



Trabajo de Fin de Máster
“Máster Universitario en Microelectrónica:
Diseño y Aplicaciones de Sistemas
Micro/Nanométricos”

**Octopus image sensor with asynchronous
windowed readout**

Escrito por Sergio Palomeque Mangut
Dirigido por Juan Antonio Leñero Bardallo
3 de julio de 2023

Acknowledgments

Para mí madre, para mí padre.

Quiero agradecer al profesor Juan Antonio Leñero Bardallo la supervisión de este trabajo. Juan Antonio generosamente me dió la oportunidad de incorporarme a su equipo de investigación en el IMSE, donde trabajo con comodidad en una línea que me está apasionando. Espero poder seguir investigando junto a él en sensores de imagen basados en eventos durante muchos años venideros.

Este trabajo ha sido posible gracias a mis compañeros del IMSE. Ellos sostienen la actividad del centro, muchas veces sin el reconocimiento que merecen. Gracias a Pablo, a Valentín, a Rubén, a Carlos 'Senior', a Roberto, a Rafa, a Carlos 'Junior', a Iván, a Laura y a Javi.

Y a mi hermano, David. Mi vida es más fácil porque sigo sus pasos.

Contents

Acknowledgements	i
List of Figures	ix
List of Tables	xi
Abstract, Motivation and Objectives	xiii
Abstract, Motivations and Objectives	xiii
1 Fundamentals of Octopus Image Sensors	1
1.1 Introduction	1
1.2 Event-based image sensors	2
1.2.1 Octopus sensor	6
1.2.2 Address event representation (AER)	9
2 Asynchronous Readout	11
2.1 Tenderness towards synchronous readout techniques	11
2.2 Arbiters	13
2.2.1 Greedy arbiter	15
2.2.2 Fair arbiter	17
2.3 Our solution: windowed asynchronous readout	19
3 Octopus Pixel	23
3.1 Introduction	23
3.2 Two-stage comparator	25
3.2.1 Differential pair with active load	26
3.2.2 Common-source amplifier	28
3.2.3 Characterization	29
3.2.4 Design	30
3.3 In-pixel handshake circuit	37
3.3.1 Estimating the line impedance	38
3.4 Layout	42
4 AER Periphery	47
4.1 Sender interface	48
4.2 Encoders	49
4.3 Arbiter tree	50
5 Expected Results and Future Work	55
Bibliography	57

List of Figures

1.1	Comparison between CCD and CMOS image sensors [5]. (a) CCDs transfer simultaneously the signal charges of its photoreceptor, as received, to the next CCD. A fast shift register at the end of the lines sends them to an amplifier for external readout. (b) A CIS uses amplification in each pixel, converting the signal charge into a voltage signal. Although CIS were more sensible to fixed-pattern noise (FPN), this drawback has been eliminated.	2
1.2	Historical evolution of digital image sensors. Originally from [5], modified in this work.	3
1.3	Silicon retina designed by Mahowald [16]. (a) Diagram of the silicon retina showing the resistive network and a single pixel element. (b) Address event representation (AER) scheme. Asynchronous neurons request control of the bus when they generate action potentials and transmit their addresses.	4
1.4	Preliminary image deblurring results on high-speed scene, obtained combining Prophesee Metavision sensor and algorithms with Snapdragon platform.	5
1.5	Pulse-modulation imagers [28] These pixels transmit absolute intensity of incident light through the timing of events in the comparator's output. (a) Time-to-first spike (TTFS) architecture, based on PWM. Brighter pixels spike sooner than darker pixels. (b) Octopus pixel architecture, using PFM encoding. Larger photocurrents are converted into higher frequencies.	7
1.6	Schematics and images from the original octopus sensor [29]. (a) Asynchronous AER readout. A latch buffers the request from the pixel. The request signals are arbitered in the arbiter trees, which select the first pixel that produced an event. The encoders output the address of that selected pixel. (b) Pixel schematic, showing the current-feedback event generator and the in-pixel handshake circuit. (c) Example images with linear intensity (top) and log scale (bottom).	8
1.7	(a) Schematics of an octopus sensor as an AER sender implemented by Leñero [34]. The first stage provides buffering and signal control for requests and acknowledges pixel arbitration. Requests are sent to an arbiter tree, which selects the winning row or column through an acknowledge, which then selects a word from the encoder. (b) Signal flow of the arbitration between rows and columns, and the AER point-to-point link with a 4-phase handshaking protocol [38].	10
2.1	(Left) A 4-phase, bundled-data communication. A sender is connected to a receiver by data lines, a request line, and an acknowledge line. (Right) Its timing diagram. When the request line is low, the data is to be considered invalid and liable to change at any time. The sender usually waits for the acknowledge signal to remove the data from the data bus.	14

2.2	Block diagram of an asynchronous readout AER implementation for an event-based image sensor with a resolution of $L \times H$ pixels in a grid arrangement. The row and column interface can include pull-up elements for the wired ORs, buffers, and latches, as well as control logic for handshake signals. The main limitation we face when scaling pixel count is the size of the wired ORs. These column and row metal lines get longer and have more pull-down transistors connected to them, causing the capacitance of the lines to increase, degrading signal integrity and slowing the readout process. In the arbiter trees, each line is actually formed by two wires: a request and an acknowledge. There are $\ln(L)$ (or $\ln(H)$) levels and $L - 1$ (or $H - 1$) arbiters in each tree, in the case of square matrix dimensions.	15
2.3	Circuit implementation of a glitch free 2-way greedy arbiter [49]. The RS bistable ($req_0 = S, req_1 = R$) and metastability filter form the core mutual exclusion element. It exclusively acknowledges one of two incoming requests, only if its own outgoing request is acknowledged. The glitch filter prevents a condition where the output request switches logic briefly. The transistor count is 27.	16
2.4	Token movement in an arbiter tree to handle the requests from neurons. In a greedy arbiter tree, neurons located near each other may form a greedy path because the token does not need to reach to the top of the tree. A fair mechanism is needed to ensure that requests from an arbiter are blocked until its parent's acknowledge is cleared.	17
2.5	Asymmetric Muller C-element, an essential component for understanding the fair arbiter behavior. (a) Symbol of the circuit. (b) Transistor implementation. (c) Truth table.	18
2.6	Asynchronous fair arbiter implemented by Fok and Boahen [47]. The RS bistable with NOR gates implies the use of active-low logic for requests. A metastability filter is implemented with two inverters at the output of the latch, with pull-down source connected to the reciprocal request. The asymmetric C-element guarantees that no greedy path is created. Transistor count is higher than for the greedy arbiter.	18
2.7	Image artifact occurs by the mismatch between event generation time and readout time for high event rates [51].	18
2.8	The two readout modes proposed in this work. The octopus sensor we designed implements a switch to decouple event generation from transmission through a signal called Window. (Left) The sparse readout mode is based on the works of Karen Adam [53], and will enable us to scan the events of those pixels in the matrix that are not receiving the attention of the periphery. (Right) The quanta image mode, based on the sensor conceived by Fossum [55], will use the window to randomly sample the Poisson process of the spikes produced in the matrix.	20
2.9	Results from the Matlab models developed by Méndez-Romero [54]. In the top, an example bright image is used to evaluate the saturation in the AER channel and the impact of quanta-based acquisition. In the bottom, a 3D representation of the spikes in each pixel. In the original image, the most illuminated pixels have 55000 spikes, whereas its darker parts have around 2000. Roberto used a linear transformation that assigned spiking frequencies between the brightest and to the darkest pixel. Then, he modeled saturation by comparing the sum of all spikes in the matrix with the common saturation rate of the AER periphery. If the sum is bigger, he erased spikes of the less illuminated pixels. Last, he modeled quanta acquisition by dividing the original image in a cube of binary bit planes, each indicating whether the pixels had spiked at least once, and randomly eliminated spikes of each plane. The number of spikes is better distributed among all pixels.	21

2.10	Results from a quanta-based post-processing using an image acquired by an octopus sensor [55]. The original images were acquired using an octopus sensor, and are formed by around 150.000 events each. After post-processing, using a window with a uniform distribution and 0.7 average duty cycle to sample the spikes, the images are still recognizable. The result is that each image can be represented with half the events. For each one of the images, histograms and the applied tone mapping curves are shown.	22
3.1	Pixel schematic and signal flow [30]. The incident light is encoded in the frequency of V_{SPIKE} . Spikes produce a self-reset in the comparator's input. The circuit implements an integrate-and-fire neuron model. The in-pixel handshake interface with memory capacity implements the logic for arbitration between pixels and AER communication. The window decouples spike generation from communication. Transistor sizes ($\mu\text{m}/\mu\text{m}$): M1 = 0.24/0.34, M2 = 0.45/0.18, M3 = M4 = 0.24/0.18, M5 = 2/0.18, M6 = M7 = 1.6/0.18. Total transistor count: 37. C = 20 fF. Discharging time of the capacitance due to current leakage: 336.77 μs	24
3.2	(a) Transfer characteristic of a comparator considering finite static gain and offset voltage (left) and hysteresis (right). (b) We use the two-stage comparator with NMOS input. Transistor sizes ($\mu\text{m}/\mu\text{m}$): M1 = M2 = 0.67/1.1, M3 = M4 = 1.34/2.2, M5 = M7 = 0.37/0.48, M6 = 0.3/0.6.	25
3.3	(a) Large-signal analysis of the first stage of the comparator, with the two signal paths affecting the single-ended output. Note that V_{in} polarity respect to a 5-T OTA depiction is reversed, as the second stage of the comparator inverts the output. (b) The V_{in}^+ signal path can be studied as a NMOS common source amplifier. (c) Influence of V_{in} in V_{o1} and in the current of the right branch.	26
3.4	(a) Small-signal circuit of the comparator's first stage. Although V_{bot} is a constant DC reference, we include its AC component for the calculation procedure. (b) Small-signal equivalent of a diode-connected PMOS.	27
3.5	(a) Small-signal equivalent circuit of the common-source amplifier. (b) High-frequency model of the stage with Miller's approximation.	29
3.6	Histogram and its corresponding probability distribution for the output voltage in the offset Monte Carlo testbench, with I_{bias} equal to 50 nA. The output contains the common-mode value $V_{\text{bot}} = 1.5 \text{ V}$ and the offset, which is represented by the standard deviation value $\sigma = 6.66 \text{ mV}$	33
3.7	Magnitude and phase in a corner AC analysis of the open-loop comparator, without the reset feedback. For nominal values the DC gain is 66.78 dB and the pole is located at 47.11 kHz. We define the process corners at -40 $^{\circ}\text{C}$ and 85 $^{\circ}\text{C}$ for fast-fast (FF), slow-slow (SS), fast-slow (FS), and slow-fast (SF) transistor models.	34
3.8	Histogram of V_{bot} with two different photocurrents in a Monte Carlo simulation. The variations are the effect of mismatch. The displacement to the left is produced at high frequencies. The comparison point is set at 1 V.	35
3.9	Histogram of the spiking frequency of a pixel with two different photocurrents in a Monte Carlo simulation. Same conditions as above.	36
3.10	(Left) Original test image in greyscale. (Right) Test image after adding random frequency variations with the achieved standard deviation.	36
3.11	Transient simulation characterizing the behavior of the comparator. The fall time is 60 ns and the rise time is 15 ns.	37

3.12	(a) Wired-NOR circuit implemented in the pixels' rows and columns, and the sender block in the periphery (the circuits in the sender are explained later). C_{out} and R_{out} represent the drain capacitance, the line crosstalk, the output resistance and the line resistance, respectively. (b) Equivalent circuit of the wired-NOR, neglecting the influence of r_o and R_{GND} . $M_{\text{PD}}=2/0.18$, $M_{\text{PU}}=18/0.7$, $R_{\text{line}}=235.6\Omega$ and $C_{\text{line}}=600.1\text{ fF}$	38
3.13	Transient simulation of the circuit in Fig. 3.12(b), implementing a wired-NOR in a row of 95 pixels.	39
3.14	Handshaking protocol in a transient simulation for a cluster of four pixels, with two transmitted spikes. First, the voltage V_P in the input C_{ph} capacitance decreases until it reaches the comparison value V_{bot} . The output of the comparator resets V_P to V_{DD} . The spike at $t = 105\mu\text{s}$ is transmitted if it occurs while WINDOW is high. The spike generates the <code>_REQ_X<0></code> , which is answered by the row arbiters with a <code>ACK_X<0></code> and <code>RESET_X<0></code> . Then, <code>_REQ_Y<0></code> produces <code>BUS_REQ</code> and <code>RESET_Y<0></code> ; the sensor (AER transmitter) sends its address to a external processor (AER receiver), while the spike stored in the pixel is terminated.	40
3.15	Transient simulation with a sinusoidal input photocurrent being codified in spikes at different frequencies.	41
3.16	Transient simulation to represent the window masking capability, in conjunction with its in-pixel memory. Events are stored for $334.44\mu\text{s}$ before they are discarded due to current leakage. Spikes and requests use active-low logic in the in-pixel handshake protocol.	42
3.17	Layout of a pixel cluster, showing the NWELL, PPLUS and active layers. Pixels in the cluster share the NWELL, as well as with their neighbors. PWELL is connected to ground in every cluster. By seeing the active areas we can interpret the arrangement of the transistors.	43
3.18	Layout of a pixel. We achieved a pixel pitch of $18.63\mu\text{m} \times 12.38\mu\text{m}$, and a fill factor of 47.5%. The corners of the photodiode are chamfered. M6 and M5 layers are used for power supply, ground and MIMCAP. M4 and M3 are used for routing signals that cross the pixel matrix.	44
3.19	Layout of a cluster with four pixels. Pixels have a mirror disposal to facilitate the fulfillment of design rule checks (DRC) and achieve a more compact arrangement.	45
3.20	Layout of the pixel matrix with a 96×64 resolution.	46
4.1	Block diagram of the AER periphery and its signals. The wired-NORs are marked in red. The pixel matrix sends a request in both dimensions and receives a row acknowledgment (there is no need for column acknowledgment) and a reset in both dimensions. The pixel matrix is omitted for brevity.	47
4.2	Schematic of the sender. It implements the buffering of the wired-NOR signal, which goes directly to the arbiter tree. An acknowledge arrives and is latched, triggering a chip request in the AER bus for its corresponding dimension. The acknowledge is also inverted to activate certain bits of the encoder. When the acknowledge from the AER sender is received, the pixel is reset. Notice that all signals are related to their dimension, since there is a sender for rows and columns.	48
4.3	Schematic of the row encoder. All transistors have minimum length and $W = 3\mu\text{m}$. The request line crosses the block without connections to reach the arbiter tree. The output is an N bit signal to encode 2^N addresses. There is another encoder for columns with one more bit. Each encoder has an output bus.	49
4.4	(a) Schematic of a single greedy arbiter. (b) Last three levels of the greedy arbiter column tree. We need to use a dummy connection because there are 96 columns, which is not an exponential number with base 2.	50

-
- 4.5 Post-layout verification of the AER blocks. The column circuits (sender, encoder, and arbitration tree) are tested with three requests coming from neighbor and extreme pixels. ACK1 comes 6.45 ns after REQ1 goes high, ACK2 comes 0.98 ns after REQ1 goes low (with REQ2 active), and ACK3 comes 6.63 ns after REQ2 goes low (with REQ3 active). 51
- 4.6 Layout of the row periphery, with the sender (left), the encoder (middle), and the arbiter tree (right). This block is connected directly to the pixel matrix and contains 64 senders, 64 encoders, and 63 arbiters in a 6 stages tree. The column periphery is not shown for brevity, but the dimensions are almost identical: 96 senders, 96 encoders, and 96 arbiters in a 7 stages tree. The sixth stage has a dummy connection. 52
- 4.7 Layout of the chip's top view, with the size of two mini-ASICs. 53

List of Tables

2.1	Comparison Between Asynchronous and Synchronous Readout Techniques in Dynamic Vision Sensors in Works Reported in the Literature	13
2.2	Temporal Characteristics of our Octopus Sensor	21
5.1	Comparison to Other Octopus Sensors reported in the Literature	55

Abstract, Motivation and Objectives

Abstract

This Master's Thesis focuses on the design of an octopus image sensor using UMC0.18 technology. We incorporated a logic for decoupling spike generation from readout, aiming to solve the main issues that affect octopus architectures. This type of event-based sensor encodes the intensity of the light hitting each pixel in a frequency of pulses, known as events or spikes. Spikes are subsequently read out using asynchronous logic involving wired-NORs, arbiters, and encoders. The operation of the imager is divided into three main parts: (1) integration of light and conversion into a one-bit pixel request signal, (2) row and column arbitration to select the output pixel, and (3) encoding of the pixel address, acknowledgment, and reset.

Chapter 1 provides an overview of the development of event-based imagers and compares them with mainstream digital image sensors such as CCD or CIS APS. The unique characteristics of event-based image sensors are presented, with a specific focus on the octopus sensor and the AER asynchronous readout protocol.

In Chapter 2, the limitations of the asynchronous readout scheme are discussed in greater detail compared to conventional synchronous readout. Various types of arbiters used in asynchronous systems are explained. The final section of Chapter 2 encloses the theoretical framework supporting the implementation of the decoupling logic and motivating the integration of the chip described in this work. The two readout modes to be tested in the laboratory in future research, namely the sparse readout mode and the quanta image mode, are presented.

Chapter 3 delves into the design details of the pixel, providing an explanation of its operation and a step-by-step guide to designing the comparator, which is the most sensible component within the octopus pixel. The functioning of the in-pixel handshake interface is highlighted, and layout and post-layout verifications are performed.

Chapter 4 revisits the AER circuits discussed in the previous chapters, showcasing their implementation in the designed sensor along with the layout and post-layout results.

Finally, Chapter 5 describes the results we expect to obtain in the laboratory and the potential of spike decoupling in asynchronous readout for all classes of event-based vision systems.

Motivation and Objectives

A well-known feature of octopus sensors is that brighter pixels receive more attention in asynchronous readout. Because the integration threshold is reached faster in brighter pixels, they request bus access more frequently. This can lead to motion artifacts in the image, as a bunch of pixels hoards the whole bus bandwidth for themselves. Traditional synchronous readout schemes allocate an equal portion of the bandwidth to all pixels regardless of their activity, effectively preventing such congestion. The limited bandwidth of the arbiters in the asynchronous periphery has hindered the application of octopus sensors to specific tasks, such as tracking small and intensely luminous light sources.

In this work, we propose a simple solution using an asynchronous windowed readout scheme to alleviate congestion in the AER channel, reducing the bias toward brighter pixels in favor of darker ones. Our solution is based in the assumption that it is more advantageous to have spatially sparse information, meaning that it is preferable to have more pixels firing, even if each pixel fires less frequently. By decoupling event generation from readout and leveraging the in-pixel aging storage mechanism, we anticipate being able to capture spike data from a larger number of pixels in high-illuminance scenarios, regardless of any spikes that may be missed from more active pixels.

Also, the sensor we designed aims to test the hypothesis that by employing observation windows to capture spikes, we can effectively preserve the most relevant aspects of the image by leveraging the Poisson distribution of the spiking matrix output. This method is inspired by the quanta image sensor (QIS). While QIS relies on single photon detection and justifies its usage of Poisson arrival statistics based on the random nature of photon arrivals, we seek to experimentally verify whether we can replicate a similar readout scheme in the laboratory by constructing binary bit planes using the spikes from our octopus pixels.

The result of this work is a design that was sent for manufacturing, and that will allow us to specifically investigate these approaches in an octopus sensor. The testing of the fabricated sensor and of the proposed readout schemes is beyond the scopes of this work. Although this is the only class of event-based sensor that performs light-to-frequency conversion, we expect that future studies on our sensor can have an impact on related aspects of other event-based sensors, such as motion artifacts, scalability, and data compression. To achieve this result, this work have dealt with the following tasks:

- The successful design and development of an octopus image sensor in UMC0.18. The design process required careful consideration to ensure its viability for manufacturing and future studying and dissemination of results.
- The primary purpose of this sensor is to serve as an experimental platform for validating the proposed readout schemes. Future laboratory experiments will evaluate the extent of data reduction achievable through the decoupling technique and examine the trade-offs between data reduction and image quality.
- A thorough examination of the evolution and current state-of-the-art in readout schemes employed in event-based imagers was conducted, both synchronous and asynchronous. This involved studying existing techniques and understanding their strengths and limitations.
- Establishing a clear and well-justified theoretical framework for the future implementation of the asynchronous windowed readout was an essential aspect of this work. The framework provides the necessary guidelines and principles for designing and implementing the decoupling logic and exploring its potential benefits in the future.

CHAPTER 1

Fundamentals of Octopus Image Sensors

1.1 Introduction

The oldest surviving photograph was taken in 1826 [1]. To capture it, French inventor Nicéphore Niépce used a camera obscura containing a metal plate coated with a light-sensitive bitumen and faced it out of a window in his estate in Le Gras. Due to the low light sensitivity of the material, Niépce exposed the sensor for a few days in broad sunlight. He then dipped the plate in a diluted lavender oil bath to develop the latent image, dissolving the less-exposed bitumen parts and thus resulting in a negative image. What Niépce had conceived is a chemical process that we now call contact print, the basis of film photography. Despite its lack of visual appeal, the image stands as a testament to the early crafts of image-sensing techniques, a concept that would eventually evolve into the sophisticated image-capture technologies we enjoy today.

Modern photography is intrinsically associated with digital cameras. A digital image sensor is fabricated on a piece of semiconductor, which alter its electronic properties when struck with photons of a specific energy. From here, we can measure the intensity of the light by measuring the current or the voltage induced in the semiconductor. In 1969, AT&T Bell Labs engineers Willard Boyle and George Smith developed the charge-coupled device (CCD), leading to the first digital image sensor to take off to the mass market [2]. CCD image sensors originally used MOS capacitors to accumulate the photo-generated charge in silicon, which were unaffected by the manufacturing issues that troubled early passive and active pixel sensors. The packets of charge in a line of MOS capacitors were then shifted to the next line, until reaching a fast shift register for amplification and readout. This architecture was improved in 1980 with the pinned photodiode (PPD), that solved its inherent shutter lag. Due to their invention, Boyle and Smith won the 2009 Nobel Prize in Physics, along with Michael Tompsett [3].

CCDs had a long list of drawbacks that seem unsolvable. The thousands of charge transfer steps required to readout each pixel consumed considerable energy, making them hard to scale to larger pixel counts. Also, they required external processing components outside the chip, which made them bulky, heavy, and power-hungry; and suffered from undesired effects such as smearing or blooming. These constraints paved the way in 1992 for Eric Fossum to develop the active pixel sensor (APS) or CMOS image sensor (CIS) [4]. While he was working for the NASA Jet Propulsion Laboratory at Caltech, he figured out an architecture that had in-pixel charge transfer and amplification, so NASA could embed lower-powered and more compact image systems in their missions. Also, CMOS sensors could be manufactured in standard CMOS fabrication processes, which made production a lot easier and more affordable than CCDs.

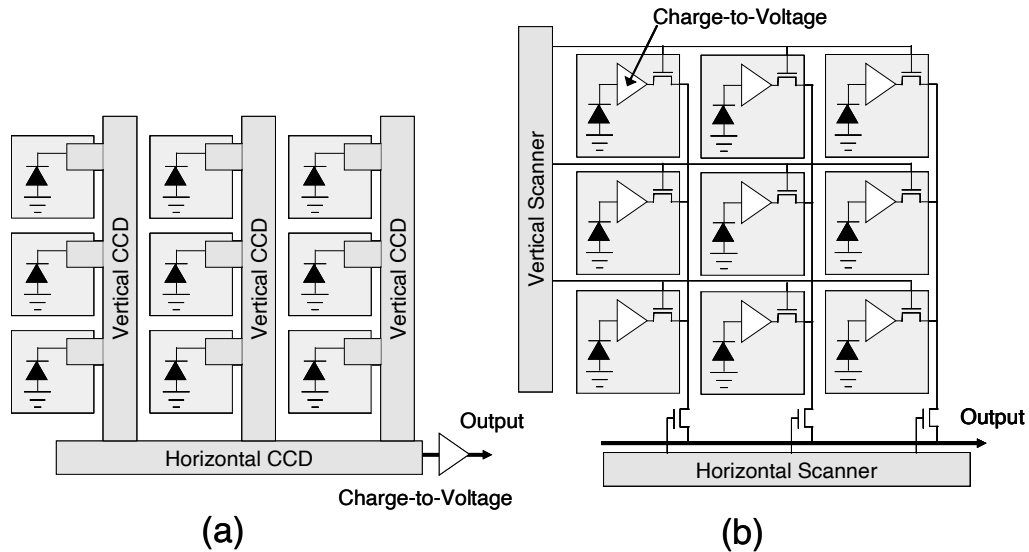


Figure 1.1: Comparison between CCD and CMOS image sensors [5]. (a) CCDs transfer simultaneously the signal charges of its photoreceptor, as received, to the next CCD. A fast shift register at the end of the lines sends them to an amplifier for external readout. (b) A CIS uses amplification in each pixel, converting the signal charge into a voltage signal. Although CIS were more sensible to fixed-pattern noise (FPN), this drawback has been eliminated.

Designers could integrate all the timing, control, and signal-processing CMOS circuits on the same CIS. Fossum founded the spinoff company Photobit in 1995 to develop and commercialize the CMOS APS. Only six years later, Photobit was acquired by Micron. During those days, Micron became the world’s largest supplier of image sensors [6]. As of 2023, those charts are led by Sony and Samsung, but the industry is still largely dependent on IPs licensed by Caltech [7, 8]. It is estimated that over 6.5 billion CMOS APS were shipped worldwide in 2020.

As we have seen, mainstream image sensors consist of a photosensitive array that provides absolute illumination values at each point of an image, typically acquired frame-by-frame at a fixed sequential rate. These systems differ greatly from biological systems found in most animals, where cells operate independently and asynchronously, primarily focused on reporting changes. In an attempt to mimic the behavior of biological systems, the development of neuromorphic silicon retinas or event-based image sensors has run parallel to the development of CIS [9]. These endeavors languished in front of CIS chips for three decades. However, event-based sensors are now gaining industrial traction due to their high temporal resolution, very high dynamic range, low power consumption, and high pixel bandwidth [10]. The arrival of industrial giants in this field is bringing new methods and techniques for event-based sensing, with promising applications in machine vision and related AI tasks in which image quality is not essential.

