AER communication protocol is the current-feedback event generator [6] shown in Fig. 2. This element is incorporated in the pixel cell and is responsible for requesting access to the output bus. It operates as follows. Upon reset, the photodiode capacitance is charged to V_{DD}, then allowed to discharge with the incident optical illumination; this until it reaches the threshold voltage of the inverter formed by (M3-M4). At this point, a spike or pulse is generated at the output (*Event* signal in Figure 2). The time to the first spike (TFS) encodes pixel brightness [7], with brighter pixels firing first. Figure 3 depicts the circuitry used to block requests if the pixel value has not significantly changed since the last frame capture. The circuitry operates as follows. First, a global reset signal discharges the sampling capacitors which are used to represent the Time to First Spike for the current and previous frames, represented by voltages V_1 and V_2, respectively. At each pixel reset (i.e. a frame transition), current and previous sampling capacitors values are updated. The circuit then waits for an event to occur. Simultaneously, a sampling capacitor is charged with a ramp voltage (V_{sweep}) to measure the new pixel value, which is encoded in the time required to generate a new spike (Figure 2). Each time an event occurs, V_1 is compared to V_2, and if they are equal within a window of tolerance, the pixel self-resets asynchronously without generating a row-request. Otherwise, the event blocking circuitry will not interrupt the normal handshaking procedure. After self-reset, the pixel remains dormant until the next integration phase. A single transition can thus occur per pixel during a single integration phase, thereby greatly reducing power consumption with an average current of 10nA/pixel, which is 3 orders of magnitude lower than conventional spiking pixels [7].

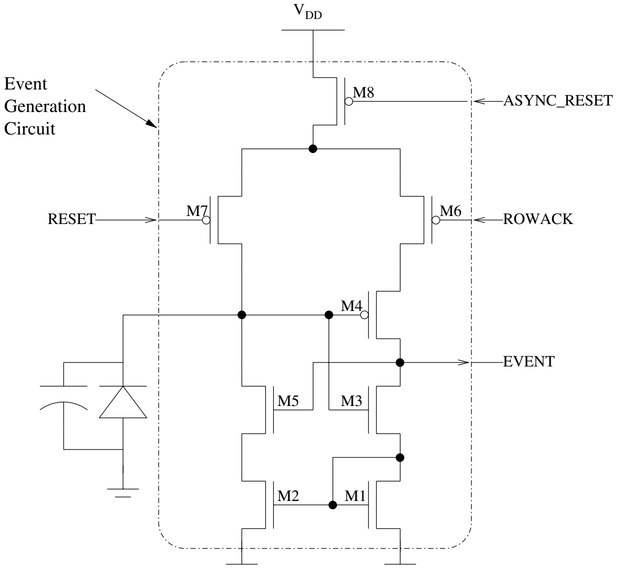

Figure 2. Event-generator circuit.

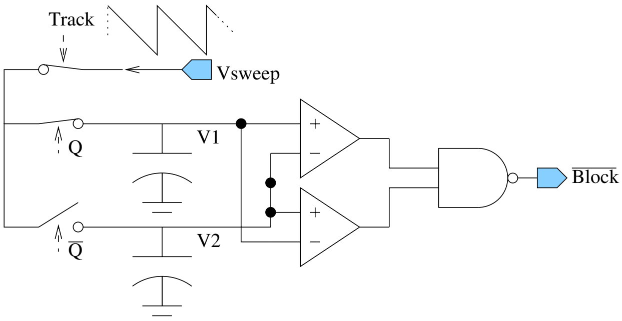

Figure 3. Event Blocking Circuitry.

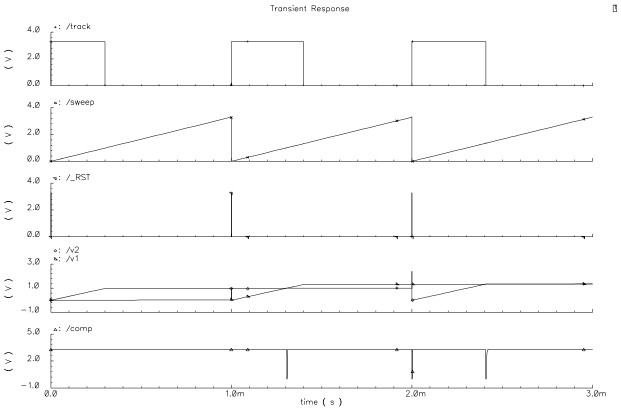

Figure 4. Control signals for the Event Blocking circuitry.

