

Imaging sparse events at high speed

Gaozhan Cai, Bart Dierickx, Bert Luysaert, Nick Witvrouw, Gerlinde Ruttens
Caeleste CVBA, Hendrik Consciencestraat 1 b, 2800 Mechelen, Belgium
Email: gaozhan.cai@caeleste.be Tel. +32489840507

Introduction

In many scientific imaging applications one needs to read out images with sparse information at extremely high frame rate. In such frames a finite number of pixels are illuminated on a dark (empty) background. A pixel or a group of pixels which is hit by a single or group of photons or particles is called an “event”. Of interest is the center position of the event, much less the size, peak intensity or integrated intensity of the event. Of additional interest is to obtain a better-than-pixel-resolution of the center of the event by combining the information of all pixels of the event. For the application where high resolution (1 to 16 Mpixels) and high speed (1000 to 100000 frames per second) is required, the system builder will be confronted with the almost impossible task to distill individual events sufficiently separated in space and time from a pixel data rate beyond any standard.

The classic solution is to limit the flux of radiations or to time-integrate the events, thus losing position, shape or number information of individual events. More advanced solutions may use smart pixels which count photons or particles in the pixel, and execute operators in the pixel or a local group of pixels to obtain sufficient resolution. The drawback is that normally these pixels are huge (pixel size $\gg 30\mu\text{m}$), thus cannot result in a high resolution high yield imager. We propose a solution to solve this problem.

Architecture: the “SLICE” concept

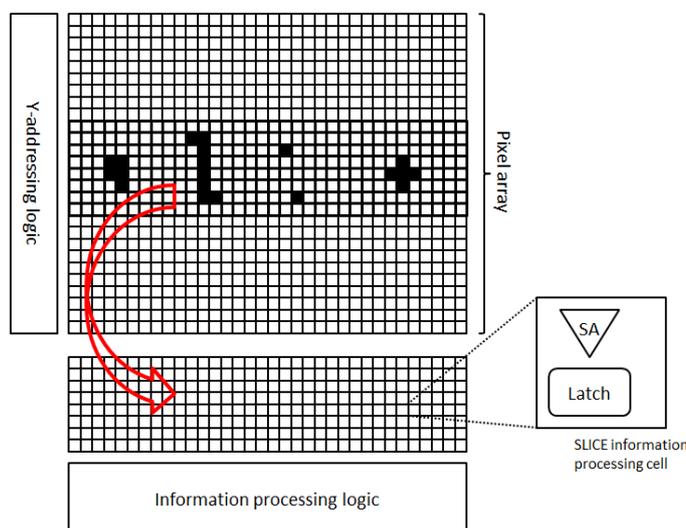


Figure 1 Architecture of the imager actually implementing the “SLICE” algorithm

In order to have small pixels and high readout speed, resulting in the capability to see individual events with the optional capability to perform sub-pixel resolution, we propose to read out a large groups of pixels (actually 16 rows of pixels, called a “SLICE”) to an processing area, which consists of “processing cells” at the bottom of the pixel array, and execute the algorithm there. Figure 1 shows the floor plan, having a $1136(\text{H}) \times 912(\text{V})$ rather simple 4T pixel array, combined with an processing area with an array of physically larger processing cells. Pixel data are buffered and binarized by 16 rows of sense amplifiers “SA”, latching binary data for further processing. Digital information processing logic is then located below.

Comparison of different SLICE algorithms

The simplest algorithm considered is the “bounding box” shown in Figure 2. This algorithm reduces the information per event into 4 numbers, namely the positions of the corners of the event’s bounding box. This algorithm is the easiest to implement, however the obvious drawback is that it may create “false big events”. Like the first event in the figure, the actual shape of the event is much smaller than the bounding box. The 5th event is a “false big event”: two separate hits will be regarded one big event by this algorithm.

