Event-based Image Sensor for low-power

Mohamed Akrarai

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : EEATS - Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)
Spécialité : Nano électronique et Nano technologies
Unité de recherche : Techniques de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés

Capteur d'image intelligent basé sur des événements pour de la basse consommation

Event-based Image Sensor for low-power

Présentée par :

Mohamed AKRARAI

Direction de thèse :

Laurent FESQUET Maître de Conférence Grenoble INP / Phelma, Université Grenoble Alpes	Directeur de thèse
Gilles SICARD Ingénieur / Chercheur, Université Grenoble Alpes	Co-directeur de thèse

Rapporteurs :

Dominique GINHAC
PROFESSEUR DES UNIVERSITES, UNIVERSITE DE BOURGOGNE
Wilfried UHRING
PROFESSEUR DES UNIVERSITES, UNIVERSITE STRASBOURG

Thèse soutenue publiquement le **19 juin 2023**, devant le jury composé de :

Laurent FESQUET MAITRE DE CONFERENCES HDR, GRENOBLE INP	Directeur de thèse
Gilles SICARD DIRECTEUR DE RECHERCHE, CEA CENTRE DE GRENOBLE	Co-directeur de thèse
Alain SYLVESTRE PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Examinateur
François BERRY PROFESSEUR DES UNIVERSITES, UNIVERSITE CLERMONT AUVERGNE	Président
Dominique GINHAC PROFESSEUR DES UNIVERSITES, UNIVERSITE DE BOURGOGNE	Rapporteur
Wilfried UHRING PROFESSEUR DES UNIVERSITES, UNIVERSITE STRASBOURG	Rapporteur

This work is dedicated to my family and my past self

Acknowledgments

I extend my heartfelt gratitude to my supervisors, Professor Laurent Fesquet and Professor Gilles Sicard, for offering me the invaluable opportunity to undertake this Ph.D. journey and for their unwavering support and guidance. I would also like to express my appreciation to Professor François Berry, Professor Wilfried Uhring, Professor Dominique Ginhac, and Professor Alain Sylvestre for their participation in my thesis committee, their keen interest, and recognition of the efforts I put into my research during these years. My sincere thanks also go to the research engineers Nils Margotat and Marco Passy, as well as to all my colleagues in the CDSI team.

I am also grateful to the administrative staff at the TIMA laboratory for their assistance in navigating bureaucratic processes.

Last but not least, I want to acknowledge my family for their unwavering emotional and financial support throughout this journey. I also commend myself for enduring the challenges that came my way, for not opting for an easier path, and for making decisions that will stand the test of time.

Résumé

Dans le cadre du projet européen OCEAN 12, cette thèse de doctorat a réalisé la conception, la mise en œuvre, les tests d'un capteur d'image basé sur les événements, ainsi que la publication de plusieurs articles scientifiques dans des conférences internationales, y compris des conférences renommées telles que le Symposium international sur les circuits et systèmes asynchrones (ASYNC). La conception de capteurs d'images basés sur les événements, qui sont sans trame, nécessite une architecture dédiée et une logique asynchrone réagissant aux événements. Tout d'abord, cette thèse donne un aperçu des architectures basées sur une matrice de pixels hybrides comprenant des pixels TFS et DVS. En effet, ces deux types de pixels sont capables de gérer respectivement la redondance spatiale et la redondance temporelle. L'un des principaux résultats de ce travail est de tirer parti de la présence des deux types de pixels dans un capteur d'image afin de réduire le débit de bits de sortie et la consommation d'énergie. Ensuite, la conception des pixels et de la lecture en technologie FDSOI 28 nm de STMicroelectronics est détaillée. Enfin, deux capteurs d'image ont été implémentés dans une puce de test et testés.

Abstract

In the framework of the OCEAN 12 European project, this PhD achieved the design, the implementation, the testing of an event based image sensor, and the publication of several scientific papers in international conferences, including renowned ones like the International Symposium on Asynchronous Circuits and Systems (ASYNC). The design of event-based image sensors, which are frameless, require a dedicated architecture and an asynchronous logic reacting to events. First, this PhD gives an overview of architectures based on a hybrid pixel matrix including TFS and DVS pixels. Indeed, this two kind of pixels are able to manage the spatial redundancy and the temporal redundancy respectively. One of the main achievement of this work is to take advantage of having both pixels inside an imager in order to reduce its output bitstream and its power consumption. Then, the design of the pixels and readout in FDSOI 28 nm technology from STMicroelectronics is detailed. Finally, two image sensors have been implemented in a testchip and tested.

Contents

ACKNOWLEDGEMENTS i

RESUME iii

ABSTRACT v

Contents vii

1 Introduction **1**

1.1 Context and Motivations 3

1.2 Thesis organization 4

2 Image sensors **7**

2.1 Standard image sensors 9

2.1.1 Photodiode 10

2.1.2 CCD image sensor 11

2.1.3 Standard CMOS image sensors (CIS) 12

2.2 Event based image sensors 14

2.2.1 The Bio-inspired retina 14

2.2.2 The Dynamic Vision Sensor pixel 15

2.2.3 The spiking pixel 16

2.2.4 Address Event Representation (AER) 18

2.2.5 The Time to first spike (TFS) image sensor 19

2.2.6 The Asynchronous Time Based Image Sensor (ATIS) 21

2.2.7 The DAVIS sensor 22

2.2.8 Event-based image sensor challenges 23

2.2.9 Conclusion 26

3 High Level Image Sensor Architecture 29

3.1 Introduction to the proposed architecture 31

3.2 High level architecture 32

3.2.1 Simulation setup and stimuli 33

3.2.2 Readout verifications and spatial redundancy suppression 35

3.2.3 Events collision 36

3.3 Simulation results 37

3.3.1 Events reduction performance 37

3.3.2 Verification simulation results 39

3.3.3 Image quality and readout performances 40

3.3.4 Conclusion 42

4 Pixel Design and Architecture 45

4.1 The 28 nm FDSOI technology 47

4.1.1 Image sensor technologies 48

4.2 Pixel Design 49

4.2.1 DVS 49

4.2.2 TFS 50

4.3 DVS and TFS pixel combination design 53

4.3.1 Averaging pixel architecture 53

4.3.2 the averaging circuit 54

4.3.3 The DVS pixel part 55

4.3.4 The 4 TFS pixels 56

4.3.5 The trigger 58

4.4 Circuit design and power estimation 59

4.4.1 Pixels and circuits design 59

4.4.2 Image Sensor Power Estimation 60

4.5 Conclusion 63

5 The asynchronous readout design: 65

5.0.1 The Readout System 67

5.1 The readout system components 69
5.1.1 The Processing Element 69
5.1.2 The Time-to-Digital Converter 71
5.1.3 Global reset block 72
5.2 Readout system simulation 72
5.2.1 Simulation results 74
5.3 Conclusion 78

6 Test chip **81**
6.1 Event based image sensor (EBIS) 83
6.1.1 testing setup 83
6.1.2 testing methodology 85
6.2 Results 86
6.3 The hybrid event based image sensor testing 88
6.3.1 The testing setup 88
6.3.2 The testing methodology 89
6.3.3 The testing results 91
6.4 Conclusion 94

7 Winner Take All (WTA) arbiter **97**
7.1 WTA cells (Lazzaro cells) 99
7.2 Proposed architecture 100
7.2.1 scaling 103
7.3 Chip test results and comparison 103
7.4 Conclusion 106

8 Conclusion **109**

Publications **115**

References **I**

List of figures **VII**

List of tables **XI**

1

Introduction

We present our context and motivation.

Sommaire

1.1 Context and Motivations 3

1.2 Thesis organization 4

1.1 Context and Motivations

Electronics enhance every aspect of our lives: transportation, health, education, entertainment, and economy. Almost everything around us has a microchip built-in, to augment its utility. Silicon literally powers the human civilization today, and the demand for electronic devices is increasing, in proportions to the ongoing digitization and connectivity[1] in the developing world. This, coupled with the populations increase, has surged the need for micro-chips. The latest pandemic of 2019 accelerated this trend, as billions of people started working and studying from home, with phones, laptops and tablets, one year later this strong demand generated the 2021 semiconductor shortage. The power consumed by these devices is also associated with this trend, which among other factors puts our energy resources at a significant stress. For example, according to the International Energy Agency (IEA) report of 2017, 50% of household electricity appliances by 2040 is expected to come from connected devices, and by the same year, the global energy use of active controls in buildings will be 8 times more than the year 2020 [1]. This begs the question: is the current level of power consumption in electronic devices sustainable? The answer is absolutely not. Tackling this issue, requires lots of combined effort from academia and the industry. Every electronic device around us, should be designed to consume at least an eighth of what it is consuming today, according to the IEA numbers above.

In our field of microelectronics, scaling down the lithography technology according to Moore's law, reduces the power consumption in bulk CMOS. This, coupled with innovative techniques like body biasing, power gating, and clock gating for very large scale digital systems, enabled significant gains in terms of dynamic power consumption. However, reducing transistor dimensions according to Moore's law, does not offer dynamic power consumption gains bellow 100 nm, and a lot of analog circuit designs, in most cases do not benefit from the technology node reduction. In fact the current leakage increases, which notably increases the static power consumption. To overcome this, the Fully Depleted Silicon on Insulator (FDSOI) was proposed, which consists on inserting an insulator layer between the transistor contacts and the bulk, it reduces current leakage

[1] exchange of data between humans, devices and machines (including machine-to-machine), through digital communications networks.

and is more adapted to low power designs [2]. For this thesis, we used the 28 nm FDSOI technology provided by STMicroelectronics. This technology does not specifically provide advantages to image sensor design or photo-diodes. The thesis was also fully funded by the European project OCEAN12.

Our contribution in the field is only focused on image sensors. Image sensor chips are present in vision applications, video communications, robotics, surveillance, and many more. These fall into the category of electronic devices that tend to consume too much power and need an important reduction of their power consumption. The image sensor improvements we propose are not directly benefiting from the technological node or the power reduction techniques above. It exploits a completely new approach, which is based on a non-conventional sampling scheme and implies a redesign of the image sensors. In this thesis, we capitalised on the work previously done on silicon retinas. Our approach consists in reducing the data flow generated by the image sensor, while maintaining a relevant resolution to properly display the viewed scene. The data flow reduction is achieved through eliminating redundancies. The event based pixels, we used, exploit a pixel matrix readout, which has intrinsic proprieties enabling redundancy suppression. Moreover, these pixels are arranged in different architectures and evaluated thanks to high level simulations. The studied architectures show high performance in term of activity and data flow reduction.

1.2 Thesis organization

The second chapter of our thesis presents the working principles of image sensors, particularly standard CMOS image sensors, and highlights their major flaws that limit low power operations. We introduce event-based image sensors and describe their main differences from usual image sensors, as well as their improved readout. Additionally, we explore the biological retina from which event-based image sensing was inspired, and delve into two event-based pixels that form the fundamental building blocks of our novel

architecture, citing and explaining several works in the art.

The third chapter focuses on the main work of our thesis. Before addressing circuit design, we perform high-level simulations of the different pixel architectures we intend to design. The results help us define our pixel matrix architecture and the performance requirements we need to meet. Subsequently, we designed our image sensor using Computer Aided Optimization (CAO) tools from Cadence and the Process Development Kit (PDK) of STMicroelectronics, in chapter four. In chapter 5, we examine our redesigned readout that was previously developed in our lab, and explain in detail each block and its role. This readout was essential for reading data from our image sensor architecture. For our chip test, the readout was implemented outside the image sensor microchip and inside an FPGA for testing and flexibility reasons.

In Chapter 6, we summarize our image sensor test setup as well as procedure, and present the results of our tests, in this same, chapter we present the testing results of our arbiter architecture, that we designed in the same chip as the image sensor. This arbiter is our suggested solution for the arbitration bottle neck and scalability issues in event based image sensors readout.

Finally, we concluded our manuscript by reflecting on the potential of our results, limitations and the pitfalls to be avoided in any future implementations.

2

Image sensors

We define the main characteristics of an image sensor. We compare standard and event based image sensors.

Sommaire

2.1 Standard image sensors 9

2.1.1 Photodiode 10

2.1.2 CCD image sensor 11

2.1.3 Standard CMOS image sensors (CIS) 12

2.2 Event based image sensors 14

2.2.1 The Bio-inspired retina 14

2.2.2 The Dynamic Vision Sensor pixel 15

2.2.3 The spiking pixel 16

2.2.4 Address Event Representation (AER) 18
2.2.5 The Time to first spike (TFS) image sensor 19
2.2.6 The Asynchronous Time Based Image Sensor (ATIS) 21
2.2.7 The DAVIS sensor 22
2.2.8 Event-based image sensor challenges 23
2.2.9 Conclusion 26

2.1 Standard image sensors

Also called imagers, these are devices that retrieve projected light, and convert it to an electrical information. This information can be processed to generate a human readable image through a screen, or interpreted by a system. The information can be conveyed in the form of digital or analog signals. Visible light is not the only electromagnetic wave image sensor react to. Using dedicated pixel, these devices can be designed to detect other wavelengths such as in infra-red, X-ray and ultraviolet imagers. Prior to CMOS image sensor, there was a device called camera tube, it enabled the capture of electron based images and videos and their display [3]. Imaging started with these devices, but since the technology is not widely used anymore it will not be presented in this thesis, but it is worth mentioning.