1.2 Event-based image sensors

“These guys are saying that a nerve membrane works like a transistor. Is this right? —Delbruck asked Mead brusquely” [11]. It is 1967, and Carver Mead is a teacher at the California Institute of Technology (Caltech), in Pasadena. Mead has a growing reputation as a world-beating expert on transistor physics. He was an advisor and collaborator of Gordon Moore while working at Fairchild Semiconductor, who credits him for coining the term Moore’s Law [12]. Max Delbruck,

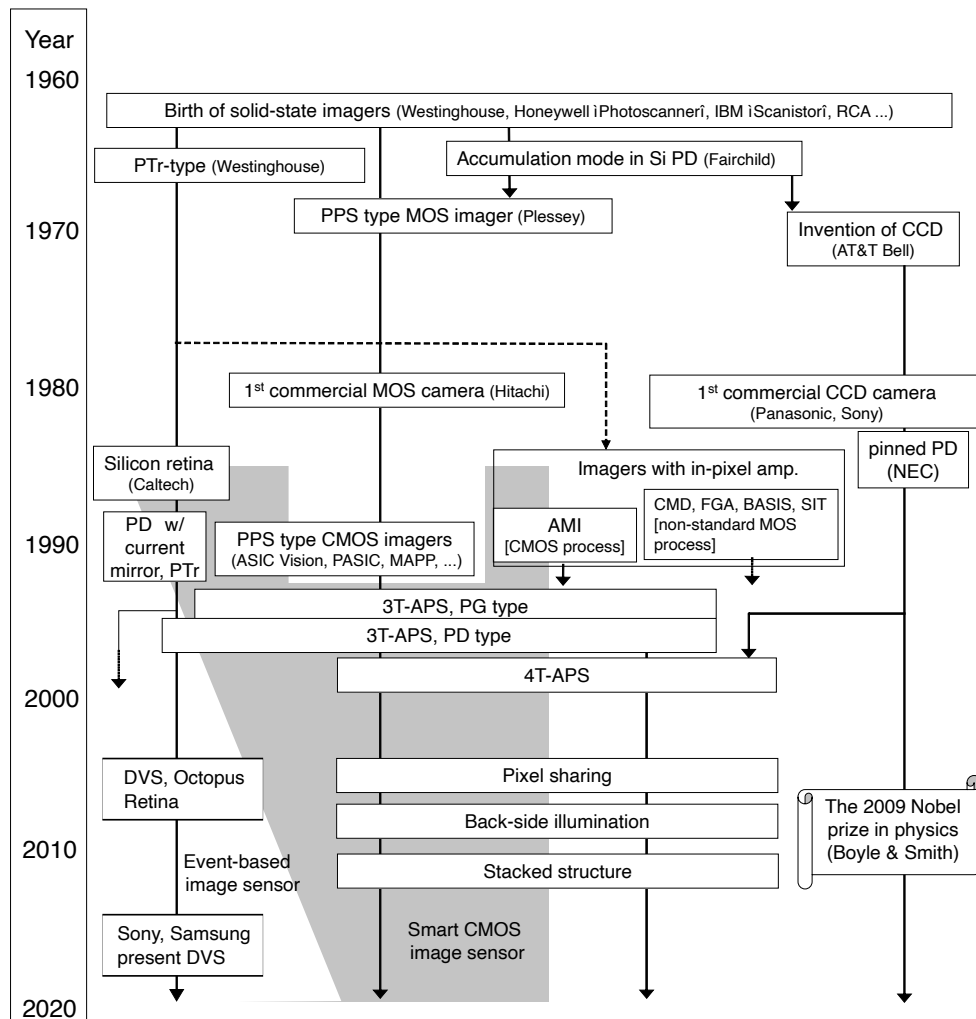


Figure 1.2: Historical evolution of digital image sensors. Originally from [5], modified in this work.

a biophysicist also working at Caltech on bacteria and their viruses, earned the 1969 Nobel Prize in Physiology or Medicine. After that initial meeting, Mead would dedicate his research efforts to neuromorphic engineering: a field for analog, digital, and mixed-signal VLSI implementation of neural systems models. Mead's group at Caltech gathered the most relevant names in the field, with the likes of Misha Mahowald, who implemented the first silicon VLSI retina; Tobi Delbruck, son of Max and developer of the dynamic vision sensor (DVS) [13]; or Kwabena Boahen, who formalized the address event representation (AER) communication channels between neuromorphic chips [14]. They were strongly influenced by the works of Fukusima and its neocognitron. To this day, the greatest success of neuromorphic systems has been the emulation of vision signal acquisition and transduction, resulting in a family of event-based image sensors.

Neurons play a fundamental role in the brain by receiving sensory information, processing it, and transmitting the results to other neurons, ultimately influencing bodily functions and movement. Neuronal communication occurs through action potentials, also known as spikes, which are sudden changes in a neuron's membrane potential. These spikes have a fixed shape and amplitude and can be chemically transmitted to other neurons. Simplified models of neurons suggest that spikes are generated when the integration of electrical current input exceeds a certain threshold. This

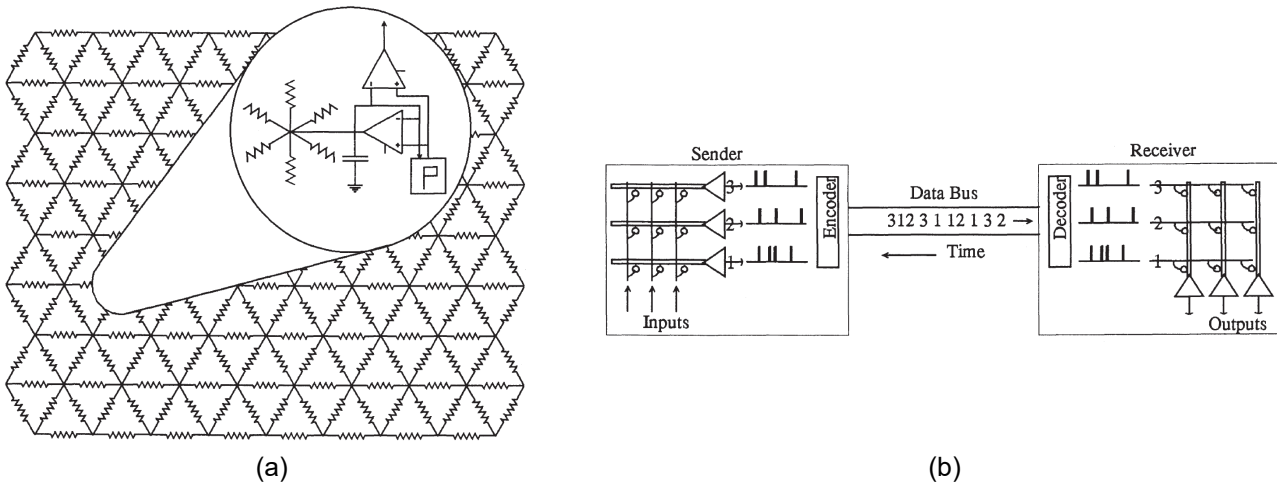


Figure 1.3: Silicon retina designed by Mahowald [16]. (a) Diagram of the silicon retina showing the resistive network and a single pixel element. (b) Address event representation (AER) scheme. Asynchronous neurons request control of the bus when they generate action potentials and transmit their addresses.

process, often referred as the integrate-and-fire neuron model, can be compared to a tipping bucket, where the neuron collects and integrates input current until it reaches a threshold, at which point a spike is emitted. Consequently, the neuron’s output is encoded in the timing of these spikes, creating a stream of spikes with a constant amplitude that is transmitted to other neurons.

Event-based image sensors, or event-driven image sensors, work by compressing light intensity values within their pixels. Unlike mainstream image sensors, these pixels do not transmit analog or digital absolute light intensity information. Instead, they perform all the necessary analog processing at the pixel level and transmit an action potential called an event or spike. As a result, an event-based pixel matrix produces a set of spikes, or events, based on the pixel architecture. Although we can think of the spike as a 1-bit digital signal, it behaves more like an impulse that carries the time at which the event occurred. The sensor typically encodes the timestamp of the spike either on-chip or externally while transmitting the address of the firing pixel. Consequently, event-based sensors are intrinsically asynchronous, with each pixel autonomously detecting and transmitting events. This event-driven operation frees the sensor from a fixed time-step, allowing it to remain idle until a spike occurs, naturally responding to the visual scene. Efficient spike encoding ensures that the number of updates is minimized.

We can distinguish several classes of event-based sensors depending on how events are produced [15]. The original works published by Mahowald and her colleagues implemented spatial contrast detection by computing the average illumination values between neighboring pixels through a diffusive network of MOS variable resistors and comparing it with local pixels [16]. If they differ, the pixel spikes. Mahowald already implemented the AER protocol in its silicon retina.

Mahowald’s design was later improved by Zaghoul and Boahen [9], with additional features modeled after layers of the biological retina. But these silicon retinas had pixels that were vastly too big and noisy. Circuit complexity, silicon area, fill factors, noise levels, mismatch: they all languished in front of its CIS rival. The neuromorphic community used these first chips to demonstrate neurobiological models and theories, but did not find traction in real-world applications.

Eventually, the class of event-based sensor that stood out was the DVS. This family of sensors are only sensitive to temporal contrast, achieving extremely high temporal resolution (in the order

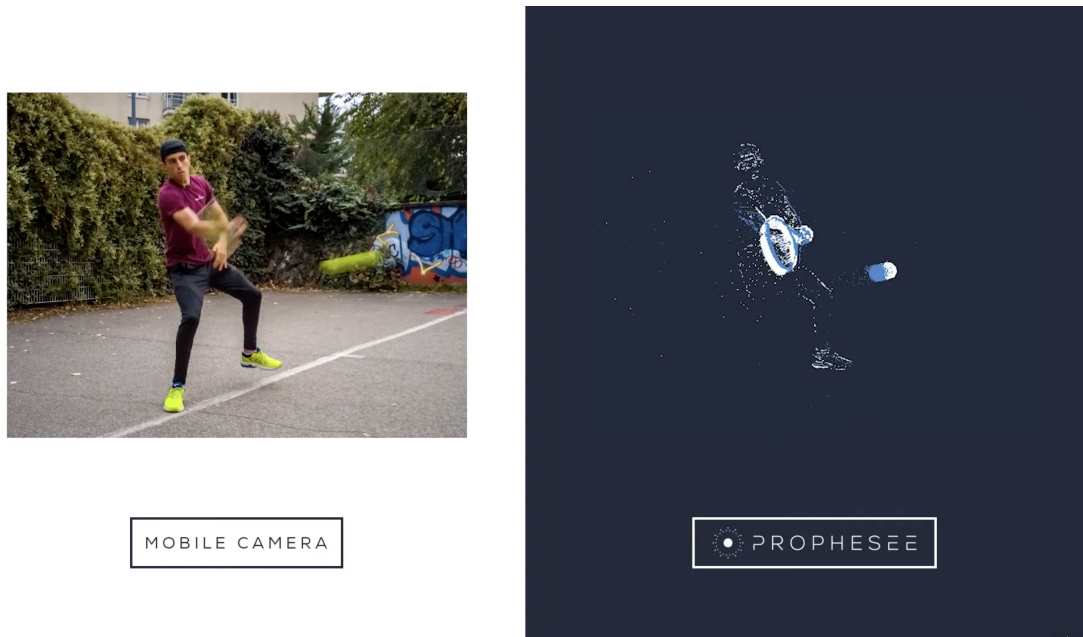


Figure 1.4: Preliminary image deblurring results on high-speed scene, obtained combining Prophesee Metavision sensor and algorithms with Snapdragon platform.

of μs), intrascene dynamic range (up to 140 dB) and low latency (also μs). More importantly, these sensors squeeze a lot of information about the visual scene without the need of external processors, optimizing data transfer, storage, and processing, hence increasing power efficiency and compactness of the vision system. It is a prime example of sparse computing: the removal of useless and redundant data, computing only where and when needed. The foundational paradigm of neuromorphic engineering. Besides, by shifting performance constraints from the voltage domain into the time domain, dynamic range is no longer limited by the power-supply rails, thus providing relative immunity to the aggressive supply voltage scaling of modern CMOS technologies. Nowadays, DVS are find frequent applications in robotics, tactile sensing, high-speed control, driving, space and computational photography.

Delbruck and Patrick Lichtsteiner came up with the DVS while working on the European project CAVIAR [17]. Their pixel design had three main blocks: an active unity-gain logarithmic photoreceptor, which is buffered to a capacitive-feedback amplifier that computes the temporal derivative of the voltage induced by the photoreceptor, and a couple of comparators that monitor the amplified voltage. This pixel produces ON and OFF events, depending on whether there is a positive or negative temporal contrast. These sensors have received considerable attention by the industry, to the point where they have become synonymous with event sensors, and are well suited for high-speed motion detection and analysis, object tracking, and shape recognition, among many others [10].

DVS cameras are already available on the shelves. Actually, some companies already have several generations in their line of products. Sony's IMX636ES was designed in collaboration with Prophesee, a French company with four sensors in their catalogue. Then there is iniVation, a spin-off of the ETH Zurich participated by Delbruck, with three manufactured sensors. Samsung also has three, whereas the Chinese start-up has launched two. All of them have published their results in different editions of the ISSCC [18, 19, 20, 21, 22, 23, 24].

Delbruck's pixel architecture served as the foundation for subsequent generations of DVS developed by the major players, which have been incorporating new features into their designs.

Delbruck itself reported the DAVIS in 2014 [25], a hybrid sensor that combines the DVS and APS. The DAVIS enabled both conventional frame-based sampling of intensity and asynchronous detection of logarithmic intensity changes. Besides, a significant breakthrough seems to be underway, as Prophesee garnered considerable attention at the 2023 Edition of the Mobile World Congress. Event-based sensors have emerged as a viable option to complement frame-based methods, particularly in consumer-facing products. In light of this, Prophesee has partnered with Qualcomm to enhance the quality of smartphone cameras in fast-moving dynamic scenes. It won't be long before we start seeing these products integrated into wearables or AR/VR headsets.

However, recent literature indicates a shift in the principles of these sensors. There is a growing trend among companies towards synchronous readout schemes, deviating from the traditional asynchronous arbitration of spikes using wired-NOR circuits. The limitations of large sensor formats and the unpredictable arbitration time of asynchronous systems have led companies to favor more deterministic approaches. Although Prophesee and Sony previously utilized a mix of asynchronous arbitration between pixel rows and sequential scanning of columns, their most recent research reveals a shift towards event-based vision sensing combined with conventional framed image acquisition.

A seemingly inevitable drawback of event-based sensors is their lower spatial resolution when compared to CIS imagers: pixel count is usually below 1 MP. Besides the limits imposed by current asynchronous handshake circuits, the complexity of the pixels' circuitry is a constraint for reduced pixel pitch. Traditionally, pixel sizes have ranged from $17\ \mu\text{m}$ to $50\ \mu\text{m}$, although new designs have achieved sizes as small as $4.95\ \mu\text{m}$ through vertical integration [26]. In this work, we address the issue of scalability in event-based sensors. Before delving into the limits of the AER readout scheme, let's provide more details about the specific class of event-based sensors implemented in our work.

1.2.1 Octopus sensor

We have already discussed two classes of event-based sensors: those that encode spatial contrast in their events and those that are sensitive only to temporal contrast. A third class consists of bioinspired image sensors that transmit pixel absolute intensity in the timing of events. While many of these sensors do not achieve redundancy suppression or latency reduction, they do benefit from other properties of event-based sensing, such as high dynamic range, low-power operation, and high SNR. Pulse-modulated (PM) imagers, with various architectures, usually employ a comparator as a fundamental building block [27], and benefit from smaller pixel size compared to other event-based sensors. Unlike conventional CIS imagers, which integrate photocurrent for a specific scanning period and then read the integrated voltage in a raster scan, PM image sensors integrate photocurrent until it reaches a fixed voltage threshold, autonomously generating a spike when crossed.

One implementation of time-domain encoding is the time-to-first spike (TTFS) image sensor, which uses pulse-width modulation (PWM) schemes by measuring the time between a pixel's reset and the switching of the comparator. Time is measured using pixel-level or global timers. A measurement cycle starts when the photodiode voltage is reset to a defined voltage level, which is then discharged by the photo-generated current. A comparator continuously compares the integration voltage ramp to a voltage reference. The incident light intensity is inversely proportional to the integration time. Then again, the measurement is not externally governed as in a conventional APS or CCD, and each pixel is allowed to autonomously choose its own optimal integration time. Once the threshold is reached, a pulse is generated.

In 2011, Christoph Posch introduced the asynchronous time-based image sensor (ATIS) [28], which combined a DVS architecture for change detection with absolute exposure measurement carried out locally by individual pixels using a TTFS circuit triggered by change detection. The

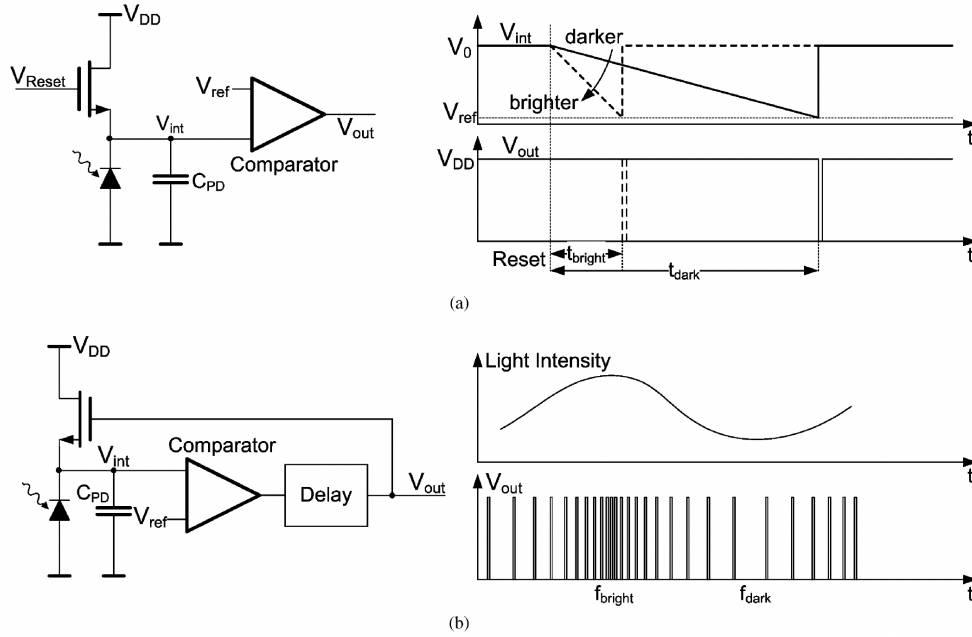


Figure 1.5: Pulse-modulation imagers [28] These pixels transmit absolute intensity of incident light through the timing of events in the comparator’s output. (a) Time-to-first spike (TTFS) architecture, based on PWM. Brighter pixels spike sooner than darker pixels. (b) Octopus pixel architecture, using PFM encoding. Larger photocurrents are converted into higher frequencies.

authors reported an intra-scene dynamic range of 143 dB static, 125 dB at 30 fps equivalent temporal resolution, a typical SNR of 56 dB, and a FPN under 0.25%. The sensor was fabricated on UMC180, with a supply voltage of 3.3 V and 1.8 V for analog and digital circuits, achieving QVGA resolution with a pixel pitch of $30\ \mu\text{m} \times 30\ \mu\text{m}$ and a fill factor of 30%. They implemented asynchronous readout through an AER channel. The pixels comprised 77 transistors, 3 capacitors, and 2 photodiodes, and power consumption of the entire sensor was 175 mW.

In this work, we implement the octopus sensor, which utilizes pulse-frequency modulation (PFM). Octopus pixels encode their absolute light intensity in the frequency of spikes emitted by each of them. Their implementation is similar to that of TTFS imagers, but with a self-reset mechanism in the voltage ramp, initiating a new integration cycle each time the voltage threshold is reached. The interspike interval is inversely proportional to the incident light’s photocurrent, so that a higher event rate corresponds to a brighter pixel. Octopus pixels can achieve spiking frequencies of dozens of kHz. However, the precision of frequency measurement is limited by the voltage offset of the comparator induced by mismatch, which can lead to FPN, as well as by kTC noise, by charge injection in the reset transistor, and by the switching delay of the comparator.

Curciello and Boahen introduced the first octopus sensor in 2003 [29]. They described a current-feedback event generator for monitoring the integrated voltage and generating spikes upon reaching the threshold. The architecture aimed to minimize power consumption while maximizing gain and bandwidth. Readout was asynchronous using the AER protocol with wired-ORs for arbitration. The matrix resolution was 80×60 , with a pitch of $32\ \mu\text{m} \times 30\ \mu\text{m}$, and a fill factor of 14%. They achieved a dynamic range of 120 dB and a maximum bandwidth of 40 MHz for the entire array, with a maximum update rate per pixel of 8300.

One notable feature of octopus sensors, as observed by the authors, is that brighter pixels receive more attention in asynchronous readout. Because the integration threshold is reached

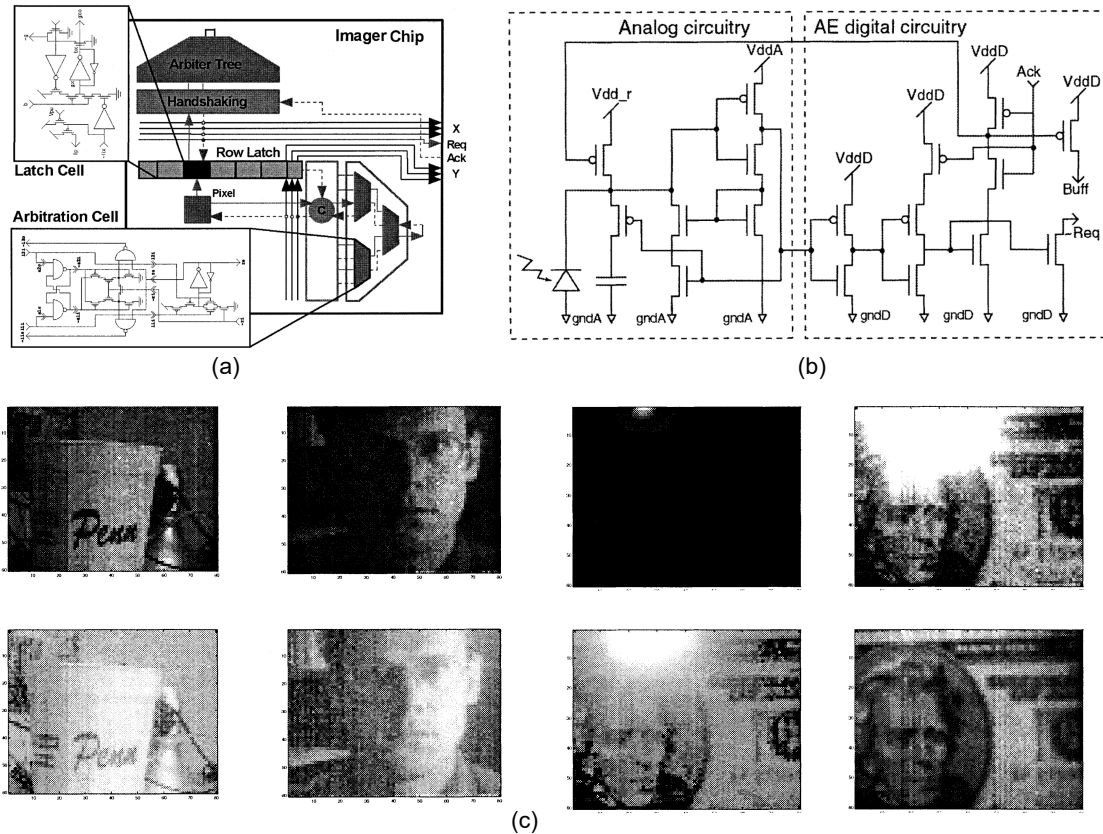


Figure 1.6: Schematics and images from the original octopus sensor [29]. (a) Asynchronous AER readout. A latch buffers the request from the pixel. The request signals are arbitrated in the arbiter trees, which select the first pixel that produced an event. The encoders output the address of that selected pixel. (b) Pixel schematic, showing the current-feedback event generator and the in-pixel handshake circuit. (c) Example images with linear intensity (top) and log scale (bottom).

faster in brighter pixels, they request bus access more frequently. This can lead to motion artifacts in the image, as a bunch of pixels hoards the whole bus bandwidth for themselves. Traditional synchronous readout schemes allocate an equal portion of the bandwidth to all pixels regardless of their activity, effectively preventing such congestion. The limited bandwidth of the arbiters in the periphery has hindered the application of octopus sensors to specific tasks, such as tracking small and intensely luminous light sources. In this work, we propose a simple solution using an asynchronous windowed readout scheme to alleviate congestion in the AER channel, reducing the bias toward brighter pixels in favor of darker ones.

Octopus sensors have demonstrated usefulness in tracking small and bright light sources [15]. Their excellent temporal resolution, low power consumption, and reduced data output make them suitable for space navigation, precisely gauging the sun’s position. [30]. These digital sun sensors have evolved into more compact and efficient solutions, incorporating TTFS operation [31] or utilizing photodiodes in the photovoltaic region for self-powering [32]. Octopus sensors have also been proposed for flame monitoring using NIR filters [33] and color detection [34]. Additionally, in the early 2000s, they were considered for retinal prosthesis applications [35].

While readout saturation and large FPN have limited their use in conventional imaging applications, efforts have been made to address these drawbacks. For example, authors in [36] incorporated

in-pixel spike counting and memory in 8-bit Gray code format for parallel counting and readout. Their synchronous architecture allowed them to capture snapshots of the scene during the integration period. Leñero integrated several hybrid pixels with pulse-frequency modulation and contrast computation [37], APS [38], and frame-based readout [39]. Recently, a method involving synchronous readout of spikes in octopus image sensors was proposed [40].

1.2.2 Address event representation (AER)

Event-based image sensors draw inspiration from the computational processes of biological systems, but some of these can not be translated into silicon in a feasible way. While our brains are composed of a highly-dense 3D network of neurons, axons, and synapses, CMOS technology is fundamentally 2D and cannot replicate such intricate wiring. In a neural network, a single neuron is typically interconnected with thousands of other components, whereas standard digital logic gates typically connect to only a few inputs. This physical disparity presents a constraint for artificial vision systems, where each pixel would require a dedicated wire to convey its data out of the array. However, designers overcome this limitation by capitalizing on the remarkable speed of transistors. Unlike the spiking activity of a neuron, which lasts for milliseconds, the switching delay of a transistor is on the order of picoseconds. Taking advantage of this discrepancy in timescales, communication in neuromorphic systems is often implemented using a time-multiplexed, packet-switched communication method called address event representation (AER).

A packet-switch network operates by time-multiplexing individual segments of the network, with packets being communicated requesting access to shared resources on the fly. Such protocols suit networks where two endpoints exchange bursts of small amounts of information, such as spikes in a neuromorphic system. AER provides the multiplexing and demultiplexing functionality for the spikes that are generated by or delivered to a cluster of neurons. Notice that AER is also used to create networks of neuromorphic processors that can send and receive spikes, but in the context of event-based image sensing there is no need to communicate spikes to the pixels. In our application, a standard digital processor acts as the AER receiver, processing spikes as they arrive. Practically all bioinspired vision sensors with spiking output reported in the literature use the AER protocol, or some modified version, to communicate their data.

Given that spikes are generated asynchronously, the AER sender circuits in an event-based sensor must accept spikes as they are generated, arbitrate between simultaneous spiking pixels, encode the address of the first firing pixel, and multiplex the address in the shared bus. Additionally, a routing topology within the pixel matrix is required to transmit the spikes to the AER circuits. Several schemes exist for each of these blocks, but we will focus on those implemented in octopus sensors described in the literature. For more detailed information on arbitration schemes, routing topologies, and AER receivers, we recommend referring to [41].

The AER sender of our octopus sensor is constructed as a 2D matrix of address-event sender elements, or pixels. The address event is represented by the row and column of the spiking pixel in a word-serial addressing scheme. Encoding and arbitration are done at the row level first and at the column level later. When both dimensions are finished, the pixel's row and column addresses are transmitted through a shared digital bus in a bundled-data 4-phase handshake asynchronous transmission, using a request signal (REQ), an acknowledge (ACK), and the data buses required to encode its location. Addresses transmit their implicit timing information of the spike, timestamped on the sensor itself or in the AER receiver. These blocks are represented in Fig. 1.7.

The AER circuits in both dimensions are almost identical. The only difference is that the pixel does not need to receive the acknowledge column token. The first stage in the AER readout is a sender with four functions: buffering the request and acknowledge signals coming from and going to the pixel matrix, implementing the pull-up transistor of the wired-NOR routing topology, and controlling the logic for the AER multiplexing and pixel reset. There is a sender for each array

in the dimension, which is equal to the square root of the number of rows or columns, \sqrt{L} . The second stage is a tree-shape arbitration logic to solve contention and queuing, formed by $\ln(\sqrt{L})$ number of stages. The third stage is the encoder, which is activated with the acknowledge signal. When both dimensions are arbitrated, the sensor sends a request signal with inverted polarity to the AER receiver, which answers with an acknowledge to let know the sensor that it has access to the data bus. After that, the action potential of the spiking pixel is reset.

The scheme we described is the standard dubbed AER 0.02, used for single-sender to single-receiver communications; that is, a point-to-point AER link. There are other schemes to implement the AER readout, with minor modifications in the arbitration block using ring arbitration [42], or radically different paradigms for the entire multiplexing sequence, such as the burst-mode operation [43]. The AER 0.02 standard was loosely defined, without specification of voltages, bus width, signal polarities, signal setup and hold times, or any kind of connector standard.

