

# Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization

Chen, Shoushun; Amine, Bermak

2007

Chen, S. S., & Amine, B. (2007). Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 15(3), 346-357.

<https://hdl.handle.net/10356/91132>

<https://doi.org/10.1109/TVLSI.2007.893624>

---

© 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder. <http://www.ieee.org/portal/site> This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

# Arbitrated Time-to-First Spike CMOS Image Sensor With On-Chip Histogram Equalization

Chen Shoushun, *Student Member, IEEE*, and Amine Bermak, *Senior Member, IEEE*

**Abstract**—This paper presents a time-to-first spike (TFS) and address event representation (AER)-based CMOS vision sensor performing image capture and on-chip histogram equalization (HE). The pixel values are read-out using an asynchronous hand-shaking type of read-out, while the HE processing is carried out using simple and yet robust digital timer occupying a very small silicon area ( $0.1 \times 0.6 \text{ mm}^2$ ). Low-power operation (10 nA per pixel) is achieved since the pixels are only allowed to switch once per frame. Once the pixel is acknowledged, it is granted access to the bus and then forced into a stand-by mode until the next frame cycle starts again. Timing errors inherent in AER-type of imagers are reduced using a number of novel techniques such as fair and fast arbitration using toggled priority (TP), higher-radix, and pipelined arbitration. A verilog simulator was developed in order to simulate the effect of timing errors encountered in AER-based imagers. A prototype chip was implemented in AMIS  $0.35 \text{ }\mu\text{m}$  process with a silicon area of  $3.1 \times 3.2 \text{ mm}^2$ . Successful operation of the prototype is illustrated through experimental measurements.

**Index Terms**—Address event representation (AER), CMOS image sensors, on-chip histogram equalization, time-to-first spike (TFS) vision sensor.

## I. INTRODUCTION

THE LAST decade has witnessed significant technological advancement of CMOS image sensors. CMOS imagers are undoubtedly gaining more territory when compared to their charge-coupled device (CCD) counterpart. This is mainly 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. Deep submicron technologies have contributed significantly to paving the way to more novel on-chip processing. The concept of “Camera-on-a-chip” has already been introduced in the 1990s [1] and new developments have seen more complex image processing such as image compression, motion and edge detection [2], [3].

A particularly interesting processing, which is required as a preprocessing stage in many applications, is image histogramming. A number of applications related to object and face recognition require histogram equalization (HE) as a preprocessing stage. Traditionally, HE is performed off-chip, by first capturing the image using CCD or CMOS camera and then buffering the entire frame before processing each frame

sequentially. In [4], the authors proposed an interesting analog cellular adaptive image sensor based on current mode active pixel. The obtained cumulative histogram is computed in analog domain using current sources. This is achieved in reversed order and is also nonlinear in time due to the reverse relationship between integration time and the photocurrent. In addition, the design suffers from mismatch in current sources and limited flexibility since the processing is performed in analog domain. In conventional digital signal processing (DSP)-based vision systems, images are read-out using a clock, which switches the multiplexer from one sensor to another, reading a brightness value from each and every sensor at a fixed interval, hence, called “scanner.” Images are, therefore, produced by sequentially scanning the array using column and row scanners. Once the pixel values are scanned they are sorted in order to perform HE. Scanning read-out strategies will soon fall short of meeting higher resolution and frame rate requirements, and hence new approaches are, therefore, required to overcome these limitations. Address event representation (AER) [5] combined with the spiking pixel architecture was proposed in order to provide efficient allocation of the transmission channel to only active pixels [7].

Recent biological studies [8] reviewed a number of arguments for taking into account the temporal information that can be derived from the very first spikes in the retinal spike trains. The study suggests that retinal encoding can be performed in the time-to-first spike (TFS) rather than the frequency of the spikes. In building CMOS vision sensors, the two approaches can be equally used to convert luminance into a pulse train signal. In the TFS case the information is encoded in the spike latency [9], while in the spiking pixel case the information is encoded in the firing frequency of the resulting oscillator. While both concepts provide a viable mean to build a vision sensor, both the operation of the pixel and the read-out strategy are fundamentally different. In the spiking pixel based AER, brighter pixels are favored because their integration threshold is reached faster than darker pixels. Consequently, brighter pixels request the output bus more often than darker ones. This results in an unfair allocation of the bandwidth as well as congested read-out bus because of the periodical request due to the spiking nature of the pixel. This imposes higher constraints on the AER processing speed and induces more dynamic power consumption and temporal jitter affecting the signal-to-noise ratio (SNR). Another very interesting property of TFS-based arbitrated vision sensor is the inherent ordering property of the pixels’ brightness at the output bus, allowing to greatly facilitate the VLSI implementation of HE processing.

This paper first presents TFS-based arbitrated vision sensor followed by on-chip HE processing. A number of novel de-

Manuscript received April 18, 2006; revised September 6, 2006. This work was supported by the Research Grant Council of Hong Kong SAR, China, under Project HKUST610405.

The authors are with the Smart Sensory Integrated Systems (S2IS) Laboratory, Electronic and Computer Engineering Department, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: eechenss@ust.hk; bermak@ust.hk).

Digital Object Identifier 10.1109/TVLSI.2007.893624

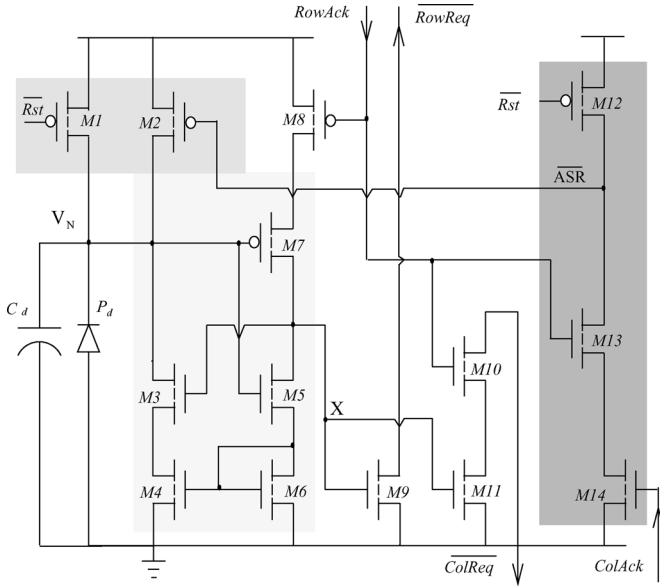


Fig. 1. TFS pixel schematic. The pixel consists of mainly four building blocks: photodetector, reset circuit, event generator [7], and the handshaking circuit.

sign concepts such as fair, high radix and pipelined arbitration are introduced. The arbitrated TFS-based sensor is compared to TFS-based digital pixel sensor (DPS) [10] and its potential scaling in deep submicron technologies is also studied. Section II introduces the TFS-based pixel concept together with its simulation results. Section III introduces the AER architecture and HE processing. This section also introduces various design strategies used for reducing the timing errors such as fair and high radix arbitration as well as pipelining. Section IV reports the simulation results used to validate the previous concepts. Section V describes the VLSI implementation and the experimental results while Section VI concludes this paper.

## II. TFS-BASED PIXEL

Recent biological studies [8] demonstrated that TFS is an important and useful information in retinal encoding. TFS encoding scheme can offer very interesting features when implemented in hardware. One interesting feature is the fact that the illumination can be encoded in a single transition resulting in lower dynamic power consumption and more effective imager bandwidth [11]. In addition, TFS-based encoding results in a natural ordering of the pixel illumination values, which facilitates the implementation of various image processing such as HE. Fig. 1 shows the schematic circuit diagram of our proposed TFS-based sensor. The circuit includes four main building blocks namely photodetector (PD) 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). Finally, transistors (M8-M14) are used in order to implement the 2-D handshaking protocol with the column and row arbitration circuits. The TFS information is, therefore, multiplexed and arbitrated using a column and row arbitration circuits, which constitute the AER read-out [5]. In [7], a thorough comparison is carried-out to compare this event generator with various structures including

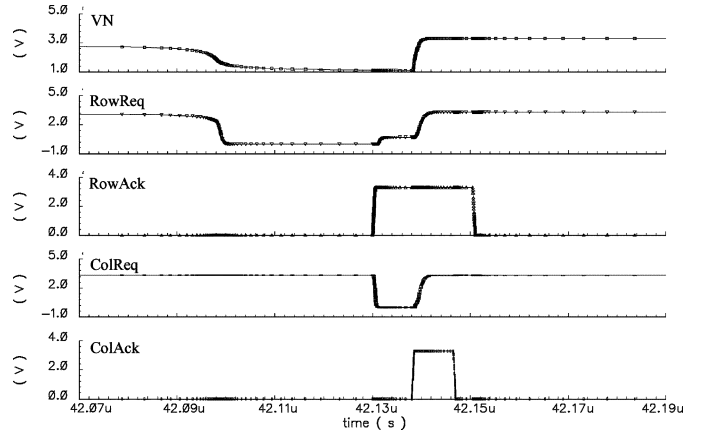


Fig. 2. Simulation results of the pixel operation. Signals from top to bottom are photodiode voltage  $V_N$ , row request, row acknowledgment, column request, and column acknowledgment, respectively.

the simple inverter, the capacitive-feedback inverter and the starved inverter. It was demonstrated that the current feedback inverter presents superior performance in terms of energy consumption by several orders of magnitude [7]. The current-feedback presents an energy consumption and a switching speed that is independent of the input slew rate because of the positive feedback, hence, offering a very good tradeoff between speed and energy consumption [7].