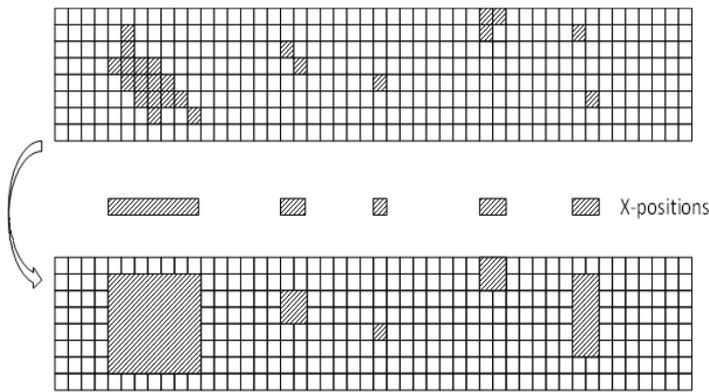


Figure 2 Concept of the “Bounding box” SLICE algorithm

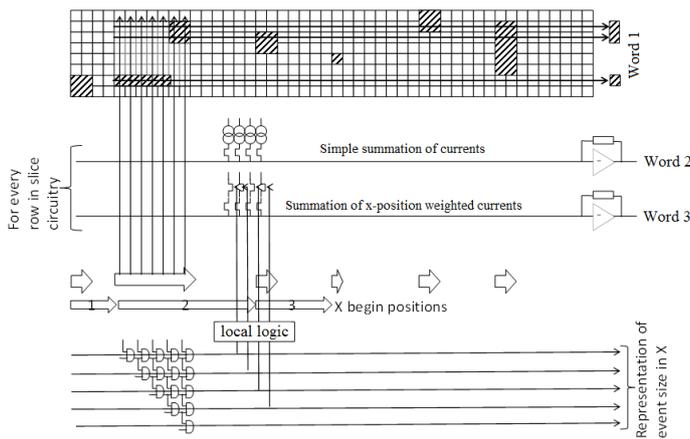


Figure 3 Concept of analog centroid SLICE algorithm reaching sub-pixel resolution

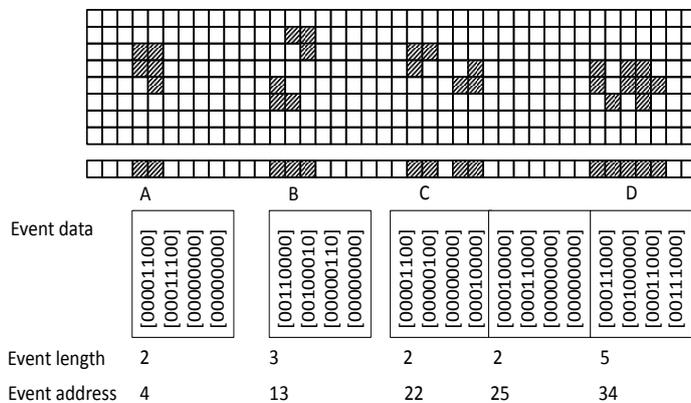


Figure 4 Concept of “binary centroid” SLICE algorithm

Actual implementation

In the actual implementation, one slice contains 16 rows. $4(H) \times 16(V)$ event-data bits are read out since most event length is smaller than 4 columns and available bond pad count is limited. When an event is larger than 4 columns, event length which is a 4bit signal is used to give the actual size of that event. Event address is 11bit. Pixel readout circuit is shown in Figure 5. The readout circuit consists of a unity-gain buffer, a threshold circuit, a sense amplifier and D-latch. Timing diagram of the readout circuit is also shown in Figure 5.

In order to reach high processing speed, we use priority encoder instead of a shift register to scan geometrically distant events and “black” columns of pixels will be skipped. The implementation of the priority encoder is given in Figure 6. The input of the priority encoder is COLUMN<0:8> which come from a wired-OR circuit to indicate which columns contain an event. UPDATE pulse signal will let the

In order to reach sub-pixel resolution, a more advanced “analog centroid” algorithm is proposed in (Figure 3). In this algorithm 3 analog “words” are created and read out for each event. For each event in X-axis in sequence and separately, local logic and a cascade of AND gates will determine which columns belong to that event. The projection of the occurrence of an event on Y-axis will be “word 1”. The output of the local logic will activate a bank of equal current sources (sum of them forming “word 2”) and a bank of position dependent current sources (sum of them forming “word 3”). In this way, the position and exact shape of each event is known. Image is formed by post processing all the event information. A disadvantage of this algorithm is that its performance depends on the fast readout of accurate analog values.

Therefore we actually implemented a “binary centroid algorithm” (Figure 4). In this algorithm, there are three binary words, i.e. event data, event length and event address, instead of analog words to the “represent every event. Event data represent all the hit pixels in the slice. When a pixel is hit above certain threshold level, its event data will be “1”, otherwise it is “0”. Event length represents the number of columns occupied by the event. Event address gives the position of the first column of the event. Original image can be reconstructed by using these three words. The advantage of this algorithm is that digital signals can be readout much faster than analog signals.

COLUMN<0:8> signals stored in the Set-Reset (SR) latch for further processing. ASR is used to reset all the D flip-flops. Each event can be found by simply clocking NEXT_EVENT. The “carry-look-ahead” technique is used in this circuit to shorten the critical path. The actual implementation is much more complex than the circuit shown in the figure since we have 1136 columns of pixels, but their principles are the same. The layout of the prototype and CoB test board is shown in Figure 7. Figure 8 shows part of the layout of the test slice and obtained image. The “black pixels” in the test slice are electrically made insensitive. The pixels indicated in red, are normally sensitive 4T pixel. Since we can only process 4 columns of pixels for each event, when an event length is larger than 4 columns, the remaining columns of that event will be filled with gray color as shown in this figure.