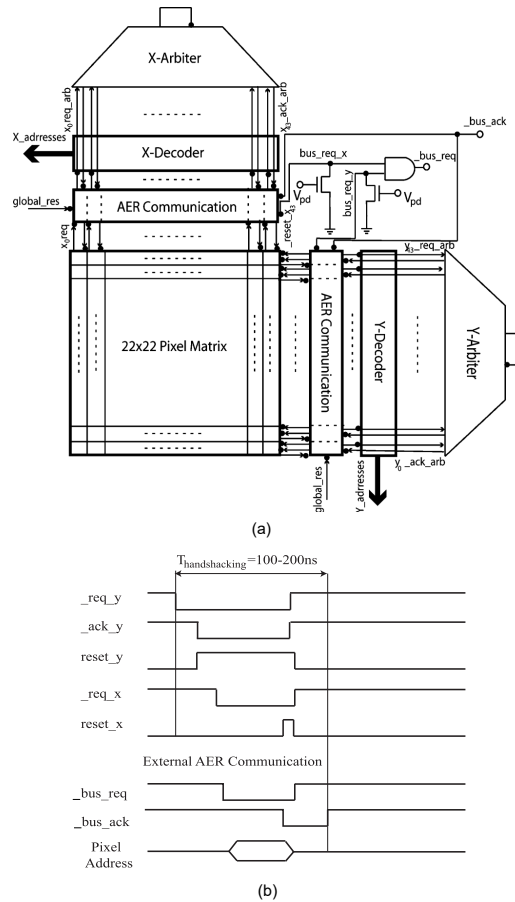


Figure 1.7: (a) Schematics of an octopus sensor as an AER sender implemented by Leñero [34]. The first stage provides buffering and signal control for requests and acknowledges pixel arbitration. Requests are sent to an arbiter tree, which selects the winning row or column through an acknowledge, which then selects a word from the encoder. (b) Signal flow of the arbitration between rows and columns, and the AER point-to-point link with a 4-phase handshaking protocol [38].

CHAPTER 2

Asynchronous Readout

This chapter describes the theoretical framework that is the basis for the research conducted in this work. Section 2.1 outlines the limitations of the asynchronous AER readout scheme in comparison to conventional synchronous readout methods. Section 2.2 delves into further detail regarding the various types of arbiters utilized in asynchronous systems, highlighting the reasons why fair arbiters fail to address the congestion issues in the readout channel. Finally, Section 2.3 introduces the technique proposed in this work, offering mathematical justification for its application and showcasing Matlab simulations to support its validity.

2.1 Tenderness towards synchronous readout techniques

A neuromorphic system should outperform conventional technology. Historically, most neuromorphic applications have failed to convincingly demonstrate that the bioinspired approach is better than simply scaling logic and exploiting parallelism. Emulating the data-driven computation and communication architecture used in brains may not necessarily exceed the capabilities of the digital clocked paradigm. Vaguely stating that metronomic schedules are less resource-efficient than an event-driven approach do not guarantee superior performance in practical usage and adherence to constraints.

Designers at industry leaders such as Sony and Samsung have likely contemplated this issue. As discussed in Chapter 1, the industry has recently shown interest in event-based sensors, particularly to the DVS architecture. Initial implementations of DVS employed asynchronous AER logic for collision handling and event transmission. While subsequent designs, like the DAVIS sensor developed by Delbruck, integrated additional functionality such as global shutter with active pixel sensing (APS), the event reading still relied on asynchronous logic. These designs achieved event rates of up to 50 Meps and a pixel size of $18.5\ \mu\text{m}$.

In 2017, Samsung introduced a slightly modified approach in a VGA format DVS employing Grouped-AER (G-AER) [44]. Their scheme involved grouping 8 neighboring pixels within a column and treating them as a single entity. The ON and OFF events of the pixels within a group were processed in parallel. The readout scheme combined asynchronous handshake arbitration between columns with burst-mode clocked readout of rows for groups containing at least one firing pixel. Using a digital logic synthesized with a 50 MHz clock, they reported an event rate of 300 Meps, a pixel pitch of $9\ \mu\text{m}$, and 50 mW of power consumption.

In 2018, the Swiss company iniVation implemented a low-resolution DVS with synchronous AER (SAER) [45]. They introduced an innovative and intricate readout scheme where the pixel matrix was controlled synchronously by external circuitry. The scheme utilized two distinct pulses:

one to capture an event frame into each pixel’s internal event memory and another to reset the pixel. Pixels were organized into 2x2 groups that shared a digital logic responsible for reading their internal memory and determining whether to transmit their events. In this way, they performed pre-readout pixel-parallel suppression of noise and spatial redundancy. The readout process occurred sequentially in both dimensions, following a token-ring scheme where columns or rows that had spiked were read while those that did not were skipped. They reported an event rate of 180 Meps, a pixel pitch of 10 μm , and 4.9 mW of power consumption.

In 2020, Samsung made advancements in their DVS technology and scaled it up to a resolution of 1280x960 pixels [26]. The researchers acknowledged that higher resolution led to motion artifacts caused by timing errors within the limited bandwidth of event-driven readout. To address this, they took two approaches. First, they completely removed all asynchronous readout logic and replaced it with sequential column readout. The authors criticized the ”unpredictable process timing inaccuracy” of the arbiter tree. Second, they introduced a global event-holding function using in-pixel storage cells, similar to the global shutter in CIS. This modification resulted in an event rate of 1.3 Geps, a pixel pitch of 4.95 μm , and 150 mW of power consumption.

Also in 2020, Sony collaborated with Prophesee to develop a 1280x720 DVS sensor that retained asynchronous arbitration in one dimension [22]. Rows were connected via a low-latency interface to an asynchronous selection tree. Events from the active row were promptly timestamped using an auxiliary time-out column located at the end of the pixel matrix. Subsequently, the entire row was scanned through an asynchronous-to-synchronous interface. The selection operation for the active row was pipelined, enabling parallel processing of previous data while new row arbitration took place. Additionally, their architecture was filled with on-chip digital processing of events. The reported specifications for this sensor were an event rate of 1.066 Geps, a pixel pitch of 4.86 μm , and 32 mW of power consumption.

In 2023, the ISSCC featured three papers describing modified DVS sensors with hybrid pixels in a stacked configuration, featuring significant architectural innovations. OmniVision presented a 1032x928 DVS matrix with a readout scheme similar to Sony’s 2020 implementation [18]. However, the row-selection tree in OmniVision’s design operated synchronously and was governed by a 250 MHz clock. The selected row was scanned entirely, and skip logic was implemented in both dimensions to bypass rows or columns that did not require reading. This sensor achieved an event rate of 4.6 Geps, a pixel pitch of 8.8 μm , and 46 mW of power consumption. The other two papers were designs by Sony [19, 20]. Both sensors followed the industry’s trend of abandoning asynchronous readout in any form. In both cases, the output of the logarithmic amplifier was stored in a sample-and-hold circuit, which was then processed by a comparator using a three-phase logic. Events were read per row through a scan access mechanism, with a skip logic to ensure that only rows with at least one event were read.

The position of these industrial giants towards asynchronous readout techniques can be summarized by statements found in Sony’s latest published work [19]. According to them, ”the unpredictable arbitration time [of asynchronous readout] degrades time accuracy and affects system-level features such as recognition accuracy in high-activity scenes. There have been several efforts to increase event throughput with innovative circuit techniques, but the additional processing increases the post-processing cost to reconstruct a frame from asynchronous event data. Moreover, conventional image acquisition with very low noise is still helpful for the complex tasks combined with event-based vision sensing”. As these companies introduce the DVS into the market, they are moving away from circuits implementing the AER scheme. However, a thorough comparison between asynchronous and synchronous readout schemes is still lacking in the literature.

The arbitration circuits in asynchronous readout are often considered the main bottleneck in these schemes, which have not been able to exceed rates of 50 Meps. At high event rates, arbiter trees tend to saturate, giving more priority to certain regions in the matrix and causing unreliable timestamps for events. It is worth noting that most of the asynchronous readout AER logic

Table 2.1: Comparison Between Asynchronous and Synchronous Readout Techniques in Dynamic Vision Sensors in Works Reported in the Literature

	Asynchronous	Synchronous
Pros		
	Low power consumption and readout latency at small event rates	Maximum readout speed with constant latency
	Suitable for low resolution sensors	Solution for larger formats
	Per pixel timestamp	Frame-like behavior is appropriate for machine vision applications
Cons		
	Limited readout speed due to arbitration delays	Considerable static power and latency at low rates
	Non-deterministic behavior	Temporal resolution is solely dependant on frame rate

used today is based on the arbiters proposed by Boahen in 2001 and 2004, which have not been widely challenged by other researchers. Even Prophesee’s 2020 design [22] uses a fair arbiter tree. Additionally, the 2D routing arrangement in rows and columns presents a significant limitation, as highlighted by Gómez-Merchan [46].

A new approach has recently been proposed by Boahen [47], introducing a tree-shaped alternative where arbiters are integrated inside the pixel matrix rather than in the periphery. This novel approach aims to address the limitations of conventional 2D routing arrangements. Although these topics are beyond the scope of this work, it is valuable to explore the circuit implementations of asynchronous arbiters. Understanding the circuit-level details can shed light on the challenges and potential solutions related to asynchronous readout schemes.

2.2 Arbiters

Arbiters play a crucial role in asynchronous circuit design, alongside other components such as latches, Muller C-elements, MUTEX, and MUX components [48]. Handshake links and data tokens serve as abstractions that are equivalent to the register transfer level (RTL) used in synchronous circuit design. Arbiters are relatively expensive in terms of speed. However, arbiters are relatively expensive in terms of speed, and the loss of determinism in circuits with arbitration can introduce challenges in testing and verifying design correctness. Consequently, the neuromorphic community has explored various approaches to collision prevention. Hafliker identified three main schemes: full arbitration, where collisions are resolved in the AER sender; discarding, where collisions are resolved in the receiver; and aging, which involves discarding old pulses. This work focuses on the first scheme, which is the only that guarantees that the 4-phase asynchronous handshake between the AER sender (the octopus sensor) and the AER receiver (an external processor) seen in Fig. 2.2 communicates unaltered addresses.

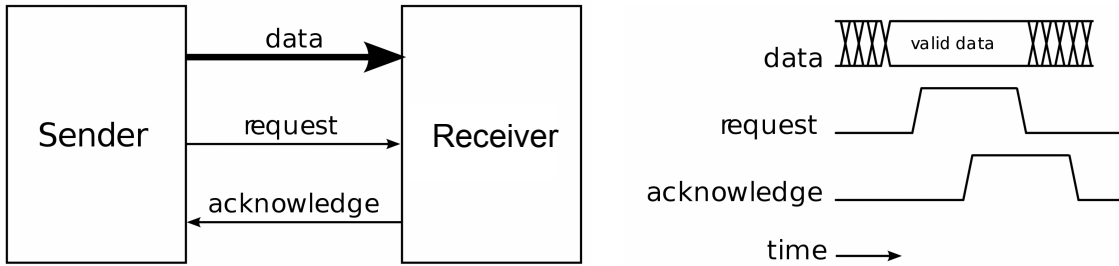


Figure 2.1: (Left) A 4-phase, bundled-data communication. A sender is connected to a receiver by data lines, a request line, and an acknowledge line. (Right) Its timing diagram. When the request line is low, the data is to be considered invalid and liable to change at any time. The sender usually waits for the acknowledge signal to remove the data from the data bus.

Why is arbitration between spikes so important? The time at which the pixel address is generated corresponds to the time at which the pixel produced a spike, plus a small delay due to the encoding process. As long as spikes are sufficiently separated in time, the encoding process ensures that the addresses are correctly ordered. If each pixel in the 2D matrix were guaranteed to only produce a spike when no other pixel in the same matrix was spiking, then the multiplexing circuits would correspond to a standard asynchronous encoder circuit. However, this is not a valid constraint as groups of pixels could have overlapping firing times. AER encoders therefore generally use arbitration logic to handle potentially simultaneous spike arrival times from multiple pixels.

Employing arbitered access to the shared bus prevents collisions, thereby increasing throughput. An arbiter grants only one request at a time and outputs the address of the granted pixel through the encoder. The arbiter receives row or column request lines from the pixel array via a routing interface and performs arbitration among the active lines. The arbiter manages the handshaking signals from the row (or column) requests and provides acknowledgments. For each dimension of row and column (assuming N rows or columns), the arbiter consists of a tree-shaped structure of $N - 1$ two-input arbiter cells and $\ln(N)$ tree levels required to arbitrate between N pixels. Each two-input arbiter gate outputs a request to the two-input arbiter at the next level if any one of its two input request signals is active. These circuits are located in the periphery of the matrix and are usually routed to the pixels with a grid arrangement as seen in Fig. ??.

The bandwidth of communication links in image sensors is a crucial specification, and asynchronous readouts in event-based sensors have struggled to surpass rates of 50 Meps. Boahen introduced five criteria to evaluate the performance of event-driven communication links: capacity, throughput, latency, integrity, and dispersion [14]. Capacity represents the maximum event transmission rate, while throughput reflects the sustainable event rate in practical conditions. Latency measures the average transmission delay, integrity indicates the fraction of correctly delivered spikes, and dispersion quantifies the standard deviation of the latency distribution. These parameters are influenced by various factors, including the design of the arbiter block, routing scheme, encoders, and the external AER receiver.

Arbiter circuits are primarily designed to meet two main criteria: low latencies and fairness. The former ensures that incoming requests are serviced promptly, while the latter guarantees that requests are serviced in the order they are received, regardless of their position on the arbiter tree. In the following discussion, we explore two different arbiter designs that illustrate the trade-offs implemented in the circuits as they have evolved over time.

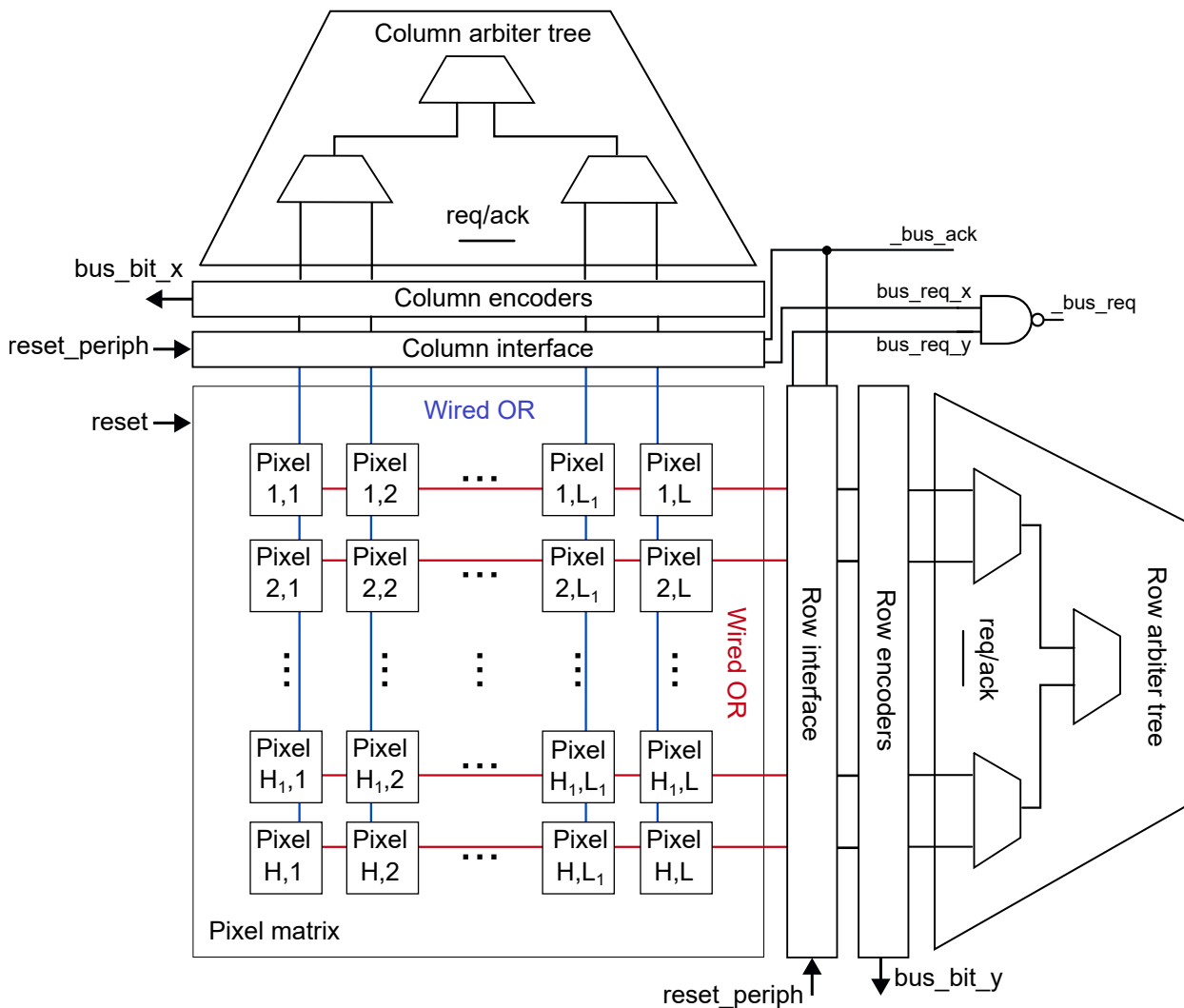


Figure 2.2: Block diagram of an asynchronous readout AER implementation for an event-based image sensor with a resolution of $L \times H$ pixels in a grid arrangement. The row and column interface can include pull-up elements for the wired ORs, buffers, and latches, as well as control logic for handshake signals. The main limitation we face when scaling pixel count is the size of the wired ORs. These column and row metal lines get longer and have more pull-down transistors connected to them, causing the capacitance of the lines to increase, degrading signal integrity and slowing the readout process. In the arbiter trees, each line is actually formed by two wires: a request and an acknowledge. There are $\ln(L)$ (or $\ln(H)$) levels and $L - 1$ (or $H - 1$) arbiters in each tree, in the case of square matrix dimensions.

2.2.1 Greedy arbiter

All arbiters are 2-channel, meaning that they can attend to two input requests simultaneously. They adhere to a “first come first served” functionality. However, the tree topology might introduce an unfair mechanism, as subtrees do not withdraw their request until all their leaves are serviced, causing the pixels closer to the previous winner to gain an advantage over more distant pixels, regardless of which pixel requested first. Consequently, a row with high activity tends to hold onto

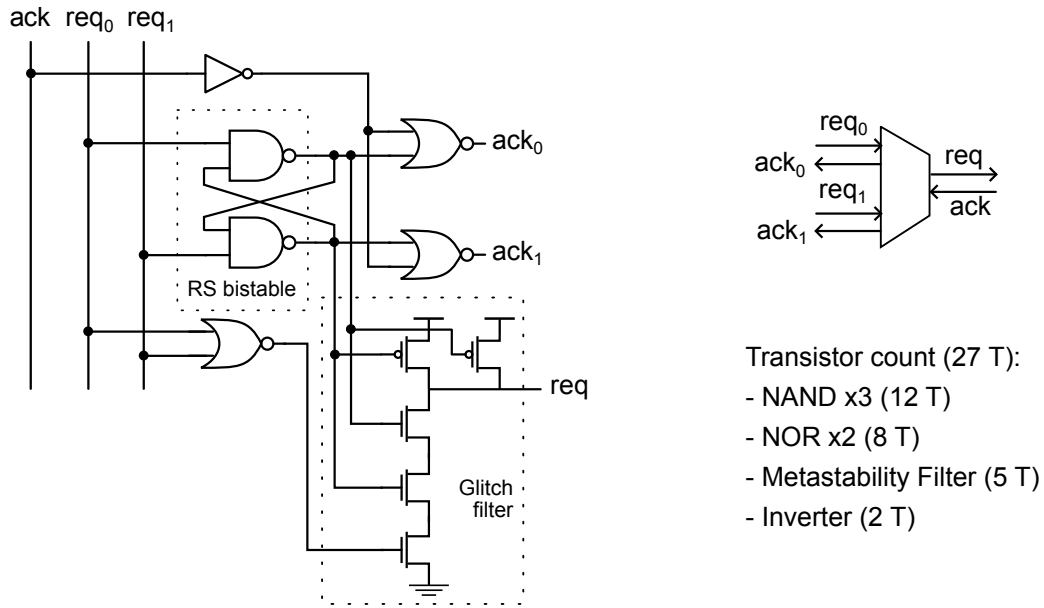


Figure 2.3: Circuit implementation of a glitch free 2-way greedy arbiter [49]. The RS bistable ($req_0 = S$, $req_1 = R$) and metastability filter form the core mutual exclusion element. It exclusively acknowledges one of two incoming requests, only if its own outgoing request is acknowledged. The glitch filter prevents a condition where the output request switches logic briefly. The transistor count is 27.

the arbitration bus leading to the transmission of events from only a part of the chip. Depending on the fairness of the architecture, we differentiate between greedy arbiters and fair arbiters.

In the octopus sensor, we have implemented greedy arbiters, which are simpler and employ fewer transistors, thus minimizing area. Although several circuit implementations of fair arbiters exist, we opted to adopt the architecture proposed by Haflinger [49]. This particular architecture, seen in Fig. 2.3, incorporates a mutual exclusion (MUTEX or ME) element, a common component found in all asynchronous arbiter architectures, whether they are greedy or fair. Typically, a MUTEX is implemented using a bistable element and a metastability filter.

The RS latch used in the arbiter cell operates with active low inputs. In its idle state, both outputs of the latch are set high. When a child sends a request, depending on which request signal arrives first, one of the outputs switches and propagates the request through the glitch filter to the next level of the tree. If this request is granted by an incoming acknowledge, then its corresponding acknowledge is propagated back. This arbiter cell does not need to wait for requests from its parents to be cleared before resetting its acknowledge signals. Instead, the cell clears its acknowledge as soon as its daughter's request clears. Only when there are no more requests from lower levels (req_0 and req_1) are req and ack withdrawn, allowing other arbiter cells on the same level to be acknowledged. However, if there is a simultaneous request; that is, req_0 and req_1 are both active during handshaking, a greedy path is formed. In this path, the token is granted to that specific request without propagation to the top of the tree, thus ignoring other requests in the tree that might have occurred earlier. This situation is illustrated in Fig. 2.4.

In addition to the issue of greediness, this circuit also presents a potential problem of glitches at the output request when both req_0 and req_1 are high, and one of them transitions to low after the acknowledge signal goes high. When req_0 (or req_1) is withdrawn, the RS latch transitions from (0,1) to (1,0), passing through a transient state of (1,1). During this state, req briefly switches to low. To prevent this situation, the circuit includes a glitch filter with an additional input that

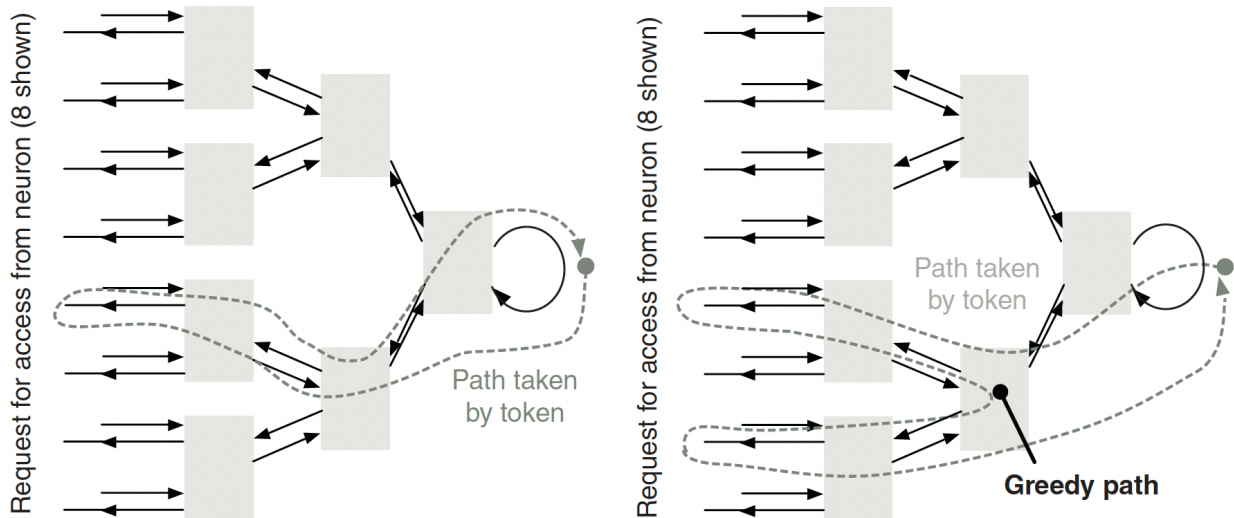


Figure 2.4: Token movement in an arbiter tree to handle the requests from neurons. In a greedy arbiter tree, neurons located near each other may form a greedy path because the token does not need to reach to the top of the tree. A fair mechanism is needed to ensure that requests from an arbiter are blocked until its parent’s acknowledge is cleared.

leaves the output floating while the state lasts. It’s important to note that this circuit does not implement a metastability filter.

2.2.2 Fair arbiter

Boahen introduced both the original greedy arbiter [14] and the fair arbiter circuit [43], and their schematics and detailed behavior can be found in [41] for further reference. In a recent development, Fok and Boahen presented a new fair arbiter implementation [47]. An asymmetric C-element ensures fairness by blocking new requests until its parent’s acknowledge signal clears, which occurs immediately after the cell clears its own request. Therefore, requests propagate upwards in the tree until they reach a cell where the other request is currently selected. This implementation addresses the issue of fairness in the arbiter circuit design.

The 2-way fair arbiter employs a mutual exclusion element to determine which child’s request is served first. The latch component ensures the selection of one request, while the metastability filter prevents any metastable signals before toggling the outputs. The two NOR gates play a crucial role in preventing overlapping handshakes, ensuring that a new request from the arbiter’s children does not receive unfair attention. These components work together to maintain fairness in the arbitration process.

In [50], a comparison between different implementations of 2-way fair arbiters can be found. Regardless of fairness, arbiters can introduce motion artifacts due to their limited bandwidth, unpredictable nature, and latency. These factors can lead to errors in timestamp accuracy [26]. As shown in Fig. 2.7, arbitration is the preferred choice for event-based applications with sparse spatial and temporal activity. While temporal dispersion decreases with advancements in technology and faster logic, the collision probability remains the same. Current arbitration circuits have drawbacks such as occupying area and introducing additional time constraints, which impact the pixel count and the achievable event rate readout in event-based imagers.

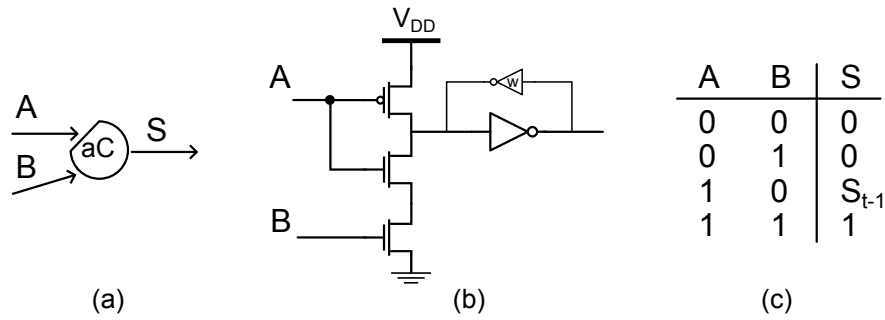


Figure 2.5: Asymmetric Muller C-element, an essential component for understanding the fair arbiter behavior. (a) Symbol of the circuit. (b) Transistor implementation. (c) Truth table.