Image capture process is initiated by pulsing an active low  $\overline{Rst}$  pulse, which is used to reset the pixels and start the integration process. The light falling onto the photodiode  $P_d$  will start discharging the internal capacitor of the photodiode  $C_d$ . This results in a linearly decreasing voltage  $V_N$  across the node of the photodiode. Once this voltage reaches the threshold voltage of the inverter (M5,M7), a spike corresponding to the time to reach the threshold, will be generated at node X. Assuming the photocurrent is constant during a frame read-out period, the TFS is given by

$$TFS = \frac{(V_{dd} - V_{TH}) \times C_d}{I_d} \quad (1)$$

where  $I_d$  and  $V_{TH}$  are the photocurrent and the threshold voltage of the inverter (M5,M7), respectively. The time required for the photodiode voltage to reach the threshold voltage of the inverter and, hence, to generate the event can be interpreted as the TFS. The spike generated at node X, is used to initiate the handshaking procedure by turning ON transistor M9 responsible for pulling down a row request signal  $\overline{RowReq}$ , which is sent to the row AER. As a consequence, the row AER is activated and all the row requests are processed and only a single acknowledgment signal (RowAck) is granted to one and only one row. At this stage, all pixels that generated an event within the acknowledged row, will send a new request  $\overline{ColReq}$  to the column AER and will asynchronously self-reset the photodiode node by turning on transistor M2 once an acknowledgment signal is received. The process is initiated again at the end of each frame capture, by the  $\overline{Rst}$  signal, which will start the next frame cycle. It is important to note that within a frame capture cycle, an acknowledged pixel is forced to a stand-by mode until the next frame cycle starts again. This feature not only reduces the

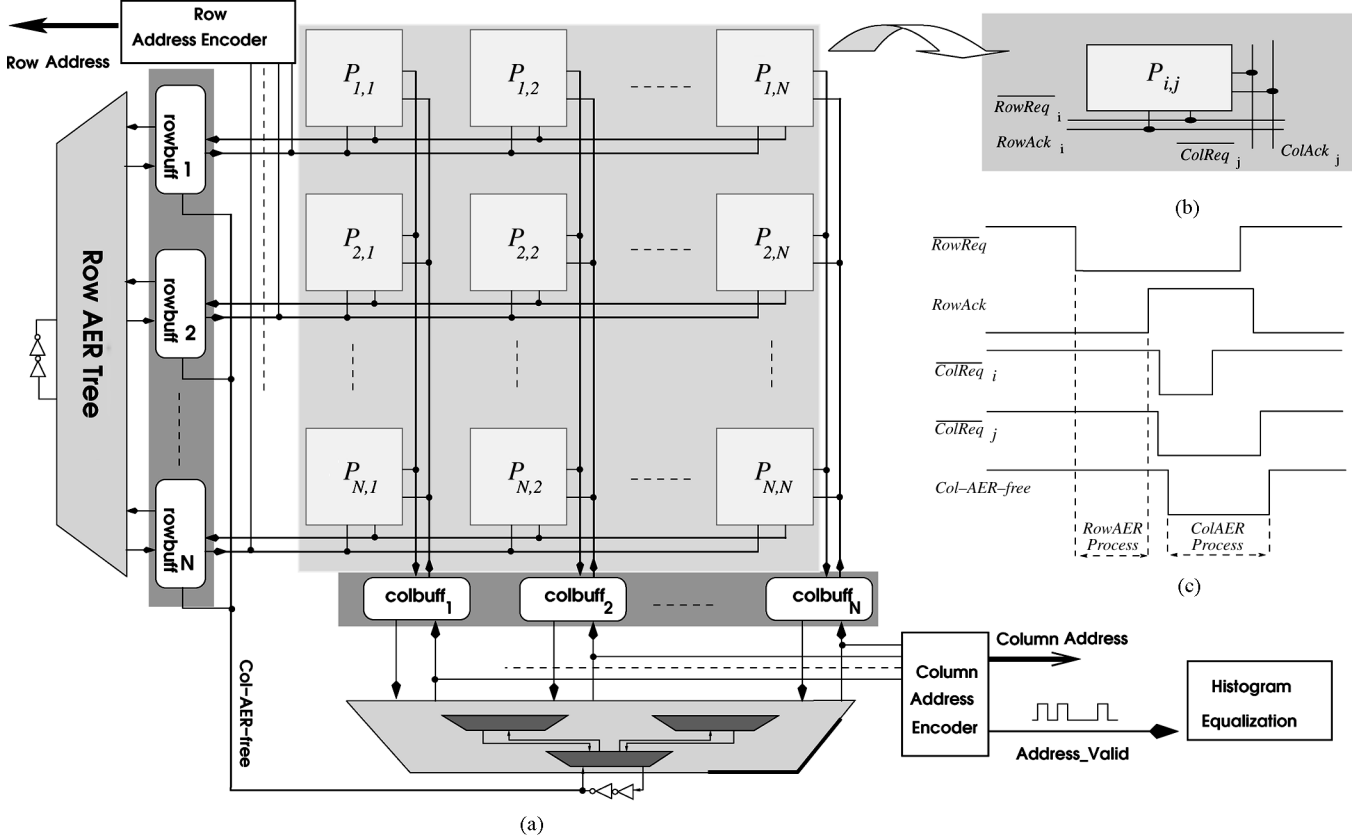


Fig. 3. (a) Vision sensor architecture. The sensor includes an array of TFS pixels, column and row buffers and arbiter, as well as column and row address encoders. Once the address is encoded, the address valid signal is used as a clock input to the HE counter circuit. (b) Input/output signals to each pixel within the array. (c) Sequence of handshaking signals.

consumed power and switching activity, but also reduces the amount of requests processed by both the column and row AER.

Fig. 2 reports the simulation results of the TFS-based image sensor illustrating the photodiode voltage, the event generation process and the handshaking signals. The figure shows the sequence required in a full pixel operation cycle, which can be described as: *Start Integration*  $\mapsto$  *Event Generation*  $\mapsto$  *Row Request*  $\mapsto$  *Row Acknowledgment*  $\mapsto$  *Column Request*  $\mapsto$  *Column Acknowledgment*  $\mapsto$  *Self Reset*. It is very important to note that each pixel in the proposed scheme is responsible for self-resetting itself, after which it enters a stand-by mode until a new integration cycle is initiated. The row and column acknowledgement signals are encoded as an address data for the event. An asynchronous event-driven imager is, therefore, realized based on “single transition per pixel and self-reset procedure.” It should be also noted that in this proposed scheme, the charge-up current required to reset the sensing node is kept minimum as the complete discharge of the sensing node is prevented. The charge-discharge swing is kept constant at about  $(V_{dd} - V_{TH})$  for all pixels within the array.

### III. AER IMAGER AND HE

#### A. Imager Architecture

The architecture of the arbitrated TFS CMOS image sensor is shown in Fig. 3. The imager includes an array of  $128 \times 128$

pixels converting illumination into TFS information. TFS information acquired from the 2-D array needs to be read-out and eventually digitized. One way to achieve this is to use a pixel-based memory, which can be quite effective however will result in increased pixel size and reduced fill-factor. Another solution consists of placing the pixel-generated spikes into a bus. This requires both row and column arbitration circuitries to ensure multiplexing the 2-D array information into a single output bus. This is referred to as “Address Event Representation” read-out strategy [5]. In contrast to conventional image sensors, images are not acquired using a scanner reading a brightness value from each sensor at a fixed interval, but instead acquisition is event driven. Only active pixels will be granted access to the output bus. In this kind of imager, the readout process is initiated by the pixel itself by sending out a request signal. Pixels are organized into rows and columns sharing the same request and acknowledgment buses. When one or more pixels within a row fire, a request row signal  $\overline{\text{RowReq}}$  is sent to the row AER for arbitration. The row AER may receive several requests at the same time. After arbitration, only one row will be acknowledged by RowAck. The fired pixels within the acknowledged row will send request  $\overline{\text{ColReq}}$  to the column AER. Instead of waiting for the column AER to acknowledge the requests one by one, column buffers are inserted as a pipeline stage between the pixel array and the column AER enabling the pipelining of the overall array operation. The AER-based vision sensor includes row and



column address encoders used to encode the address of the acknowledged pixels. An output address valid signal is used as a clock signal for the HE circuit as will be explained in the next section.

