A CMOS vision sensor with on-the-fly histogram equalization using TFS encoding and AER read-out

Chen Shoushun†, Amine Bermak† and Dominique Martinez‡
†Hong Kong University of Science and Technology, EEE Department
Clear Water Bay, Kowloon, Hong Kong, SAR.
‡LORIA-CNRS, Vandoeuvre-Les-Nancy 54506, France.
Email: dazui, eebermak@ust.hk,dominique.martinez@loria.fr

***Abstract*— This paper presents a CMOS vision sensor performing image capture and on-chip histogram equalization. Both image capture and the processing are based on time-to-first spike encoding combined with Address Event Representation (AER) read-out scheme. The proposed approach not only provides a compact and efficient VLSI implementation but also eliminates voltage/current domain analog processing and hence provides immunity to dynamic range reduction caused by low voltage operation in advanced sub-micron technologies. The pixel values are read-out using an asynchronous handshaking type of read-out while the processing is carried out using simple and yet robust digital timer occupying a very small silicon area ($0.1 \times 0.6mm^2$ in $0.35\mu m$ CMOS). Low power operation (10nA per pixel) is achieved since the pixels are only allowed to switch once per frame and only active pixels are allowed to asynchronously access the bus. The prototype was implemented in $0.35\mu m$ CMOS technology with a silicon area of $3.1 \times 3.2mm^2$.**

I. Introduction

CMOS image sensors are gaining an increasing attention due to their inherent advantages of low power, low cost and more importantly their ability to integrate image capture together with on-chip image processing. This is becoming even more possible due to the aggressive device scaling found in today's CMOS technologies which has opened the door to even more on-chip signal processing. CMOS imagers are currently providing real "Camera-on-a-chip" solutions [1] together with more complex image processing such as image compression, motion and edge detection [2]. A particularly interesting processing which is required as a pre-processing stage in many applications is image histogramming. Indeed, a number of applications such as face recognition and finger print identification, the captured image under different illumination environment need to be preprocessed to explore the distinction of features in the scene. This will increase the efficiency and accuracy of image classification and interpretation. Histogram equalization is traditionally performed by first scanning the image into a digital memory and then processing it sequentially. This requires extensive computations, storage and additional processing time. In most real-time applications such delay is unacceptable as histogram equalization is only a pre-processing stage. In this paper, we propose to implement a CMOS vision sensor based on Address Event Representation (AER) read out protocol and on chip histogram equalization performed while reading out the pixels. In our imager, pixels are naturally sorted according to their illumination levels and hence histogram equalization can be easily performed by segmenting the pixels into any desired quantization levels. In addition, in our proposed architecture, each pixel is only permitted to switch once per frame which reduces dynamic power consumption and the bandwidth resulting in improved performance. The remainder of the paper is organized as follows. Section II introduces the pixel operation, the imager architecture as well as the histogram equalization circuity. Section III introduces the results and VLSI implementation. Section IV presents a conclusion.

II. Vision sensor based on pulse width modulation

A. Pixel description

The proposed pixel is shown in figure 1. It contains a reverse biased photodiode P_d with its internal capacitance C_d, a reset circuit, composed of the parallel combination of the PMOS transistors *M1* and *M2* followed by a current feedback event generator *(M3-M7)*. Transistors *(M8-M14)* are used in order to implement the asynchronous communication with the column and row AER circuit.

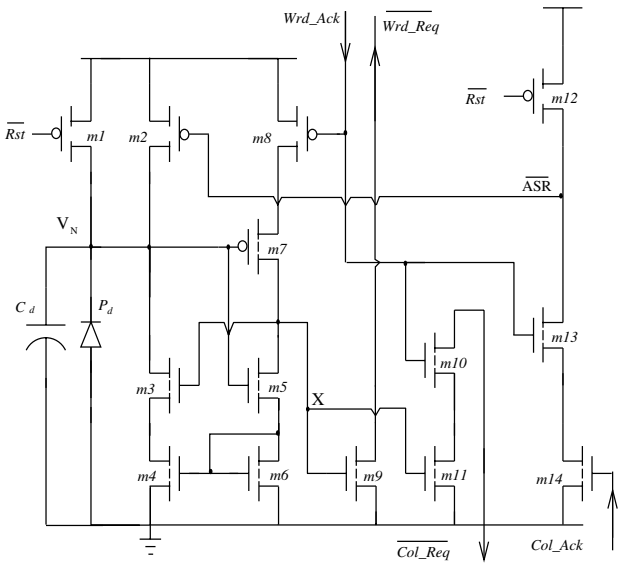

Fig. 1. Pixel Schematic

A global reset "$\overline{Rst}$" will precharge the capacitor C_d to V_{DD}. After this reset phase, the light falling onto the