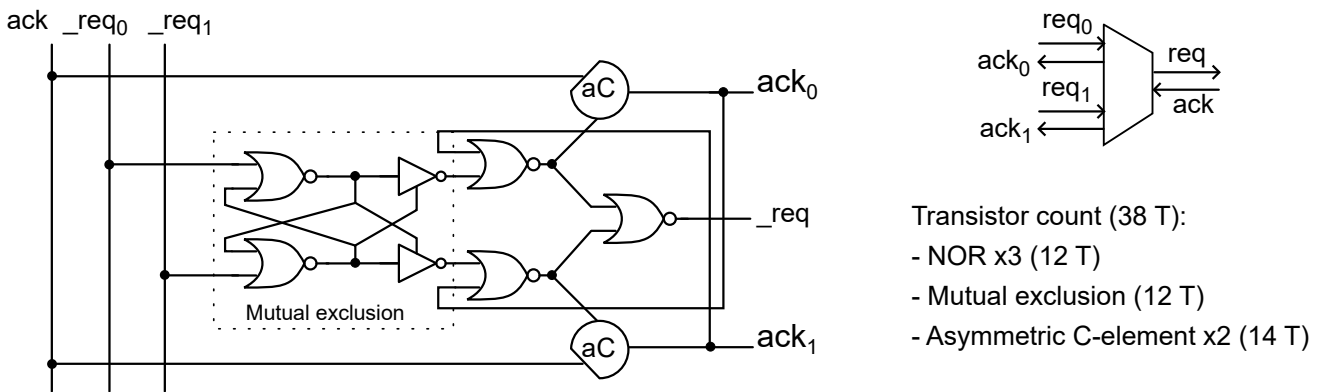


Figure 2.6: Asynchronous fair arbiter implemented by Fok and Boahen [47]. The RS bistable with NOR gates implies the use of active-low logic for requests. A metastability filter is implemented with two inverters at the output of the latch, with pull-down source connected to the reciprocal request. The asymmetric C-element guarantees that no greedy path is created. Transistor count is higher than for the greedy arbiter.

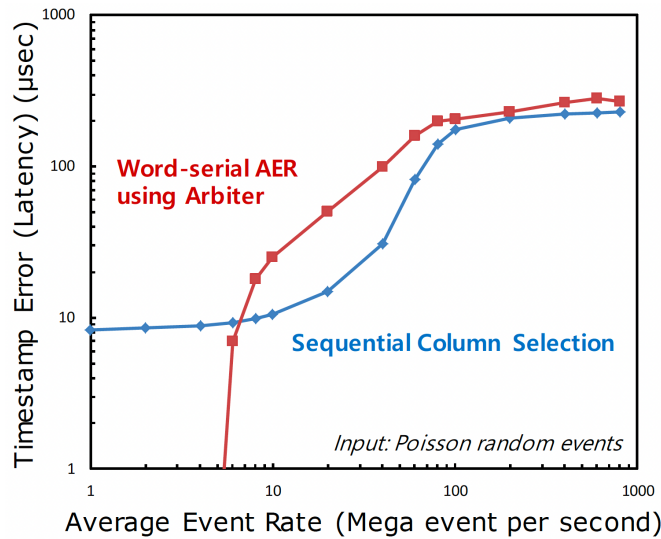


Figure 2.7: Image artifact occurs by the mismatch between event generation time and readout time for high event rates [51].

2.3 Our solution: windowed asynchronous readout

This work introduces a novel approach to decouple event transmission from readout. We achieve it by inserting a switch immediately after the comparator in a octopus pixel. In Chapter 3, we will delve into the details of the pixel’s design and this switch, which is activated by an external signal and acts as a time window to selectively allow or dismiss spikes for communication. The primary objective of this technique is to mitigate congestion in the AER channel, particularly in high-illuminance scenarios where there is a bias towards brighter pixels. The application of this approach is specifically investigated for an octopus sensor that performs light-to-frequency conversion, as described in the preceding chapter.

We also expect to study whether this technique can address the issue of scalability in all kind of event-based sensors. As resolution increases, the event rate goes up and the time between events is reduced. Authors in [52] demonstrated that there are some major drawbacks in using high-resolution DVS to solve standard computer vision tasks, and that low-resolution cameras can outperform high-resolution ones in low-illumination and high-speed conditions. Nevertheless, DVS development will inexorably trend toward higher resolution sensors (HD, Full HD or even 4K) required in certain applications, so the question of scalability that also motivates this work remains relevant.

The outcomes of this research might contribute to further develop the work conducted by Karen Adam at EPFL, in her recently published thesis titled “Timing is Everything” [53]. According to her own words: “Spikes provide a power-efficient way to encode information and can provide better sample efficiency in comparison to clocked and synchronous sampling schemes”. Through her examination of time encoding machines (TEM), which are mathematical models of integrate-and-fire neurons, Adam provided compelling arguments in favor of asynchronous readout in imagers. These assertions can be experimentally validated using our windowed readout technique. Specifically, the statements put forth by Adam include: (1) timing-based spiking devices offer advantages over traditional uniform sampling in multi-channel encoding by eliminating the need to align clocks; (2) the asynchrony of spikes across different spiking devices leads to less redundant and more efficient data, fostering collaborative operations among pixels; and (3) asynchronous readout enables the entanglement of temporal and spatial resolution in video, allowing for an increase in both aspects by expanding the number of pixels.

Our technique primarily addresses the second point mentioned above—the concept of group work between pixels. The principle of group work suggests that pixels that exhibit low spike rates can be compensated for by other pixels that spike more frequently, but only up to a certain threshold. Once a pixel effectively characterizes its illuminance using, for instance, 15 spikes, generating additional spikes (20, 30, or 40) becomes redundant. Therefore, beyond that threshold, it is more advantageous to have spatially sparse information, meaning that it is preferable to have more pixels firing, even if each pixel fires less frequently. By decoupling event generation from readout and leveraging the in-pixel aging storage mechanism, we anticipate being able to capture spike data from a larger number of pixels in high-illuminance scenarios, regardless of any spikes that may be missed from more active pixels.

The inclusion of window logic in our approach also provides an opportunity to investigate another hypothesis regarding the behavior of the pixels. Because the pixels act independently, the readout queue of spikes occur according to a Poisson process [29], in which the probability that a certain pixel’s address is communicated is proportional to the light intensity of its area. This is an interesting property, that Culurciello highlighted in the original paper, where it was noted as the “the first reported example of a probabilistic APS, where the output activity reflects the statistics of the scene”.

We aim to test the hypothesis that by employing observation windows to capture spikes, we can effectively preserve the most relevant aspects of the image by leveraging the Poisson distribution

of spikes' output. This technique has been previously explored within our research group, with Méndez-Romero investigating it in his Master's thesis [54] and subsequently publishing a related study in the 2022 edition of ISCAS [55]. The concept draws inspiration from the quanta image sensor (QIS), a novel sensor design introduced by Fossum in 2005 [56]. A QIS generates images by aggregating binary bit planes, where each bit represents the presence or absence of at least one photoelectron in a photoreceptor. While QIS relies on single photon detection and justifies its usage of Poisson arrival statistics based on the random nature of photon arrivals, we seek to experimentally verify whether we can replicate a similar readout scheme in the laboratory by constructing binary bit planes using the spikes from our octopus pixels.

The proposed reading schemes are illustrated in Fig. 2.8. The sparse readout mode aims to provide channel access to pixels with low spike frequencies. While fair arbiters could potentially achieve a similar effect, we believe that this readout scheme is better suited for octopus sensors as it is expected to minimize motion artifacts. Additionally, we plan to experimentally validate the quanta-based readout mode, which extends beyond the previous Matlab simulations conducted by Méndez-Romero. The promising results shown in Fig. 2.9 and Fig. 2.10 motivate further investigation into its practical applications. A summary of the timing constraints for the sensor's internal signals can be found Table 2.2.

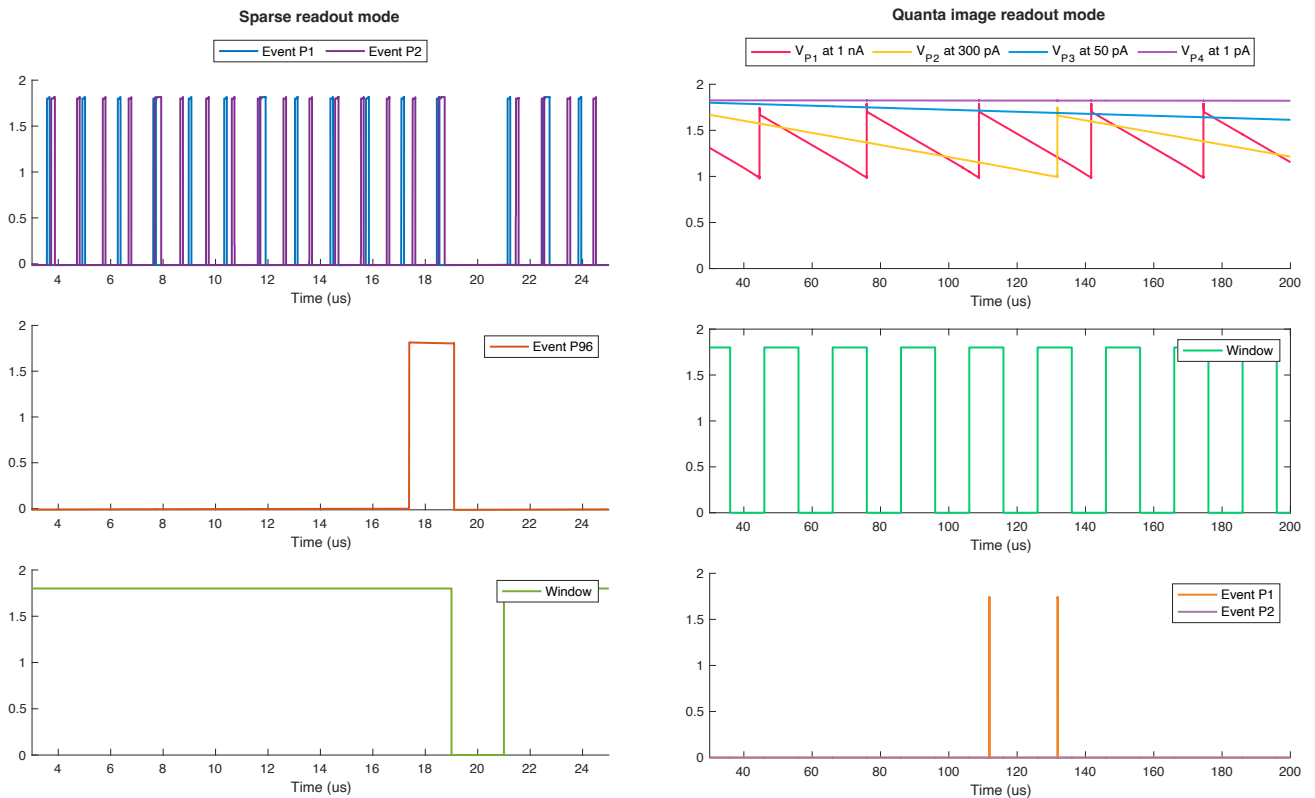


Figure 2.8: The two readout modes proposed in this work. The octopus sensor we designed implements a switch to decouple event generation from transmission through a signal called Window. (Left) The sparse readout mode is based on the works of Karen Adam [53], and will enable us to scan the events of those pixels in the matrix that are not receiving the attention of the periphery. (Right) The quanta image mode, based on the sensor conceived by Fossum [55], will use the window to randomly sample the Poisson process of the spikes produced in the matrix.

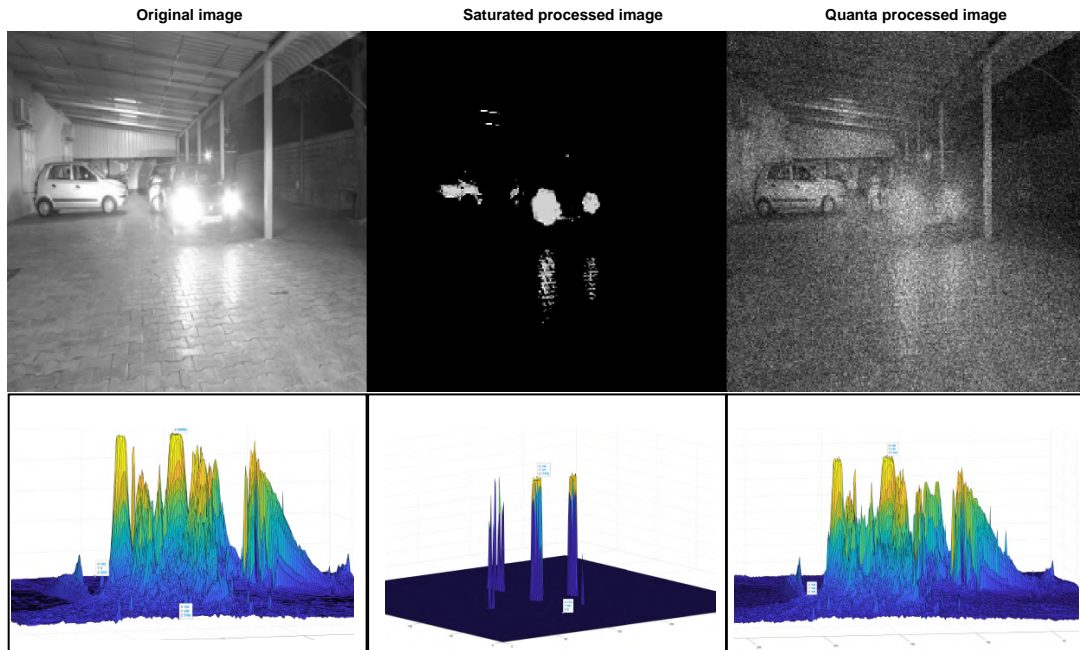


Figure 2.9: Results from the Matlab models developed by Méndez-Romero [54]. In the top, an example bright image is used to evaluate the saturation in the AER channel and the impact of quanta-based acquisition. In the bottom, a 3D representation of the spikes in each pixel. In the original image, the most illuminated pixels have 55000 spikes, whereas its darker parts have around 2000. Roberto used a linear transformation that assigned spiking frequencies between the brightest and to the darkest pixel. Then, he modeled saturation by comparing the sum of all spikes in the matrix with the common saturation rate of the AER periphery. If the sum is bigger, he erased spikes of the less illuminated pixels. Last, he modeled quanta acquisition by dividing the original image in a cube of binary bit planes, each indicating whether the pixels had spiked at least once, and randomly eliminated spikes of each plane. The number of spikes is better distributed among all pixels.

Table 2.2: Temporal Characteristics of our Octopus Sensor

Spike duration	Pixel spike frequency range	Spike aging	Arbiter delay
15 ns	155.92 Hz - 59.19 kHz	334 μ s	6.63 ns

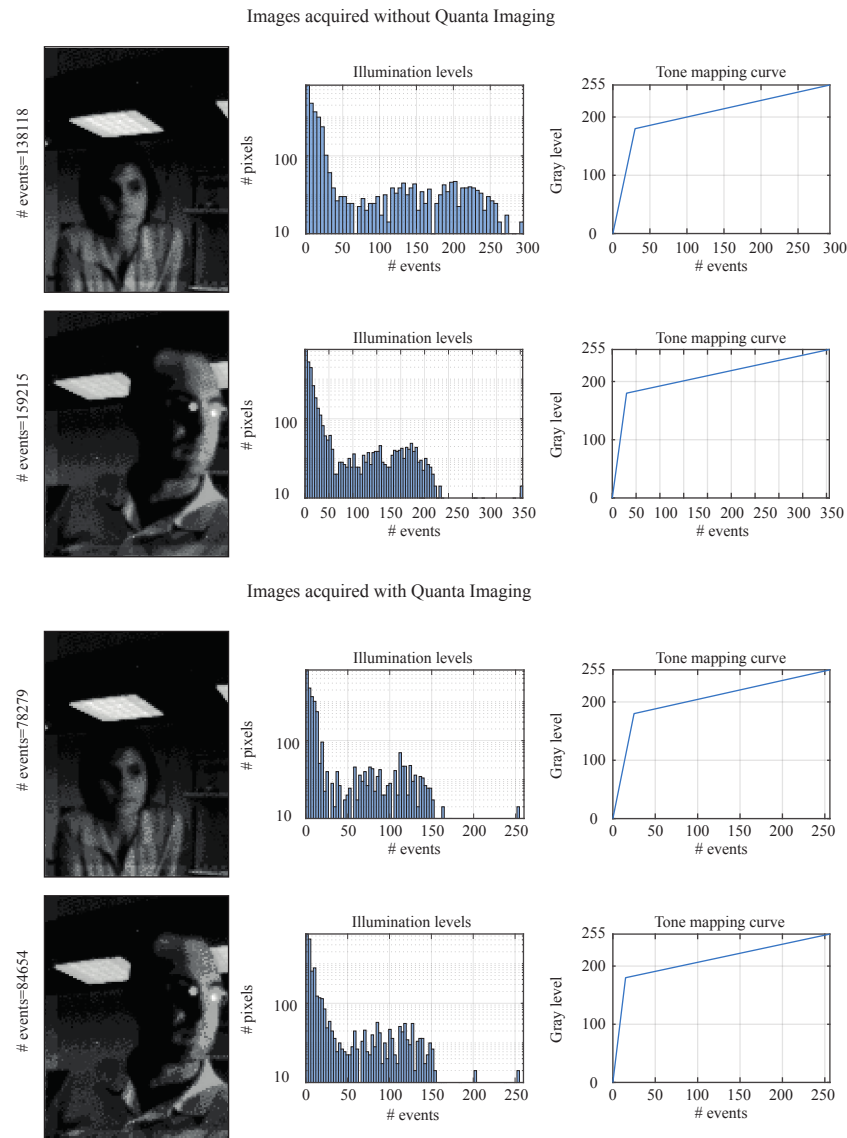


Figure 2.10: Results from a quanta-based post-processing using an image acquired by an octopus sensor [55]. The original images were acquired using an octopus sensor, and are formed by around 150.000 events each. After post-processing, using a window with a uniform distribution and 0.7 average duty cycle to sample the spikes, the images are still recognizable. The result is that each image can be represented with half the events. For each one of the images, histograms and the applied tone mapping curves are shown.

CHAPTER 3

Octopus Pixel

This chapter describes the octopus pixel designed and fabricated in a standard 180 nm CMOS process. Section 3.1 introduces the pixel architecture. After providing a solid theoretical framework for the acquisition technique with decoupling logic, the sensor is design and send for fabrication. Section 3.2 provides insight into the most sensible component within the pixel, the comparator, and its design process. Section 3.3 deals with the in-pixel handshake interface for pixel row and column arbitration, and event transmission. Finally, Section 3.4 details the resulting layout and post-layout verifications.

3.1 Introduction

The pixel circuit is portrayed in Fig. 3.1. Its architecture is not novel, as both the light-to-frequency block and the in-pixel handshake interface have been covered in the literature [30, 57]. But we introduced a modification for decoupling both blocks with an external signal that acts as a time window for data transmission.

The main circuit in the light-to-frequency conversion block is the comparator. It functions as a 1-bit ADC, producing a voltage transition at its output when the input analog signal V_P crosses the reference V_{bot} . This transition triggers a signal to the gate of a PMOS transistor, which resets the input voltage V_P and returns the output to its idle state. We refer to this pulse as a spike or an event.

The spiking frequency f is affected by the capacitance at P, which acts as a sensing capacitor with a value given by the parasitic capacitance in the photodiode C_{ph} . Following a first-order approximation, C_{ph} is discharged at a rate given by the photocurrent I_{ph} following

$$V_P = V_{\text{bot}} \approx V_{\text{DD}} - \frac{I_{\text{ph}}}{C_{\text{ph}}} \cdot \Delta t \quad (3.1)$$

where Δt is the time it takes for V_P to cross with V_{bot} . We can approximate the spiking frequency as

$$f \approx \frac{I_{\text{ph}}}{C_{\text{ph}}(V_{\text{DD}} - V_{\text{bot}})} \quad (3.2)$$

where the parasitic capacitances of the comparator's input and the dark current have been neglected. In this way, the spiking frequency at the comparator's output encodes the pixel's exposure, assuming that the intensity of the light during the integration process is constant. The comparator continuously changes between the two states in this architecture, behaving like an astable multivibrator.

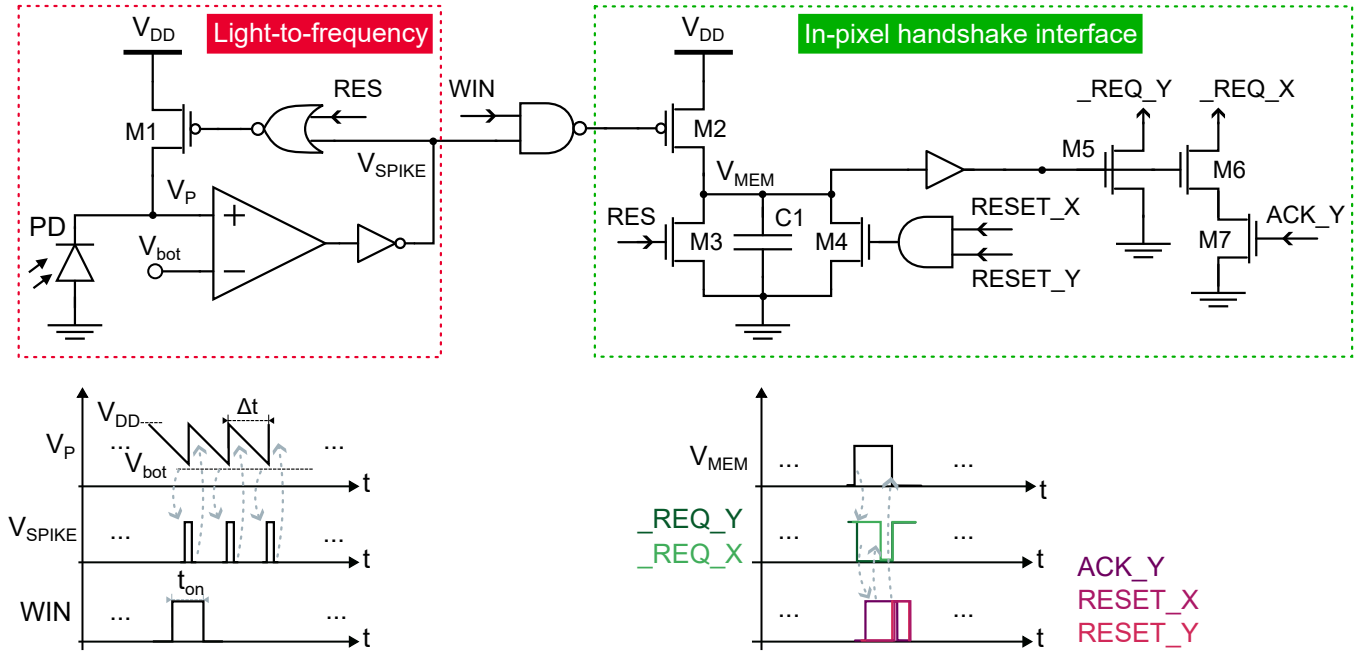


Figure 3.1: Pixel schematic and signal flow [30]. The incident light is encoded in the frequency of V_{SPIKE} . Spikes produce a self-reset in the comparator's input. The circuit implements an integrate-and-fire neuron model. The in-pixel handshake interface with memory capacity implements the logic for arbitration between pixels and AER communication. The window decouples spike generation from communication. Transistor sizes ($\mu\text{m}/\mu\text{m}$): $M1 = 0.24/0.34$, $M2 = 0.45/0.18$, $M3 = M4 = 0.24/0.18$, $M5 = 2/0.18$, $M6 = M7 = 1.6/0.18$. Total transistor count: 37. $C = 20$ fF. Discharging time of the capacitance due to current leakage: 336.77 μs .

The rest of the circuit is digital. As seen in Fig. 3.1, the spike is processed by a NOR gate along with the global reset, to avoid having two PMOS hanging at P which would increase current leaks. M1 is an I/O transistor (3.3 V). Also, the NOR and the inverter act as a buffer stage, delaying the reset signal for a few nanoseconds and thus improving the stability of the comparator. The window logic is implemented with a NAND gate, acting as a shut-off valve governed by an external signal that disables spike transmission (WIN).

The spikes are transmitted through an asynchronous handshake logic interface, which was presented in [57]. The pixels in the sensor are arranged in a grid, with each sharing two wired NORs for sending a request signal through its row and column. C1 stores the event until it gets the attention of the periphery in a short period, or the event gets discarded. Rows are arbitrated first ($_REQ_Y$). The periphery will answer an acknowledge (ACK_Y) to the row that first transmitted the request, which triggers column arbitration ($_REQ_X$). Then, the sensor will transmit the pixel address (row and column) and wait for an acknowledge signal from the exterior. When received, reset signals are sent to the pixel (RESET_X and RESET_Y). M3 and M4 have minimal dimensions.

More details about the behaviour of this logic are provided in Section 3.3 and Chapter 4. Now that we have covered the basics of the architecture, let's dive in and describe the performance of the comparator.

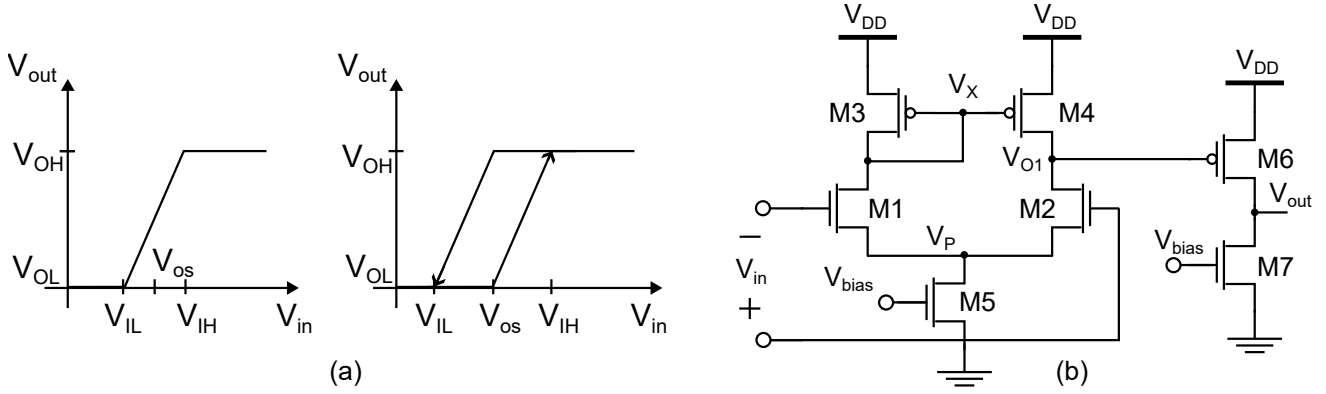


Figure 3.2: (a) Transfer characteristic of a comparator considering finite static gain and offset voltage (left) and hysteresis (right). (b) We use the two-stage comparator with NMOS input. Transistor sizes ($\mu\text{m}/\mu\text{m}$): $M1 = M2 = 0.67/1.1$, $M3 = M4 = 1.34/2.2$, $M5 = M7 = 0.37/0.48$, $M6 = 0.3/0.6$.

3.2 Two-stage comparator

Comparators are widely used in ADCs. They can be thought of as decision-making circuits that sample two analog signals and detect whether one is larger or smaller than the other, codifying the outcome as a digital signal. The input signals may well be differential voltages or currents, but their output is usually a voltage.

Voltage comparators are basically voltage gain devices. Therefore, its mechanism to achieve voltage gain is multiplying a small-signal transconductance g_m by a small-signal resistance r_o . Depending on their architecture, we can broadly categorize comparators as open-loop or regenerative [58]. Open-loop comparators are used without compensation. A simple implementation uses an operational amplifier, letting its high gain produce comparator operation between two saturation levels, along with other stages. Regenerative comparators use positive feedback to build faster unstable operation with intrinsic hysteresis, and have been extensively used for implementing bistables. For more information about comparator operation and architectures, we suggest the interested reader to [59].