While AER-based read-out has its own merits as it introduces the idea of low-power asynchronous pixel-driven read-out, however the approach does suffer from the inherent disadvantage of the event driven read-out nature of the pixel, which results in collision problems occurring when multiple requests occur at the same time. Assume that at a given time,  $\rho$  pixels fire and request access to the bus. An arbiter will grant access to the bus to a given pixel and will place the remaining  $\rho - 1$  pixels in a processing queue. A timing error is, therefore, induced, which is proportional to the processing time of each request in the arbitration tree, as well as the number of requests received at any given time. This will introduce delay in processing some requests, which results in jitter and timing errors. Another issue when dealing with AER-based read-out is to provide a fair allocation of the shared bus to all pixels. Fixed priority often results in an unfair allocation of the output bus to only “privileged” rows and columns. To overcome these problems we propose a number of novel design concepts such as high radix and pipelined arbitration scheme. Fair arbitration is also proposed using toggled-priority (TP) and free metastate SR-based arbiter cell.

### B. Fair Arbitration

In an AER-based read-out, the arbiter is traditionally realized using a tree. Each building block within the tree processes two incoming requests and propagates the decision to the layer below. Each building block is typically realized using an SR latch in which the S and R inputs are connected to the two input requests. The sizing of the two NOR gates can be biased such that higher priority is allocated to one specific request input. We propose to avoid biasing the arbitration by using a novel SR-latch circuit featuring TP processing and free metastability. Fig. 4 shows the 2-input single building block in the AER tree, which includes our proposed TP feature.

Each cell within the tree arbiter is constituted of 3 basic units, namely: arbitration unit, propagation unit and an acknowledgment unit. The arbitration unit is constituted of an SR latch composed of two cross-coupled NOR2 gates and five additional transistors used to provide fair arbitration. Initially, M16 is turned ON by the global reset, providing the top NOR2 gate a larger pulling down capability compared to the bottom NOR2 gate. If the two requests  $\overline{req0}$  and  $\overline{req1}$  are initially received at the same time, competition will occur and the top NOR2 gate will gain priority over the bottom gate, i.e.,  $x_0 = 1$  and  $x_1 = 0$ . The result is maintained until the arbiter receives an acknowledgment from higher stages and then  $\overline{ack0}$  will be activated. At this stage, transistor M16 is turned off and the bottom NOR2 gate gains priority over its counterpart. The priority is, therefore, toggled as the pulling down capability of the top NOR2 gate depends on the *switch* signal, which is toggled after an arbitration process took place. Fig. 5(a) shows the fair arbitration unit while Fig. 5(b) shows the equivalent circuit for the SR latch when the *switch* signal toggles. It should also be noted that the two NOR2 gates always have different pulling down capabilities and this allows to avoid the metastate of the SR latch. The simulation results

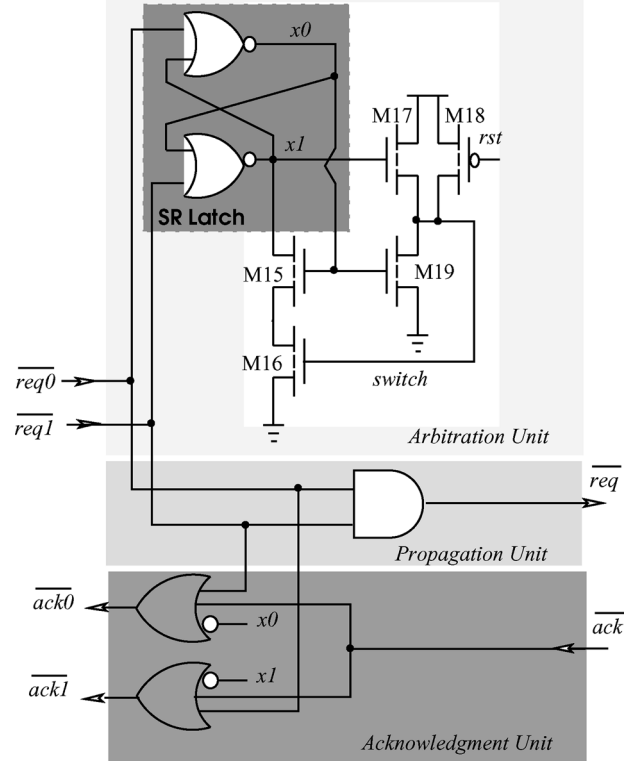


Fig. 4. 2-input fair arbiter building block. Each cell consists of three building blocks, namely: (i) arbitration unit; (ii) propagation unit; and (iii) acknowledgment unit.

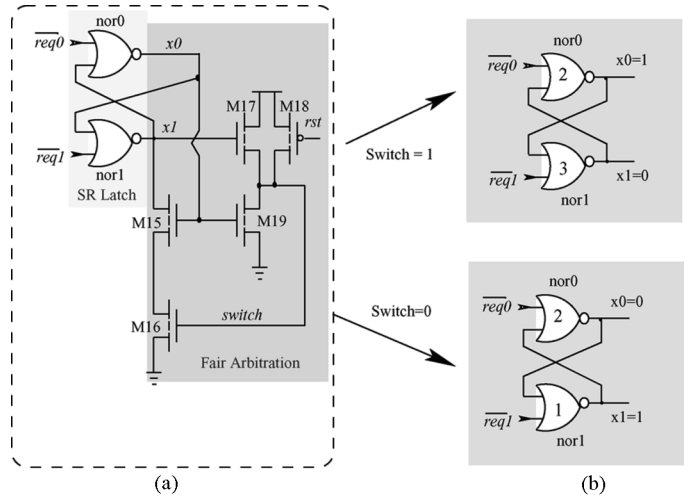


Fig. 5. Operating principle of the fair arbitration. Priority is toggled after arbitration has taken place as the pulling down capability of the bottom NOR2 gate depends on the state of the *switch* signal.

of this fair arbitration process is shown in Fig. 6. One can note from this figure that initially  $\overline{req0}$  and  $\overline{req1}$  arrive at the same time and  $\overline{req0}$  is acknowledged first ( $\overline{ack0} = 0$ ) followed by  $\overline{req1}$  ( $\overline{ack1} = 0$ ). A second  $\overline{req0}$  is received and processed. The priority is, hence, toggled to  $\overline{req1}$ , which explains why  $\overline{req1}$  is processed first in the third cycle.

Depending on the illumination intensity, one row may request access to the tree multiple times. Fixed priority [5], [12]–[16] often results in an unfair allocation of the output bus to only

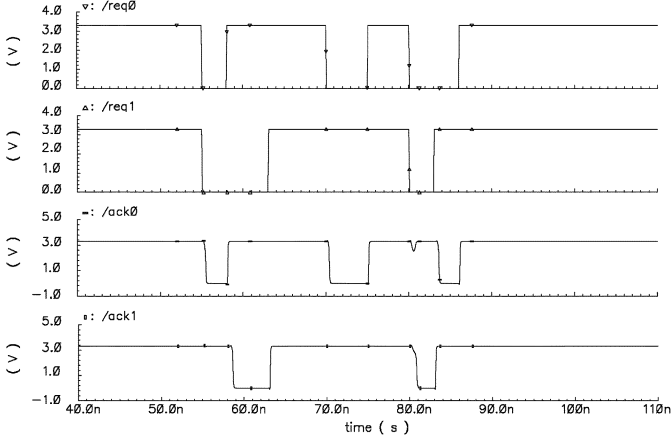


Fig. 6. Simulation of a 2-input fair arbitration scheme. The priority is toggled after an arbitration process has taken place.

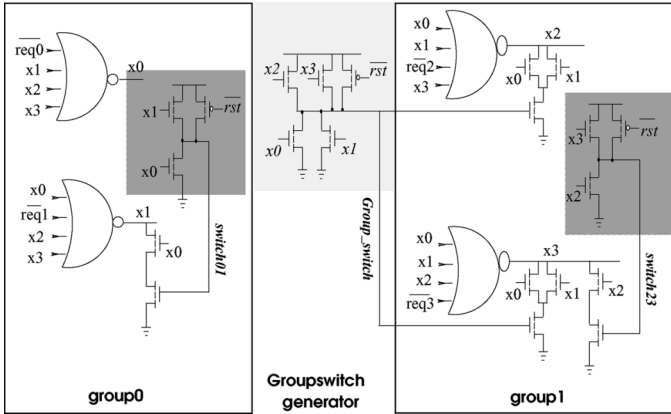


Fig. 7. Schematic of the 4-input fair arbitration unit. Four cross-coupled NOR4 gates are organized into two groups: group0 and group1.

“privileged” rows thus resulting in an unbalanced timing error, i.e., for rows with higher priority, the timing error is small and for rows with lower priority, the timing error is large.