In our computers we can view images, and if we zoom in, we can see that they are made of tiny units of grey color (a distribution between black and white) also called pixels, in the case of a black and white image, see figure 2.1. So basically an image is a matrix of uniformly distributed pixels.

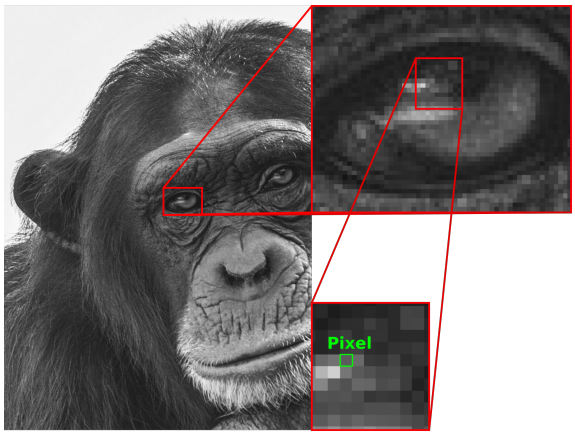


Figure 2.1: Individual pixels in an image

Image sensors have the same geometrical distribution of pixels in the pixel matrix. However, in an image sensor, pixels measure light intensity. In a camera system, the image sensor lays behind the optics at the focal distance, these focus the viewed scene into an optical image on top of the image sensor area, and every pixel reads its local light intensity, projected by the lens. It is very important that the image sensor is at the focal distance, and that the optical image projected by the lens fits the image sensor area, otherwise the viewed scene won't be complete. A complete imaging system (optics,

image sensor and electronic readout) is built to capture the image information which can be summed up into: light intensity, space (position), wavelength, and time [4]. Before introducing how visible image sensors[1] work, it is important to talk about its essential electronic component, which is the photodiode.

2.1.1 Photodiode

It is a reverse biased pn junction. Unlike diodes, photodiode are light exposed pn junctions. The incident light, shouts photons at the semiconductor composing the photodiode. Electrons are free from the covalence bond upon impact from photons. A free electron leaves behind a hole. The amount of free electrons and wholes are theoretically directly proportional to the photons penetrating the photodiode, but limited by the photodiode responsivity. Basically, for a given constant power light, the photodiode could be modeled in a first order by a simple current generator with a capacitance in parallel.

The electronic part of the pixel in an image sensor collects these charges, for later measurement or quantification. In the pixel the collected charge quantity is proportional to the current across the photodiode. [5]. Photocurrent through the photodiode is given bellow:

$$I_{ph} = R_{\lambda}.P \tag{2.1}$$

With R_{λ} being the spectral Responsitivity of a silicon photodiode, it is a measure of the effectiveness of the conversion of the light power into electrical current. It varies with the wavelength of the incident light as well as applied reverse bias and temperature. P is the power of the incident light. The responsitivity also defines the quantum efficiency which is given by the ratio of the observed responsitivity $R_{\lambda ideal}$ by the ideal one $R_{\lambda ideal}$:

$$Q.E = \frac{R_{\lambda observed}}{R_{\lambda ideal}} \tag{2.2}$$

The total current flowing through the photodiode is composed of the reverse-biased current of the pn junction and the photocurrent generated by the incident light.

$$I_{total} = I_{sat}(e^{\frac{q.V_r}{k_b T}} - 1) - I_{ph} \tag{2.3}$$

[1] Visible image sensors that operate in visible light wavelengths (400 to 700nm)

Vr is the reverse biais voltage, q the electron charge, T the absolute temperature in kelvin, K_B Boltzmann constant, and I_{sat} is the saturation current.

2.1.2 CCD image sensor

This sensor uses the Charge Coupled Devices (CCD) architecture to channel charges collected from photodiodes, into the output of the pixel matrix. The charges are then converted into a voltage, current or frequency depending on the type of conversion. The first CCD device was invented by Bell Labs in 1970 [6]. The principle of CCDs is using an array of capacitors with a common terminal connected to ground, as displayed in subfigure (a) 2.2. Shifting charges from the capacitor on the left in an array of capacitors requires propagating a control signal. This signal generates charges in the top terminal (black dots) that have the opposing charge of the charges in the bottom common terminal (white dots). This way, charges at the bottom shift and follow the control signal.

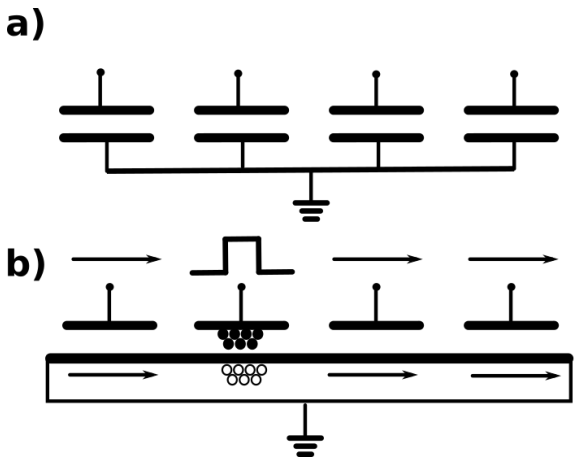


Figure 2.2: Charge Coupled Device operating principle

In a CCD image sensor, invented by Bell Labs also, the same principle is applied to the rows and columns of the pixel matrix. Figure 2.3 is a principle schematic of the most common CCD image sensor (the interline CCD). Here, every pixel has a photodiode and an electrode to evacuate charges, generated in the photodiode during integration time, to the Vertical Charge Coupled Device (VCCD), the charges are then channeled to the Horizontal CCD (HCCD), before finishing at a charge quantity measurement circuit, that will quantify charges and will convert them to an electrical signal.

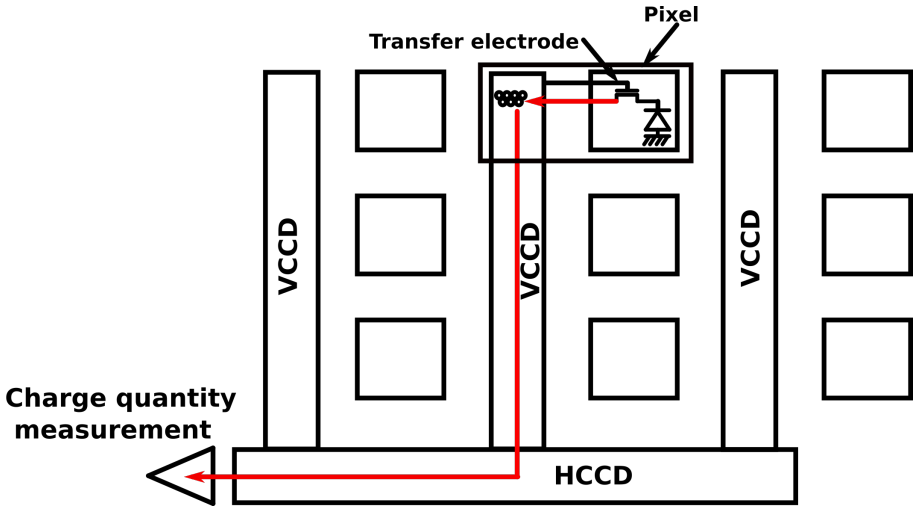


Figure 2.3: Schematic diagram of the most common CCD image sensor architecture [4]

2.1.3 Standard CMOS image sensors (CIS)

Several improvements were introduced, like reduced power consumption and a wide functionality [2]. Production of CMOS sensors for digital cameras was first started in 1997 by Toshiba [4], and since then were implemented in a wide range of products. CIS have evolved from MOS image sensors, also called passive pixel sensors (PPS), these sensors where a transition between CIS and CCD image sensors. As they kept the passive pixel architecture with one transistor per pixel (see Figure 2.3), but replaced the vertical and horizontal CCDs with direct horizontal and vertical lines. Their major flaw was due to the direct connection through a MOS between the photodiode and the vertical line, induced high kTC noise[2], a low frame rate and a limitation at low resolution. These issues were solved in CIS sensors, as an active amplification stage (simple source follower amplifier) was added inside each pixel, to decouple the photodiode node and the pixel output node.

The schematic of figure 2.4 shows the schematic diagram of 3-Tr pixel CMOS sensor. In a CIS, All the pixels have the same exposure time (integration time) to integrate the photo-generated charges, but are reset at different times, depending the matrix line they are, and the readout order (reset is done after the readout). The exposure time is the duration between the reset and the readout of the pixel. During this, a voltage drop occurs across the terminals of the photodiode, it is proportional to the quantity of charges the

[2]kTC noise is a phenomenon caused at the time of setting a capacitor to a certain voltage by switching it on and off

photodiode collects (and so the incident light power). After the exposure time, the row select signal is activated to connect the source follower amplifier (SFA) (composed of the drive and load transistors) to the vertical buses. The SFA copies the voltage value of the photodiode node. Finally the column select transistor is switched ON, to connect the pixel output to the vertical output and transfers the pixel analog value to the column readout architecture. The most occurring noise type in a CIS is the Fixed Pattern Noise (FPN), this noise is due to the fabrication process variations in each pixel and column readout electronic. Hence, the need for an FPN cancellation circuit. This circuit differentiates the signal level of the pixel during the reset time with the signal level at the readout time, therefore suppressing the offset noise variations for every pixel.

The schematic of figure 2.4 highlights the necessary components of CIS sensors for analog display.

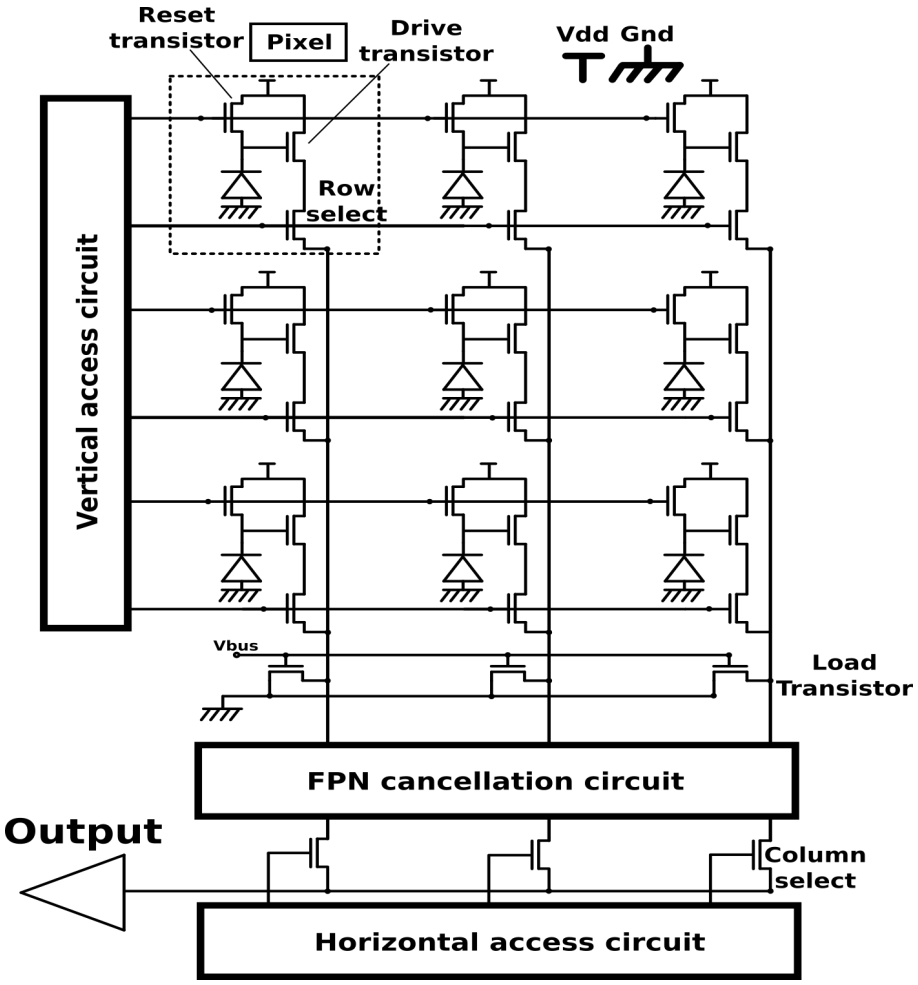


Figure 2.4: 3 transistor CMOS image sensor schematic

All industrial CIS sensors today interface with digital circuits for post-processing or

a digital display. Hence an Analog to Digital Converted (ADC) is necessary. This circuit converts individual signal information into 8 to 14 bit digital data. Basically all current industrial CIS sensors have an ADC per column. Most also include a post-processing circuit in the same chip: colorisation process, tone mapping, JPG compression, ect. An example is the CIS from *Omnivision* that contains a 200 Mega pixels pixel array, an image processor and many communication interfaces, this sensor operates at 8 frames per second, which is reasonable considering the massive pixel matrix [7]. Thanks to 3D integration technology, some industrial circuits implement a CIS and complex image processing bellow the sensor, like AI architecture.