The DC transfer curve in Fig. 3.2(a) illustrates the behaviour of a comparator. It shows the effects of two non-idealities: the finite static gain around the input transition point A_v and the input offset voltage V_{OS} . The static gain can be expressed as

$$A_v = \frac{V_{OH} - V_{OL}}{V_{IH} - V_{IL}} \quad (3.3)$$

where $V_{IH} - V_{IL}$ is the static resolution of the comparator. The minimum and maximum values are given by

$$V_{IH} = V_{OS} + \frac{V_{OH} + V_{OL}}{2A_v} \quad (3.4)$$

and

$$V_{IL} = V_{OS} - \frac{V_{OH} + V_{OL}}{2A_v} \quad (3.5)$$

For any input level inside this range, the digital output state is uncertain.

A relevant feature in comparators, particularly useful in noisy environments, is hysteresis. This quality is the variation of the input threshold as a function of the output level due to the circuit's

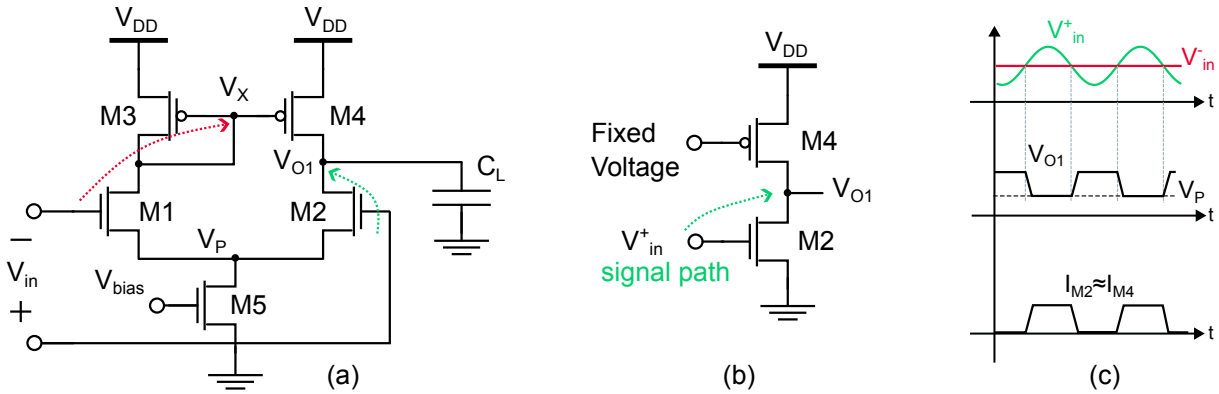


Figure 3.3: (a) Large-signal analysis of the first stage of the comparator, with the two signal paths affecting the single-ended output. Note that V_{in} polarity respect to a 5-T OTA depiction is reversed, as the second stage of the comparator inverts the output. (b) The V_{in}^+ signal path can be studied as a NMOS common source amplifier. (c) Influence of V_{in} in V_{O1} and in the current of the right branch.

inertia. The DC transfer curve considering hysteresis is also seen in Fig. 3.2(a). Depending on the comparators topology, the static gain in each transition—that is, the slope in the transfer curve—may differ.

In our design, we opted for the simple open-loop, two-stage comparator in Fig. 3.2(b). This topology is a two-stage Miller OP-AMP without compensation that uses a NMOS input pair. Because reducing the current consumption is a major concern, we biased the circuit to keep all transistors operating at weak inversion. Both design choices fit perfectly well for reducing the number of transistors in the pixel. Unfortunately, these also have a negative impact on speed. The next few sections are dedicated to analyzing the two stages of the comparator and its design process.

3.2.1 Differential pair with active load

The differential pair has a current mirror as an active load, implementing a simple 5-T OTA which serves as a single-ended gain stage. This topology has been around for decades, at first implemented with bipolar technologies. One particular feature is that the current mirror is not used for biasing purposes, but it processes signals from the differential pair. Also, the stage is not symmetric, because M3 is diode-connected and M4 is not, greatly affecting the properties of the circuit. Moreover, the topology contains two signal paths with different transfer functions.

These two dynamics determine the output voltage at the same time: (1) a decrease (increase) in V_{in}^+ causes M2 to draw less (more) current, which elevates (lowers) the output voltage—consider M2 and M4 as a NMOS common source amplifier—, and (2) an increase (decrease) in V_{in}^- augments (reduces) the current through M1, which is copied to M4 and thus rises the output voltage—then again, consider M4 and M2 as a PMOS common source amplifier—. Therefore, both signal paths demand the output voltage to go up (or down). Because all transistors should be well-matched, the DC current flowing through all transistors is the same and equal to half the bias current.

The two paths enhance each other in V_{O1} . Note that they also have opposite effects on I_{M2} . These influences collapse when the differential input V_{in} becomes negative, so V_{O1} increases enough to make $V_{SD,M4} < V_{SG} - V_{th}$, M4 leaves saturation and I_{M4} is no longer the copy of I_{M3} . For a moment, I_{M2} is determined by $V_{O1} - V_P$. Because the PMOS load voltage drop is close to zero,

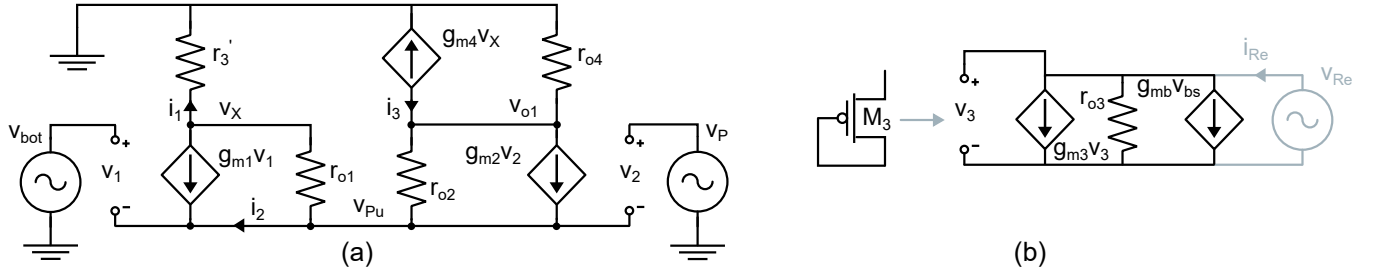


Figure 3.4: (a) Small-signal circuit of the comparator's first stage. Although V_{bot} is a constant DC reference, we include its AC component for the calculation procedure. (b) Small-signal equivalent of a diode-connected PMOS.

V_{o1} is equal to V_{DD} . Eventually, both M2 and M4 are cut off. In any case, the V_{in}^- signal path does not have to be considered when using the circuit as a comparator, as it only processes the DC reference V_{bot} .

With this configuration, the lower limit of V_{o1} (when $V_{in}^+ > V_{in}^-$) is limited by the almost invariable V_P —mostly determined by the current source M5 and the common mode input voltage—, because $V_{DS,M2} = 0$ and cannot sink V_{o1} further. Also, when V_{o1} is high ($V_{in}^+ < V_{in}^-$), V_P cuts off M2 because $V_{GS,M2} < 0$. The second stage will widen the output range.

Besides the large-signal behavior description, it is interesting to analyze the small-signal properties of the comparator, since many of its parameters are a function of how the comparator amplifies when operating in the linear region; that is, when the differential input is around zero. Fig. 3.4(a) shows the small-signal equivalent circuit. Note that the diode-connected PMOS, shown in Fig. 3.4(b), behaves like a small-signal two-terminal resistor with value

$$(g_{m3} + g_{mb3})v_{Re} + \frac{v_{Re}}{r_{o3}} = i_{Re} \rightarrow \frac{v_{Re}}{i_{Re}} = r_3' = \frac{1}{g_{m3} + g_{mb3} + r_{o3}^{-1}} = \frac{1}{g_{m3} + g_{mb3}} \parallel r_{o3} \quad (3.6)$$

which can be approximated as

$$r_3' \approx \frac{1}{g_{m3}} \quad (3.7)$$

When small differential inputs are applied, the swings in V_{o1} and V_x are vastly different. This is because the diode-connected device M3 yields a much lower voltage gain from the input to O1 than that from the input to X. As a result, the effects of v_{o1} and v_x at P (through r_{o1} and r_{o2} , respectively) do not cancel each other, and this node cannot be considered a virtual ground.

To find the small-signal gain, we start by assuming that, because P is not grounded, the currents i_1 , i_2 and i_3 are equal

$$i_1 = v_x g_{m3} = i_2 = i_3 \quad (3.8)$$

Kirchhoff's current law (KCL) at O1 yields

$$v_x g_{m3} + \frac{v_{o1}}{r_{o4}} + g_{m4} v_x = 0 \rightarrow v_x = \frac{-v_{o1}}{r_{o4}(g_{m4} + g_{m3})} \quad (3.9)$$

and

$$\frac{v_{o1} - v_P}{r_{o2}} + g_{m2} v_2 - v_x g_{m3} = 0 \quad (3.10)$$

KCL at X yields

$$\frac{v_x - v_P}{r_{o1}} + g_{m1} v_1 + v_x g_{m3} = 0 \quad (3.11)$$

Considering that pairs of transistors M1-M2 and M3-M4 are matched with equal W/L ratios, and that input voltages follow

$$v_{\text{bot}} - v_1 = v_P - v_2 = v_{P_u} \quad (3.12)$$

we can write

$$\frac{v_x - v_{o1}}{r_{\text{ON}}} - g_{\text{mN}}(v_P - v_{\text{bot}}) + 2v_x g_{\text{mP}} = 0 \quad (3.13)$$

where the subindex N and P refer to NMOS and PMOS, respectively.

Substituting the value of v_x in Eq. (3.13) with that obtained in Eq. (3.9)

$$g_{\text{mN}}(v_P - v_{\text{bot}}) = -v_{o1} \left(\frac{1}{2r_{\text{OP}}r_{\text{ON}}g_{\text{mP}}} + \frac{1}{r_{\text{ON}}} + \frac{1}{r_{\text{OP}}} \right) \quad (3.14)$$

We can neglect the influence of the first term in the right-side sum. Therefore, the small-signal voltage gain at low frequencies is approximately

$$A_{V1} = -g_{\text{mN}}(r_{\text{ON}} \parallel r_{\text{OP}}) = -\frac{g_{\text{mN}}}{g_{\text{dsN}} + g_{\text{dsP}}} \quad (3.15)$$

The negative sign is related to how we consider inverting and non-inverting inputs during the analysis. Remember that the second stage inverts the output. Because we can also compute the voltage gain as the product between the short-circuit transconductance G_m and the output resistance R_{o1} , we get

$$G_m = g_{\text{mN}} \quad (3.16)$$

and

$$R_{o1} = (r_{\text{ON}} \parallel r_{\text{OP}}) \quad (3.17)$$

A more detailed analysis of these expressions can be found in [60, p. 152].

The frequency response of the circuit is also worth to consider. There are two poles of interest: (1) at X , referred as the mirror pole, with a capacitance C_x equal to the sum of C_{gd1} , C_{gd4} , C_{gs3} , C_{gs4} , C_{db3} and C_{db1} , and (2) C_{o1} accounting for C_{gd2} , C_{gd4} , C_{db2} , C_{db4} and C_L . The mirror pole is typically much higher in magnitude than the output pole, an assumption that can be easily justified by considering that the small-signal resistance at X is approximately $1/g_{\text{m3}}$, much smaller than the resistance at $O1$.

Then again, a detailed analysis of the frequency response of the circuit was done by Razavi [60, p. 201], resulting in the following expressions for both poles

$$\omega_{p1} \approx \frac{1}{(r_{\text{ON}} \parallel r_{\text{OP}})C_{o1}} = \frac{g_{\text{dsN}} + g_{\text{dsP}}}{C_{o1}} \quad (3.18)$$

and

$$\omega_{p2} \approx \frac{g_{\text{mP}}}{C_x} \quad (3.19)$$

3.2.2 Common-source amplifier

As we mentioned earlier, the signs of the input terminals we consider for the comparator are the inverse of those traditional assigned to a 5-T OTA. The second stage is a current-sink inverter, producing a positive gain in the comparator.

This PMOS common-source amplifier converts the changes in its V_{SG} to a small-signal drain current, which passes through a NMOS load to generate an output voltage. The NMOS load, therefore, acts as a current source that provides a path to ground to the bias current flowing

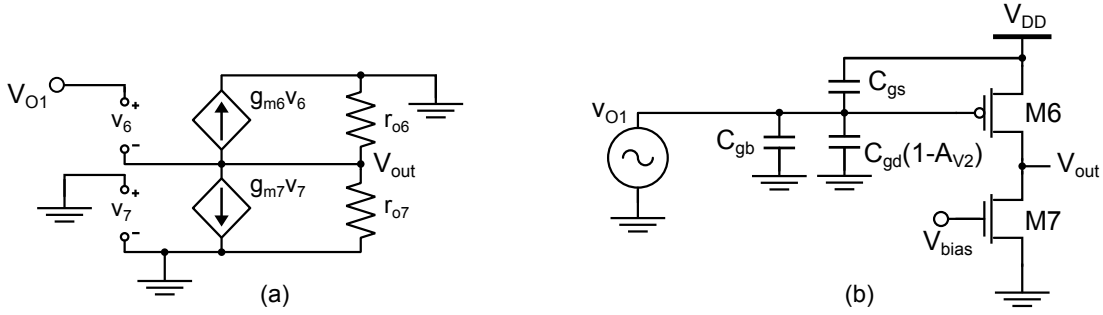


Figure 3.5: (a) Small-signal equivalent circuit of the common-source amplifier. (b) High-frequency model of the stage with Miller's approximation.

through the PMOS transistor, while also providing a large load impedance with a small voltage drop. Furthermore, this stage has a large input impedance.

The small-signal model of the stage can be seen in Fig. 3.5(a). KCL at V_{OUT} yields

$$g_{m6}V_{in} + \frac{V_{out}}{r_{o6}} + \frac{V_{out}}{r_{o7}} = 0 \rightarrow A_{V2} = -g_{m6}(r_{o7} \parallel r_{o6}) = -\frac{g_{m6}}{g_{ds6} + g_{ds7}} \quad (3.20)$$

This stage has one pole located at

$$\omega_{p3} = \frac{g_{dsN} + g_{dsP}}{C_{out}} \quad (3.21)$$

where C_{out} is the sum of C_{gdN} , C_{gdP} , C_{bdN} , C_{bdP} and C_L . Fig. 3.5(b) shows the high-frequency model of the stage, with Miller's approximation for studying the equivalent capacitance at its input. The capacitances shown have a strong influence in the location of the pole ω_{p1} , which is the dominant pole of the system.

3.2.3 Characterization

We define the propagation delay of the comparator as the time required to switch the state of its output when the input signal has crossed the comparison value. In our design, this value is limited by the comparator's slew rate, and follows the expression

$$\Delta t_{o1} = C_{o1} \frac{\Delta V_{o1}}{I_{bias}/2} \quad (3.22)$$

for the output of the first stage, in which Δt_{o1} is measured between the steady value and the threshold point, or trip point, of the second stage. Also, for the output of the second

$$\Delta t_{out} = C_{out} \frac{\Delta V_{out}}{I_{bias}} \quad (3.23)$$

The resulting rising and falling delay is the sum of both.

Our application is required to have a low-power supply, thus the operation at 1.8 V and weak-inversion. We consider all transistors to be saturated in common mode.

A simple expression for the low-frequency gain is obtained from Eq. (3.15) and Eq. (3.20)

$$A_V = \frac{g_{m2}}{g_{ds2} + g_{ds4}} \cdot \frac{g_{m6}}{g_{ds6} + g_{ds7}} \quad (3.24)$$

where subscripts refer to transistors at each stage.

Our design process starts by selecting drain currents, inversion coefficients IC, and channel length L for desired tradeoffs in performance [61]. An overview of such process is available in Annex. Considering weak-inversion operation, we obtain

$$A_V = \frac{V_{A2}V_{A4}V_{A6}V_{A7}}{n^2U_T^2(V_{A2} + V_{A4})(V_{A6} + V_{A7})} \quad (3.25)$$

Early voltages V_A are a function of transistors L and IC , thus the only variables we as designers have to control the low-frequency gain when devices are set to work on weak-inversion [58, p. 398]. V_A values as a function of IC and L can be found in [61, p. 153].

To calculate the location of poles, we ought to determine the value of the capacitances at C_{O1} and C_{OUT} . Following the expressions

$$C_{o1} = C_{gd2} + C_{gd4} + C_{db2} + C_{db4} + C_{gd6} + C_{gs6} \quad (3.26)$$

and

$$C_{out} = C_{gd7} + C_{gd6} + C_{db7} + C_{db6} + C_L \quad (3.27)$$

Values for C_{gd} are proportional to 0.94 fF/ μm in NMOS and 0.64 fF/ μm in PMOS for a 180 nm process, following the parameters provided in [61, p. 45]. Besides, C_{db} follows

$$C_{db} = W \left(\frac{1}{2} W_{DIF} C_J C'_J + C_{JSW} C'_{JSW} \right) \quad (3.28)$$

where C'_J and C'_{JSW} are reduction factors for V_{db} [61, p. 177]. In any case, considering the Miller effect for C_{gd6} and neglecting parasitics limited to a few fF, we get

$$C_{o1,eq} = C_{gd2} + C_{gd4} + C_{db2} + C_{db4} + C_{gs6} + C_{gd6}(1 - A_{V2}) \quad (3.29)$$

and

$$C_{out} \approx C_L \quad (3.30)$$

We neglect the influence of ω_{p2} and ω_{p3} because they are far from the dominant pole.

3.2.4 Design

The comparator performance is crucial in our application. Its gain, bandwidth, offset and current consumption will determine the behaviour of the pixel. It should be notice that the feedback does not allow the output to move between rails, as the spike produces the reset in the comparator's input before its output saturates.

Based in the group's experience, the photodiode in our design will not produce currents higher than 1 nA. As shown later, this current is encoded in a frequency of 83.33 kHz following Eq. (3.2). But peak values like this are rare in normal conditions. Therefore, we design the comparator seeking to reach a cutoff frequency $f_{-3dB} = 10$ kHz and a DC gain $A_V = 70$ dB. Here are the steps of the design process:

1. **Determining the bias current from a selected propagation delay time for a slewing response.** As mentioned earlier, the propagation delay in both edges is calculated separately for the two stages. We will approximate the input threshold of the second stage in Eq. (3.22) as half the swing at V_{o1} , which is limited by V_{DD} and V_P . Because it is located at the midpoint, the rising edge has the same value as the falling edge

$$\Delta t_{o1} = C_{o1} \frac{V_{DD} - V_P}{I_{bias}} \quad (3.31)$$

For the second stage, the voltage swing occupies the two rails. Therefore

$$\Delta t_{\text{out}} = C_{\text{out}} \frac{V_{\text{DD}}}{I_{\text{bias}}} \quad (3.32)$$

The total delay will be the sum of both. A first approximation to the value of the parasitic capacitances yields

$$C_{\text{o1}} \approx 2 \text{ fF} \quad (3.33)$$

and

$$C_{\text{out}} \approx C_{\text{L}} \approx 15 \text{ fF} \quad (3.34)$$

Using Eq. (3.2) with a maximum current of 1 nA, $C_{\text{ph}} = 15 \text{ fF}$, $V_{\text{DD}} = 1.8 \text{ V}$ and $V_{\text{bot}} = 1.0 \text{ V}$, we obtain a maximum spike frequency of 83.33 kHz, that is, a period of 12 μs . We want to keep the propagation delay at around 5% of this value, $\Delta t_{\text{out}} \approx 0.6 \mu\text{s}$.

From Eq. (3.31-3.34), and approximating $V_{\text{P}} \approx V_{\text{bot}} - V_{\text{thN}}$ we obtain

$$I_{\text{bias}} \approx 50 \text{ nA} \quad (3.35)$$

2. **Sizing current sources from the bias current and weak-inversion operation.** Following the procedure explained in Section [previous section Binkley], we use the drain current, the IC and the channel length as the three independent design variables. We already approximated the bias current of the comparator, so we are in a position where we can calculate the dimensions of M5 and M7.

The IC value for weak-inversion operation is 0.1 [61, p. 54]. Therefore, considering the technology current I_0 is 0.64 μA for our 0.18 μm process we obtain

$$\frac{W_{5,7}}{L_{5,7}} = \frac{I_{\text{D}}}{I_0 \cdot \text{IC}} = \frac{50 \text{ nA}}{0.64 \mu\text{A} \cdot 0.1} = 0.78 \quad (3.36)$$

There is a relevant tradeoff regarding the selection of L , which involves gain, bandwidth, size and I_{D} dependency with respect to V_{DS} . We seek to maximize both gain and bandwidth, while keeping the last two as low as possible. The I_{D} dependency is given by the Early voltage V_{A} . Assuming V_{DS} will be around 0.5 V, we can use the values from [61, Fig. 3.45]. With a V_{A} equal to 10, we obtain a value of $L_{5,7} = 0.48 \mu\text{m}$, and thus

$$W_{5,7} = 0.37 \mu\text{m} \quad (3.37)$$

3. **Sizing the common-source amplifier.** This stage behaves as an inverter: its trip voltage can be changed by varying the width of its transistors. By reducing the width of the NMOS in proportion, the midpoint is moved lower in its transfer voltage characteristic [62, p. 181]. Therefore, M6 has a slightly smaller width than M7

$$W_6 = 0.30 \mu\text{m} \quad (3.38)$$

Whereas we want to increase M6 ratio to improve the stage gain, as expressed in Eq. (3.20), we must also be aware of systematic offset and its impact on M6 ratio.

4. **Sizing the PMOS current mirror so that systematic offset is eliminated.** To ensure there is no inherent input-offset voltage present in the comparator, certain conditions must be met [63, p. 252].

When the differential input voltage is null, the output of the first stage V_{o1} must ensure that V_{SG6} makes I_6 equal to I_7

$$V_{\text{SG6}} = \ln \left(\frac{I_{\text{D7}}}{I_0 W_6 / L_6} \right) \cdot n \cdot U_{\text{T}} + V_{\text{T}} \quad (3.39)$$

It is clear that $V_{SG6} = V_{SD4}$. Without the offset presence, the first stage should be balance when the input difference is null, so $V_{SD4} = V_{SD3}$, and thus $V_{SD3} = V_{SG3} = V_{SG4}$. We can express this last voltage as

$$V_{SG4} = \ln\left(\frac{I_{D5}/2}{I_0 W_4/L_4}\right) \cdot n \cdot U_T + V_T \quad (3.40)$$

Considering these equations and the results from step 2, it yields

$$\frac{I_{D5}/2}{W_4/L_4} = \frac{I_{D7}}{W_6/L_6} \rightarrow \frac{W_6}{L_6} = 2 \frac{W_4}{L_4} \quad (3.41)$$

This simple analysis ensures offset voltages in the order of a few mV. Also, offset is conditioned by mismatch between transistors, which is known to be inversely proportional to \sqrt{WL} .

These constraints allow us to start performing simulations with our design, in order to fine-tune the dimensions of the transistors.

5. **Sizing all transistors for keeping mismatch as low as possible, without occupying too much area.** With the help of Cadence, we will design the comparator with a target input offset specification using the numerical relationships we have considered until now.

We perform a DC analysis, which finds the bias point of all devices. By connecting the comparator in a buffer configuration (with negative feedback and a common-mode input signal), we obtain at the output the value of the input-referred offset. Mathematically, this is justified by

$$\begin{aligned} V_{out} &= A_0(V^+ - V^-) = A_0(V_{CM} + V_{OS} - V_{out}) \rightarrow V_{out} = \frac{A_0}{1 + A_0}(V_{CM} + V_{OS}) \\ &\approx V_{CM} + V_{OS} \end{aligned} \quad (3.42)$$

The common-mode input signal will be the comparison voltage reference V_{bot} . Offset sources are the systematic offset, as expressed in Eq. (3.41), and mismatch between transistors. To study its influence, we configure a Monte Carlo simulation, a simple method of variability analysis useful for simulating the random results of the manufacturing process. After a Monte Carlo analysis, the variable is represented in a histogram and its standard deviation σ .

We design the comparator to reach a σ value of around 6 mV at $I_{bias} = 50$ nA. Also, we need to minimize the input capacitance of the differential pair, as this parasitic will be added to C_{ph} . The results can be seen in Fig. 3.6. The contribution to mismatch is distributed among M1 (45%), M2 (40%), M3 (8%) and M4 (6%). The transistors' dimensions are shown in Fig. 3.1. The resulting gate capacitance of M3 and M4 is 3.92 fF. All aspect ratios and currents ensure weak-inversion operation.

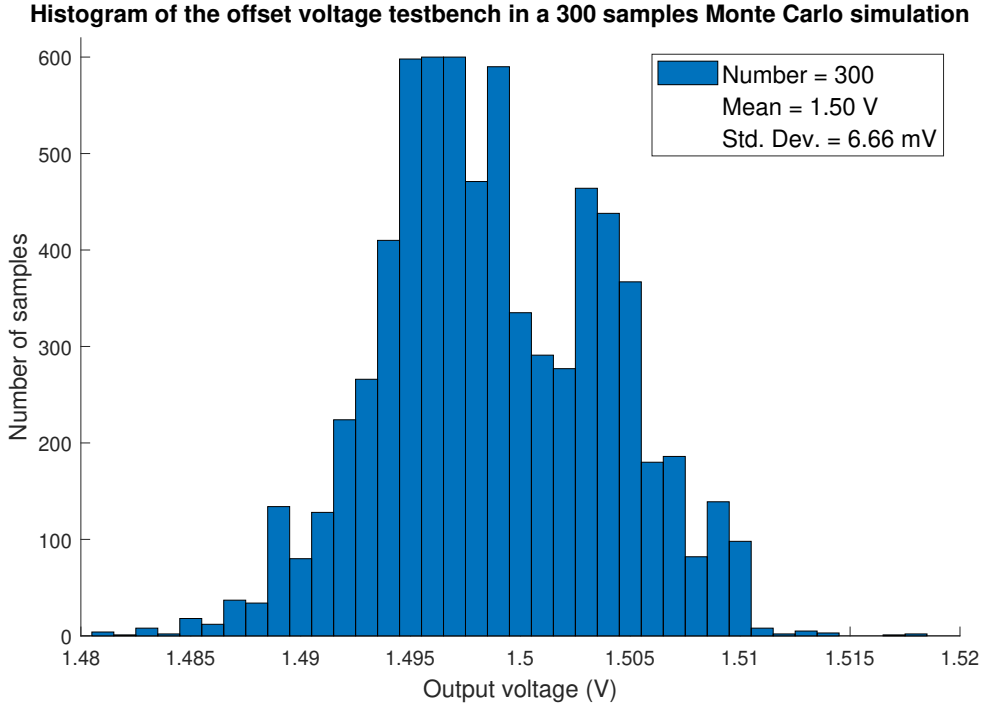


Figure 3.6: Histogram and its corresponding probability distribution for the output voltage in the offset Monte Carlo testbench, with I_{bias} equal to 50 nA. The output contains the common-mode value $V_{\text{bot}} = 1.5 \text{ V}$ and the offset, which is represented by the standard deviation value $\sigma = 6.66 \text{ mV}$.

The input offset voltage is a property of the comparator in an open-loop configuration, caused by the mismatch between transistors. The comparator's offset is a crucial metric in image sensor design since its presence is related to Fixed Pattern Noise (FPN). If adjacent pixels have slightly different transition voltages, they will codify the same illuminance with different frequency values, degrading the image. It is common practice to limit the offset at 5% of the comparison value.