### C. Higher Radix Arbiter Tree

Timing errors are introduced due to the delay in the arbitration tree. One way to reduce this delay is to build a higher radix arbitration tree, which permits to reduce the depth of the tree. The delay in the arbitration tree can be expressed as  $\Theta = \theta \times \log_r m$ , where  $\theta$ ,  $m$  and  $r$  are the delay of the basic building block, the number of columns and the radix (or the number of inputs per arbiter cell), respectively. By increasing the radix  $r$ , the depth of the tree  $\log_r m$  is hence reduced. This will improve the global delay  $\Theta$  if the delay of the new higher radix arbiter cell  $\theta$  is maintained to an acceptable level. Using higher radix building blocks will allow processing more than 2 requests per cell at the same time. With such arbiters, the depth of the AER tree is reduced and, therefore, the overall delay can be reduced as long as the delay of a single higher radix cell is maintained to a reasonable level. Based on the architecture of 2-input fair arbiter, we expanded the concept to build a 4-input building block, as shown in Fig. 7.

Four cross-coupled NOR4 gates are organized into two groups, group0 ( $\text{req0}$  and  $\text{req1}$ ) and group1 ( $\text{req2}$  and  $\text{req3}$ ). Within

TABLE I  
DELAYS OF A SINGLE BUILDING BLOCK AND AN ARBITER TREE FOR DIFFERENT RADIX ( $r$ ), DIFFERENT ARRAY SIZE ( $m$ ) AND FOR FP AND TP

Operation type	$\theta$		$\Theta$			
	$r = 2$	$r = 4$	$r = 2$ $m = 16$	$r = 2$ $m = 64$	$r = 4$ $m = 16$	$r = 4$ $m = 64$
FP single Request	0.33n	0.49n	1.32n	1.98n	0.98n	1.47n
FP multi. Requests	0.39n	0.55n	1.56n	2.34n	1.1n	1.65n
TP single Request	0.37n	0.55n	1.48n	2.22n	1.1n	1.65n
TP multi. Requests	0.41n	0.59n	1.64n	2.46n	1.18n	1.77n

each group, the principle of toggling the priority is similar to the 2-input building block discussed earlier. A group priority signal *Groupswitch* is used to switch the priority between *group0* and *group1*. For example, if the current priority order is  $x0 \mapsto x1 \mapsto x3 \mapsto x2$ , then after  $\text{req0}$  is received and processed, the priority order will be toggled at the next cycle to  $x3 \mapsto x2 \mapsto x1 \mapsto x0$ . An AER building blocks with  $r = 4$  was designed and its delay was evaluated and compared with the case where  $r = 2$  for both TP and fixed priority (FP). In addition the global performance of the tree based on the two building blocks and for different array sizes are reported in Table I. One can note that for larger array size, the higher radix arbiter tree and TP scheme reduces the global delay by more than 25%.

### D. Pipelining the Row and Column AER Processing

Fig. 8 shows the schematic of the column buffer. It is important to note that the column buffer is responsible for generating the acknowledgment back to the pixel after a certain delay. When the  $\text{ColReq}$  is received by the column buffer from the array, transistor M20 is turned ON and, therefore, an active high  $\text{ColAck}$  signal is sent back to the array and at the same time the request is propagated to the column AER through  $\text{ColAERReq}$  signal. The same signal is delayed through the inverter chain IC2 allowing to kill the request signal of the array by pulling high the  $\text{ColReq}$  through transistor M23. Once the request is processed by the column AER, an acknowledgment signal  $\text{ColAERAck}$  is received by the buffer allowing to turn ON transistor M21, which will in turn disable the request signal  $\text{ColAERReq}$ . It should also be noted that the  $\text{rst}$  signal is used to reset the column buses to the correct initial state by disabling all acknowledgment and request signals. One very interesting fact about this novel column buffer circuitry is its important role of isolating the array from the column AER and, hence, avoiding the charge and discharge of large capacitances of the column buses by the column AER. This will improve further the arbitration speed particularly for large pixel array.

In parallel with the column arbitration, the row arbitration process is carried out at the same time. This is realized by pulling-up the  $\text{RowReq}$  signal using a row buffer circuit shown in Fig. 9. This will permit to the row AER to start processing the next row arbitration while the column AER is still processing the current row. A key issue here is to make sure that the

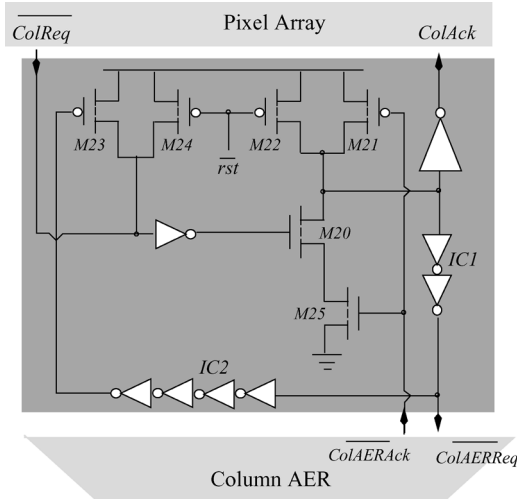


Fig. 8. Schematic of the column buffer, which acts as an interface circuit between the array and the column AER.

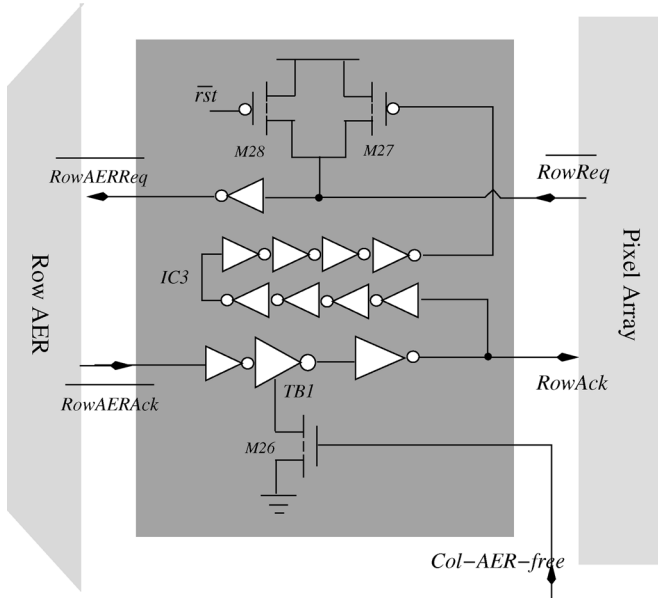


Fig. 9. Schematic of the row buffer. Pipelined processing is achieved by initiating the row arbitration in parallel with the column arbitration through the monitoring of ColAERFree signal.

address of the newly selected row is not propagated to the array until the column AER has finalized its current processing. This can be achieved using a ColAERFree signal, which indicates the status of the column AER. In fact this signal corresponds to the propagated request signal at the root of the tree, which is the ANDed signal of all active low requests of the column buffer. This signal is used to control a tristate buffer TB1 through transistor M26 as shown in Fig. 9. This permits to prevent the newly selected RowAERAck from propagating to the array. The delay of the inverter chain IC3 in the row buffer is carefully designed in order to ensure that the acknowledged row has sufficient time to successfully send the column requests before the RowAck is disabled by turning ON transistor M27 after a delay set by IC3. Turning ON M27 will also disable the request signal to the row AER ( $\overline{\text{RowAERReq}}$ ). At this stage,

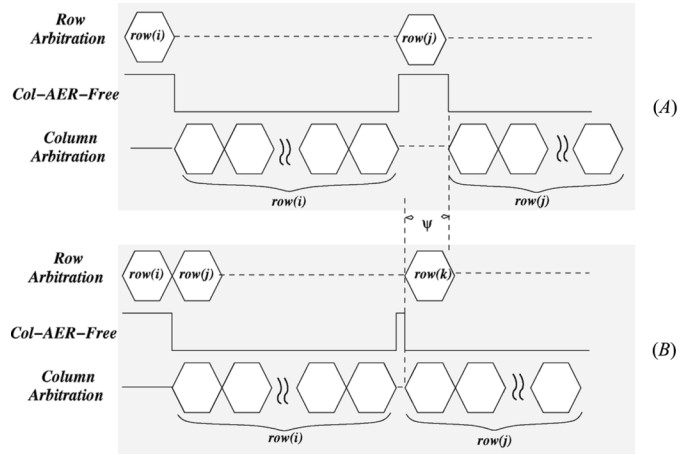


Fig. 10. Pipelining principle between the row and column AER. (a) Represents the nonpipelined processing, while (b) corresponds to the pipelined one. In the pipelined version, the delay corresponding to the row arbitration is avoided as row arbitration is performed in parallel with column arbitration. This results in a time saving corresponding to the row arbitration time denoted as  $\psi$ .

a new round of arbitration process can start in parallel with the column AER. Thus a pipeline processing of the row and column AER is obtained. In most cases, when the column AER finishes processing the current row, a new decision in the row AER can be ready and minimum slack can be achieved.