2.2 Event based image sensors

Two decades ago, a new imaging paradigm emerged in research. In contrary to CIS sensors, that rely on a clock signal to synchronize the readout of data from the IS, event based IS do not require a global synchronization signal, here, each pixel functions autonomously and only communicates data, when it has it. These IS came along with new algorithms for image processing, that are also event based. Some imagers are bio-inspired, meaning they have an architecture inspired by the biological retina, others still maintain the architecture of a standard pixel, but use Time to Digital Conversion (TDC) and non-uniform sampling. Throughout this section we will look into these imagers, their internal components and operating principles.

2.2.1 The Bio-inspired retina

In divers fields of engineering, nature has been a source of inspiration. Bio-inspired engineering is the practice of designing systems that mimic nature, as nature is proven to be more efficient in many aspects, like power consumption. For example the human brain is much more power efficient than a computer in 2008 [8], as it consumes 90% less power to do certain calculation. In imaging the source of inspiration can only be the biological retina, which evolved around 400 million years ago [9]. Before the design of the first bio-inspired retina circuit, a methodology was set up for large scale integration of analog

circuits to emulate biological systems, by Maher [10], inspired by the work of Mead in Neuromorphic circuits [3] [11]. Afterwards, the first retina emulation was attempted by Mahowald and Mead, their circuit had all the major cell types that contributed to the task of vision in a biological retina [12]. Figure 2.5 displays the known cells in the retina. On the top the photoreceptor cells convert light into an electrical signal similar to a photodiode. The ON and OFF bipolar cells, separately code for bright spatio–temporal contrast and dark spatio–temporal contrast changes, these cells operate by comparing the photoreceptor signals to spatio–temporal averages computed by the laterally connected layer of horizontal cells. The amacrine cells mediate the signal transmission process between the bipolar and the ganglion cells, these later communicate the signals to the optical nerve.

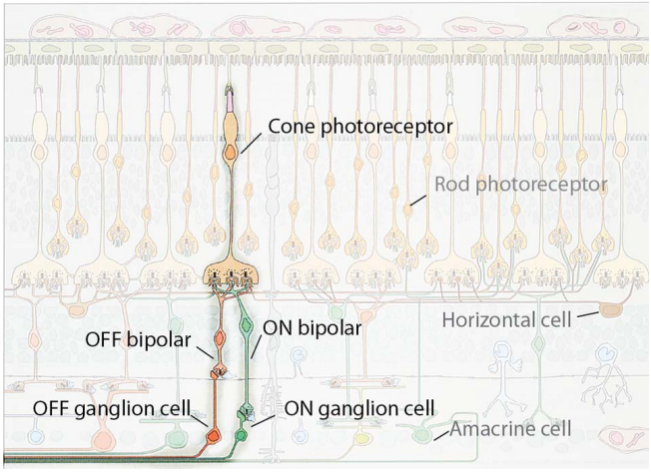


Figure 2.5: The main cells in the biological retina [13]

The first CMOS implementation of the biological retina is the silicon retina by Mahowald [12], and several improvements were introduced to this model by, for example, Sicard [14]. In the next subsection we present its operating principle and inner circuitry, while highlighting the similarity with the biological retina cells.

2.2.2 The Dynamic Vision Sensor pixel

The Dynamic Vision Sensor (DVS) pixel is an event based pixel. It is sensitive to luminance intensity change. Its block schematic is presented in Figure 2.6. This pixel uses a switched capacitor differentiator circuit to compute the difference between two successive

[3] analog circuits that mimic biological neurons in operation

samples of the photo-detector voltage, Vp. This difference is compared afterwards to two thresholds to generate ON or OFF events according to the polarity of the slope of Vlog. A positive change yields to an ON Event and a negative change to an OFF event. In the right side of the figure 2.6, there's a diagram displaying the signals operating in the pixel. In blue, the photodiode current generated transduced and logarithmically compressed into Vlog. The luminance slope changes are detected in the pixel thanks to a differentiator circuit, which translates to Vdiff spikes (Green). The latter go into two comparators, to generate ON or OFF events. The reset signal is used after the detection of an event to clear the process of the event generation, and drive Vdiff back to its reset level, before the next detection. The full circuit schematic of the pixel will be presented in the sequel because our pixel architecture uses a DVS pixel. In the right bottom of the figure 2.6, we can see a sample image of the DVS sensor output after readout. The movements in the scene translate to luminance change. These changes are detected by the sensor and are sent as Address Event Requests (AER) (address of the event in the pixel matrix, encoded as the logic coordinates of the pixel in the pixel matrix), that are read and displayed in the final image. As you can see in the figure bellow, the pixel duplicates the three vertical cell types of a biological retina.

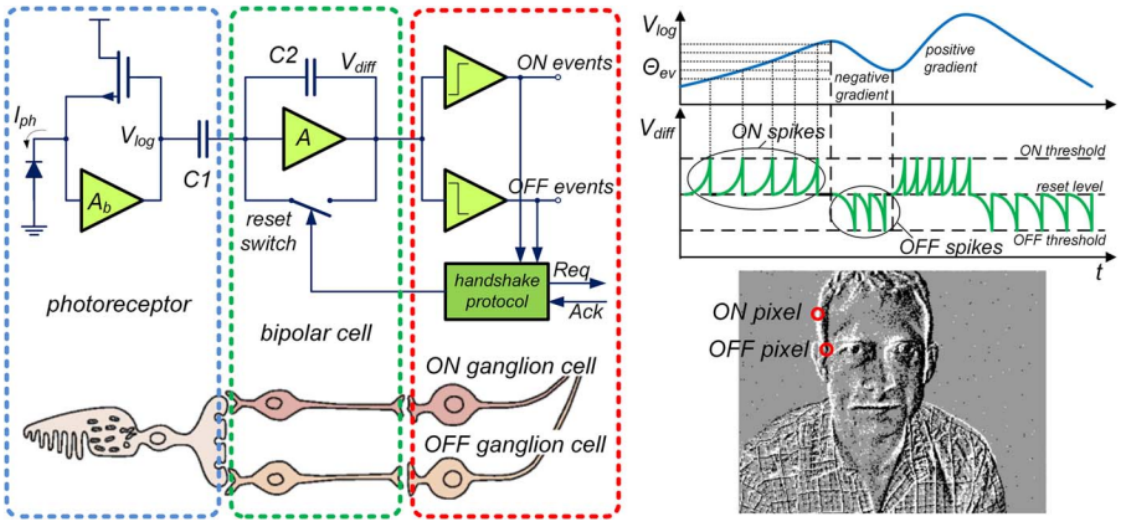


Figure 2.6: DVS pixel block schematic [13], [15]

2.2.3 The spiking pixel

In standard CIS, the position of the ADC relative to the pixel matrix is very important for the performance. Putting the ADC outside of the pixel matrix, for the whole matrix re-

quires a high frequency of operation, and limits the frame rate and the enhance of the array resolution. An ADC per column architecture answers these drawbacks and offers better performance than a whole chip ADC, like lower noise, low load capacitance and lower frequency of operation [16], when operated in parallel. In this perspective, a dedicated ADC per pixel is more beneficial for the SNR , as the load capacitance is even smaller since the ADC is in-pixel, and reduces even further the frequency of operation [17], the only issue that emerges from this, is a low fill factor, since fitting an ADC in the pixel takes a lot of area around the photodiode. Finally, the position of the ADC is decided by the compromise of power consumption, performance, fill factor, and noise tolerance.

The principle of the Spiking pixel was first presented by A.Bermak [18]. This pixel is an effective solution to the compromise mentioned above. Here, the in-pixel ADC was replaced by a comparator and feedback circuit to do pulse frequency modulation (PFM). This pixel operates as a one level crossing sampling circuit, basically when the photodiode voltage Vd (figure 2.7) crosses an externally defined threshold Vref, a spike is generated, and the Counter/Register outputs data encoding the number of spikes generated by the comparator. Every time a spike is generated the photodiode is reset to start another cycle. This way, under the same light intensity during the enabled count duration Tcnt, the pixel will spike at a fixed rate.

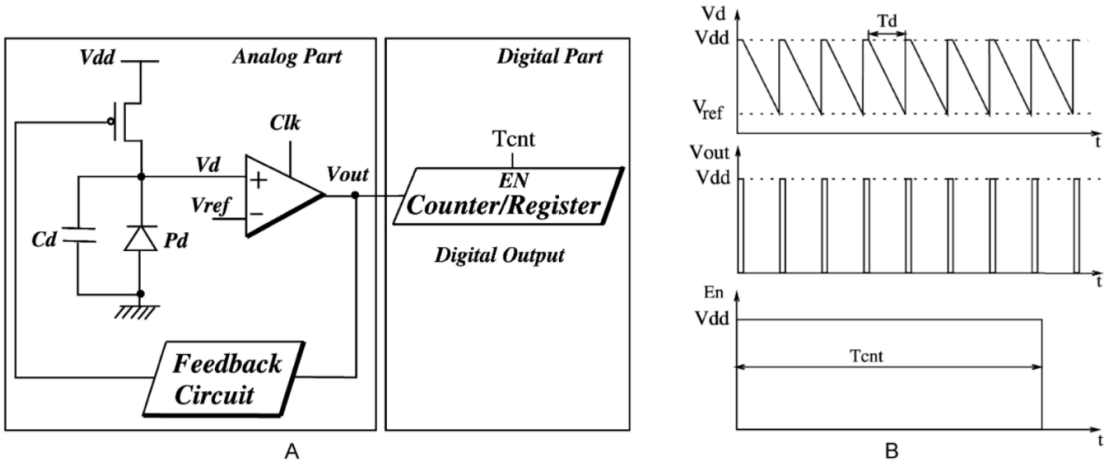


Figure 2.7: (A) Block diagram of the proposed pixel based ADC (B) Voltages of the different nodes of the circuit [18]

The frequency of spiking is given as:

$$f = \frac{i_d}{(V_{dd} - V_{ref}) \times C_d} \tag{2.4}$$

Where, i_d is the diode photo-current, and C_d is the photodiode capacitance. $T_d = \frac{1}{f}$ defines the integration time. This pixel architecture improved the fill factor issue when trying to insert the ADC in-pixel, while also keeping the advantages of low noise, low load capacitance and low ADC/TDC frequency, thanks to parallel operation. The only drawback of this pixel architecture, is the repeated spikes required to count data, and the frame dependant readout.

A study done by biologists proved that retinal encoding can be performed in the Time to First Spike (TFS) information rather than the frequency of the spikes [19], knowing this has led to the first implementations of a Time to First Spike pixel matrix with AER encoding [20], [21] and [22]. Most event-based image sensors relay on the Address Event Representation (AER) to communicate the spikes to the outside of the pixel matrix.

2.2.4 Address Event Representation (AER)

Event based image sensors are Neuromorphic chips, designed to mimic the biological structure of a neural network. The major challenge that arise in Neuromorphic circuit implementations, is the duplication of neuron connectivity. In silicon, a transistor or logic gate can only drive a few others, while a biological neuron can have up to 15 000 synapses (neuronal junction) [23]. This is impossible to directly reproduce on silicon. However, in silicon we can mitigate this limitation with high switching frequencies and multiplexing in time. Hence, the usage of AER encoding. This communication channel, is an efficient way of communicating spikes, between chips or circuits. Multiplexing leverages the five-decade difference in bandwidth between a neuron (hundreds of hertz) and a digital bus (tens of megahertz), enabling the replace of thousands of dedicated point-to-point connections with a handful of high-speed metal wires and thousands of switches (transistors) [24]. It is dedicated for inter-chip communications, like image sensor chips [25]. Figure 2.8, displays the basic structure on an AER communication. Here, sender address-encoder generates a unique binary address for each neuron whenever it spikes. A bus transmits these addresses to the receiving chip, where an address decoder selects the corresponding location [24].

By considering a single pixel as a neuron, a spike (event) generated from a pixel can be sent using the logic coordinates of the pixel in the matrix, using one AER interface for row spikes and a second for column events.