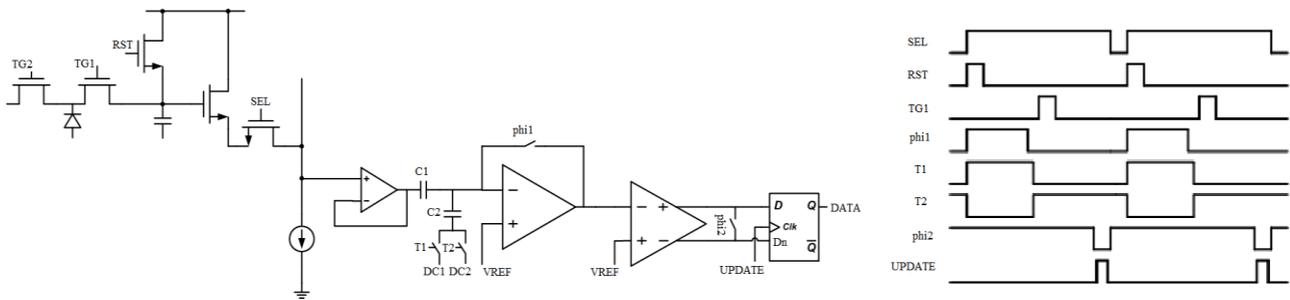


Figure 5 Schematic of analog data path and timing diagram

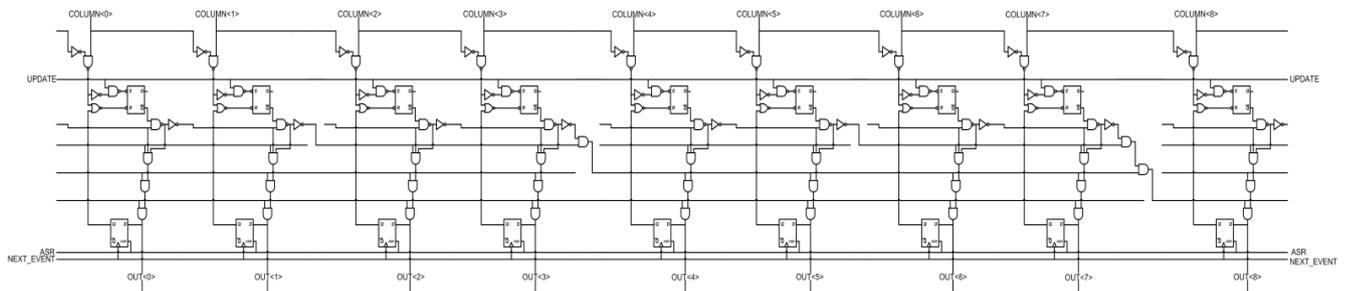


Figure 6 Implementation of priority encoder

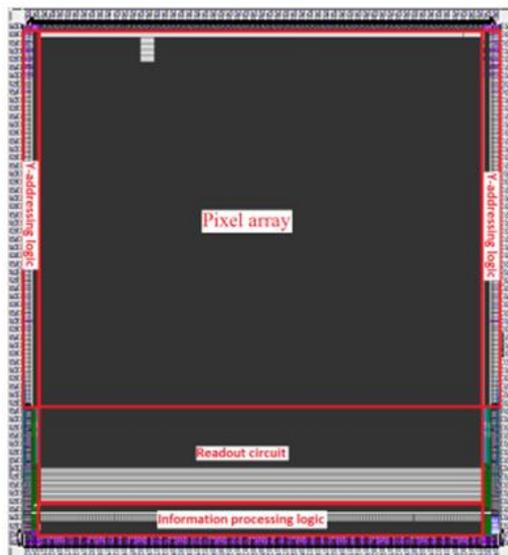


Figure 7 picture of layout and the IC on the CoB test board



Figure 8 Layout of the test slice and obtained + processed image

Measurement results

The actual device can process $50 \text{ events}/\mu\text{s}$ and needs $2\mu\text{s}$ for the analog data path of a full slice to convert the analog signal from the pixel to the digitized event data. In the prototype, we have 57 slices, so the frame rate we can achieve is 8770fps. The threshold for detection is at about 300 electrons: charge packets beyond this value are reliably binarized as a “hit”.

References

- [1] A. Dupret, B. Dupont, M. Vasiliu, B. Dierickx, A. Defernez, “CMOS image sensor architecture for high-speed sparse image content readout”, IISW, Bergen, Norway, June 2009
- [2] J. Kramer, “An on/off transient imager with event-driven, asynchronous read-out”, IEEE International Symposium on Circuits and Systems, vol. 2, p165, May 2002
- [3] B. Dierickx, “Position encoding smart pixel arrays”, Nuclear Instruments and Methods, vol.A305, p.561, 1991.
- [4] Dupret A., Vasiliu M., Devos F., "Performance and power analysis on asynchronous reading of binary arrays", IEEE International Symposium on Circuits and Systems, p5724, May 2006
- [5] O.Barkan et al, “Sparse readout method and apparatus for a pixel array”, US patent 5,107,122, April 1992
- [6] J.Crooks et al, “A CMOS image sensor with in-pixel ADC, timestamp, and sparse readout”, IEEE Sensors Journal, vol. 9, p20, Jan 2009