Fig. 10 compares the signal sequencing in the row and column AER with and without pipelining strategy. In the pipelined case, the row AER can start to process new request before the column AER has completed its current task, while in the nonpipelined AER the row arbitration is held in a wait mode until the column AER finishes its current processing. The overall saving in one single arbitration cycle using AER processing with pipelining strategy is equal to the time required to perform the row arbitration denoted as  $\psi$  (refer to Fig. 10). This represents a significant saving as processing a row arbitration requires propagating forward and backward the entire row arbitration tree.

#### E. Histogram Equalization

In a TFS-based sensor, 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. HE can, therefore, be performed simply by associating the same quantization level to a number of pixels firing within a given time slot.

The  $128 \times 128$  pixels are equally segmented into 256 quantization bins resulting in an equalized image capture with uniform intensity histogram (64 pixels in each bin). Fig. 11 shows the block diagram of HE. Address Valid 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 flip-flop every 32 cycles. The output of the T flip-flop is then used to drive an 8-bit down-counter, which will decrement by 1 once 64 pixels have been counted. The second counter is a down-counter as illumination is inversely proportional to the TFS pulse signal. The 8-bit counter value combined with the pixels address constitute the output of HE circuit. Compared to the HE proposed in [4], our approach shows

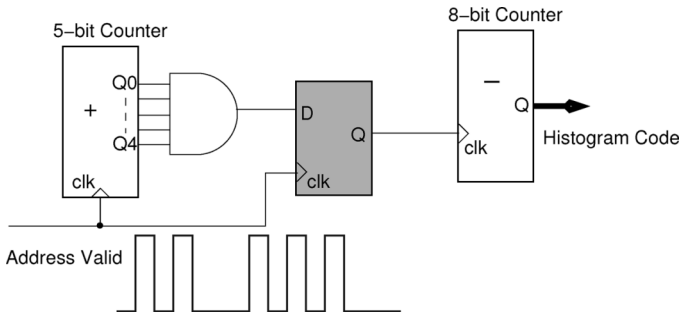


Fig. 11. Building block diagram of HE processing which is realized using only two counters.

several advantages. First, in our pixel, the pixel's illumination information is encoded into a digital spike instead of analog current or voltage signal. Early analog-to-digital (A/D) conversion is obtained and no post analog signal processing is needed. Since our scheme uses a digital encoding and read-out, it also offers flexibility and easy post processing. For example instead of evenly distributing pixel values into uniform quantization levels one could adapt the quantization levels to perform adaptive quantization. Secondly, the histogram values are obtained on the fly and can be transmitted out of the array thus no temporary storage is needed. In addition, in contrast to previous implementations, our imager can operate in two modes: 1) image capture mode or 2) HE mode.

#### IV. SIMULATION RESULTS

In order to simulate the different techniques proposed in this paper as well as the AER imaging concept in general, we developed a Verilog based simulator. The Verilog program simulates all stages of the AER processing including photodetection, TFS pulse generation, handshaking communication protocol, as well as the row and arbitration processing. The input of the simulator is a 2-D image, which is first translated into an original TFS matrix. The original image undergo all processing stages including handshaking and arbitration. These processing stages will introduce distortion in the form of jitter and mismatch to the TFS matrix due to the timing errors explained earlier. The evaluation of the proposed techniques was carried out by first simulating the effect of such distortion on different sample images with and without introducing the various circuit techniques proposed. In a second stage, we expressed this distortion in terms of peak signal-to-noise ratio (PSNR) for a wide range of  $256 \times 256$  sample images. Fig. 12 shows the simulation results for a sample image using the proposed techniques discussed in this paper. Fig. 12(a) is the original image while Fig. 12(d)–(k) is the AER reconstructed images using the various approaches introduced earlier. Fig. 12(d)–(g) is the nonpipelined reconstructed images using 2-input FP arbiter, 2-input TP, 4-input FP, and 4-input TP, respectively. Fig. 12(h)–(k) represents the same simulations but for the pipelined AER processing. It is clearly shown from this simulation that the pipelined and higher radix fair arbitration scheme permits to reduce the mismatch in the captured AER image. One can also note a row based mismatch mainly explained by the fact that the read-out process is row based. Once a row is acknowledged, all pixels that fired within the row are read-out. This induces a larger row based mismatch as compared

to the column based mismatch. It is also clear from the simulation results that the pipelining scheme permits to significantly reduce the row based mismatch. It is very important to note that the timing error is illumination dependant. For higher illumination range, TFS values are relatively small and any timing error will have a greater effect on the AER output as compared to low illumination environment. In order to express the gain in using our proposed circuit techniques for acquiring AER images, we evaluated the PSNR figures for different dynamic ranges of the input original image using our Verilog simulator. The input image is first spread over a given range, which will result in a set of TFS dynamic range expressed in dB. The acquired AER images for different input dynamic are compared to the original image and the mismatch between the two images is expressed in terms of PSNR as shown in Fig. 12(b). It is clear from this figure that for low dynamic range (50–75 dB), the PSNR values are quite large for all AER images, which suggests that for low illumination range, timing errors are very negligible even without using the various techniques proposed in this paper. On the other end of the illumination range and for wider dynamic ( $> 100$  dB), the PSNR values are drastically reduced and even using all of the proposed circuit techniques will not help that much. This is mainly due to the fact that at higher illuminations, TFS timing resolution becomes much smaller due to the inverse illumination-TFS relationship [see (1)]. This makes the AER image acquisition very vulnerable to timing errors introduced in the arbitration circuitry. At high level of illumination, the AER bus request queue becomes prohibitively large resulting in poor PSNR values. In the midrange dynamic (75–95 dB), the proposed techniques are very effective in improving the quality of the acquired images. An improvement of up to 15 dB is found in this range of illumination. The same simulation was repeated but this time for HE processing. Fig. 12(c) illustrates the results for HE processing. PSNR figures are reported with respect to the original HE image. It is very interesting to note that HE processing permits to improve the performance by an average of 18 dB across the midrange dynamic (75–95 dB). In addition, PSNR figures for HE are slightly higher when compared to AER output. This is explained by the fact that HE is not sensitive to the absolute timing mismatch. Indeed, a shift of all illumination values due to timing errors will not introduce any error in the obtained HE image. HE is only sensitive to the relative timing errors which may cause swapping of pixel read-out order located at the boundary of the HE quantization bins. Table II reports the PSNR figures for all proposed techniques and for different sample images. The previous results are clearly confirmed for a large set of images with an average PSNR improvement of 9 and 12 dB for normal AER images and HE images, respectively. Combining fair and fast arbitration using TP, higher-radix and pipelined arbitration permit to reduce the timing error in midrange illumination (75–95 dB) and improve PSNR figures for AER images with and without HE.

#### V. VLSI IMPLEMENTATION AND EXPERIMENTAL RESULTS

##### A. VLSI Implementation and Comparison With DPS

The prototype chip including the AER image sensor and HE processing was implemented using 0.35- $\mu\text{m}$  AMIS CMOS dig-