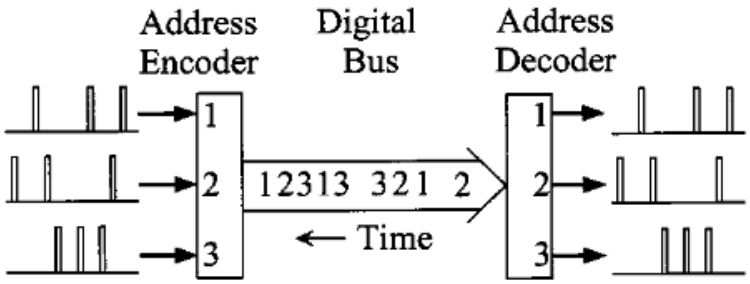


Figure 2.8: The AER pulses from spiking neurons are transmitted serially by broadcasting addresses on a digital bus. Multiplexing is transparent if the encoding, transmission, and decoding processes cycle in less than $\Delta = n.s$, where Δ is the desired spike-timing precision and n is the maximum number of neurons that are active during this time [24]

2.2.5 The Time to first spike (TFS) image sensor

As presented in subsection The spiking pixel, The TFS pixel uses a biologically proven method to communicate data, by only relying on the first spike. The first implementations of a Time to First Spike pixel matrix with AER communication occurred in [20] and [22]. In contrast to the spiking pixel of figure 2.7, the TFS pixel doesn't dispose of the feedback between the comparator output and the reset of the photodiode, hence this pixel only fires one time, until the photodiode is reset using a global synchronization signal or autonomously using an asynchronous digital logic, like in the TFS pixels of [20] and [22]. Here, the information that would help construct an image lies in the integration time, defined simply as the duration between the reset of the pixel and the spike generation:

$$Tf = \frac{(V_{dd} - V_{ref}) \times C_d}{i_d} \tag{2.5}$$

Time to digital conversion is done externally. On the level of the image sensor, the TFS pixel matrix is read asynchronously without the need for a global synchronization signal or the notion of a frame. Figure 2.9a illustrates the basic architecture of an event based IS such as the TFS IS. Here, the array of pixels is connected to row and column encoders, to reduce the amount of output data from m+n bits to $\log_2(m) + \log_2(n)$ bits. However, simply encoding pixel requests limits the image sensor output bandwidth to only one request at a time. This is not guaranteed since pixels under the same luminance will generate requests simultaneously, hence vent collision and the need for arbitration. As shown in figure 2.9a, an event based IS has to dispose of row and column arbiters, to mitigate the simultaneous multiple requests. The readout is done in the following order:

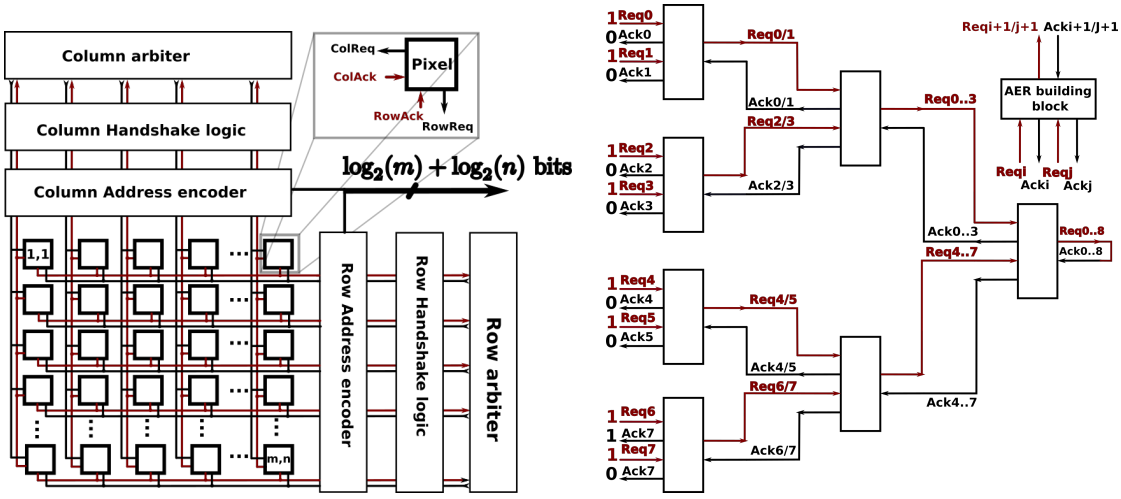
first the requests (RowReqi, RowReqi+1...) from multiple row are sent to the arbiter through the handshaking logic block, the row arbiter selects one row, the pixels on this row receive a RowAck signal from the handshaking logic block. The row arbiter enables the column arbiter to arbitrate and acknowledge one pixel using the ColAck signal, finally the selected pixel address is encoded and sent to the output. The active pixels (the generated a request) in the same row are acknowledged by the column arbiter, they are reset to start another integration cycle.

The arbiter tree internal circuitry (presented in figure 2.9b) is a binary tree made of arbitration blocks (the upper right of the figure), each arbitration block has three inputs and three outputs, two input requests and the input acknowledge received from the arbitration block in the upper stage, two output acknowledge signals to the stages bellow, and one output request to the the upper stage. A single arbitration block works in the following way: at the start two request signals are received, either from the lower arbitration stage or input signals, depending on the position of the block in the arbitration tree, then the arbiter decides randomly[4] to channel only one of the request signals to the upper stage. When in a tree, the input requests propagate through the tree upwards until the top block is reached, while losing half of the request signals in each stage, the final arbitration block has its output request and input acknowledge signals shorted, and the only acknowledge signal propagates from the top to the bottom of the tree, thereby generating the acknowledge address of the winning input request.

The image sensor readout using the arbitration mechanism simply chooses a row address and encodes it, and the column arbiter chooses one pixel from that row and encodes. This way, we get the address of the pixel with the request. If this process continues non stop like in [22], it can increase event collision(3.2.3) in the Imager since highly illuminated pixels will fire several times before a less illuminated pixel fires one time. This is prevented in [20], thanks to a global synchronization signal (global reset), here pixels can only fire one time max in a global reset period.

The TFS IS basic architecture explained here, has also been used for the readout of the DVS pixel in [15].

[4]Most arbitration circuits in event based sensors, rely on no priority arbitration blocks, to give equal access to all row or columns of the pixel matrix



(a) Event based IS basic architecture *(b) AER arbiter operating principle*

Figure 2.9: Event based IS basic architecture: many detail communication signals in figure (a) were omitted for the sake of simplicity

2.2.6 The Asynchronous Time Based Image Sensor (ATIS)

As discussed in subsections 2.2.2 and 2.2.5 The TFS pixel (subsection 2.2.5) measures light intensity, and the DVS pixel (subsection 2.2.2) measures light intensity change and slope. A paper in the art [26] presents a combination of these two pixels to suppress temporal redundancies. Since the TFS pixel will always communicate the same information (integration time) even if this information doesn't change. The idea presented in this paper is to use the DVS pixel as a change detector to trigger photo-integration of the TFS pixel. This means, whenever the DVS pixel detects light intensity change and slope, it will trigger the TFS pixel to measure the light intensity. Here, the change in light intensity is considered as the relevant information to capture, otherwise no integration is done and the previous information in the image memory is kept. This way, a significant part of data emitted from each pixel, and consequently the imager is reduced significantly.

In the ATIS circuit a pixel comprises two blocks, which are the change detector (DVS) and the second is the photo-integrator(TFS). This imager improves on the biggest flaw of a standard CIS, which is data redundancy in time.

Figure 2.10 bellow, displays the type of data each block inside the ATIS pixel generates: on the left is the output of the DVS based change detector, it detects the light intensity change slope, a decreasing intensity is a black pixel and an increasing intensity

is a white pixel on the resulting images, the background unchanging light intensity is in grey. In the middle, the output of the TFS based photo-measurement block, it measures light intensity when triggered by its corresponding change detector. In contrast to the left image, everywhere the change detector got triggered, a photo measurement was run, which yield an image of grey value pixels only in activity areas. Elsewhere the photo-measurement was not conducted is the inactive background in black. Finally, on the right is the resulting final image, which is the middle output image updating a background image. The background image was taken at the beginning of the capture, by triggering all the pixels simultaneously. This image will keep updating only the areas where a relevant information occurs, meaning an activity, which also translates into a light change.

The ATIS imager, thanks to its DVS based change detector manages to completely suppress temporal redundancy. The drawback of this IS compared to standard images sensors is the pixel complexity (89 transistors, 4 capacitors, 2 photodiodes) and area (30umx30um in the process: 0.18μm 1P6M MiM CMOS), and as a result, the fill factor is considerably small with 9% for the change detector and 14% for photo-measurement, since the two blocks have separate photodiodes. For comparison, standard image sensors have pixels with single digit square micrometer area, a fill factor that exceeds 70% and a circuit complexity of a few transistors.

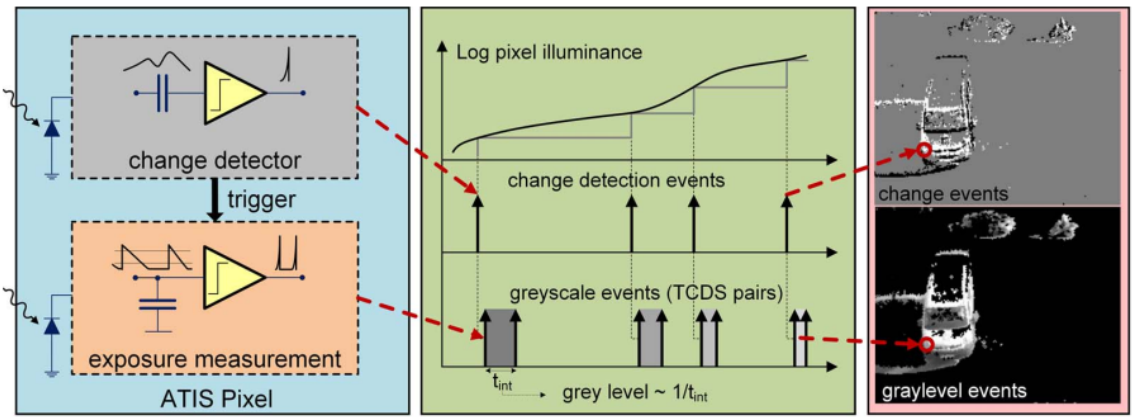


Figure 2.10: ATIS readout output [26], [13]

2.2.7 The DAVIS sensor

Another architecture that implements triggered capture is the DAVIS image sensor [27]. Its pixel is basically a DVS pixel, plus an Active Pixel Sensor (APS) mounted on top of

the logarithmic compressor transistor of the DVS, the two share the same photo-diode(see figure 2.11). The resulting pixel combines conventional frame-based sampling of intensity with asynchronous detection of log intensity changes. The goal of this architecture is to be able to capture the static data of the scene with APS readout(transitor MN1 to reset the voltage Vaps, MN2 is the source follower amplifier, MN3 is the column select and MN4 protects the drain of MN5 from voltage transients due to the reset of Vaps), and the area of action with intensity changes. This pixel has a 60% area reduction compared to the ATIS in the same process (0.18um). Even though, this image sensor uses frame based readout in its APS part, the sensor as a whole consumed 90% less power than the ATIS [27].

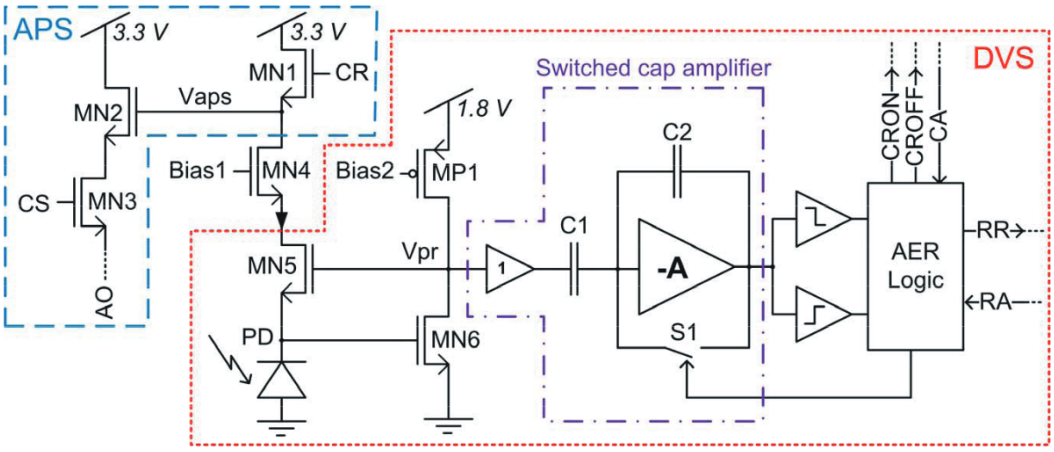


Figure 2.11: DAVIS: APS pixel schematic and the DVS block schematic [27].