6. **Verify open-loop gain, bandwidth and transition point.** We can measure the small-signal characteristics of the open-loop comparator in a simulation with a simple testbench, in which we connect a high-value RC feedback to the negative input. The Bode plot is portrayed in Fig. 3.7. Using the analysis we did in the previous section we can verify the results and check whether the simulation values are in accordance with the theoretical expressions. In order to find the pole location and gain, we first perform the following calculations

$$g_{m6} = \frac{I_{\text{bias}}}{nU_T} = 1.29 \mu\text{A/V}$$

$$r_{o6} = \frac{V_{A6}}{I_{\text{bias}}} = 283.8 \text{ M}\Omega$$

$$r_{o7} = \frac{V_{A7}}{I_{\text{bias}}} = 53.78 \text{ M}\Omega$$

$$C_{\text{gd}2} = 0.64 \cdot 0.67 = 0.43 \text{ fF}$$

$$C_{\text{gd}4} = 0.94 \cdot 1.34 = 1.26 \text{ fF}$$

$$\begin{aligned}
C_{db2} &= 0.67 \cdot \left(\frac{0.6 \cdot 0.96 \cdot C'_{JN}}{2} + 0.27 \cdot C'_{JSW_N} \right) = 0.29 \text{ fF} \\
C_{db4} &= 1.34 \cdot \left(\frac{0.6 \cdot 1.2 \cdot C'_{JP}}{2} + 0.24 \cdot C'_{JSW_P} \right) = 0.57 \text{ fF} \\
C_{gs6} &= 0.94 \cdot 0.3 = 0.28 \text{ fF} \\
C_{miller6} &= C_{gd6}(1 - A_{V2th}) = 0.94 \cdot 0.3(1 + g_{m6}(r_{o6} \parallel r_{o7})) = 16.69 \text{ fF} \\
C_{O1,eq} &= 19.52 \text{ fF}
\end{aligned}$$

and thus from Eq. (3.18) we find

$$\omega_{p1} = 79.6 \text{ kHz} \quad (3.43)$$

and Eq. (3.25) yields

$$A_V = 70.62 \text{ dB} \quad (3.44)$$

Several parameters have been taken from [61], which probably are not the same as those of the technology of the design (UMC180). This reason justifies the slight differences between the results in Fig. 3.7 and the theoretical ones. In any case, they are precise enough to be considered a good approximation.

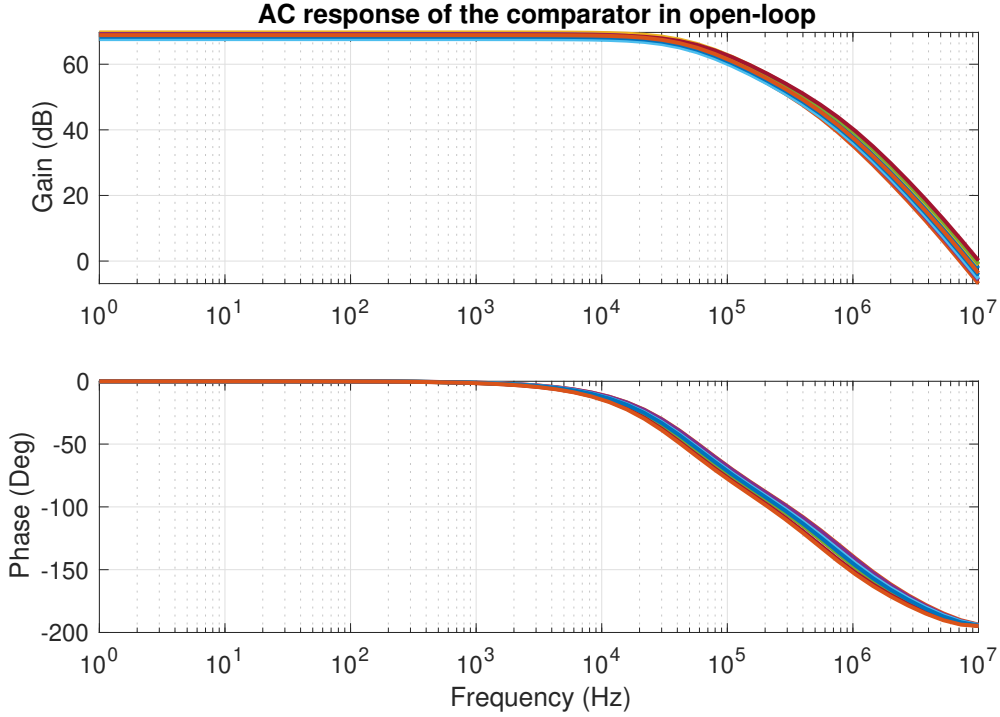


Figure 3.7: Magnitude and phase in a corner AC analysis of the open-loop comparator, without the reset feedback. For nominal values the DC gain is 66.78 dB and the pole is located at 47.11 kHz. We define the process corners at -40°C and 85°C for fast-fast (FF), slow-slow (SS), fast-slow (FS), and slow-fast (SF) transistor models.

- 7. Verify peak currents and common-mode response.** In a DC analysis, the pixel demands 135.0 nA when differential input is large and 12.46 μA in common-mode. We are also interested in current peaks, which can be found by performing a transient simulation of a cluster with four pixels and their respective periphery circuit (its details are described in

Section 4). When transmitting an event, a pixel consumes $136.04 \mu\text{A}$, and its comparator just $0.89 \mu\text{A}$. The entire cluster and its periphery draws 1.87mA at that time, although this number will increase as the periphery digital circuitry escalates with the number of pixels.

It is also worth to verify the correct behaviour of the PMOS reset transistor under two conditions. First, we want to evaluate its leak current when turned off. For values between $V_{\text{DS}} = 0.8 \text{V}$ and $V_{\text{DS}} = 1.8 \text{V}$, the leak current ranges 0.80pF and 1.80pF . These are only relevant when working at low-light conditions. To minimize its impact, we can increase V_{bot} .

Secondly, when the transistor is on it might produce a significant voltage drop if its current is too high. Considering $I_{\text{ph}} = 200 \text{nA}$, the voltage drop is 7mV , which should not cause any trouble. With these verifications, we demonstrate that our comparator reaches the desired specifications with a safe and energy-efficient operation.

8. Verify spiking frequency variations in the looped comparator due to mismatch.

The main purpose of spiking luminance sensors is to codify a photocurrent value in a frequency of voltage pulses, following Eq. (3.2). However, mismatch make the same photocurrent to be transformed into a range of frequencies, resulting in FPN. Mathematically, this is justified by considering that mismatch produces an input-referred offset, adding a contribution to V_{bot} and moving the comparison point, as seen in the simulations below. Although we tried to minimize offset, its impact is unavoidable.

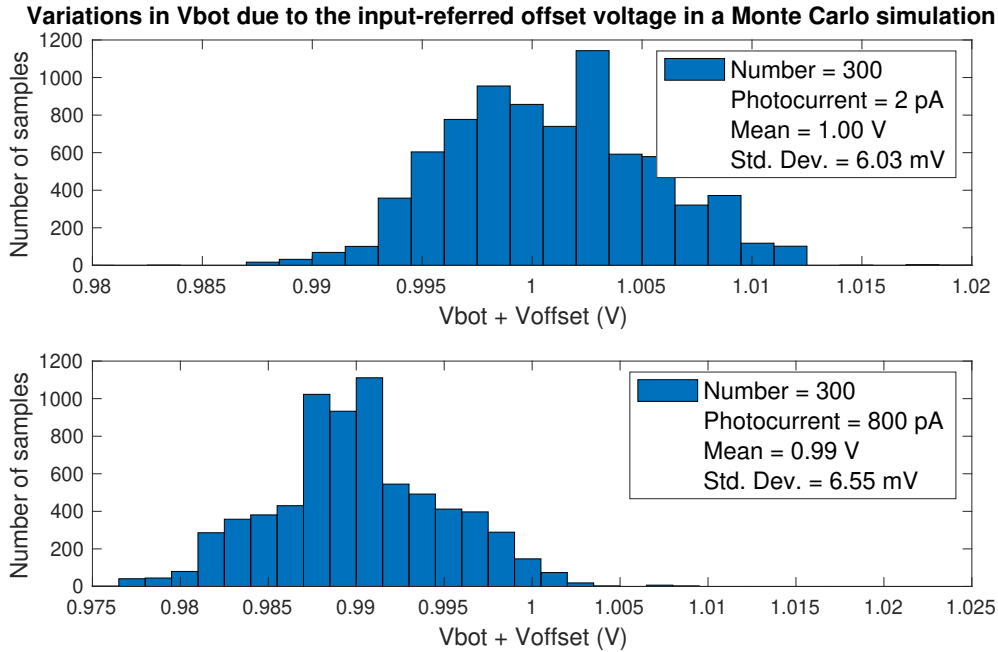


Figure 3.8: Histogram of V_{bot} with two different photocurrents in a Monte Carlo simulation. The variations are the effect of mismatch. The displacement to the left is produced at high frequencies. The comparison point is set at 1V .

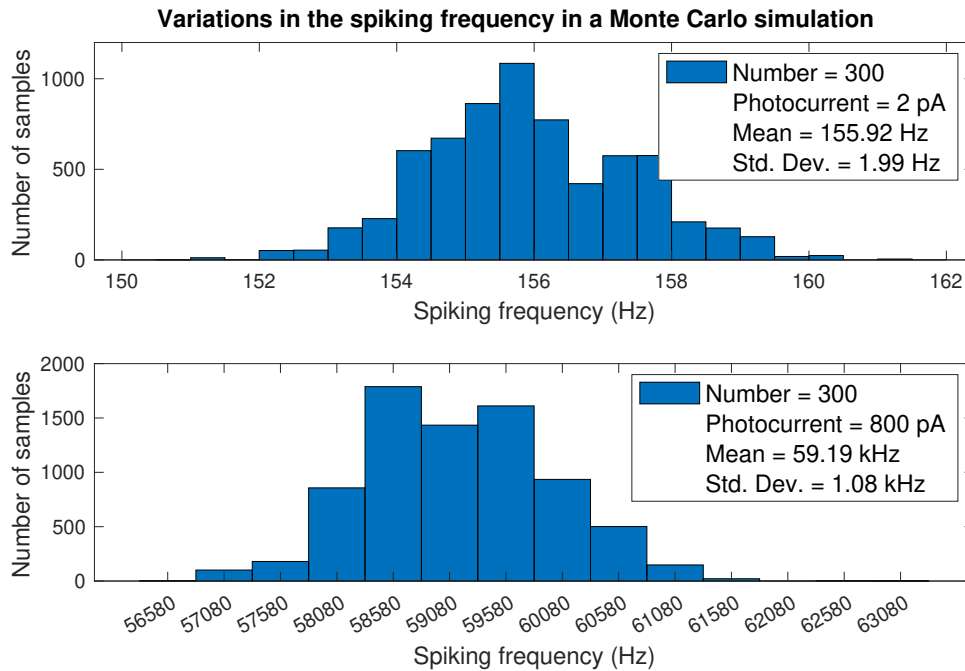


Figure 3.9: Histogram of the spiking frequency of a pixel with two different photocurrents in a Monte Carlo simulation. Same conditions as above.

The Matlab models in Annex allow us to evaluate if noise is present with the measured frequency variations. Fig. 3.10 shows no major differences between both images, and no visible noise.



Figure 3.10: (Left) Original test image in greyscale. (Right) Test image after adding random frequency variations with the achieved standard deviation.

The comparator reaches a switch time of 60 ns, as seen in Fig. 3.11. An inverter at the output of the comparator buffers the signal. The dynamic of the feedback reset is faster than the signal transition, and thus it is not able to reach the top supply rail. In any case, the signal is strong enough to transmit the spike to the next stage.

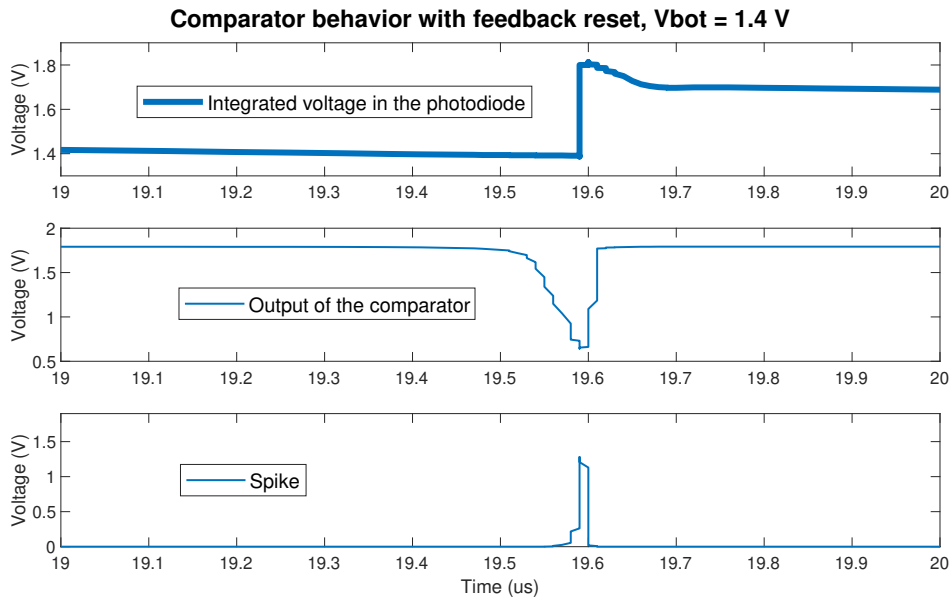


Figure 3.11: Transient simulation characterizing the behavior of the comparator. The fall time is 60 ns and the rise time is 15 ns.

3.3 In-pixel handshake circuit

The asynchronous event-reading scheme is implemented with a 4-phase handshaking protocol. As we explained in Section 2.2, this occurs at three different system levels, which from bottom-up are (1) the pixel row arbitration, (2) the pixel column arbitration, and (3) the communication of pixel's address between the AER sender and receiver. In the first and second processes, the request is implemented with circuits in the pixel, as seen in Fig. 3.1. This architecture was described in [57], and uses a couple of NMOS for global reset and pixel reset after readout, a MIMCAP for holding the spike during a few microseconds, a buffer stage, and the request logic. The receivers, which send the acknowledge back to the pixels, are the arbitrators located in the periphery. The behavior of this logic, working along with the window signal, is represented in Fig. 3.16.

Our design uses a row-column arrangement, with pixels in the 2D matrix sharing the request and acknowledge lines in both dimensions. When a pixel generates an event, it sends a request signal to the arbiter tree in the periphery through a wired-NOR. This occurs for the request lines in rows and columns. But it is not the case for acknowledge lines, which are shared among pixels but are not wired-NORs. These acknowledge lines are the ones that connect to the encoder, where the address of the pixel that produced the event is generated. Because addresses are encoded and transmitted serially, this whole scheme is classified as a word-serial protocol.

The wired-NOR circuit has several electrical constraints [46]. These limit the size of the sensor, as an increase on rows or columns might produce critical timing issues in the wired-NORs. The resolution of our sensor is 96x64 pixels, which is small enough to avoid any inconvenience. Nevertheless, this circuit must be designed cautiously if we want to achieve proper behavior.

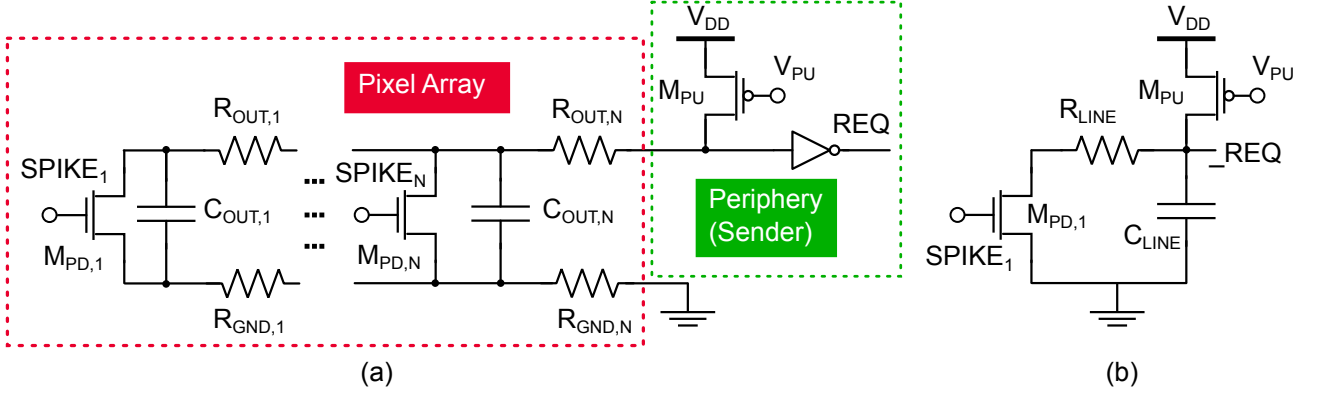


Figure 3.12: (a) Wired-NOR circuit implemented in the pixels' rows and columns, and the sender block in the periphery (the circuits in the sender are explained later). C_{out} and R_{out} represent the drain capacitance, the line crosstalk, the output resistance and the line resistance, respectively. (b) Equivalent circuit of the wired-NOR, neglecting the influence of r_o and R_{GND} . $M_{PD}=2/0.18$, $M_{PU}=18/0.7$, $R_{line}=235.6 \Omega$ and $C_{line}=600.1$ fF.

3.3.1 Estimating the line impedance

Dimensions of M5, M6, and M7 from Fig. 3.1 must be able to pull down the request line when an event is produced. The implementation of the wired-NOR in the request line and its equivalent circuit is seen in Fig. 3.12(a) and Fig. 3.12(b). The PMOS at the end of the line works in the triode region, implementing a pull-up current source.

We can neglect the influence of the output resistance r_o of all NMOS, since they operate in strong inversion and their values are above hundreds of $k\Omega$. Therefore, we are left with a RC network with a current source at one edge. The impedance of the line affects the pull-down time constant τ and the worst-case scenario occurs when an event is produced in the last pixel of a row array, that is, the furthest from the periphery circuits.

We should note that, following Elmore's delay model, the capacitance is always the same, regardless of which pixel tries to pull-down the line. But the resistance is not, and that is why we consider the most distant pixel. In that scenario, and neglecting the ground resistance R_{GND} , the pull-down time can be expressed as

$$\begin{aligned} \tau &= (1 + 2 + 3 + \dots + N)R_{line}(C_{cross} + C_{drain}) = \frac{N(N+1)}{2}R_{line}(C_{cross} + C_{drain}) \\ &\approx \frac{N^2}{2}R_{line}(C_{cross} + C_{drain}) \end{aligned} \quad (3.45)$$

where N is the number of pixels in the array. We neglected the capacitance and the resistance associated with the pull-up transistor because its impact is much smaller than that of the array. This equation highlights the importance of producing a good layout, as the values of R_{line} and C_{cross} depend on the width and length of the metal line, in the first case, and the crosstalk with other metal lines in the second. It also reinforces the idea that the 2D matrix with wired-NORs is not compatible with high resolution imagers.

However, Eq. (3.45) does not give us an exact expression on the time it takes for the line to switch logic. This time is also strongly influenced by the drain current of the pull-up and pull-down transistors, which is a function of their operating regions and dimensions.

Transistor sizes are studied in a simple Cadence simulation of the circuit shown in Fig. 3.12(b). We did this analysis considering the extracted parasitic values of the layout, since it

is a common practice in analog design to modify the circuit after post-layout verification. The value of R_{line} is given by the sheet resistance of the M4 layer (a parameter of the technology, in our case $62 \text{ m}\Omega/\mu\text{m}^2$) times the relation between the length of the line and its width

$$R_{\text{line}} = R_S \frac{L}{W} = 62 \text{ m}\Omega/\mu\text{m}^2 \cdot \frac{1900 \mu\text{m}}{0.5 \mu\text{m}} = 235.6 \Omega \quad (3.46)$$

Also from the extracted layout we obtain $C_{\text{cross}} = 611 \text{ fF}$. It is worth to check the capacitance of the column-request line (377.23 fF), as we need to dimension that line too. The drain capacitance of transistor M5 ($2 \mu\text{m}/0.18 \mu\text{m}$) is found with a DC analysis in cut-off with $V_{\text{DS}} = 1.8 \text{ V}$. The value is 2.35 fF , which multiplied by the 95 transistors of the row yields 222.87 fF . Thus, the resulting capacitance in `_REQ_Y` is approximately 600.10 fF .

With these values, we simulate in Virtuoso and obtain a high-to-low transition—the time it takes for the output to reach 50% of its value after a pixel spike crosses the same value—of 569.9 ps , and a low-to-high transition of 2.89 ns . The fall and rise times are 1.40 ns and 5.53 ns . These values, seen in Fig. 3.13 can be tuned by changing the gate voltage of the pull-up PMOS. The request line is limited at 0.092 V in its logic 1, since the PMOS is never cut-off. Thus, when sending a request signal there is a current consumption of $177.01 \mu\text{A}$.

In the next section, we provide details of the layout and the postlayout verifications performed at the pixel level. Although the circuits in the periphery are presented in the next chapter, it is worth to consider the behaviour of the pixel with the periphery implemented, for the sake of clarity. Fig. 3.14 shows the signal flow in the handshake asynchronous protocol for a cluster of four pixels with different photocurrents. In Fig. 3.15 we can clearly understand the behavior of pulse-frequency modulation in octopus sensors.

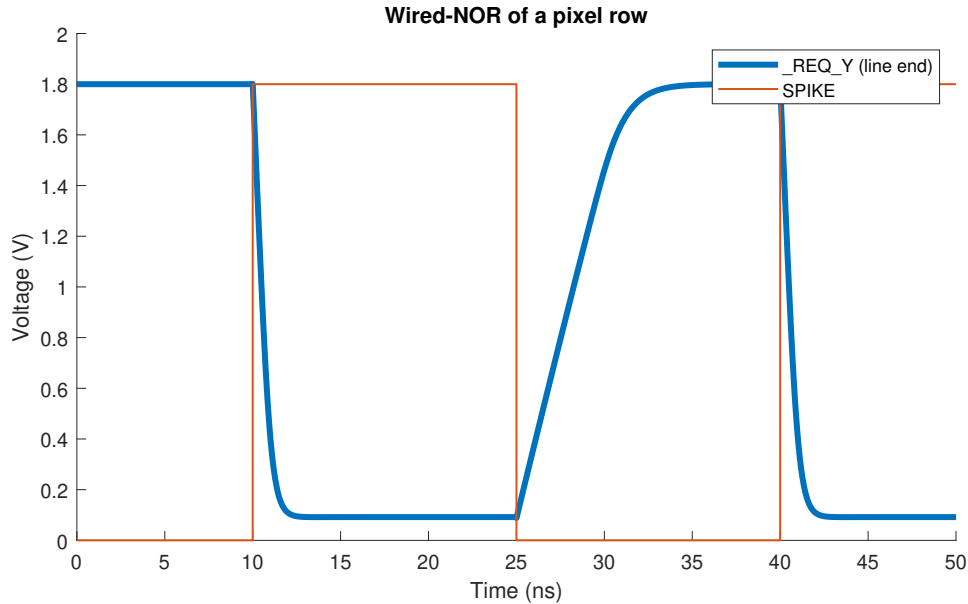


Figure 3.13: Transient simulation of the circuit in Fig. 3.12(b), implementing a wired-NOR in a row of 95 pixels.

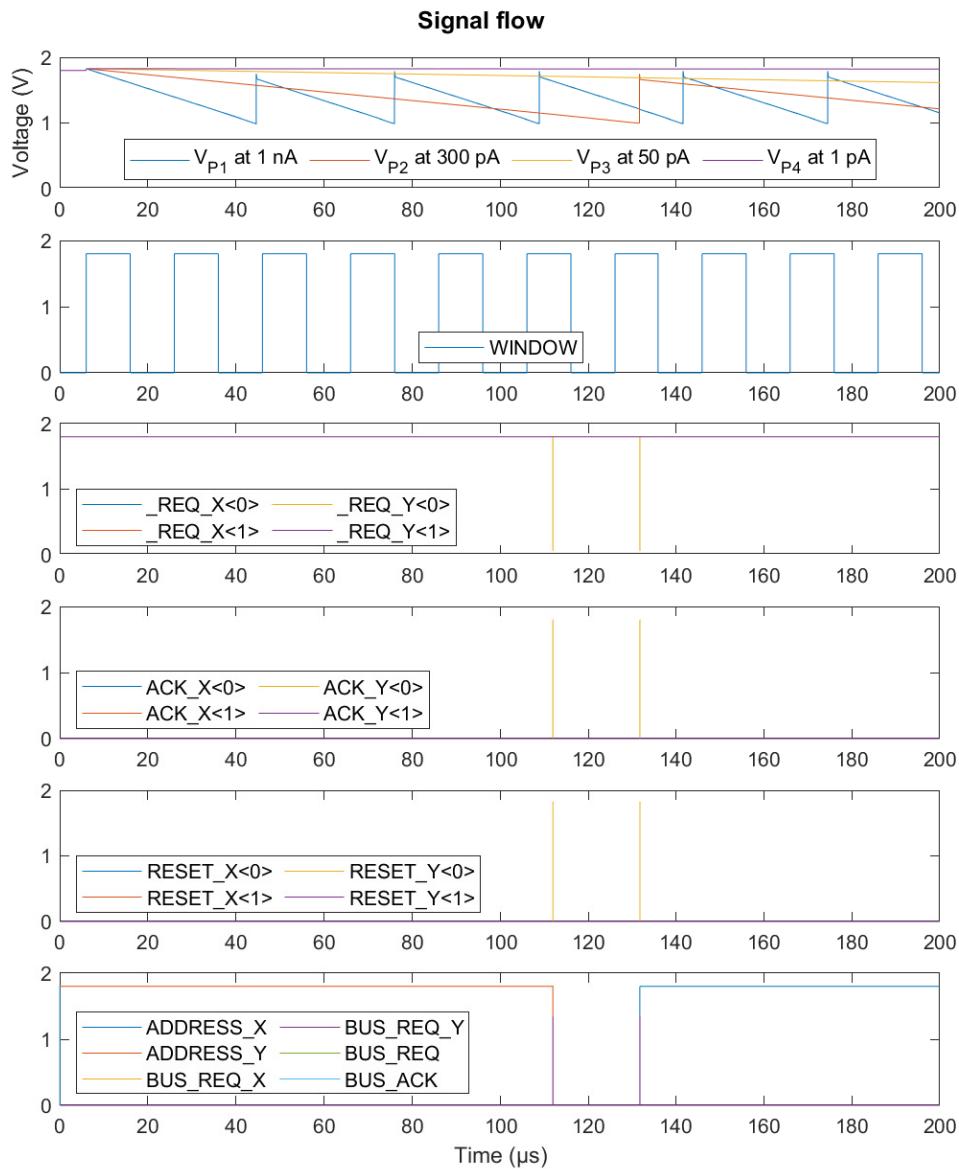


Figure 3.14: Handshaking protocol in a transient simulation for a cluster of four pixels, with two transmitted spikes. First, the voltage V_P in the input C_{ph} capacitance decreases until it reaches the comparison value V_{bot} . The output of the comparator resets V_P to V_{DD} . The spike at $t = 105 \mu\text{s}$ is transmitted if it occurs while WINDOW is high. The spike generates the $_REQ_X<0>$, which is answered by the row arbiters with a $ACK_X<0>$ and $RESET_X<0>$. Then, $_REQ_Y<0>$ produces BUS_REQ and $RESET_Y<0>$; the sensor (AER transmitter) sends its address to an external processor (AER receiver), while the spike stored in the pixel is terminated.