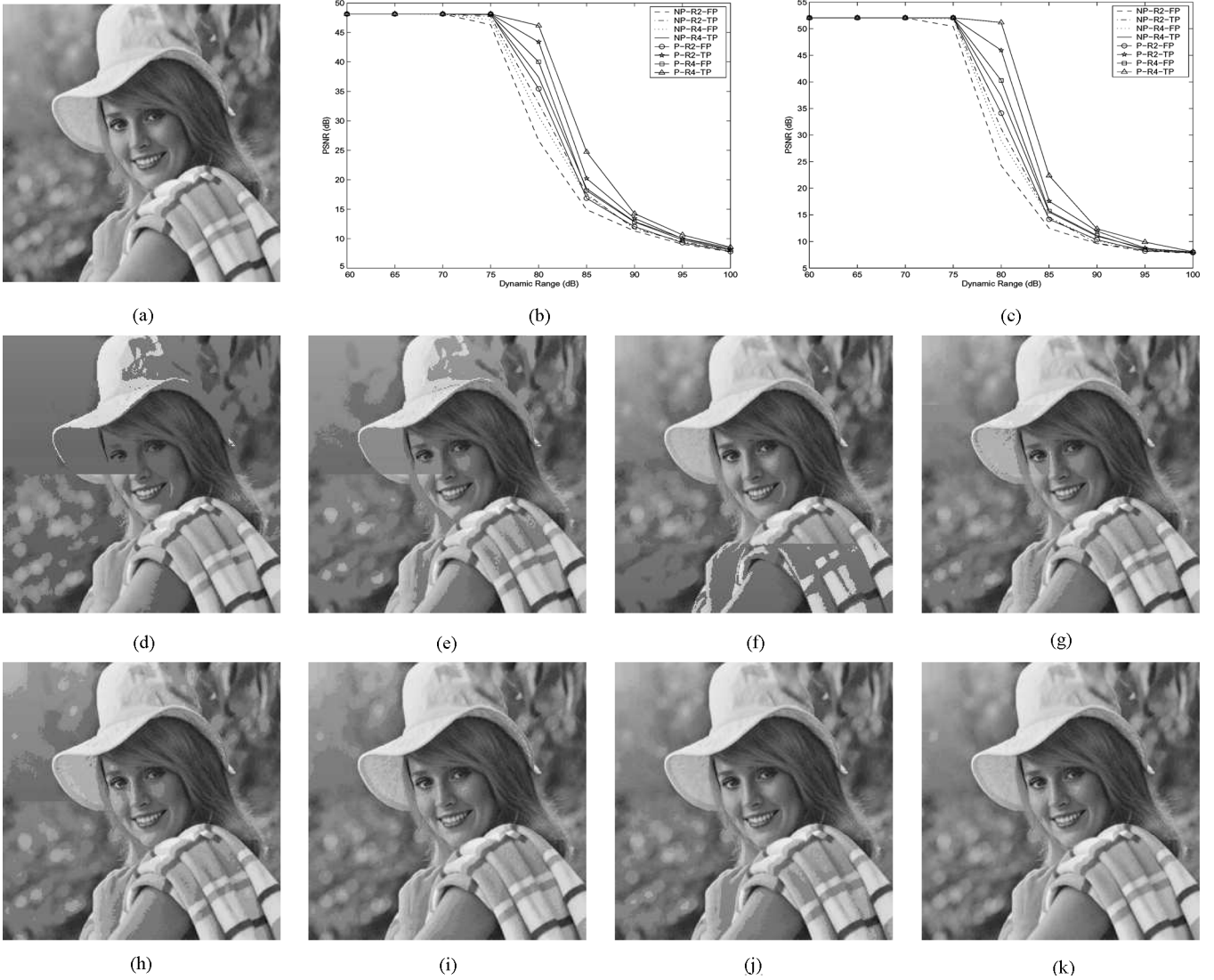


Fig. 12. Simulation results for a  $(256 \times 256)$  Elaine image under different AER operating modes. (a) Original test image. (d)–(k) Images reconstructed using different approaches, namely: Nonpipelined AER and radix-2 FP arbiter (NP2FP), nonpipelined AER and radix-2 TP arbiter (NP2TP), nonpipelined AER and radix-4 FP arbiter (NP4FP), nonpipelined AER and radix-4 TP arbiter (NP4TP), pipelined AER and radix-2 FP arbiter (P2FP), pipelined AER and radix-2 TP arbiter (P2TP), pipelined AER and radix-4 FP arbiter (P4FP), and finally pipelined AER and radix-4 TP arbiter (P4TP), respectively.

ital process (1-poly five metal layers). Fig. 13(a) shows the microphotograph of the fabricated prototype. The chip occupies a total silicon area of  $3.1 \times 3.2 \text{ mm}^2$ , with more than 95% of the active area dedicated to the pixel array. The HE circuit occupies only  $0.1 \times 0.6 \text{ mm}^2$  which corresponds to less than 1% of the active area. Fig. 13(b) shows the layout of the pixel with all building blocks highlighted. The pixel includes 14 transistors (three for reset circuit, five for the event generation, and six for handshaking operation) with a total silicon area of  $17 \times 17 \text{ } \mu\text{m}^2$  and a fill factor of 33%. This performance in terms of pixel area and fill-factor represents a major advancement as compared to TFS-based DPS reported in [10]. Fig. 13(c) illustrates the layout of TFS-based DPS realized in the same technology, where it can be noted that most of the silicon area is occupied by the memory circuitry. Table III reports the performance of arbitrated TFS and compares figure of merits to TFS-based DPS, realized in the same CMOS process [10].

It is clear from Table III that compared with TFS-based DPS, the arbitrated TFS permits to achieve a reduction of seven times in terms of pixel size and a fill-factor improvement by a factor of 2 while reducing the power consumption by more than two decades. This is explained by the fact that DPS requires writing into local memory at each firing stage, which results in significant power consumption at the pixel level. This power is scaled up with the imager resolution.

#### B. Performance Analysis and Experimental Results

The chip was mounted on a custom PCB, which provides the required control signals and captures the output signal. The performance of the imager was evaluated by measuring a number of important figure of merits. The dynamic range was first evaluated by experimentally measuring the TFS when varying the illumination across a wide range of intensities. In our first experiment, no frame limitation was imposed leading to about 100-dB

TABLE II  
PSNR (dB) FIGURES FOR THE AER AND HE OUTPUT IMAGE FOR SOME SAMPLE IMAGES USING DIFFERENT OPERATING MODES, NAMELY NONPIPELINED AER AND RADIX-2 FIXED PRIORITY ARBITER (NP2FP), NONPIPELINED AER AND RADIX-2 TP ARBITER (NP2TP), NONPIPELINED AER AND RADIX-4 FIXED PRIORITY ARBITER (NP4FP), NONPIPELINED AER AND RADIX-4 TP ARBITER (NP4TP), PIPELINED AER AND RADIX-2 FIXED PRIORITY ARBITER (P2FP), PIPELINED AER AND RADIX-2 TP ARBITER (P2TP), PIPELINED AER AND RADIX-4 FIXED PRIORITY ARBITER (P4FP), AND FINALLY, PIPELINED AER AND RADIX-4 TP ARBITER (P4TP), RESPECTIVELY. THE LATTER PERMITS TO ACHIEVE THE HIGHEST PSNR FIGURE

Quantizer	Sample Images												Average	
	Lena		Actress		Airforce		Elaine		Plane		Moon Surface		AER	HE
NP2FP	40.97	38.93	37.21	34.71	36.45	25.35	20.73	17.64	45.18	30.27	43.32	36.83	37.31	30.62
NP2TP	44.31	44.57	39.90	36.29	40.09	29.95	28.86	26.74	46.59	35.03	46.27	43.17	41.00	35.95
NP4FP	43.20	42.10	38.14	35.41	37.67	26.30	26.43	23.99	46.33	33.74	45.84	42.02	39.60	33.92
NP4TP	45.96	47.25	41.69	37.19	42.75	33.46	31.73	29.73	47.23	36.40	47.53	46.91	42.81	38.49
P2FP	46.56	48.54	42.79	37.36	39.71	29.11	31.02	28.74	46.34	32.46	47.44	46.98	42.31	37.19
P2TP	47.64	51.41	44.92	37.98	43.67	34.75	36.25	35.26	47.28	36.40	48.08	47.38	44.64	40.53
P4FP	47.62	51.10	44.71	37.93	41.99	32.11	34.60	32.90	47.17	36.18	48.09	47.38	44.03	39.60
P4TP	48.02	51.79	46.22	38.21	46.01	39.43	40.67	40.91	47.47	36.40	48.13	47.38	46.08	42.35

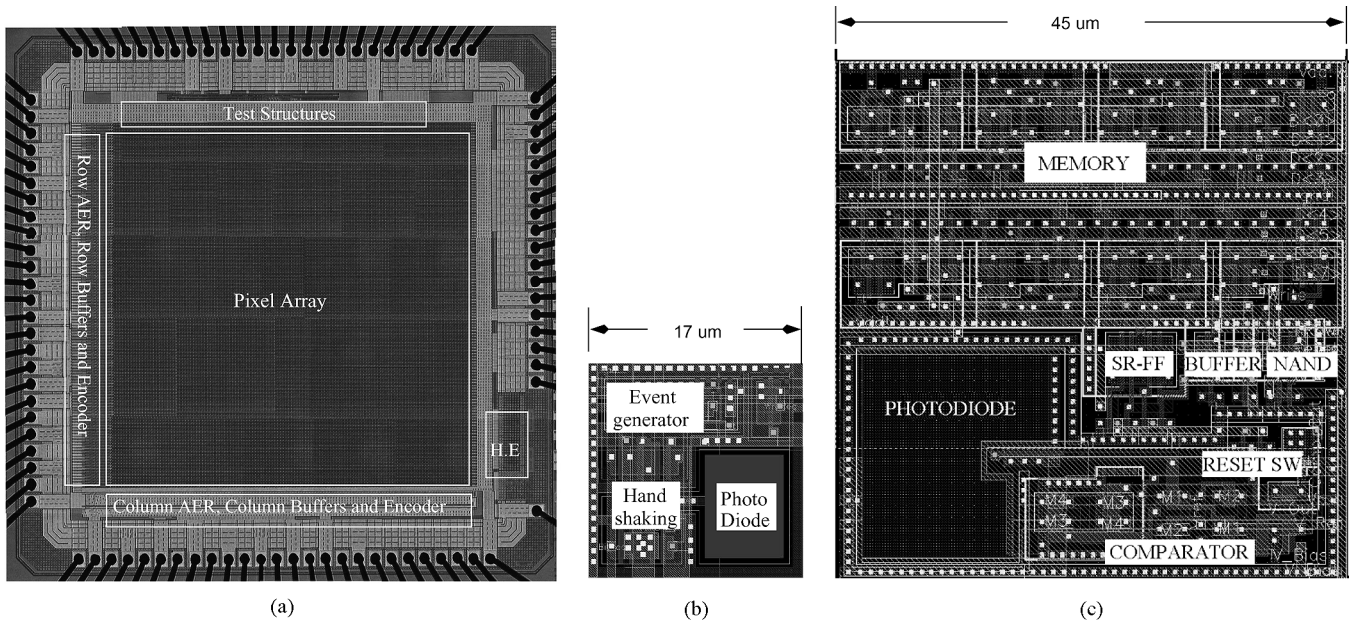


Fig. 13. (a) Microphotograph of the arbitrated TFS-based image sensor. (b) Layout of the arbitrated TFS-based pixel. (c) Layout of the DPS TFS-based pixel implemented in the same technology.

operating range. However, it is important to note that if there a minimum frame rate is imposed, the longest integration time can be low and hence the dynamic range values will be affected. For example, if a frame rate of 30 frames/s is imposed, the resulting lowest detectable illumination level is measured at about 30 lux, which implies that the lower bound of the DR is increased resulting in an effective reduction of the dynamic range down to about 70 dB.