This is mainly attributed to the reduction of circuitry in the DAVIS sensor, meaning that the ATIS circuitry dedicated to the photo-measurement (dozens of transistors) was substitute with only 4 transistors (APS part of the pixel).

2.2.8 Event-based image sensor challenges

Although the underlying principles behind event-driven image sensors are a breakthrough in imaging, since they improve upon many of the CIS limitations, such as data redundancy, dynamic range, linear photo-diode voltage limitation, they introduce their own limitations. The table 2.1 bellow, underlines event based image sensors limitations: pixel areas that are dozens of times bigger than a CIS, which is the result of the pixel complexity (dozens of transistors), and increases power consumption. The pixel complexity affects the fill factor, comparing 70% for CIS vs 20% for event based image sensors. Plus,

due to the nature (bio-inspired) of event based image sensors, they require bio-inspired communication channels, that can't be implemented on silicon, and these are replaced by AER communication protocol and arbitration circuits. A bottle neck problem arises from the latter, for instance the fastest DVS sensor implemented [28], can produce up to 333k events (3µs latency) per pixel at high intensity changes, which means that a 128x128 matrix can produce 5 Gevents per seconds, but the arbitration circuits around the matrix can only let 20Mevents per second. This, highlights the reasons why an event based image sensor is difficult to scale, while a CIS can theoretically operate at very high frame rates, needing only a high frequency clock signal and suitable ADC or ADCs per column. The following problems deter industrial semiconductor companies from adopting event based image sensor, among their plethora of solutions for high performance imaging applications. Also Table 2.1 highlights the staggering differences in performances between event based image sensor, that produce full gray level data-and not only temporal contrast like the DVS- and a standard CIS. We can clearly see that almost all metrics are in favor of standard CIS, expect the dynamic range.

	ATIS [26]	DAVIS [27]	Standard CIS [29]
CMOS technology	0.18um 1P6M MIM	0.18um 1P6M MIM	65nm 1P4M / 14nm1P8M
Array size	304 x 240	240 x 180	Dual-PD 12.2 Megapixels
Pixel area(μm^2)	30 x 30	18.5 x 18.5	1.4 x 1.4
Fill factor	20%,10%	22%	BSI
Supply voltage	3.3V analog, 1.8V digital	1.8V / 3.3V (pixels)	2.2 V/1 V
Power consumption(/pixel)	0.69-2.4 uW	0.17-0.31 uW	0.051 uW
Dynamic range	Intensity 125dB, DVS N.A.	120dB DVS, 57dB APS	65 dB

Table 2.1: Comparison of CIS with event based image sensors in the art

However, when it comes to temporal contrast vision, Prophesee (the company with the patent of the ATIS and the DVS) with Sony, released in 2020, the latest version of the DVS with outstanding perfomances that surpass all the previous implementations including the one from Samsung [30] in 2017, bringing this sensor closer to industry standards, with a pixel fill factor of more than 77%, a dynamic range higher than 124 dB, a high

definition resolution and an event rate of 1G events per second, three times the rate of the DVS from Samsung(table 2.2). Given these points, it is obvious that despite the complex design of event based image sensors, an industry is emerging around the idea of event based imaging, as a recognition of its utility and potential for machine vision applications. For consumer electronics standard CIS remain at a better position, with their Mega pixel resolutions.

	Samsung DVS [31]	Sony-Prophesee DVS [30]	Standard CIS [29]
CMOS technology	90 nm 1P5M BSI	90 nm CIS BSI	65nm 1P4M / 14nm1P8M
Array size	640x480	1280x720	Dual-PD 12.2 Megapixels
Pixel area(μm^2)	9 x 9	4.86 x 4.86	1.4 x 1.4
Fill factor	20%	77%	BSI[5]
Supply voltage	2.8V analog, 1.2V digital	2.5V / 1.1V (pixels)	2.2 V/1 V
Power consumption(/pixel)	0.088-0.16 uW	0.035-0.091 uW	0.051 uW
Dynamic range	80dB	124dB	65 dB

Table 2.2: Comparison of CIS with industry manufactured event based image sensors in the art

[5]Back Side Illumination, this will be presented in chapter 4, subsection 4.1.1.

2.2.9 Conclusion

Image sensors accumulated many improvements since their introduction. Standard image sensors throughout the years, reduced noise and increased performance, resolution and speed, but their major flaw was its frame based operation, that produced a lot of redundant data. Since the 90s, a particular attention was given to bio-inspired image sensors, by introducing the silicon retina and first time to spike based image sensors. Thereby, the concept of event-based image sensors, where pixels aren't read at a constant period of time, but every time they have a relevant information. These image sensors present the advantage of reducing the rate of operation of pixels, and redundancies suppression. However, they suffer from low fill-factor, are hard to scale, and have a limited event bandwidth due to arbitration bottleneck. The main goal of event based image sensors was to profit from modeling the biological retina, and obtain its power efficient operation. In the next chapter we try to tackle, event based image sensor issues, by proposing an event based image sensor architecture that improve upon some of the limitations presented in this chapter. Our goal is to design a low power consumption image sensor, that can the first step towards an image sensor that manages the compromise of performance, low power consumption and scalability.

3

High Level Image Sensor Architecture

We present our proposition of an event based image sensor architecture, that we evaluate through high level simulations

Sommaire

3.1 Introduction to the proposed architecture 31

3.2 High level architecture 32

3.2.1 Simulation setup and stimuli 33

3.2.2 Readout verifications and spatial redundancy suppression 35

3.2.3 Events collision 36

3.3 Simulation results 37

3.3.1 Events reduction performance 37

3.3.2 Verification simulation results 39

3.3.3 Image quality and readout performances 40

3.3.4 Conclusion 42

3.1 Introduction to the proposed architecture

In this section, We study how to enhance image sensor architecture in order to limit as much as possible the data throughput, by reducing data redundancy. We note, that there are two redundancy types in image sensor output data, spatial redundancy, refers to different pixels that keep the same grey level information at the same time, and temporal redundancy, refers to the pixels that keep the same grey level information at different times[1]. Therefore, we particularly focus on the pixel matrix, and study how changing the ratios between DVS and TFS pixels would result in a reduction in data redundancy. Inspired by the ATIS architecture, we use a DVS pixel for triggering a block of TFS pixels (against 1DVS for 1 TFS pixel in the ATIS). Several architectures implementing different pixel ratios are studied in order to quantify the drawbacks and the advantages brought by these image sensor architectures. Our work in this thesis build upon the architecture of the ATIS image sensor [26]. We will attempt to reduce the image sensor event rate even further by introducing pixel architecture improvements, solving the arbitration bottle neck and proposing a different approach to the readout. We conduct architecture evaluations through simulations. These also enable the extraction of sample videos and images, that would be generated, approximately, by these architectures if implemented. The proposed pixel architectures are evaluated through high level simulation, to determine their relative performance, like the event rate, the signal to noise ratio and the structural similarity index to measure the quality of images. Finally, a pixel architecture is chosen for implementation.

The image sensor we propose works as follows. Once the DVS pixel detects a luminance change in the scene, it activates a group of TFS pixels instead of only one. With such an approach, it is possible to define a set of kernels including a group of TFS pixels surrounding the DVS pixel (see Figure 3.1) that constitutes a pixel matrix. Depending on the targeted application and the wished behavior, it is possible to choose an appropriate kernel, which produces more or less activity. Sub-figure (a) in the figure 3.1 is the ATIS image sensor, that implements one luminance change detector pixel (DVS) with one luminance measurement pixel (TFS), kernels (b), (c), (d) and (e) show different im-

[1] In standard CIS, spatial redundancy, are spatially different pixels with the same grey levels in the same frame, temporal redundancy are pixels that keep the same grey level in two different successive frames.

plementations with 3, 5, 8 and 24 TFS pixels surrounding one DVS pixel/measurement block (a group of TFS pixels). It is important to keep in mind here, that both the DVS and TFS pixels have separate photodiodes. In kernel (f), we have a DVS pixel embedded in the middle of four TFS pixels, and takes as input, the average photo-current of the four surrounding TFS pixels. Sub-figures (g) and (h) show what a pixel matrix based on kernels (a) and (b) would look like.

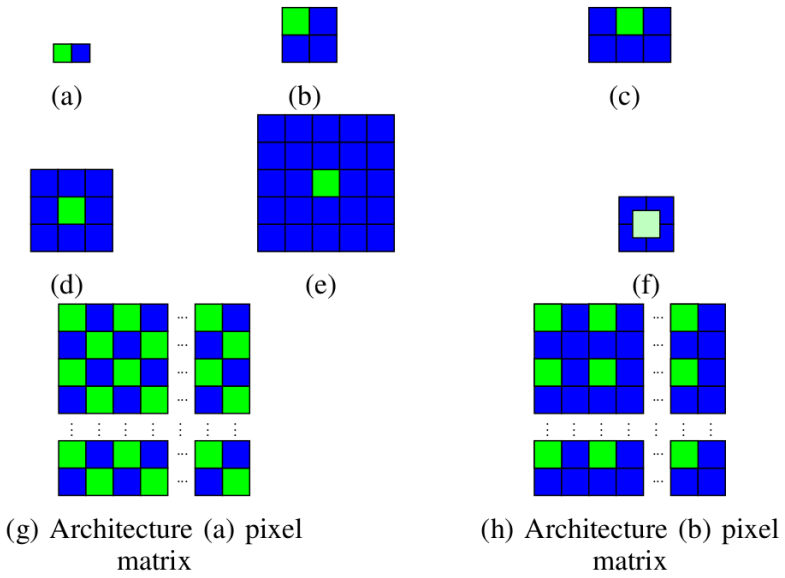


Figure 3.1: Some kernels made of TFS and DVS pixels (DVS pixel colored in green and the TFS pixel colored in blue)

Intuitively, we can see that in the kernels above, the less TFS pixels around one DVS pixel, the less data. The ratio of DVS to TFS pixels decides the maximal possible activity of the pixel matrix for the same stimuli. For example, kernel (b) will generate 50% more activity than kernel (a) for the same resolution. However, using kernels with less TFS pixels greatly affect the image quality. All these concerns are evaluated in the next section.

3.2 High level architecture

The models of each pixel were designed in Matlab and assembled to form models of the kernels in figure 3.1. Each kernel model represents its pixel matrix, and receives input from a video. Each pixel data from the video is converted into a photo-current value to be

inserted in the TFS and DVS pixel calculation formulas.

Figure 3.2: The high level simulation main steps

3.2.1 Simulation setup and stimuli

To evaluate the behavior of each kernel pixel matrix to different kinds of stimuli, we used two test cases. The first one is a video of a car driven on a road (High activity scene 3.6a). The second one is a parked car with almost no activity, except a person walking in front of the parked car for a few seconds(Low activity scene 3.6b).

The simulation input is a video. The modeled kernel is applied to every frame of the video, hence generating the output of a pixel matrix formed by the kernel in question. For each scenario, the videos last for 10 seconds at a frame rate of 30 fps and 1200×600 pixels/frame. Thus, we have a total stimulus of 2.16×10^8events, which is also the output of a standard CMOS image sensor, since it reads full frames and applies no redundancies suppression. Each kernel undergoes 6 simulations represented in figure 3.3c, 2 scenarios $\times$ 3 DVS threshold cases, we use multiple DVS thresholds to evaluate their effect on the data output of the architectures. The analog characteristics 3.3d of the photodiode used in the calculations during the simulation, applies for both DVS and TFS pixels.

Both TFS and DVS pixels receive input as a current, as the video frames are converted to logarithmic photo-current. For this simulation we set up the TFS pixel crossing threshold at 2.3 Volts(Vref), for calculations convenience since subtracting 2.3 from our 3.3 Volts Vdd is 1 volts, to avoid floating point calculations, this threshold only decides how fast a TFS pixel will integrate, and not if it will integrate or not, therefore it value is not important for our simualtion. In our TFS pixel model, the integration time that encodes the luminance value is calculated as the following:

$$Tf = \frac{(V_{dd} - V_{ref}) \times C_d}{i_d} \tag{3.1}$$

With Cd, being the photodiode capacitance a,d Id the input photo-current. For the

DVS threshold, it is defined in the equation bellow as the minimum logarithmic photo-current change capable of triggering an event, devided by the maximum dynamic of the logarithmic photo-current. (when the DVS threshold increases, less events are generated and vice versa):

$$Threshold(\%) = \mid \frac{\Delta \log(I_{ph})_{th}}{\Delta \log(I_{ph})_{max}} \mid \tag{3.2}$$

with $\Delta \log(I_{ph})_{th}$ being the threshold of the logarithmic photocurrent, that will trigger a change and $\Delta \log(I_{ph})_{max}$ the maximal range of the logarithmic photocurrent. For example if the measured logrithmic current is $\mid \Delta \log(I_{ph})_{mes} \mid > \mid \Delta \log(I_{ph})_{th} \mid$ then a trigger event can be generated.