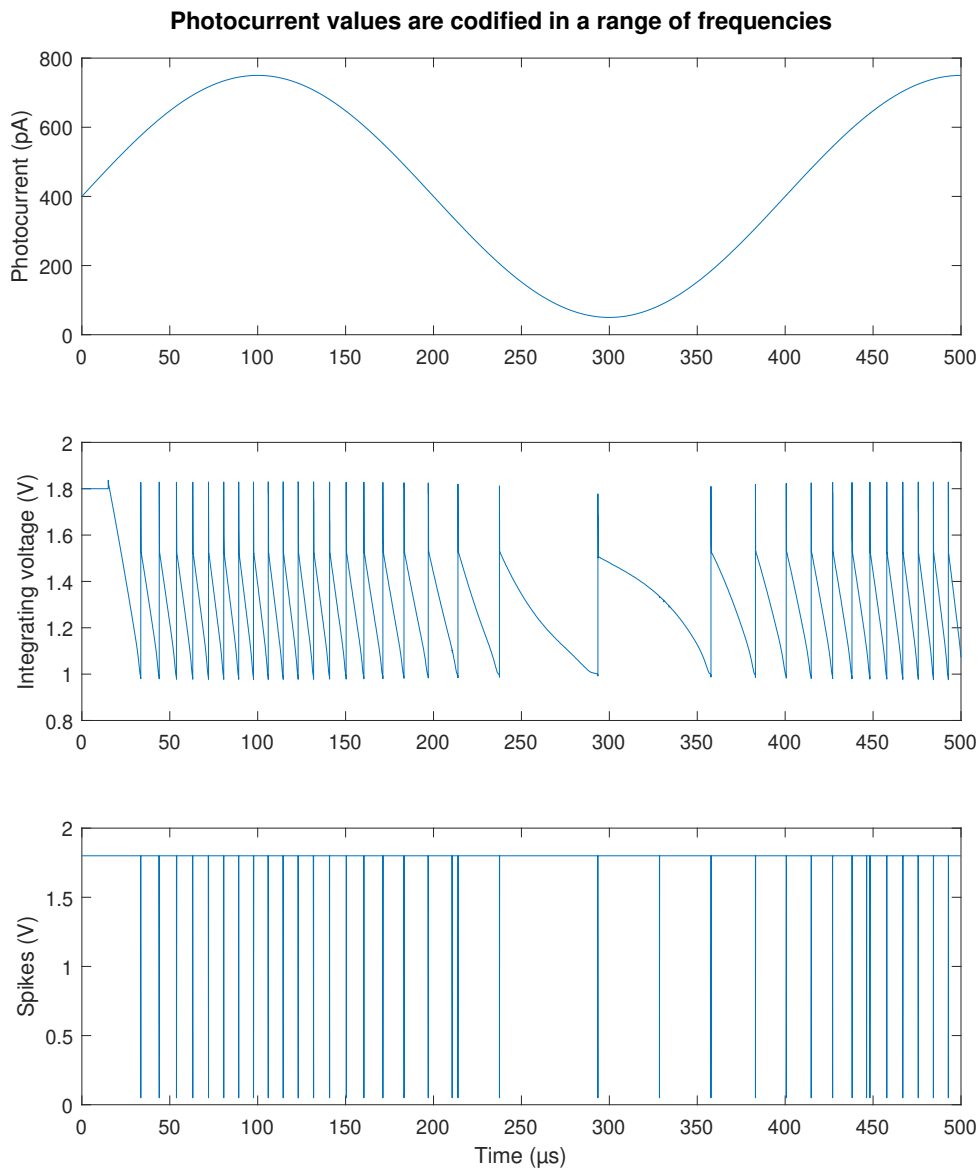


Figure 3.15: Transient simulation with a sinusoidal input photocurrent being codified in spikes at different frequencies.

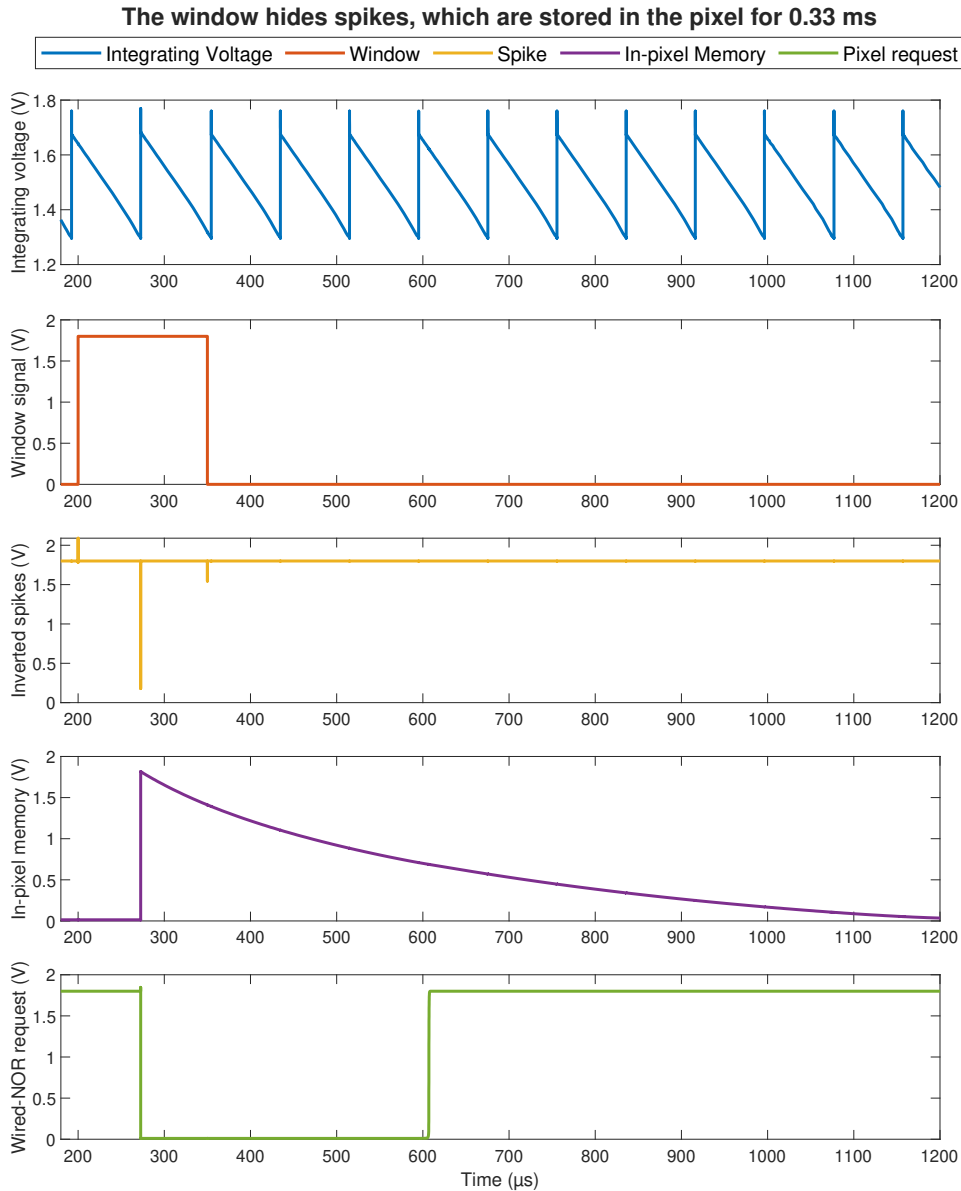


Figure 3.16: Transient simulation to represent the window masking capability, in conjunction with its in-pixel memory. Events are stored for $334.44 \mu\text{s}$ before they are discarded due to current leakage. Spikes and requests use active-low logic in the in-pixel handshake protocol.

3.4 Layout

The layout process for a pixel in our octopus sensor begins with the arrangement and organization of the circuit we had previously designed. The pixel's biggest block is the comparator, formed by seven transistors, which should be located near the photosensitive area. This

photosensitive area is located in the corner of the pixel, as we will form a cluster with four mirrored identical pixels so that N-WELLS are easily shared. This allows for a more compact layout, as DRC rules are more easily followed with such arrangements. We seek to maximize the PN junction that constitutes the photodiode, which is expressed by the fill factor. The layout process plays a crucial role in ensuring that each pixel performs its intended function accurately and reliably.

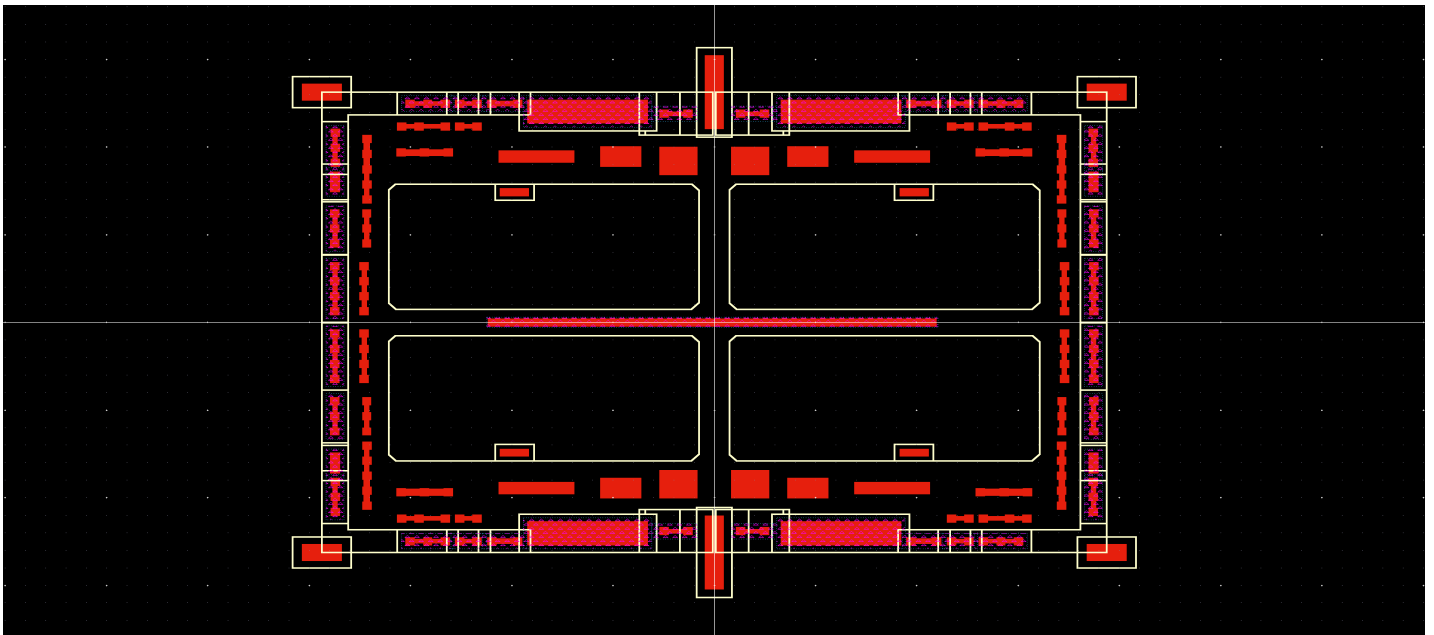


Figure 3.17: Layout of a pixel cluster, showing the NWELL, PPLUS and active layers. Pixels in the cluster share the NWELL, as well as with their neighbors. PWELL is connected to ground in every cluster. By seeing the active areas we can interpret the arrangement of the transistors.

We used metal layers M1 and M2 for routing the internal wires of the pixel. We use minimum widths ($0.24\ \mu\text{m}$ and $0.28\ \mu\text{m}$) whenever the layout forces us to be more compact and we do not care about the path resistance and wider routes for paths which should minimize resistance and have no other paths around that would increase parasitic capacitance due to crosstalk. M3 and M4 are used for those paths that cross the entire matrix. These must be laid cautiously, as these signals are a source of digital crosstalk noise which can impact the behavior of the wired-NORs. In those metal layers, wires' widths are $0.5\ \mu\text{m}$ and $0.6\ \mu\text{m}$, spaced with sufficient distance to minimize crosstalk.

We designed a test pixel, located in the location (64,96), to be able to measure the input and output of the comparator outside the array. These are connected to an analog buffer and a digital buffer, respectively. Polarizations and voltage references are all generated outside the chip.

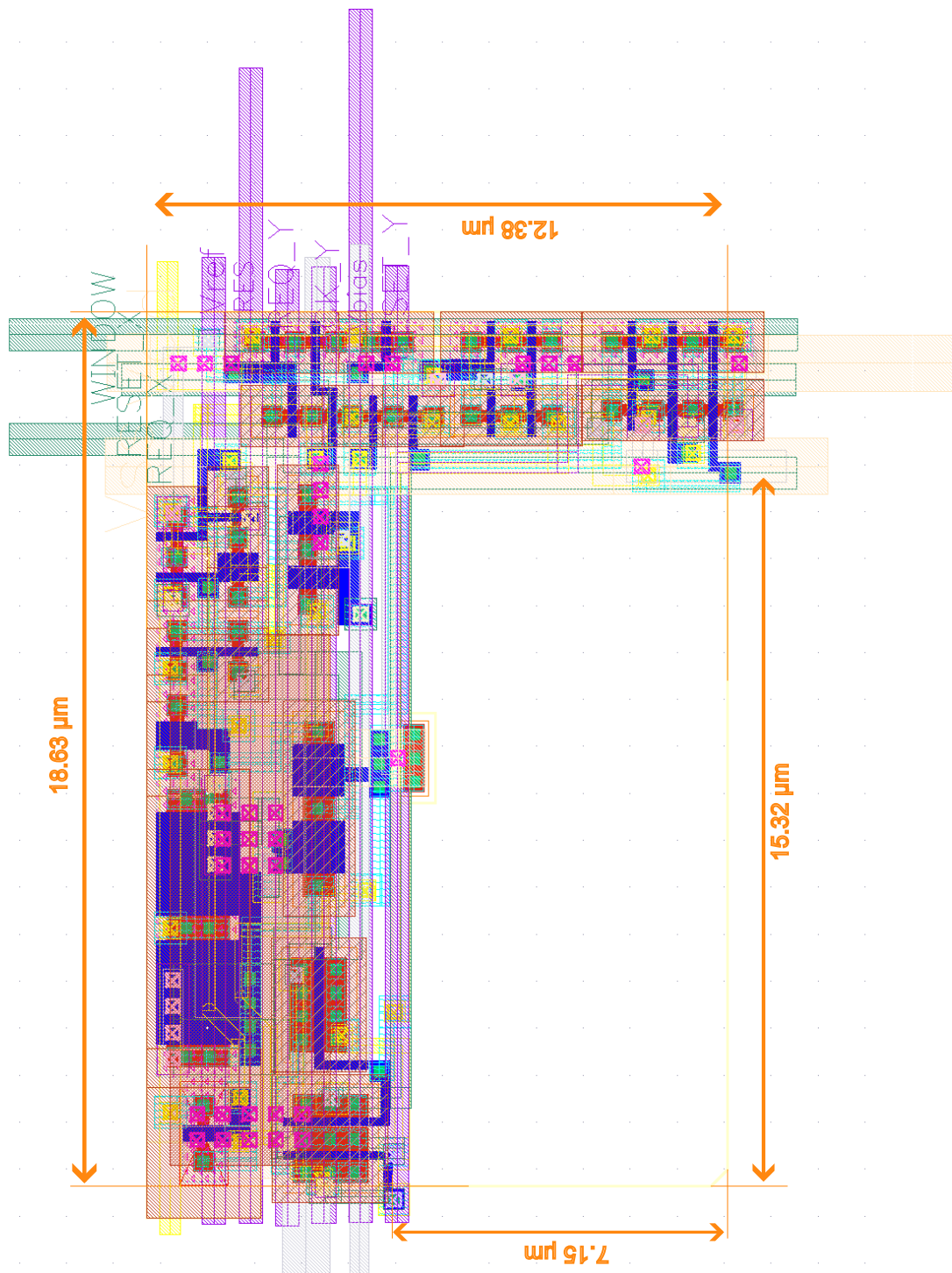


Figure 3.18: Layout of a pixel. We achieved a pixel pitch of $18.63\ \mu\text{m} \times 12.38\ \mu\text{m}$, and a fill factor of 47.5%. The corners of the photodiode are chamfered. M6 and M5 layers are used for power supply, ground and MIMCAP. M4 and M3 are used for routing signals that cross the pixel matrix.

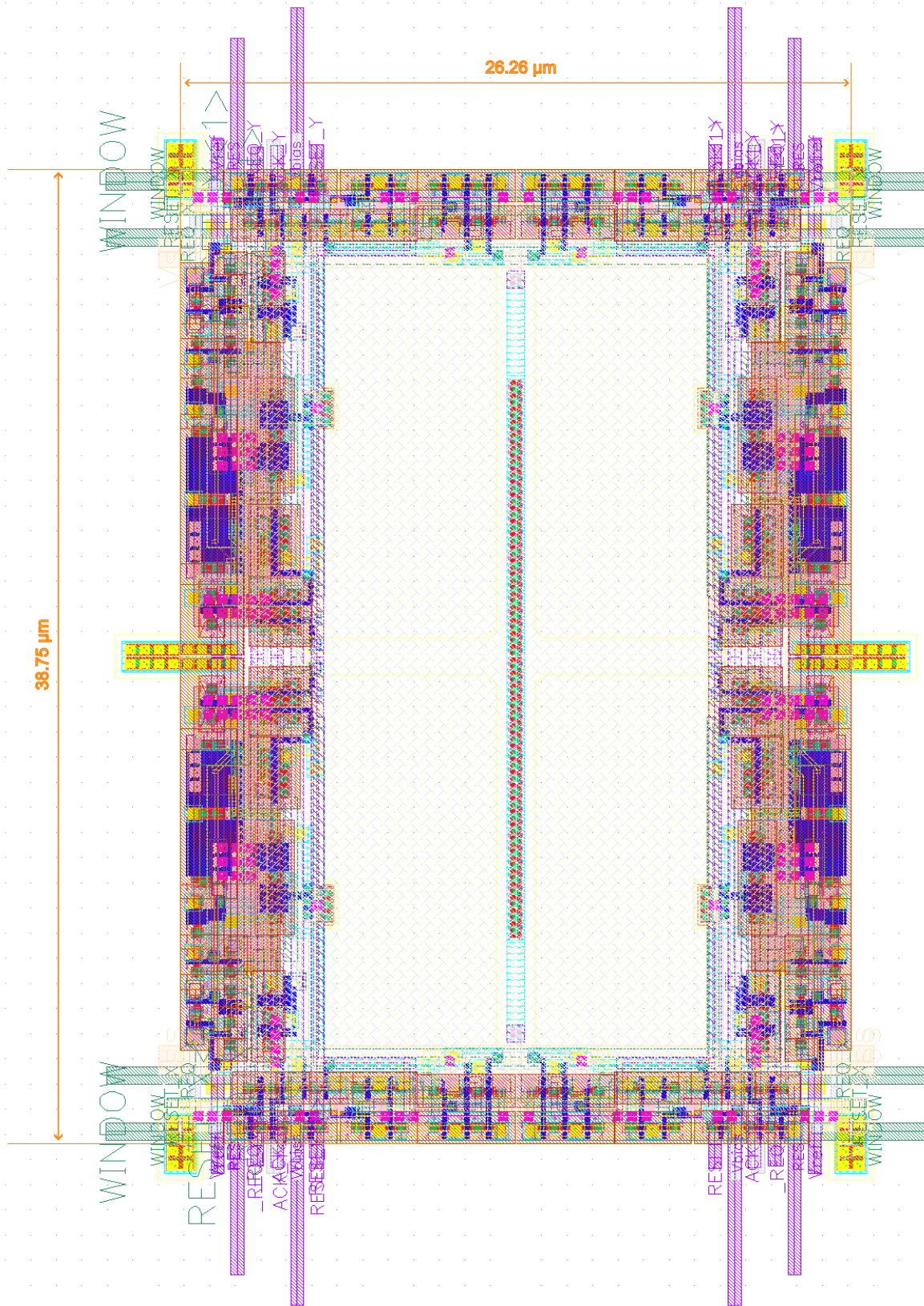


Figure 3.19: Layout of a cluster with four pixels. Pixels have a mirror disposal to facilitate the fulfillment of design rule checks (DRC) and achieve a more compact arrangement.

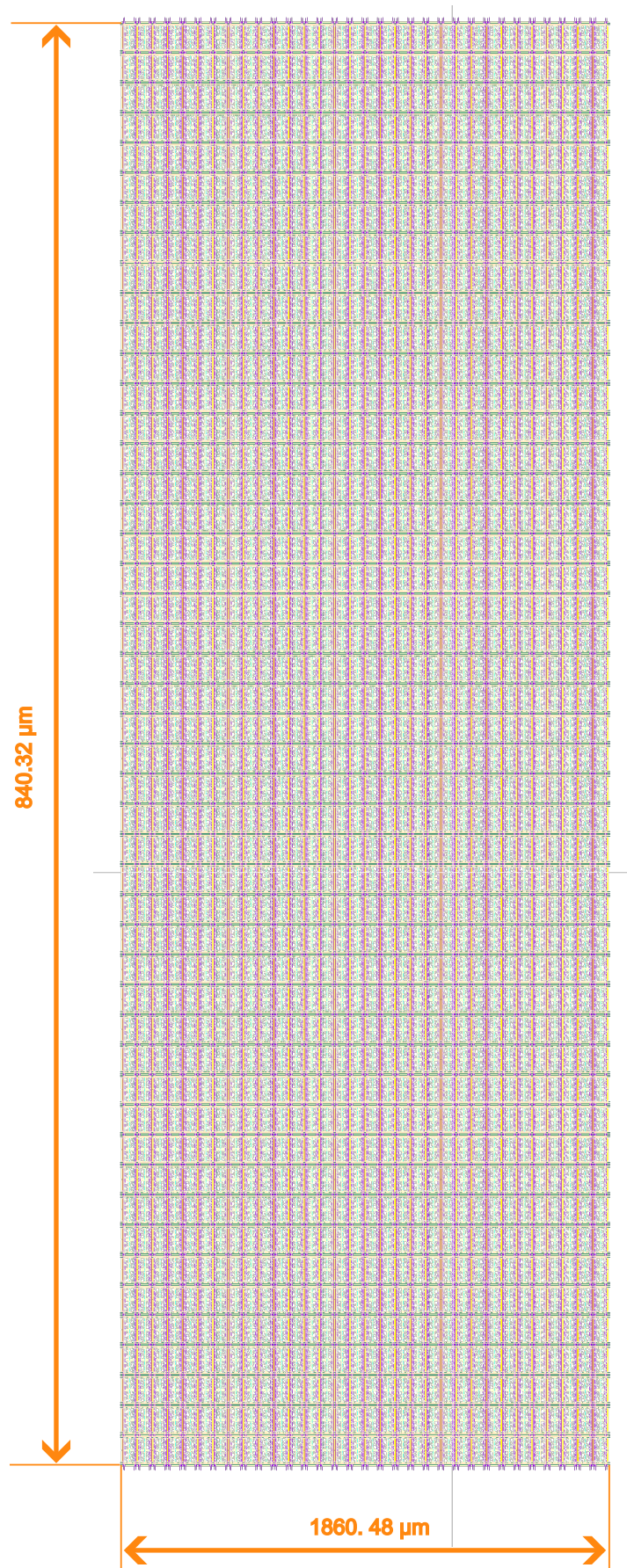


Figure 3.20: Layout of the pixel matrix with a 96 x 64 resolution.

4.2 Encoders

The encoder block uses PMOS and NMOS transistors to codify a one and a zero in each digital bit, respectively. The number of bits depends on the rows and columns that have to be represented.

As seen in Fig. 4.3, the request signal of each array crosses the encoder without any connection. This is because the encoder is usually physically located between the sender interface and the arbiter tree, so the wires need to cross it. The acknowledge signals are used to switch the transistors. They are all dimensioned with $L = 180\text{ nm}$ and $W = 3\text{ }\mu\text{m}$.

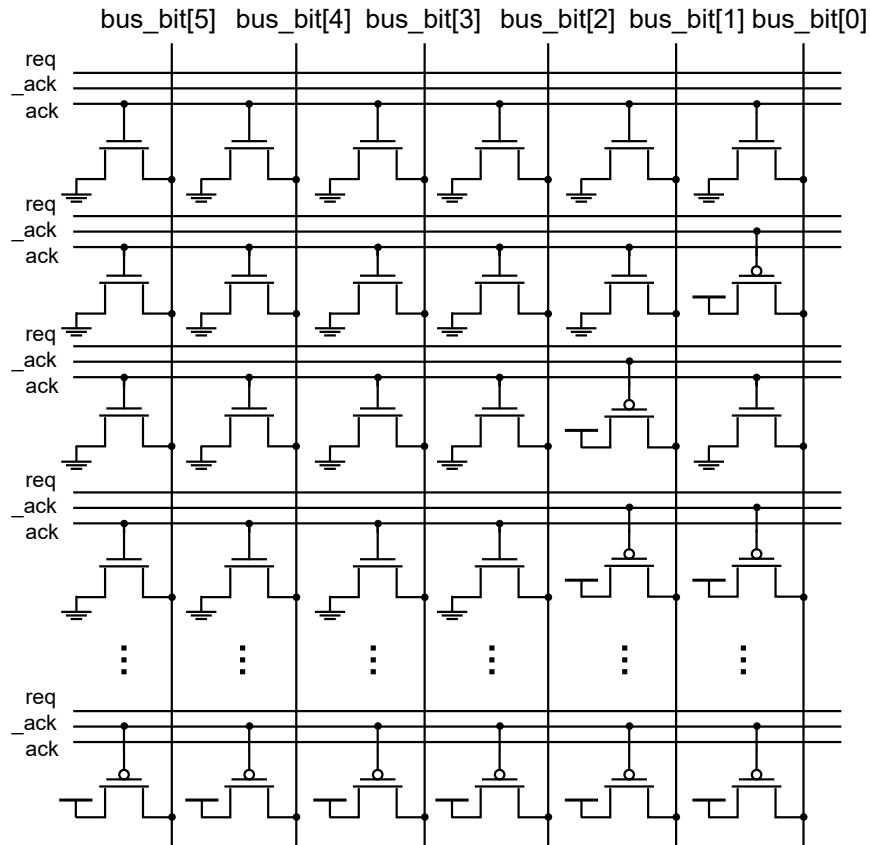


Figure 4.3: Schematic of the row encoder. All transistors have minimum length and $W = 3\text{ }\mu\text{m}$. The request line crosses the block without connections to reach the arbiter tree. The output is an N bit signal to encode 2^N addresses. There is another encoder for columns with one more bit. Each encoder has an output bus.

4.3 Arbiter tree

We already discussed the greedy arbiter in Section 2.2.1. We implemented it with PMOS widths of 660 nm and NMOS widths of 440 nm, all with minimum dimensions.

In Fig. 4.4(a), we show the circuit implementation of the greedy arbiter. Fig. 4.4(b) represents the three last levels in the column arbiter tree. Because the matrix has 96 columns, we need to use a dummy connection so that all requests and acknowledge go through the same number of arbiters.

To validate the behavior of the arbiters, we performed post-layout verifications in a simple testbench. We wanted to assess how much time the arbiter tree needed for acknowledging a request, and whether a greedy path was formed. For that, we did a transient simulation with three requests signals at different instants, and measure the time until it get acknowledged.

We did the simulation in the entire periphery circuits; that is, including the sender interface and the encoders, since their effect on timing is essential. The results can be seen in Fig. 4.5. First, a request from the first row is pulled, and an acknowledge is answered in 6.45 ns. Then, a second request is pulled by the second row, which is not answered because the request from the first is still active. After we reset the first request, an acknowledge comes 0.98 ns later. This difference in time highlights the effect of the greedy path, since the signal did not have to propagate to the top of the tree. Last, we do the same but with the last request, which yields a delay time of 6.63 ns.

Last, we show the layout result of all the blocks in the periphery for rows in Fig. 4.7. The same components are used for columns, but with more elements.