The noise figure in our proposed imager are also analyzed and characterized. The main sources of noise in this type of image sensor can be divided into two main categories [7]. One is a spatial noise caused by the device mismatch, similar to that found in conventional CMOS image sensor. The second is specific to this type of architecture and is categorized as a temporal jitter due to this time domain conversion and the arbitration circuitry. The total FPN was measured at about 4.6% for an illumination level of about 10 lux. This figure is obviously much larger than that of conventional CMOS image sensor, however it is very important to note that this represents the worst case scenario as a uniformly

illuminated scene will imply all pixels firing at approximately the same time. This will result in maximum jitter and increased overall mismatch. In real images, distributed pixel values will greatly minimize the effect of temporal jitter. FPN can also be reduced using correlated double sampling techniques, which unfortunately are not easy to implement in time domain imagers [7]. When comparing our TFS pixel with the spiking pixel reported in [7], two major points should be highlighted. First, in the spiking pixel, the jitter issue is accentuated because each pixel fires multiple times within a single frame capture. Multiple access to the bus by the same pixel will increase the probability of collision and hence will increase the jitter issue. Second, imagers that use the frequency of the spikes to calculate the pixel values can average out the error due to jitter, which reduces noise in general. Analyzing the effect of averaging will require an accurate modeling of the firing process under the proposed arbitration scheme. This problem will be analyzed in our future work.

Single pixel characterization and arbitration functionality test was performed using pixel test structures implemented at the pe-

TABLE III  
SUMMARY OF THE ARBITRATED TFS IMAGER PERFORMANCE AND  
COMPARISON WITH TFS-BASED DPS PERFORMANCE [10]

Features	Arbitrated TFS	TFS-based DPS
Technology	AMIS 0.35 $\mu$ m, 5 M, 1P CMOS	
Supply Voltage	3.3V	
DR (without frame limitation)	> 100dB	
DR (@ 30 frames/s)	70dB	
Average current/pixel/frame	$\simeq 10$ nA	1.6 $\mu$ A
Pixel pitch	17 $\mu$ m	45 $\mu$ m
FPN	4.6% (@ 10 lux)	0.8%
Number of transistors	14	80
Fill factor	33%	17%

riphery of the array [top of Fig. 13(a)]. Fig. 14 shows the experimental measurement of handshaking signals as they occur in an image capture. In this test structure, the pixel exchanges handshaking signals with its arbiter as illustrated in Fig. 14. First a row acknowledgment signal ( $\overline{\text{RowAck}}$ ) is activated. Once the row acknowledgment is sent back to the pixel it activates a column request signal which is then followed by a column acknowledgment generated by the column arbiter. Fig. 15 shows the experimental measurement of a 2-input test structure arbiter cell responding to two external request stimulus. Initially, the two input arbiter cell receives two requests  $\overline{\text{Req0}}$  and  $\overline{\text{Req1}}$  at the same time.  $\overline{\text{Req0}}$  is first processed followed by  $\overline{\text{Req1}}$ . At a later stage only  $\overline{\text{Req0}}$  is received and consequently processed. In a third cycle, both requests collide again but this time  $\overline{\text{Req1}}$  is processed first; clearly illustrating a TP. This result illustrates clearly a successful handling of request collision and fair arbitration through TP concept.

Sample 128  $\times$  128 images were acquired from the prototype under different illuminations and AER operation speeds. In our prototype, the speed of the AER can be controlled by inserting a flip-flop between the column buffer and the column AER. The column AER will be enabled to acknowledge only one request every clock cycle. The speed at which data can be read-out is limited by the speed of the data acquisition board, which can handle a maximum of about 50 MHz. Data were acquired for both AER and HE modes and at different sampling rates. Fig. 16 shows captured AER and histogram equalized images of the same scene under increasing illumination (top to bottom rows of the figure). Columns from left to right correspond to an increasing sampling rate of the data acquisition board from 10 to 50 MHz. Since TFS is an illumination-dependant encoding, for low intensity (row A), a low-frequency acquisition is sufficient to acquire the image while at high illumination levels (row C), a high-acquisition frequency is required. One can also note that HE permits to acquire a relatively illumination-independent image (as illustrated by images located at the most right column of Fig. 16).

## VI. CONCLUSION

In this paper, we have reported the theory, simulation, VLSI design, and experimental measurements of a single-chip CMOS image sensor and HE processor. Low-power image sensing is demonstrated through the use of TFS and AER. Timing

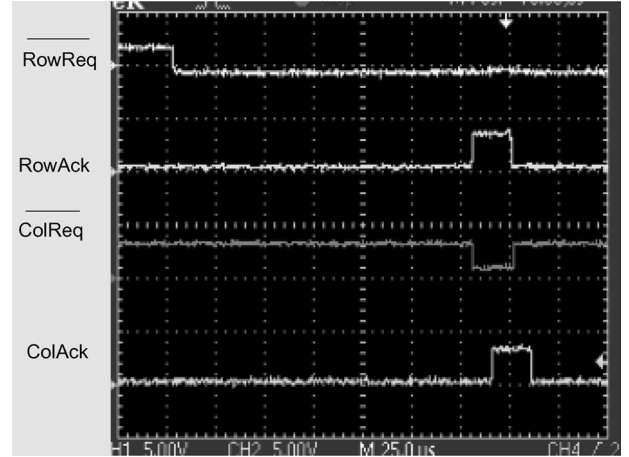


Fig. 14. Experimentally measured pixel handshaking signals. The figure clearly shows the operating sequence: *Event Generation*  $\mapsto$  *Row Request*  $\mapsto$  *Row Acknowledgment*  $\mapsto$  *Column Request*  $\mapsto$  *Column Acknowledgment*.

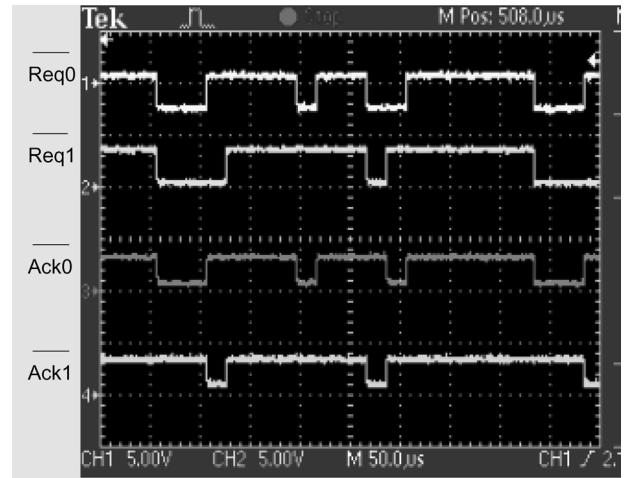


Fig. 15. Experimental results of a 2-input fair arbitration scheme. The acquired signals show that the priority is toggled after an arbitration process has taken place.

errors inherent in the AER-type of imagers were reduced using a number of novel techniques such as fair and fast arbitration using TP, higher-radix and pipelined arbitration. A verilog simulator was developed in order to provide a realistic AER model enabling us to simulate the errors induced in the AER-based imager and HE processing for a wide dynamic range of illumination. It was found that a PSNR gain of more than 12 dB can be achieved using the proposed arbitration technique for mid-range illumination (75–95 dB). Our sensor provides a significant scaling-up of the performance when compared to TFS-based DPS. Indeed the proposed arbitrated TFS permits to achieve a reduction of seven times in terms of pixel size and a fill-factor improvement by a factor of 2 while reducing the power consumption by more than two decades. This is explained by the fact that DPS requires sequential scanning of the array and writing into local memory at each firing stage, which results in significant power consumption at the pixel level. Furthermore, the output nature of the proposed TFS sensor (pixels are sorted) makes it very suitable for HE processing. A prototype chip including 128  $\times$  128 pixels, AER