(a) Road test case

(b) Parking test case

Scenario	Dimensions	Frame rate	DVS thresholds
Highway	1200×600	30 fps	1.5,5,15 %
Parking	1200×600	30 fps	1.5,5,15 %

(c) Simulation parameters

C_{ph}	Iph_{max}	Iph_{min}
10 fF	100pA	100fA

(d) Photodiode analog characteristics

Figure 3.3: Illustration of the two test cases and the simulation parameters

The photodiode characteristics in sub-figure3.3d, were obtained from the thesis of Camille Dupoiron [32], at the CEA-LETI laboratory. Here, many photodiode were implemented in the 28 nm technology from STMicroelectronics, with different areas frm

0.5x0.5 μm^2 to 5x5 μm^2. The characteristic in the table 3.3d were adopted from the characterisation of a 3x3 μm^2 N-Well/Psub photodiode.

3.2.2 Readout verifications and spatial redundancy suppression

For an event-based image sensor, AER addresses are the norm and address collision need to be avoided. Therefore, making the readout arbiter-dependant or arbiter-less is an important factor in the image sensor performance, speed, latency and power consumption. For our image sensor, we decided to use an arbiter-less readout based on the work of a former Phd student in our team [33]. This readout was implemented for an image sensor that uses a full TFS pixel matrix, meaning complete absence of trigger pixels. It is also capable of suppressing spatial redundancies[2]. This readout is presented in chapter 5, along with the sensor.

The TFS pixel cannot suppress spatial redundancies intrinsically like the DVS pixel does with the temporal redundancies. Therefore, introducing a dedicated processing circuit is mandatory. The mentioned readout above enables this by clustering integration times that are relativity close to each other. For every received event, with integration time $Tint_i$ that verifies:

$$Tint_i \leq Tint_{ref} + \Delta t \tag{3.3}$$

$Tint_i = Tint_{ref}$, with $Tint_{ref}$ being the integration time of the first event. This way different pixels (X_i , Y_i) in the resulting image will have the same gray level value corresponding to the TDC encoding of $Tint_{ref}$.

For example in figure 3.4, event e_1,e_2,e_3 with integration times $Tint_1$,$Tint_2$,$Tint_3$, can all be suppressed into $Tint_1$, since $Tint_2 \leq Tint_1 + \Delta t$ and $Tint_3 \leq Tint_1 + \Delta t$. Consequently, increasing or decreasing this duration Δt, results in a higher or lower spatial redundancy suppression. This is a fundamental parameter of the readout.

[2]These are the pixels that have the same luminance and the same time stamp when using a global pixel reset

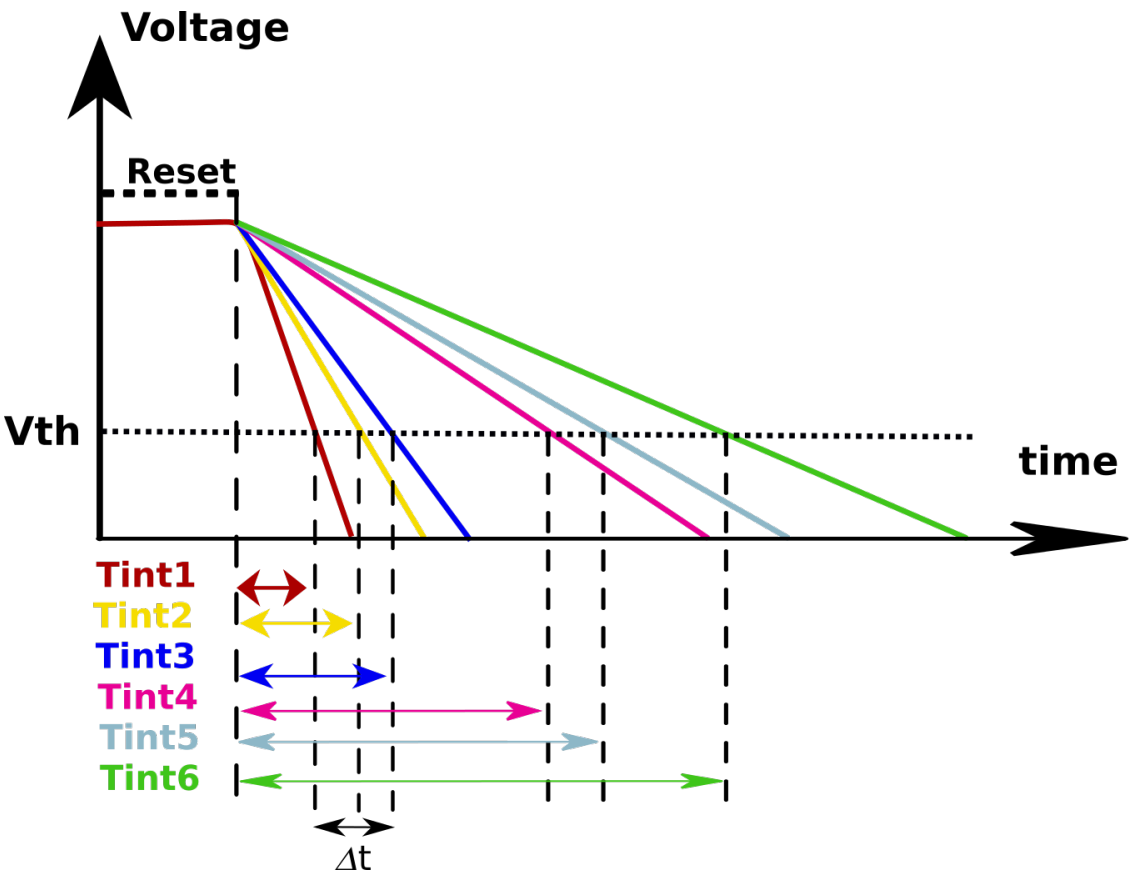


Figure 3.4: Spatial redundancy suppression principle in [33]

3.2.3 Events collision

Using an arbiter-less readout means that event collisions are common, and need to be dealt with. To illustrate this collision problem, consider figure 3.5 bellow, in which we have a 4 by 4 pixel matrix composed of event-based pixels, a TFS for instance, and lets assume that pixels in the coordinates: (X1; Y1) and (X3; Y3) trigger events, as a result, the output row AER address is 1010 and the output column AER address is 1010, these two addresses also communicate the information that pixels (X1; Y3) and (X3; Y1) (colored in yellow) triggered too, which is eventually a false positive. To resolve this, the readout in [33] has a verification circuit that polls all the suspected four pixels to know which have events and which don't, our readout presented in chapter 5, incorporates this verification process in a processing circuit 5.1.1.

In order to have an idea of the processing time overhead required to run these necessary verifications, for an arbiter-less image sensor like ours, when reading the pixel matrix, we run verification simulation for one of the architecture we presented in 3.1, and

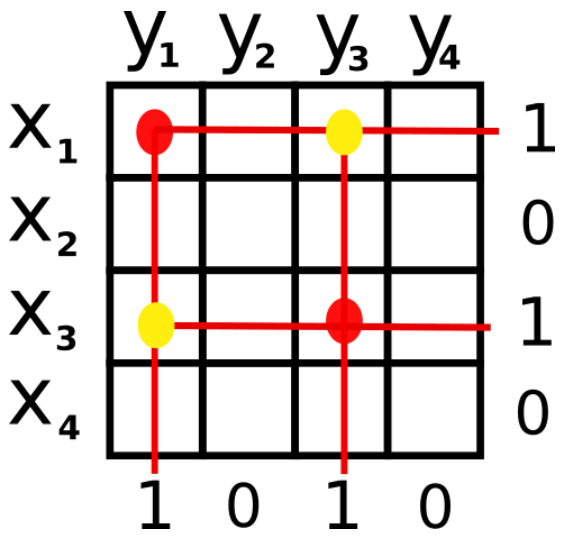


Figure 3.5: Illustration of the verification problem

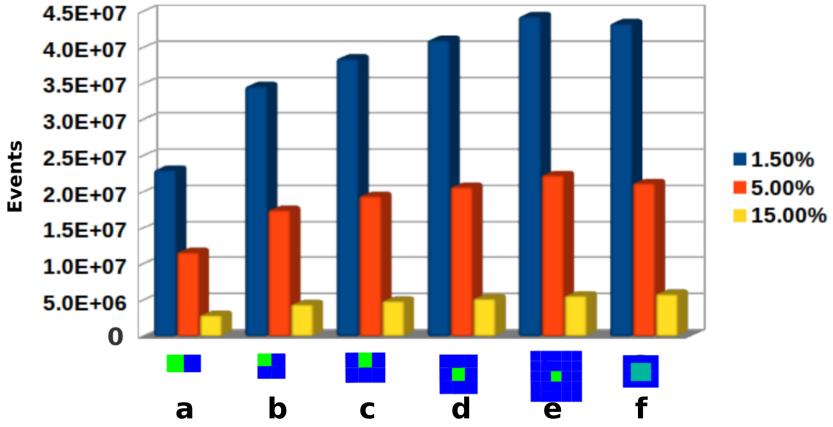
the results are given in 3.3.2. In the simulation, the number of verifications is recorded, meaning the number of pixels that were verified. Thereby, an event is a verified pixel that has actually integrated and not the result of a false address generated by a collision. It is also interesting to see how introducing spatial redundancy suppression affects the number of verifications and events. This is possible since in our Matlab model of the readout system we can vary the parameter Δt.

3.3 Simulation results

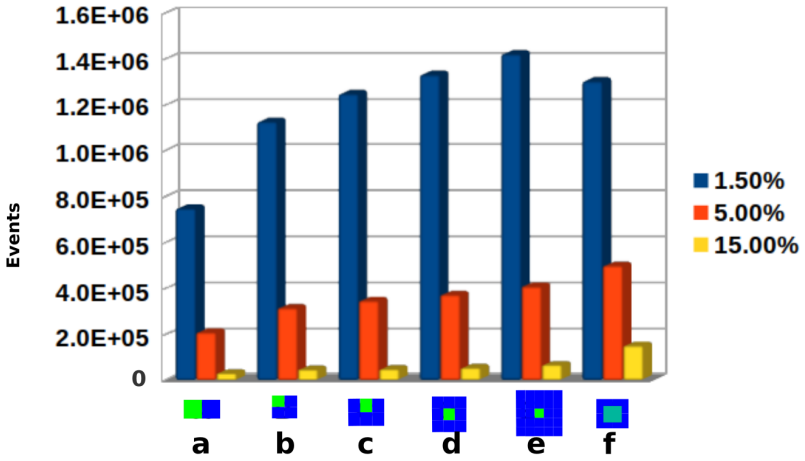
Our different architectures showed a pattern in results, with a clear compromise between the event rate and the image quality. The evolution of the number of required verification is also interesting, these results are detailed in the subsections bellow.

3.3.1 Events reduction performance

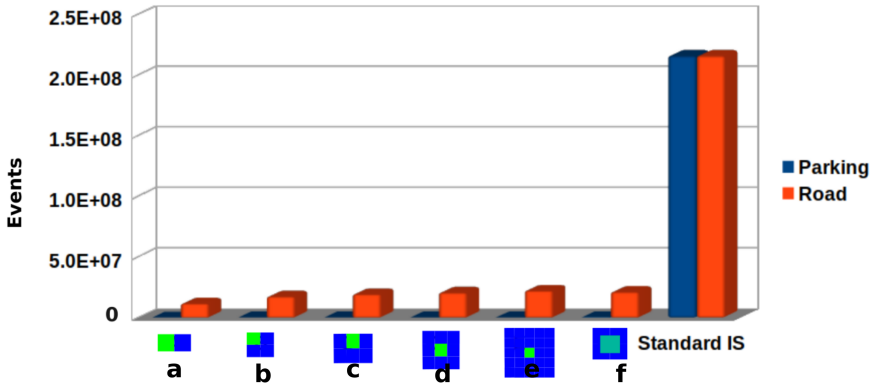
Figures 3.6a and 3.6b confirm at a first look that the event output of all architectures is indeed, proportional to the scenes activity. As expected, reducing DVS threshold results in more DVS pixels detecting luminance change and triggering more TFS pixels, hence more events are detected. Also, from the two figures, we notice that the more TFS pixels surrounding a DVS pixel, the more the generated events.