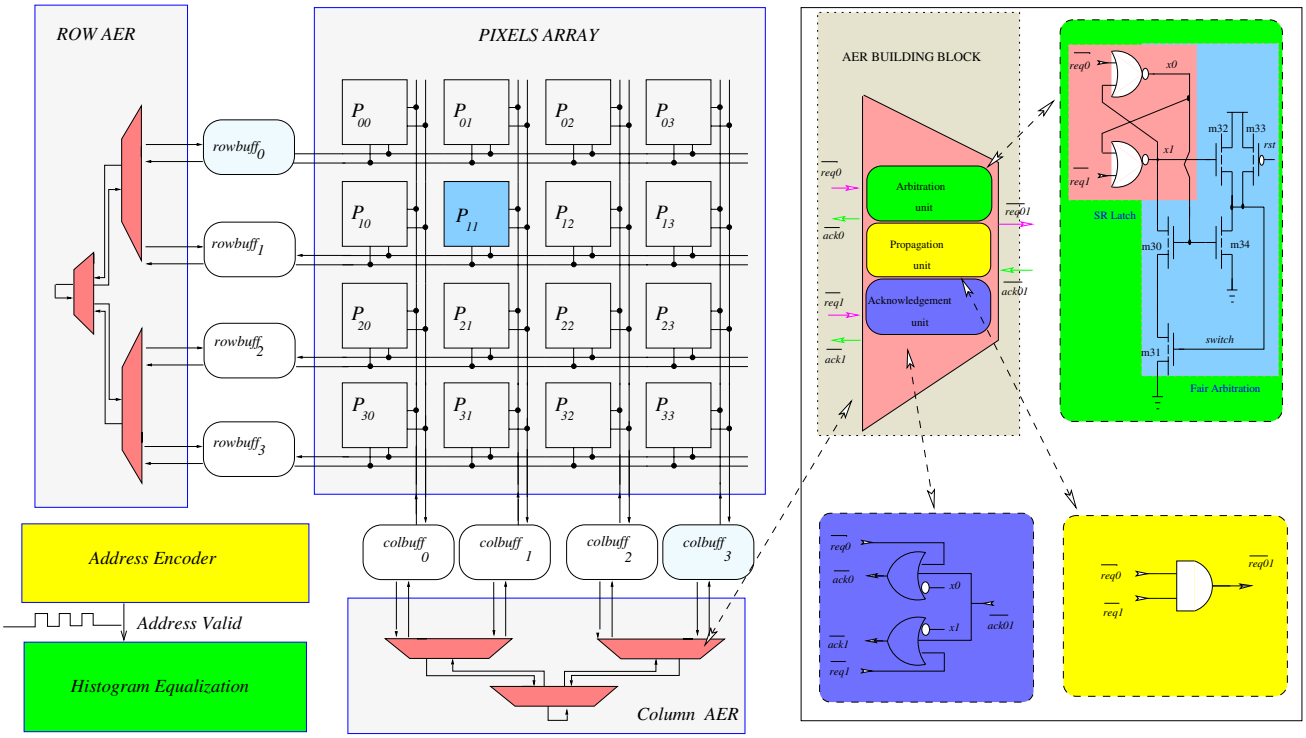

Fig. 2. Vision Sensor Architecture

photodiode starts the integration process. After a time interval, the voltage across the photodiode V_N reaches the threshold voltage of the inverter (*M5,M7*) and an event is generated at the output of the inverter (node "X"). A new upcoming event within the currently processed row will be blocked until the processing of the row is finalized. This is realized by transistor *M8* which will block the event when the row acknowledgment is active. If the row is not being processed, the event will generate an active low request "$\overline{WrdReq}$" through transistor *M9* which is then transmitted to the row AER. Many such requests can be received by the row AER at the same time and an arbitration process is required. After arbitration only one request is acknowledged through "$WrdAck$" generated by the row arbiter. This will turn on transistor *M10* and an active low request signal "$\overline{ColReq}$" is sent by all fired pixels within the acknowledged row. Another column arbitration process will take place. Once a pixel is acknowledged by "$ColAck$", transistor *M14* is turned on and the node "$\overline{ASR}$" is pulled down. This will switch on transistor *M2* and the voltage across the photodiode is recharged to V_{DD}. The event "X" as well as the request "$\overline{WrdReq}$" to the Row AER is killed. The voltage at node "$\overline{ASR}$" will be held low until a new global reset signal "$\overline{Rst}$" is received signaling a new frame capture. One can note that an asynchronous self-reset process is realized within each pixel using only 3 transistors (*M12-M14*) instead of an RS latch by 10 transistors as is the case in previous designs [3]. This greatly reduces the number of transistors thus improves the fill factor. To avoid the charge loss on node "$\overline{ASR}$" due to leakage current, transistor *M12* is made with relative large channel length. A current feed back event generator similar to that recently proposed in [4] is employed in our pixel. The positive current feedback (*M3, M4* and *M6*) is activated when V_N drops nearby the threshold voltage of the inverter. This greatly reduces the short circuit power consumption by speeding up the transition of V_N. Simulation results showed that the current consumption is about 10nA per pixel, which is 3 orders of magnitude lower compared with that of the spiking pixel.

B. Imager Architecture

The architecture of our imager is shown in figure 2. The imager includes an array of 128×128 pixels, row and column AER for event read out, row and column address encoder to generate the address of the acknowledged event, column and row buffers as well as histogram equalization circuitry.