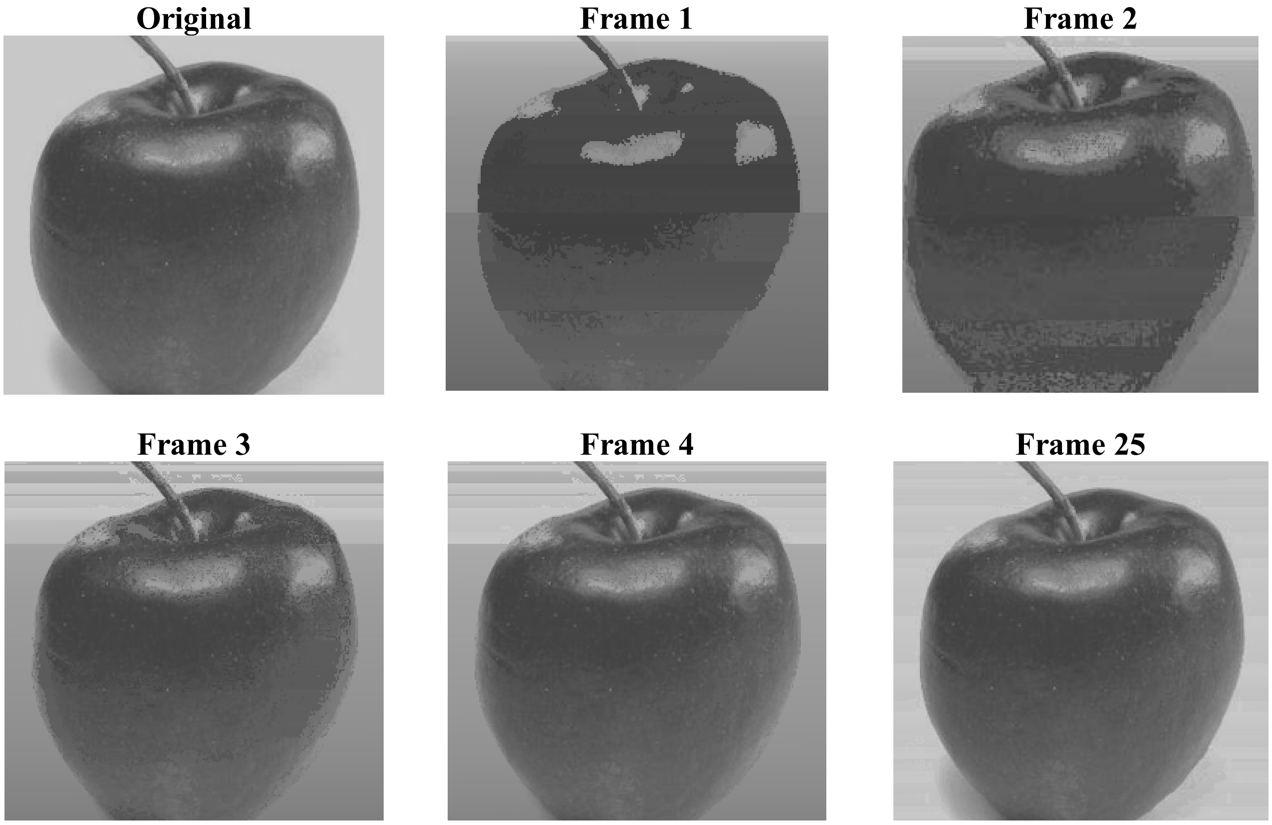

Figure 5. Pixel-driven frame capture as a function of the number of the acquired frames.

Figure 4 shows the control signals associated to the event blocking circuitry. In the first frame, the reference sampling capacitor is reset and thus the request is not blocked and is processed. The next frame has the two sampling capacitors' voltages within the window of the comparators, blocking the event and self-resetting the pixel. The final frame has a long time and thus a different end-of-frame voltage. The event is not blocked and is allowed to process as normal.

4. RESULTS AND DISCUSSION

Figure 5 shows a "still input" image, which is processed by the proposed event-driven frame capture technique. This corresponds to the worst case scenario since we have a dark object on a light background. As a result, dark pixels will request bus access relatively at the same time, the same applies of course to the brighter pixels. The first frame illustrates the effect of collisions for bright and dark pixels. Image quality is seen to improve for subsequent frames; this because the new acquired frame can be compared to the previous one. As a result, less pixels will request arbitration, thus reducing arbitration delays. Note that the collision problem is more acute for the brighter pixels (i.e. background) this because they fire after a very short time as compared to darker pixels. Figure 6 shows the Mean Squared Error (MSE) versus the number of acquired frames. It can be seen that the proposed pixel-driven frame capture technique is successful in reducing timing jitter, while making more efficient use of the transmission bandwidth.

The adopted pixel-driven frame capture technique is especially well suited to motion detection and tracking applications [8][9]. To illustrate this, consider for instance the case of an incoming vehicle's headlights in a low-lit environment. The first few pixels to fire and which are not blocked will correspond to be the new positions of the headlights; with the actual addresses of the firing pixels available on the output bus.