(a) Road test case



(b) Parking test case



(c) Comparison between our architectures (DVS threshold 5 %) and a standard CMOS in the two test cases

Figure 3.6: The total number of generated events through the whole duration of stimuli: (a) and (b) represent the total number of events generated in each test case per DVS threshold, (c) is a comparison of the different architectures with a standard image sensor (at 5% DVS threshold)

Finally, all of the hybrid architectures generate far less events than a standard TFS (100% TFS pixels) image sensor produce for the same input, at least 70 % less events in the Road test case and 96 % less events in the parking test case, as figure 3.6c demonstrates. Architecture wise, we notice that kernel (f) outperforms (in terms of generated events, less generated events the better) kernel (e) in the road test case despite having a full TFS pixels resolution, also it outperforms kernels (e) and (d) in the parking test case only at 1.5% DVS threshold, this can be attributed to the low pass property of averaging circuits, since we can clearly see at lower DVS thresholds and parking test case (both give a lower activity), kernel (f) doesn't outperform any other kernel, since the averaging low pass allows more events.

3.3.2 Verification simulation results

Figure 3.7a, shows how much verifications required for uncertain events, and most importantly how verifications scale with the pixel matrix resolution. Based on this figure, in the case of no spatial redundancy suppression applied, verifications represent 98% of the total data exchange between the pixel matrix and the readout system for a pixel matrix resolution of 600 × 1200 pixels, and 97.7% for 480 × 840 (wide VGA) pixels, and 97.3% for 360 × 630 pixels, and events only make up the rest. This also means that 97% of the processing time will be spent trying to locate correct events in the pixel matrix, in the worst case, meaning high activity, like the road test case. Consequently, suggesting an event-based image sensor architecture with AER communication without arbiters, should account for this verification overhead. Figure 3.7b [3], highlights how effective Spatial Redundancy Suppression (SRS) helps at reducing the verification overhead, Tmax is the longest integration time possible. Increasing Δt results in a higher compression and less verifications(this was the result of RTL simulations of our readout 5.1, in chapter 5), Δt is always defined as a divider of the longest integration time.

[3]The goal of this figure was to only highlight the effect of Spatial Redundancy Suppression (SRS) has on the number of verifications. Thus, we did not repeat this simulation for all the architectures, as they all behave similarly

(a) The number of events and verifications vs the resolution of the pixel matrix

(b) SRS effect, Kernel (f) based image sensor vs Full TFS image sensor

Figure 3.7: Verification scaling with the resolution, and Spatial redundancy suppression effect on verification

3.3.3 Image quality and readout performances

Directly comparing the quality of the images generated by each kernel in our context is unfair and not appropriate. For instance, since kernel f generates a full resolution image while the other kernels generate images with blind pixels corresponding to the positions of the DVS pixels, the comparison should take care of this particularity. This can be solved by a post processing filling the blind pixels. In order to give an idea, kernel (f) displayed significant performances for an image sensor: a Structural Similarity Index (SSIM) of 0.72 (ideal SSIM is 1) and a Peak Signal to Noise Ratio (PSNR) of 20.24 dB, the other kernels should present similar results if their resulting images were post processed.

According to the readout system protocol, the data generated by the pixel matrix are sent to update the image memory. At the initial instant of the simulation, all photo-diodes generate a spike of photo-current due to the sudden change in the scene luminance from darkness at $t_0 - \delta t$ to viewing the input scene at $t_0 + \delta t$, as a result almost all DVS pixels trigger the TFS pixels, which also integrate a full image of the scene. Consequently, the image memory is filled. Afterwards, the image memory is constantly updated with new data from the triggered TFS pixels in the pixel matrix. Hence, the resulting output of the image sensor is a frameless image constantly updating its pixels. Figure 3.8 illustrates the resulting image in the image memory and the flow of new events updating it.

(a) Original scene

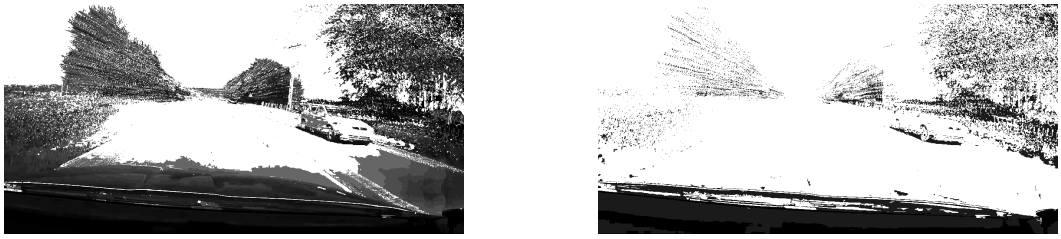
(b) The resulting image

(c) The freshly detected events

Figure 3.8: (a) The viewed scene (b) The resulting image generated by the pixel matrix of kernel (f) at 1.5% DVS threshold and no spatial redundancies suppression. (c) The flow of the new events updating the image memory

Figure 3.8, also illustrates sample images of the videos generated by our Matlab models. They clearly exhibit the desired behavior we want i.e. a low throughput image sensor. We only retrieve the relevant gray level information, meaning the area of interest, which is in our case the area of activity in a scene. Figure 3.9 demonstrates the processing effect of the generated TFS events through our readout system model. This system performs spatial redundancy suppression on the already temporally suppressed data at the output of the sensor. The two figures display the effect of proportionally increasing Δt to the maximal integration time detectable. The result of increasing Δt is a higher compression and a lower throughput and vice versa. Also, a reminder that increasing the DVS threshold

increases data compression (temporal redundancy suppression) at the sensor level, before the compression at the readout level (spatial redundancy suppression), by increasing the Δt.



(a) $\Delta t = \frac{Tint_{max}}{16}$ (b) $\Delta t = \frac{Tint_{max}}{2}$

Figure 3.9: Effect of applying spatial redundancy suppression, kernel (f)

In the figure above 3.9, we visually observe that increasing the Dt results in a higher compression, from subfigure (a) to (b), we can see a decrease in grey levels. white corresponding to the earliest events and black to the slowest events or lack of events in that area.

3.3.4 Conclusion

The proposed image sensor architectures make the event rate proportional to the scene activity and generate a completely compressed bitstream keeping most of the original scene efficiently, especially for kernel (f), without requiring any post processing. These architectures introduce the highest form of data compression compared to the state of the art since it applies both temporal(at the pixel level) and spatial redundancy suppression (at the readout level) at the same time in different stages, and no paper in the art talks about integrating these two forms of redundancy suppression together. The behavior that kernel (f) displayed in the two test cases was interesting, since the number of generated events in all other kernels, only increased with more TFS pixel in each kernel. Only kernel (f) improved significantly on the other kernels when the activity in the scene is high. An image sensor architecture based on kernel (f) will be adopted and designed, along with its closest architecture, kernel(b). Indeed, both kernels have 4 photodiodes, which is valuable for a future comparison.

4

Pixel Design and Architecture

In this chapter, we cover the pixel design steps in the FDSOI 28 nm technology from STMicroelectronics. We first start by presenting the FDSOI, its advantages and challenges when implementing image sensors. Then, we get into the design of each pixel and establish the trigger connection between the two kind of pixels. Finally, a power consumption assessment of our designs is conducted at the end, through post layout simulations.

Sommaire

4.1 The 28 nm FDSOI technology 47

4.1.1 Image sensor technologies 48

4.2 Pixel Design 49

4.2.1 DVS 49

4.2.2 TFS 50

4.3 DVS and TFS pixel combination design 53

4.3.1 Averaging pixel architecture 53

4.3.2 the averaging circuit 54

4.3.3 The DVS pixel part 55

4.3.4 The 4 TFS pixels 56

4.3.5 The trigger 58

4.4 Circuit design and power estimation 59

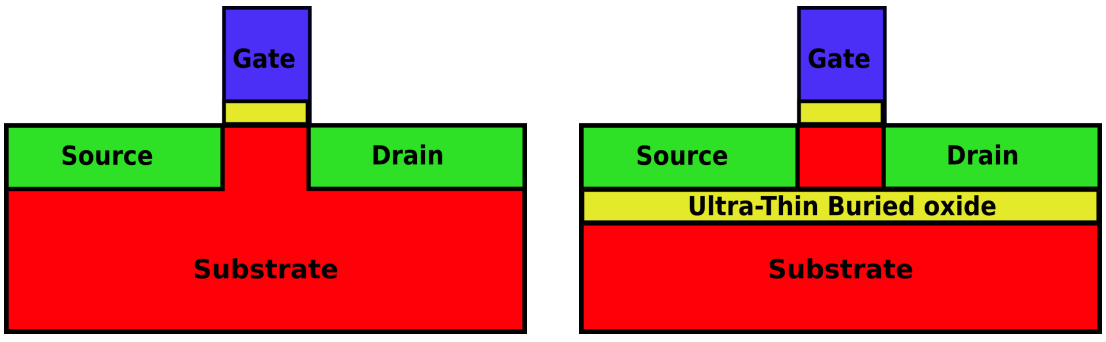
4.4.1 Pixels and circuits design 59

4.4.2 Image Sensor Power Estimation 60

4.5 Conclusion 63

4.1 The 28 nm FDSOI technology

Fully Depleted Silicon of Insulator (FDSOI) is a technology dedicated to reduce the short channel effects that emerge in bulk technologies when reducing the technological node. Figure 4.1 displays a cross section comparison between bulk and FDSOI MOS transistors. Here, we can see that FDSOI involves the insertion of an ultra thin buried oxide layer, referred as the BOX. This layer separates the bulk from the channel of the transistor. For low power circuit design, FDSOI is a good choice according to many studies. For example, Self-heating and its impact on analogue performance were studied in 28 nm technology bulk and FDSOI devices. FDSOI outperforms bulk in a wide frequency range. While thermal effects are stronger in FDSOI, their influences on device parameters are limited [34]. Moreover, simulation results indicate that FDSOI can help for below-nominal supply voltage, which is suitable for Ultra-Low Power (ULP) systems [35]. However, In this technology the BOX layer in a MOS transistor especially imposes few design challenges for image sensors when implementing photodiodes. The BOX layer adds many design rules for the designers, which increase the design time. For example, one way of implementing the photodiode in an FDSOI technology requires adding specific layers on top of the photodiode area to etch the BOX layer and achieve a high well capacity, since the presence of the BOX makes shallow the photodiode well and able to only collect few charges. A photodiode can also be co-integrated in the substrate of a fully depleted silicon on insulator transistor [36].



(a) Bulk CMOS transistor *(b) FDSOI CMOS transistor*

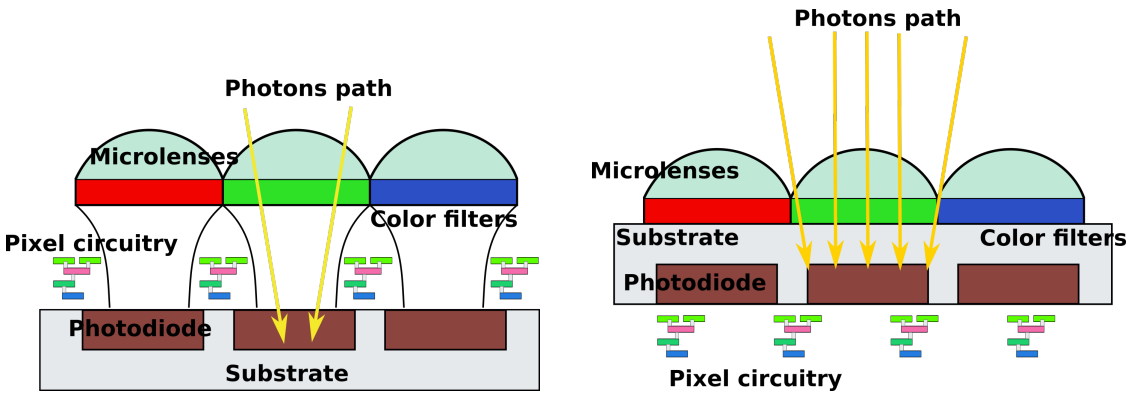
Figure 4.1: Cross-section comparison of bulk and FDOSI MOS transistors

In general, when designing an image sensor, it is required to maximise the photodiode area compared to the pixel circuitry (fill factor), to have low noise and high dynamic

range. To attain this, many technologies offer specific design kits.

4.1.1 Image sensor technologies

Before the development of efficient image sensor technologies, most were implemented in what we call Front Side Illumination(figure 4.2a) technologies (FSI), here the photo-diode is simply put between the circuitry at the same level, leaving light with a narrow path to travel through the metal levels of the circuitry, the more area the circuitry takes, the less light that gets to the photo-diode, this has prevented pixels from achieving a good fill factor. A good image sensor design in front side illumination, required maximising photo-diode area and compacting the circuitry area. To overcome this limitation, Back side illumination (BSI) technologies were introduced, it simply is the implementation of the photo-diode on a different level to the circuitry, the circuitry of the photo-diode remains in its place and now has more area, while the photo-diode is put at the back of the substrate along with its lens and color filter 4.2b. Since no circuitry is around the pixel, we can consider the fill factor to be ideal.