Pixels are organized into rows and columns sharing the same request and acknowledgment buses. Within each pixel the time required to reach the threshold voltage of the inverter and hence to generate the event can be interpreted as a pulse width modulated (*PWM*) signal which is inversely proportional to the photocurrent. In this *PWM* coding scheme, the illumination received by each pixel is coded using a single pulse. This represents a major advantage as switching activity is reduced to only a single transition in each frame for each pixel, thus allowing for lower power consumption and reduced switching noise. When one or more pixels within a row fire, their respective row will send a request "$\overline{WrdReq}$" to the Row AER. The Row AER may receive several requests at the same time. After arbitration, only one row will be acknowledged by "$WrdAck$". The fired pixels within the acknowledged row wil

send request "$\overline{ColReq}$" to the column AER. Instead of waiting for the Column AER to acknowledge the requests one by one, the Column Buffer will hold their requests and acknowledge back concurrently. This improves the processing speed of the column AER by avoiding charging and discharging the large capacitance associated with the column buses. The column AER only needs to handle the requests held by the column buffers. The combination of Row and Column acknowledgements is exploited by the row and column encoders in order to generate the address of the active pixel. The row buffer will deassert the previous "$\overline{WrdReq}$" which enables the Row AER to start another round of arbitration even while the column AER is still processing the current row. A row and column pipelining is hence implemented enabling a reduced arbitration delay.

C. Histogram Equlization

In a PWM and AER based imager, pixels with higher illumination will fire earlier compared to pixels with lower illumination and hence access to the bus is granted first to pixels with higher illuminations. This will sort pixels within the array from bright to dark pixels. Histogram equalization can therefore be performed simply by associating the same quantization level to a number of pixels firing within a given time slot. In our imager, we equally segment the 128×128 pixels into 256 bands each of which belongs to an intensity level varying from 0 to 255. Thus the equalized image will have a uniform intensity histogram (64 pixels in each band). This is realized using two counters. The first counter uses the acknowledgment signal of the column AER as its clock input. This first counter will generate an active pulse after 64 pulses have been received, which is then used to drive the second counter. The second counter provides the data output of histogram equalization which is incremented every 64 pulses. In fact this is realized using a down counter as illumination is inversely proportional to the pulse width.

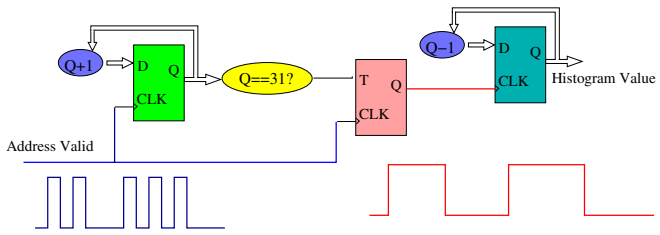

Fig. 3. Histogram equalization circuitry

Figure 3 shows the diagram of histogram equalization circuitry. "$AddressValid$" signal received from the column AER, indicates a pixel within a certain row has just been processed. It is used as the clock signal to drive a 5 bit counter which will toggle a T filp-flop every 32 cycles. The output of the T flip-flop is then used to drive a 8 bit de-counter which will decrement by 1 once 64 pixels have been processed. The 8 bit counter value combined with the pixels address constitute the output of histogram equalized image.