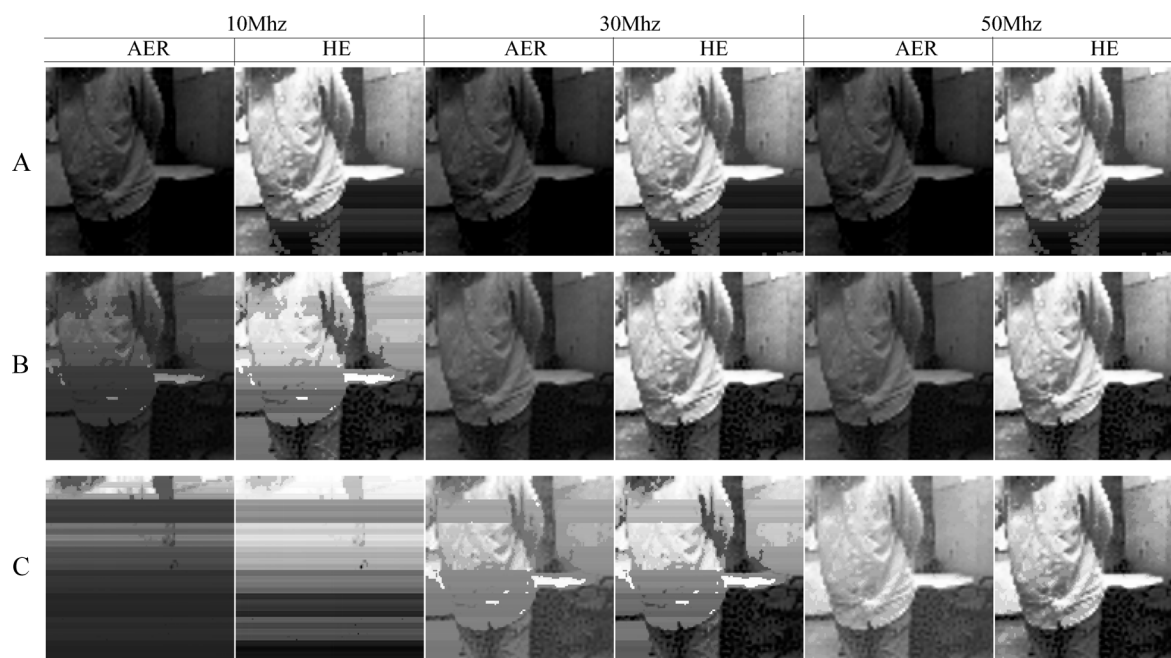


Fig. 16. Captured AER and histogram equalized images of the same scene under increasing illumination (top to bottom rows). Columns from left to right correspond to an increasing sampling rate of the data acquisition board from 10 to 50 MHz. Since TFS is illumination dependant, for low intensity (rows A and B), a low-frequency acquisition is sufficient to acquire the image, while at high-illumination level (rows C), a high-acquisition frequency is required. One can also note that HE permits to acquire a relatively illumination-independent image (as illustrated by images located at the most right column).

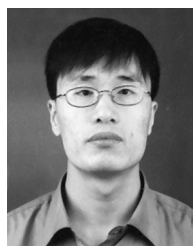
read-out and HE circuitry was implemented in  $0.35\text{-}\mu\text{m}$  CMOS technology with a silicon area of  $3.1 \times 3.2 \text{ mm}^2$ . The HE circuit occupies only a very small fraction of the total silicon area ( $0.1 \times 0.6 \text{ mm}^2$ ). While this paper illustrates the design of a very promising CMOS image sensor and time-based image processing operations, it also raises the need for addressing various new challenges such as timing errors at very high illumination ranges, efficient external interfacing circuitry, as well as improving the image quality. Resolving such issues will undoubtedly result in a very promising new generation of ultralow-power and smart vision sensors.

#### ACKNOWLEDGMENT

The authors would like to thank Dr. D. Martinez for technical discussions and support.

#### REFERENCES

- [1] E. Fossum, "CMOS image sensors: Electronic camera-on-chip," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997.
- [2] A. Bandyopadhyay, J. Lee, R. Robucci, and P. Hasler, "A  $80 \text{ uW/Frame}$   $104 \times 128$  CMOS imager front end for JPEG compression," in *Proc. IEEE Int. Symp. Circuits Syst., ISCAS*, 2005, pp. 5318–5321.
- [3] S. Kawahito *et al.*, "Low-power motion vector estimation using iterative search block-matching methods and a high-speed non-destructive CMOS image sensor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1084–1092, Dec. 2002.
- [4] Y. Ni, F. Devos, M. Boujrad, and J. H. Guan, "Histogram-equalization-based adaptive image sensor for real-time vision," *J. Solid State Circuits*, vol. 32, no. 7, pp. 1027–1036, Jul. 1997.
- [5] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.
- [6] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "Arbitrated address-event representation digital image sensor," *Electron. Lett.*, vol. 37, no. 24, pp. 1443–1445, 2001.
- [7] —, "A biomorphic digital image sensor," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.
- [8] F. Van Rullen and S. J. Thorpe, "Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex," *Neural Comput.*, vol. 13, pp. 1255–1283, 2001.
- [9] X. Qi, X. Guo, and J. G. Harris, "A time-to-first spike CMOS imager," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2004, pp. 23–26.
- [10] A. Kitchen, A. Bermak, and A. Bouzerdoum, "A digital pixel sensor array with programmable dynamic range," *IEEE Trans. Electron Device*, vol. 52, no. 12, pp. 2591–2601, Dec. 2005.
- [11] S. Chen and A. Bermak, "A low power CMOS imager based on time-to-first-spike encoding and fair AER," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2005, pp. 5306–5309.
- [12] M. B. Josephs and J. T. Yantchev, "CMOS design of the tree arbiter element," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 4, pp. 472–476, Dec. 1996.
- [13] C. L. Seitz, "Ideas about arbiters," *Lambda*, vol. 1, pp. 10–14, 1980.
- [14] A. J. Martin, On Seitz' Arbiter Comput. Sci. Dept., Calif. Inst. Technol., Pasadena, Tech. Rep. 5212, 1986.
- [15] D. L. Dill and E. M. Clarke, "Automatic verification of asynchronous circuits using temporal logic," *Proc. Inst. Electr. Eng.*, vol. 133, no. 5, pt. E, pp. 276–282, 1986.
- [16] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, Dept. Comput. Sci., Calif. Inst. Technol., Pasadena, 1992.



**Chen Shoushun** (S'04) received the B.S. degree from the Department of Microelectronics, Peking University, Beijing, China, and the M.E. degree from the Institute of Microelectronics, Chinese Academy of Sciences, Beijing, China, and the Ph.D. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, China, in 2000, 2003, and 2007, respectively. His Master's thesis was related to signal integrity in the design of the "Loogson-1" CPU, which was the first general purpose CPU designed in China. His Ph.D. research work involved the design of low power CMOS image sensors and image processing operations using time-to-first spike (TFS) encoding and asynchronous read out techniques. He is currently a Post-Doc Research Associate at Hong Kong University of Science and Technology. His research interests are in low power CMOS image sensors and on-chip image processing





**Amine Bermak** (M'99–SM'04) received the M.Eng. and Ph.D. degrees in electronic engineering from Paul Sabatier University, Toulouse, France, in 1994 and 1998, respectively.

During his Ph.D., he was part of the Microsystems and Microstructures Research Group at the French National Research Center LAAS-CNRS, where he developed a 3-D VLSI chip for artificial neural network classification and detection applications. He then joined the Advanced Computer Architecture Research Group, York University, York, England, where he was working as a Post-Doc on VLSI implementation of CMM neural network for vision applications in a project funded by British Aerospace. In 1998, he joined Edith Cowan University, Perth, Australia, first as a Research Fellow working on smart vision sensors, then as a Lecturer and a Senior Lecturer in the School of Engineering and Mathematics. He is currently an Assistant Professor with the Electronic and Computer Engineering Department, Hong Kong University of Science and Technology (HKUST), Hong Kong,

China, where he is also serving as the Associate Director of the Computer Engineering Program.

Dr. Bermak was a recipient of many distinguished awards, including the 2004 IEEE Chester Sall Award; HKUST Bechtel Foundation Engineering Teaching Excellence Award" in 2004; and the Best Paper Award at the 2005 International Workshop on System-On-Chip for Real-Time Applications. He is a member of technical program committees of a number of international conferences including the IEEE Custom Integrated Circuit Conference CICC'2006, CICC'2007, the IEEE Consumer Electronics Conference CEC'2007, and the Design Automation and Test in Europe DATE'2007. He is the general co-chair of the 2008 IEEE International Workshop on electronic design test and applications. He is also on the editorial board of IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He is a member of IEEE CAS committee on sensory systems. His research interests are related to VLSI circuits and systems for signal, image processing, sensors and microsystems applications. He has published extensively on the above topics in various journals, book chapters, and refereed international conferences.