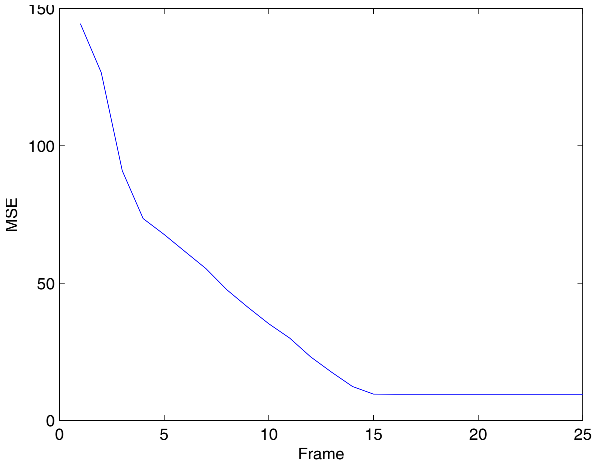

Figure 6. Mean squared error as a function of the number of acquired frames.

The proposed pixel-driven image capture technique was implemented using a 0.35μm CMOS process. The pixel layout is shown in Figure 7. The photo-sensing element, seen on the top left corner, is an n+-diff/p-well photodiode chosen for its high quantum efficiency. The handshaking circuitry is at the top right corner while the comparator/switches/NAND gate combination is at the bottom right. The sensitive nodes are the terminals of the two sampling capacitors, which hold the previous and current frame values. The layout was designed in a careful manner to limit parasitic capacitive coupling. The size and value of each sampling capacitor (55fF each) is a trade-off between charge injection and fill-factor. The total pixel size is 30×30µm for a fill factor of around 15%.

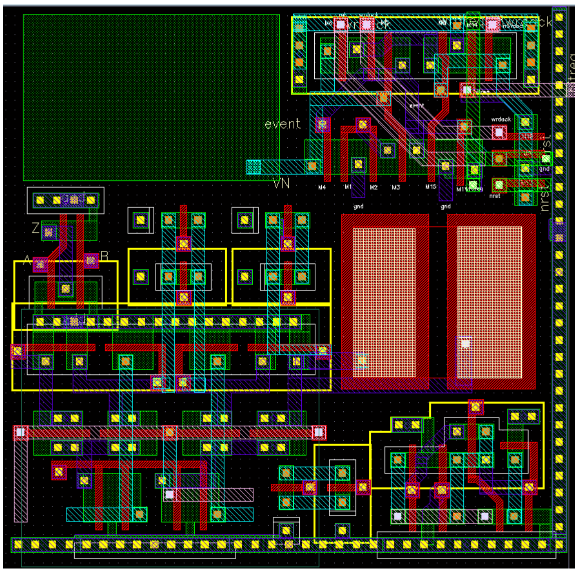

Figure 7. Pixel Layout.

5. CONCLUSION

In this paper, an efficient event-driven frame capture technique is proposed for CMOS image sensors. In the proposed scheme, read-out is initiated by each individual pixel and is only carried-out if its value has changed since the last frame capture. This frame capture technique enables low power operation, efficient use of the transmission bandwidth, and leads to a reduction in timing jitter, which translates in improved imaging quality. A 0.35µm CMOS implementation is presented together with results.

6. ACKNOWLEDGMENTS

This work was supported in part by a grant from the Australian Research Council.

REFERENCES

[1] E. Fossum, "CMOS image sensors: Electronic camera-on-a-chip," IEEE Trans. Electron Devices, vol. 44, no. 10, pp. 1689-1698, 1997.

[2] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. CAS II*, vol. 47, no. 5, pp. 416- 434, 2000.

[3] F. Van Rullen, S.J. Thorpe, "Rate Coding Versus Temporal Order Coding: What the Retinal Ganglion Cells Tell the Visual Cortex," *Neural Computation*, no. 13, pp. 1255-1283, 2001.

[4] M. Sivilotti, "Wiring considerations in analog VLSI systems with applications to field programmable networks," *Ph.D. dissertation*, California Institute of Technology, Pasadena, 1991.

[5] M. A. Mahowald, "VLSI analogs of neuronal visual processing: a synthesis of form and function," *Ph.D. dissertation*, California Institute of Technology, Pasadena, 1992.

[6] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, "A biomorphic digital image sensor," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 281-294, 2003.

[7] S. Chen and A. Bermak "A low power CMOS imager based on time-to-first-spike encoding and Fair arbitration," in *IEEE International Symposium on Circuits and Systems ISCAS2005*, Kobe, Japan, pp. 5306-5309, 2005.

[8] R. Etienne-Cummings, M.-Z. Zhang, P. Mueller, and J. Van der Spiegel, "A foveated silicon retina for two-dimensional tracking," *IEEE Transactions on Circuits and Systems II*, vol. 47, pp. 504-517, 2000.

[9] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision-processing systems," *IEEE Transactions on Circuits and Systems I*, pp. 1064-1071, 1999.