III. VLSI Implementation and Results

A prototype was designed using $0.35\mu m$ 1-poly, 5-metal CMOS process. Figure 4 shows the microphotograph of the chip including all the building blocks. The chip occupies an area of $3.1 \times 3.2mm^2$ with only $0.1 \times 0.6mm^2$ dedicated to the histogram equalization circuit.

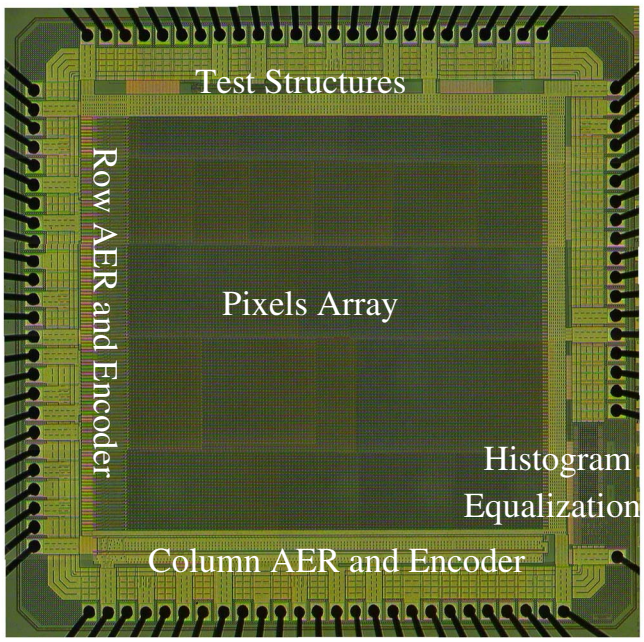

Fig. 4. Microphotograph of the chip.

Figure 5 shows the output of the AER imager with and without equalization. In order to illustrate more clearly the pixels distribution with and without histogram equalization of the test images, we also plotted the histogram of all corresponding images. One can note that the same image output is obtained when using histogram equalization (column 3) regardless of the illumination levels in the originally acquired image (column 1 and column 2). The obtained histogram equalized output images are very comparable to the ones obtained using Matlab simulation (column 4). One can note that due to the (1/x) relationship between the pulse width signal and the photocurrent, the same timing delay introduced by AER will have less effect on darker pixels as compared to brighter ones. In addition, row based mismatch is introduced due to the greedy nature of AER protocol. This can be reduced in the future by using faster hierarchical and higher radix arbitration schemes.

IV. Conclusion

In this paper, we reported the VLSI hardware implementation of a CMOS vision sensor with on chip histogram equalization circuitry. The image array is based on time to first spike encoding and the pixels are read out asynchronously