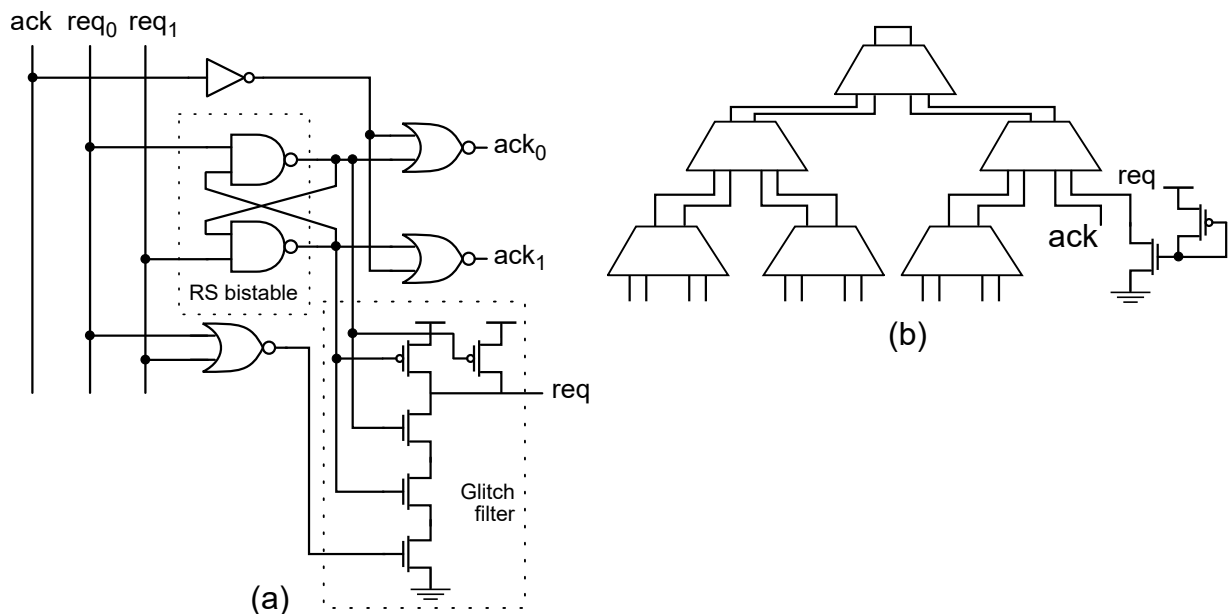


Figure 4.4: (a) Schematic of a single greedy arbiter. (b) Last three levels of the greedy arbiter column tree. We need to use a dummy connection because there are 96 columns, which is not an exponential number with base 2.

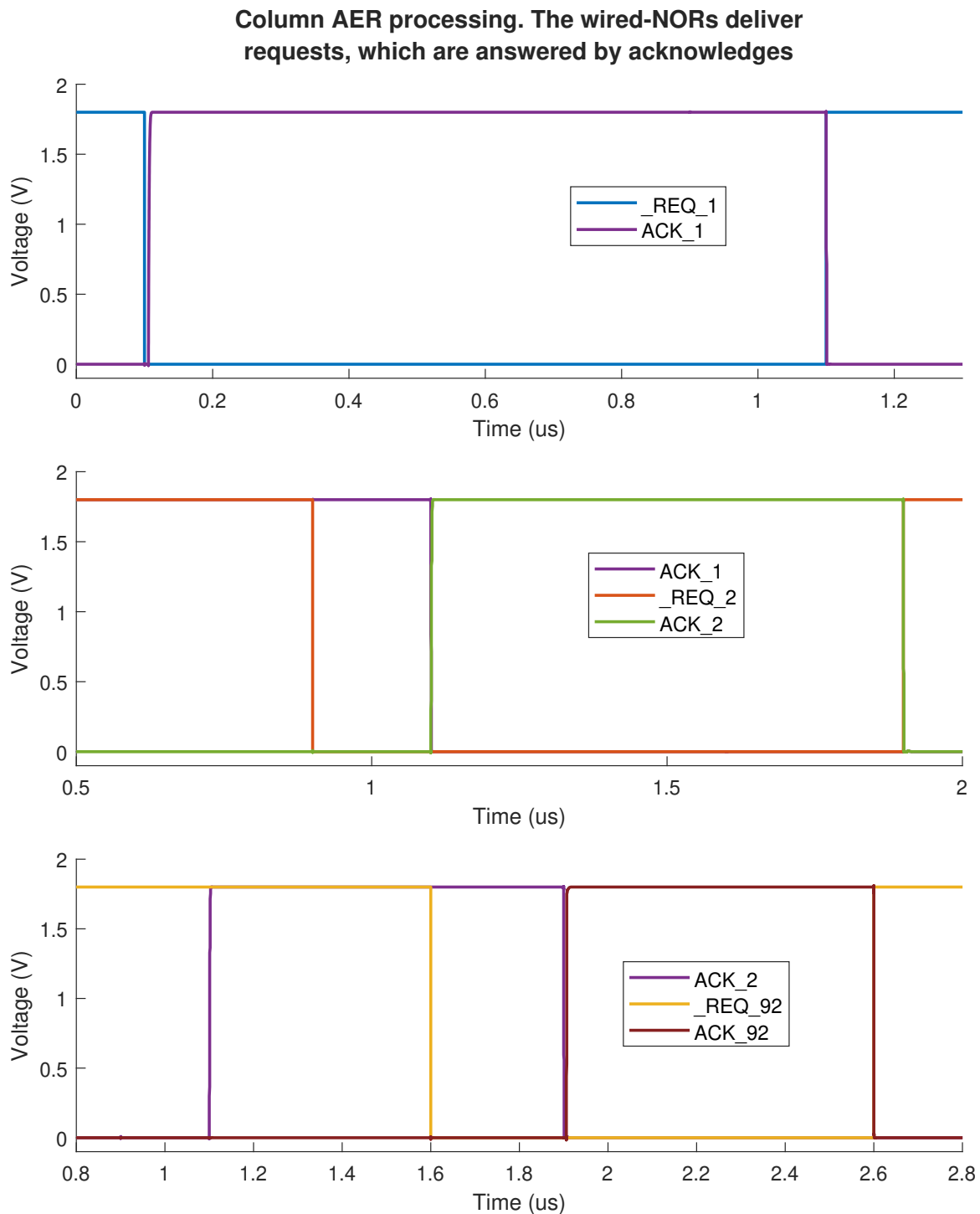


Figure 4.5: Post-layout verification of the AER blocks. The column circuits (sender, encoder, and arbitration tree) are tested with three requests coming from neighbor and extreme pixels. ACK1 comes 6.45 ns after REQ1 goes high, ACK2 comes 0.98 ns after REQ1 goes low (with REQ2 active), and ACK3 comes 6.63 ns after REQ2 goes low (with REQ3 active).

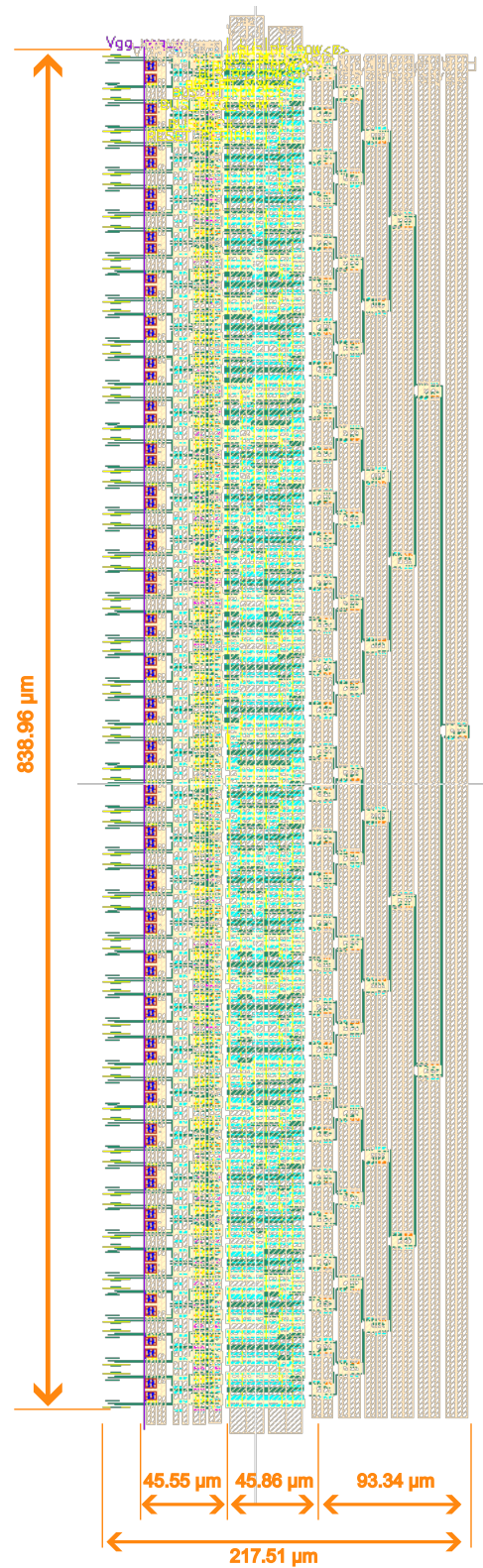


Figure 4.6: Layout of the row periphery, with the sender (left), the encoder (middle), and the arbiter tree (right). This block is connected directly to the pixel matrix and contains 64 senders, 64 encoders, and 63 arbiters in a 6 stages tree. The column periphery is not shown for brevity, but the dimensions are almost identical: 96 senders, 96 encoders, and 96 arbiters in a 7 stages tree. The sixth stage has a dummy connection.

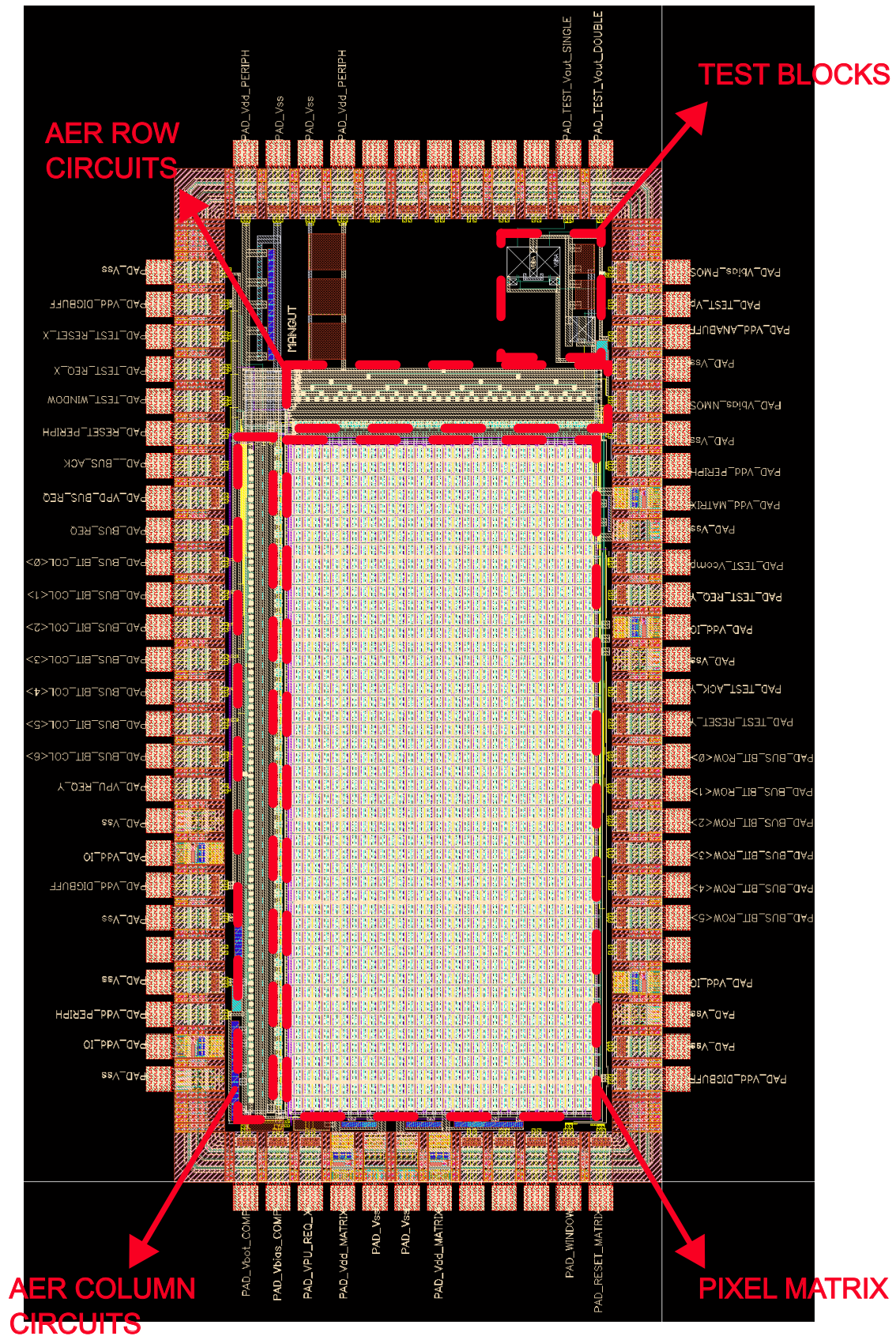


Figure 4.7: Layout of the chip's top view, with the size of two mini-ASICs.

CHAPTER 5

Expected Results and Future Work

Eric Fossum defined the perfect image sensor as an imager with “infinite resolution, dynamic range, and frame rate, together with zero pixel size and power consumption”. However, the output of this ideal sensor would come at an exorbitant cost in terms of data processing. Event-based imagers, inspired by biological systems, introduce an additional metric to the ideal imager: a perfectly balanced exchange of information between the sensor output and the vision problem at hand. These imagers employ local gain control and extensive local computational capabilities to generate an asynchronous stream of digital data that represents only the relevant information for vision.

In this study, our focus was on designing an octopus sensor with a novel decoupling paradigm, which has the potential to produce recognizable images using less data. The two proposed readout modes aim to explore how the output of a spiking matrix can be efficiently processed. Prior theoretical research has already demonstrated that a more sparse readout enhances the extraction of relevant information from visual scenes. Our objective is to test this hypothesis in a real-world application while also investigating whether a random sampling approach based on a Poisson process can generate coherent and meaningful data.

Table 5.1: Comparison to Other Octopus Sensors reported in the Literature

Work	This work	Culurciello et. al. [64]	Ohta et. al. [35]	Leñero et. al. [34]	Leñero et. al. [30, 38]
Year	2023	2001	2005	2014	2017
Technology	180 nm	0.6 μm	0.6 μm	90 nm	180 nm
Power supply	1.8 V	2.9 V Digital 2.7 V Analog	3 V	2.5 V	1.8 V Digital 5 V Analog
Resolution	96x64 pix	80x60 pix	16x16 pix	22x22 pix	96x128 pix
Fill factor	47.5%	14%	-	28%	10%
Pixel pitch	18.63 μm x 12.38 μm	32 μm x 30 μm	240 μm x 240 μm	31 μm x 31 μm	25 μm x 25 μm
Readout	Windowed asynchronous AER	Asynchronous AER	Synchronous	Asynchronous AER	Asynchronous AER
Application	Experimental	Imaging	Retinal prosthesis	Tricolor vision	Sun Sensor, Hybrid APS

By incorporating these innovative technique into the architecture of the octopus sensor, we aim to advance the understanding of event-based imaging and explore the potential benefits of sparse readout and random sampling in capturing and processing visual information with a reduction in data. This work contributes to the ongoing exploration of event-based imaging techniques and their application in various vision-related tasks.

- Development of a testbench using an FPGA board to implement the window generation. For the sparse readout mode, monitoring the AER bus and assessing the rate of off-chip requests can determine if the system is near saturation. In such cases, the FPGA could deactivate the window until all available events in the matrix have been read, indicating completion.
- For the quanta-acquisition mode, investigating and comparing different types of windows is recommended. Méndez-Romero previously studied four window schemes in Matlab simulations [54]. Conducting laboratory experiments to compare and evaluate the performance of these windowing approaches would provide valuable insights.
- A thorough comparison between asynchronous and synchronous readout schemes is still lacking in the literature. Expanding the content of the second chapter and submitting a review article to a relevant journal would contribute to addressing this gap and provide researchers with a comprehensive understanding of the strengths and weaknesses of each approach.

Bibliography

- [1] N. N. House, “Niépce and the invention of photography.”
- [2] E. R. Fossum, “The invention of cmos image sensors: A camera in every pocket,” in *2020 Pan Pacific Microelectronics Symposium (Pan Pacific)*, pp. 1–6, 2020.
- [3] I. Spectrum, “Nobel controversy: Who deserves credit for inventing the ccd?.”
- [4] E. Fossum, “Cmos image sensors: electronic camera-on-a-chip,” *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1689–1698, 1997.
- [5] J. Ohta, *Smart CMOS Image Sensors and Applications*. Optical Science and Engineering, CRC Press, 2020.
- [6] E. R. Fossum, “Camera-on-a-chip: Technology transfer from saturn to your cell phone,” *Technology Innovation*, vol. 15, no. 3, pp. 197–209, 2013.
- [7] E. Fossum and R. Nixon, “Single chip camera device having double sampling operation,” US-PATENT-6456326, Sep. 2002.
- [8] E. Fossum and R. Nixon, “Single substrate camera device with cmos image sensor,” US-Patent-7369166, Sep. 2008.
- [9] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, “Retinomorph event-based vision sensors: Bioinspired cameras with spiking output,” *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.
- [10] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 154–180, jan 2022.
- [11] G. Gilder, *The Silicon Eye: Microchip Swashbucklers and the Future of High-Tech Innovation*. W. W. Norton Company, 1th ed., 2006.
- [12] I. Spectrum, “Moore’s law - the genius lives on.”
- [13] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change,” in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, pp. 2060–2069, 2006.
- [14] K. Boahen, “Point-to-point connectivity between neuromorphic chips using address events,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 416–434, 2000.
- [15] J. A. Leñero-Bardallo, R. Carmona-Galán, and A. Rodríguez-Vázquez, “Applications of event-based image sensors—review and analysis,” *International Journal of Circuit Theory and Applications*, vol. 46, no. 9, pp. 1620–1630, 2018.

- [16] M. A. Mahowald, "An analog vlsi system for stereoscopic vision," 1994.
- [17] P. Lichtsteiner and T. Delbruck, "A 64x64 aer logarithmic temporal derivative silicon retina," in *Research in Microelectronics and Electronics, 2005 PhD*, vol. 2, pp. 202–205, 2005.
- [18] M. Guo, S. Chen, Z. Gao, W. Yang, P. Bartkovjak, Q. Qin, X. Hu, D. Zhou, M. Uchiyama, S. Fukuoka, C. Xu, H. Ebihara, A. Wang, P. Jiang, B. Jiang, B. Mu, H. Chen, J. Yang, T. Dai, A. Suess, and Y. Kudo, "A 3-wafer-stacked hybrid 15mpixel cis + 1 mpixel evs with 4.6gevent/s readout, in-pixel tdc and on-chip isp and esp function," pp. 90–92, IEEE, 2 2023.
- [19] A. Niwa, F. Mochizuki, R. Berner, T. Maruyama, T. Terano, K. Takamiya, Y. Kimura, K. Mizoguchi, T. Miyazaki, S. Kaizu, H. Takahashi, A. Suzuki, C. Brandli, H. Wakabayashi, and Y. Oike, "A 2.97m-pitch event-based vision sensor with shared pixel front-end circuitry and low-noise intensity readout mode," pp. 4–6, IEEE, 2 2023.
- [20] K. Kodama, Y. Sato, Y. Yorikado, R. Berner, K. Mizoguchi, T. Miyazaki, M. Tsukamoto, Y. Matoba, H. Shinozaki, A. Niwa, T. Yamaguchi, C. Brandli, H. Wakabayashi, and Y. Oike, "1.22m 35.6mpixel rgb hybrid event-based vision sensor with 4.88m-pitch event pixels and up to 10k event frame rate by adaptive control on event sparsity," pp. 92–94, IEEE, 2 2023.
- [21] T.-H. Hsu, Y.-K. Chen, J.-S. Wu, W.-C. Ting, C.-T. Wang, C.-F. Yeh, S.-H. Sie, Y.-R. Chen, R.-S. Liu, C.-C. Lo, K.-T. Tang, Meng-Fan, Chang, and C.-C. Hsieh, "A 0.8v multimode vision sensor for motion and saliency detection with ping-pong pwm pixel," IEEE, 2020.
- [22] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, "A 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline," pp. 112–114, 2020.
- [23] O. Kumagai, A. Niwa, K. Hanzawa, H. Kato, S. Futami, T. Ohyama, T. Imoto, M. Nakamizo, H. Murakami, T. Nishino, A. Bostamam, T. Iinuma, N. Kuzuya, K. Hatsukawa, F. Brady, W. Bidermann, T. Wakano, T. Nagano, H. Wakabayashi, and Y. Nitta, "A 1/4-inch 3.9mpixel low-power event-driven back-illuminated stacked cmos image sensor," vol. 61, pp. 86–88, Institute of Electrical and Electronics Engineers Inc., 3 2018.
- [24] B. Son, Y. Suh, S. Kim, H. Jung, J. S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsianikov, and H. Ryu, "A 640×480 dynamic vision sensor with a 9µm pixel and 300meps address-event representation," vol. 60, pp. 66–67, Institute of Electrical and Electronics Engineers Inc., 3 2017.
- [25] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.
- [26] Y. Suh, S. Choi, M. Ito, J. Kim, Y. Lee, J. Seo, H. Jung, D.-H. Yeo, S. Namgung, J. Bong, S. Yoo, S.-H. Shin, D. Kwon, P. Kang, S. Kim, H. Na, K. Hwang, C. Shin, J.-S. Kim, P. K. J. Park, J. Kim, H. Ryu, and Y. Park, "A 1280×960 dynamic vision sensor with a 4.95-µm pixel pitch and motion artifact minimization," pp. 1–5, 2020.
- [27] D. G. Chen, D. Matolin, A. Bermak, and C. Posch, "Pulse-modulation imaging - review and performance analysis," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, pp. 64–82, 2 2011.
- [28] C. Posch, D. Matolin, and R. Wohlgenannt, "A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds," vol. 46, pp. 259–275, 1 2011.

- [29] 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, 2 2003.
- [30] J. A. Leñero-Bardallo, L. Farian, J. M. Guerrero-Rodriguez, R. Carmona-Galan, and Rodríguez-Vázquez, "Sun sensor based on a luminance spiking pixel array," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6578–6578, 15 Oct.15, 2017.
- [31] Lukasz Farian, P. Häfliger, and J. A. Leñero-Bardallo, "A miniaturized two-axis ultra low latency and low-power sun sensor for attitude determination of micro space probes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, pp. 1543–1554, 5 2018.
- [32] R. Gomez-Merchan, J. A. Lenero-Bardallo, M. Lopez-Carmona, and A. Rodriguez-Vazquez, "A low-latency, low-power cmos sun sensor for attitude calculation using photo-voltaic regime and on-chip centroid computation," *IEEE Transactions on Instrumentation and Measurement*, 2023.
- [33] J. A. Lenero-Bardallo, J. M. Guerrero-Rodriguez, R. Carmona-Galan, and A. Rodriguez-Vazquez, "On the analysis and detection of flames with an asynchronous spiking image sensor," *IEEE Sensors Journal*, vol. 18, pp. 6588–6595, 8 2018.
- [34] J. A. Leñero-Bardallo, D. H. Bryn, and P. Häfliger, "Bio-inspired asynchronous pixel event tri-color vision sensor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, pp. 345–357, 2014.
- [35] J. Ohta, K. Kagawa, T. Tokuda, and M. Nunoshita, "Retinal prosthesis device based on pulse-frequency-modulation vision chip," 2005.
- [36] C. Shoushun, F. Boussaid, and A. Bermak, "Robust intermediate read-out for deep submicron technology cmos image sensors," vol. 8, pp. 286–294, 3 2008.
- [37] J. A. Leñero-Bardallo, P. Häfliger, R. Carmona-Galán, and Ángel Rodríguez-Vázquez, "A bio-inspired vision sensor with dual operation and readout modes," *IEEE Sensors Journal*, vol. 16, pp. 317–330, 1 2016.
- [38] J. A. Leñero-Bardallo, R. Carmona-Galán, and Ángel Rodríguez-Vázquez, "A wide linear dynamic range image sensor based on asynchronous self-reset and tagging of saturation events," *IEEE Journal of Solid-State Circuits*, vol. 52, pp. 1605–1617, 6 2017.
- [39] J. A. Lenero-Bardallo, M. Delgado-Restituto, R. Carmona-Galan, and A. Rodriguez-Vazquez, "Asynchronous spiking pixel with programmable sensitivity to illumination," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, pp. 3854–3863, 11 2018.
- [40] J. Xu, Z. Yang, Z. Gao, W. Zheng, and J. Ma, "A method of biomimetic visual perception and image reconstruction based on pulse sequence of events," *IEEE Sensors Journal*, vol. 19, pp. 1008–1018, 2 2019.
- [41] S. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. Wiley, 2014.
- [42] N. Imam and R. Manohar, "Address-event communication using token-ring mutual exclusion," pp. 99–108, 2011.
- [43] K. Boahen, "A burst-mode word-serial address-event link-i: transmitter design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 7, pp. 1269–1280, 2004.

- [44] B. Son, Y. Suh, S. Kim, H. Jung, J.-S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsianikov, and H. Ryu, "4.1 a 640×480 dynamic vision sensor with a 9µm pixel and 300meps address-event representation," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 66–67, 2017.
- [45] C. Li, L. Longinotti, F. Corradi, and T. Delbruck, "A 132 by 104 10m-pixel 250w 1kefps dynamic vision sensor with pixel-parallel noise and spatial redundancy suppression," 2019.
- [46] R. Gomez-Merchan, R. de la Rosa-Vidal, J. A. Leñero-Bardallo, and Rodríguez-Vázquez, "Load reduction and adaptive pull-up strategies for time delay reduction in high-resolution aer sensors," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, (Monterey, California, USA), pp. 6578–6578, 2023.
- [47] S. Fok and K. Boahen, "A serial h-tree router for two-dimensional arrays," vol. 2018-May, pp. 78–85, IEEE Computer Society, 12 2018.
- [48] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*. Springer US, 2013.
- [49] P. D. Häfliger, *A spike based learning rule and its implementation in analog hardware*. Doctoral thesis, ETH Zurich, Zürich, 2000. Diss. Naturwissenschaften ETH Zürich, Nr. 13581, 2000.
- [50] G. A. Subbarao and P. D. Häfliger, "Design and comparison of synthesizable fair asynchronous arbiter," 2020.
- [51] H. E. Ryu and S. Lsi, "Industrial dvs design; key features and applications."
- [52] D. Gehrig and D. Scaramuzza, "Are high-resolution event cameras really needed?," 2022.
- [53] K. Adam, "Timing is everything," p. 164, 2022.
- [54] R. J. Méndez Romero, "Estudio de la técnica de adquisición de datos de imagen cuántica en sensores de imagen asíncronos," 2021.
- [55] R. J. Méndez-Romero, J. A. Leñero-Bardallo, and A. Rodríguez-Vázquez, "On the application of quanta imaging acquisition to spiking luminance sensors," 2022.
- [56] E. R. Fossum, J. Ma, S. Masoodian, L. Anzagira, and R. Zizza, "The quanta image sensor: Every photon counts," 8 2016.
- [57] J. A. Leñero-Bardallo, F. Pérez-Peña, R. Carmona-Galán, and Rodríguez-Vázquez, "Pipeline aer arbitration with event aging," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2017.
- [58] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. New York Oxford, 3rd ed., 2012.
- [59] R. Domínguez-Castro, M. Delgado-Restituto, A. Rodríguez-Vázquez, J. M. de la Rosa, and F. Medeiro, "Cmos comparators," in *CMOS Telecom Data Converters* (A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, eds.), pp. 149–182, Boston, MA: Springer US, 2003.
- [60] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw Hill India, 2th ed., 2017.
- [61] D. M. Binkley, *Tradeoffs and Optimization in Analog CMOS Design*. John Wiley Sons, 1st ed., 2008.
- [62] J. M. Rabaey, *Digital integrated circuits: a design perspective*. Prentice Hall, 2nd ed., 2003.
- [63] T. C. Carusone, D. Johns, and K. Martin, *Digital integrated circuits : a design perspective*. Wiley, 2nd ed., 2011.

-
- [64] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "Arbitrated address event representation digital image sensor," p. 495, IEEE, 2001.