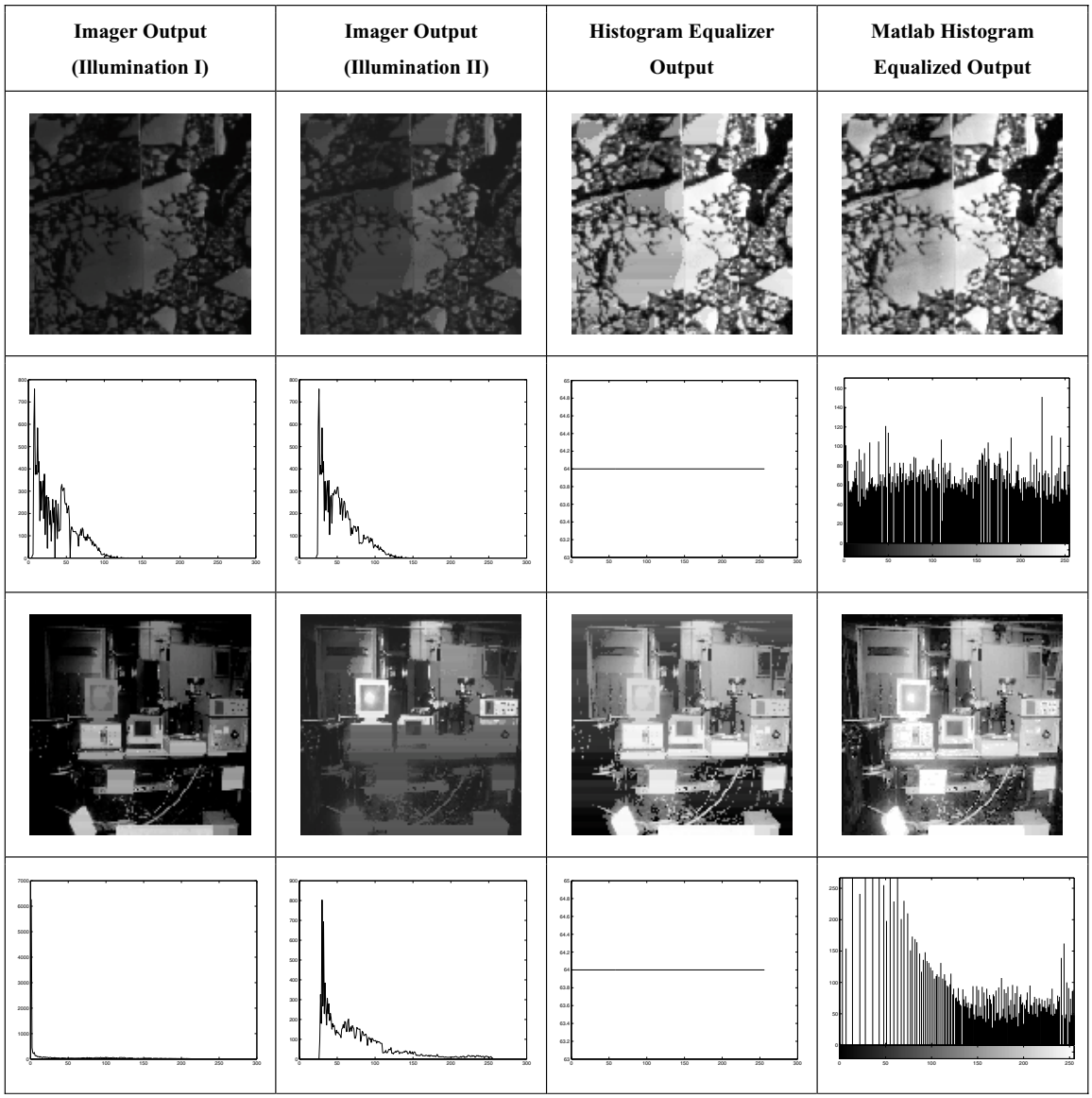

Fig. 5. Output of the AER imager. Column 1 and column 2 are images acquired at two different illuminations. Column 3 and 4 are the histogram equalization outputs from the imager and Matlab simulation, respectively.

according to their illumination levels. Low power operation (10nA per pixel) is achieved since the pixels are only allowed to switch once per frame and only active pixels are allowed to asynchronously access the bus. A prototype chip including 128×128 pixels, AER read-out and histogram equalization circuitry was implemented in $0.35\mu m$ CMOS technology with a silicon area of $3.1 \times 3.2mm^2$. The histogram equalization circuit was designed using standard cell library and automatic placement and routing technique. It occupies only a very small fraction of the total silicon area ($0.1 \times 0.6mm^2$). Histogram equalization performed on sample images with poor contrast shows enhancement of the image quality.

ACKNOWLEDGMENT

The work described in this paper was supported by a grant from the Research Grant Council of HK SAR, China (Project No HKUST6148/03E).

REFERENCES

[1] E.Fossum, "CMOS Image Sensors: Electronic Camera-on-Chip", IEEE Trans. on Electron Devices, Vol.44,No.10, pp.1689-98,1997

[2] Abhishek Bandyopadhyay, Jungwon Lee, Ryan Robucci, Paul Hasler,"A 80uW/Frame 104X128 CMOS Imager Front end for JPEG Compression", ISCAS May. 2005, Kobe, Japan

[3] A. Kitchen, A. Bermak and A. Bouzerdoum, "A PWM digital pixel sensor based on asynchronous self-resetting scheme," IEEE Electron Device Letters, Volume 25, No 7, pp. 471-473 July 2004.

[4] Culurciello, E.; Etienne-Cummings, R.; Boahen, K.A., "A biomorphic digital image sensor ", IEEE JSSC , Vol. 38, No. 2 , 2003, pp. 281 -294.