(a) Front side illumination *(b) Back side illumination*

Figure 4.2: Front and back side illumination technology comparison

In STMicroelectronics 28 nm FDSOI technology, no BSI option is available for the design of our image sensor.

4.2 Pixel Design

The two pixels TFS and DVS are the elementary blocks for the kernels presented in the previous chapter. Here, the operating principles of each pixel are presented. The DVS pixel is very similar to the one in [15]. However the TFS pixel was altered and is more complex than the integrate and fire pixel in the state of art. Our TFS pixel requires locally storing the request in order to deal with the verification process and avoid collisions, which are due to the arbiter-less readout.

4.2.1 DVS

(a) DVS block schematic *(b) DVS pixel analog simulation*

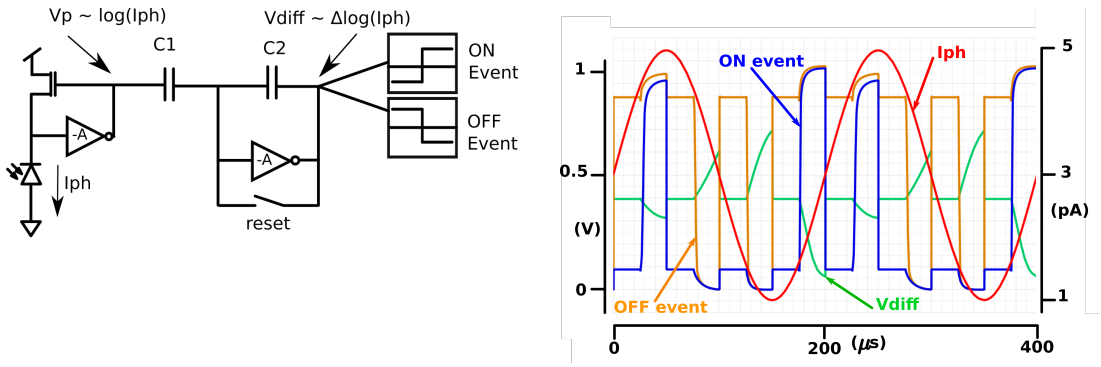


Figure 4.3: DVS pixel

The Dynamic Vision Sensor pixel is an event-based pixel. It is sensitive to the luminance change. Its block schematic is presented in Figure 4.3a. This pixel uses a switched capacitor differentiator circuit to compute the voltage difference between two successive samples on the photo-detector. This difference is compared afterwards to two thresholds to generate an ON or an OFF event according to the polarity of the slope of the photodiode voltage V_p. A positive change yields to an ON Event and a negative change to an OFF event. Figure 4.3b presents the signals operating in the pixel. In red, a photodiode current generated with a sinusoidal current source is taken as a simple stimulus. It emulates a variation of the pixel luminance. The luminance slope changes are detected in the pixel thanks to the differentiator circuit, which translates to V_{diff} spikes (Green). These latter go into two comparators, to generate ON (blue) or OFF (Orange) events. The reset signal is used after the detection of an event to clear the request, and bring V_{diff} back to its reset

level (this level corresponds to the horizontal green line, from wish V_{diff} spikes up and down), before the next detection. The full circuit schematic of the pixel will be presented in the sequel.

4.2.2 TFS

This pixel operates as a 1-level crossing sampling circuit. When the photodiode voltage V_{ph} (figure 4.4) crosses a predefined threshold, an event is generated. The pixel event is the logical coordinates address of the pixel in the pixel matrix. This is known as the Address Event Representation (AER), as defined in the second chapter 2.2.4. In order to translate this event into a gray level information, an external Time-to-Digital Converter (TDC) is used. The TDC generates the instant at which it received the event coded in a bit vector depending on the image resolution (this part of the readout circuit will be presented in the next chapter). In order to activate and communicate an event, a pixel exchanges the following signals with an external readout system (see 4.4 and Fig. 4.5):

- The pixel starts with the signals *localReset*, *LocalResetDVS* and *ON_TFS* inactive.
- Once the DVS pixel trigger is active using the signal *ON_TFS* (goes to 0), the TFS pixel notifies the external readout of the start of the photo-integration, by driving the two signals *RowStart* and *ColStart*.
- Once the photodiode voltage level crosses an externally defined threshold, the comparator drives the *TFSComp* signal to go down (active), and the latter drives the *RowReq* and *ColReq* signals down while at the same time, it saves a local request through the signal *ReqLoc*.
- The readout system (RS) detects the request signals on the row and column buses, and sends the acknowledge signals *RowAck* and *ColAck* acknowledging the pixel requests.
- After the pixel receives the acknowledgement signals, it deactivates the request signals *RowReq* and *ColReq*.
- Afterwards, the readout system proceeds to the verification process, by setting the *RowVerif* and *ColVerif* signals. The pixel receives these latter signals and checks

its local request memory. If the *ReqLoc* signal is up, the response signal *PxPos* pulls down to indicate the presence of a request. If instead there is no local request, the pixel is still integrating and the signal *PxOn* goes up to state the ongoing integration.

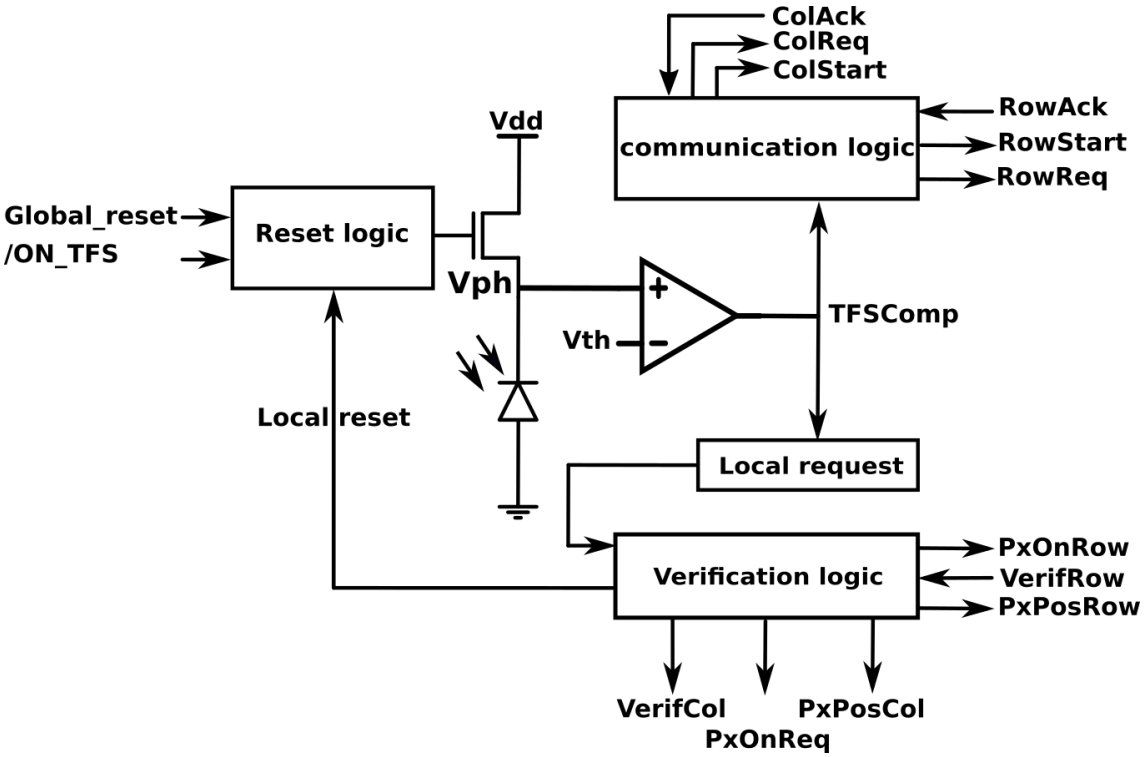


Figure 4.4: TFS pixel block schematic

Figure 4.6 illustrates an example of AER address recovery. In the figure, both pixel in red (2,3) and (3,2) generate AER events through the signal buses *RowReq* and *ColReq*, which write at the output of the AER interfaces RowReq(0,...,1,1,0) and ColReq(0,1,1,...,0). However, when the readout system receives the AER address, the address can be interprated as pixels (3,2),(3,3),(2,2) and (2,3) have all together fired. To mitigate this, the readout system checks the local request of each pixel amongst the four, to verify if they truly fired or not. The verification process is mandatory since, our AER request communication is used without any arbitration, which is the strength of our approach. Using arbiters introduces many issues, such as a limited event bandwidth, a reduced scalability and inaccurate luminance measurements. The verification process solves the AER address conflicts. Even if the verification is a sequential process, this approach is particularly interesting because it suppresses the risk of time measurement errors due to the arbitration tree. Indeed, this latter delays the requests inducing an inaccurate time stamping, espe-

cially with a large matrix array. Notice that the verification sequence is fast compared to the photodiode integration time, For example in our readout simulation in the next chapter, the maximal integration time was 32 ms, meaning that for 256 grey levels the minimum integration time is 125 us, hence the verification process takes no longer than 125 us at worst. Thus, the verification speed is not an issue for such image sensors.

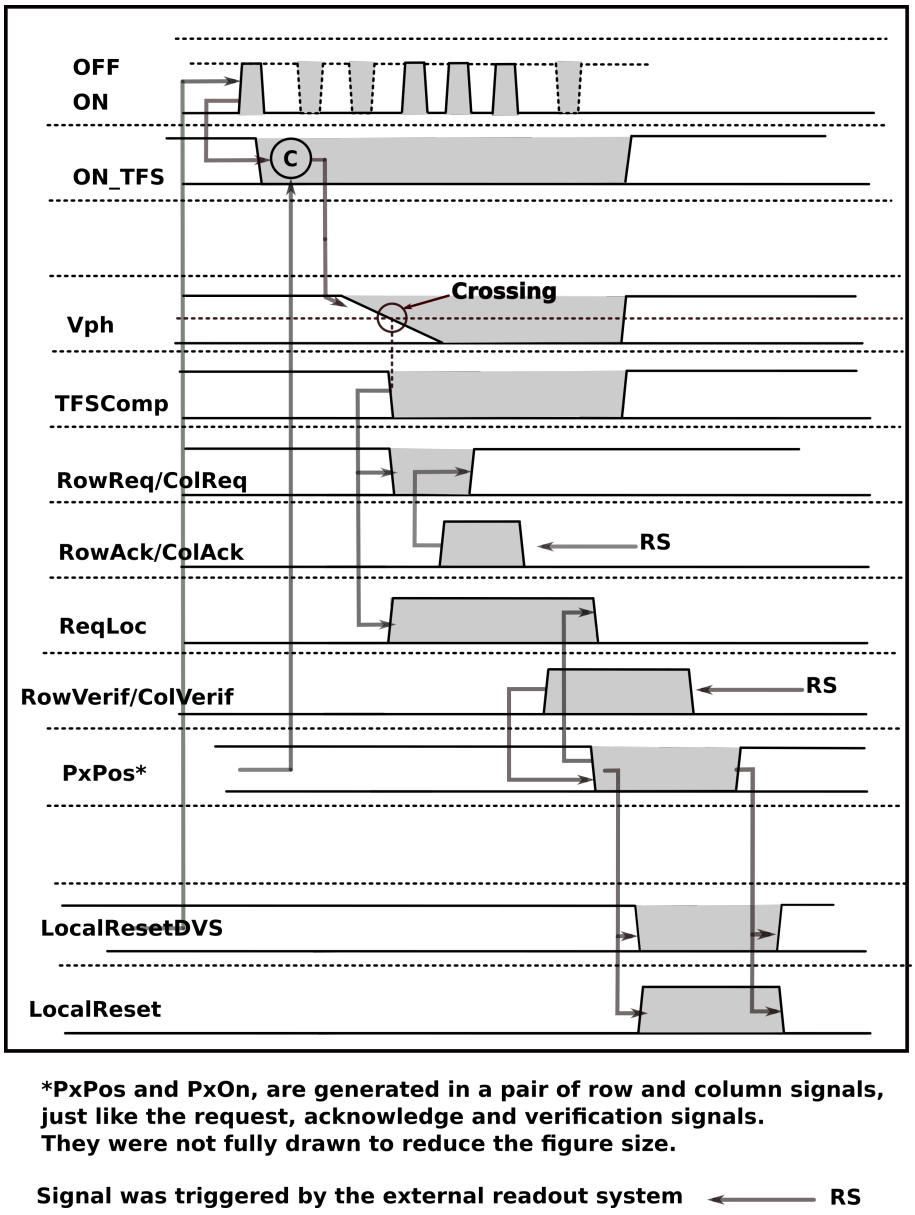


Figure 4.5: TFS pixel communication signals

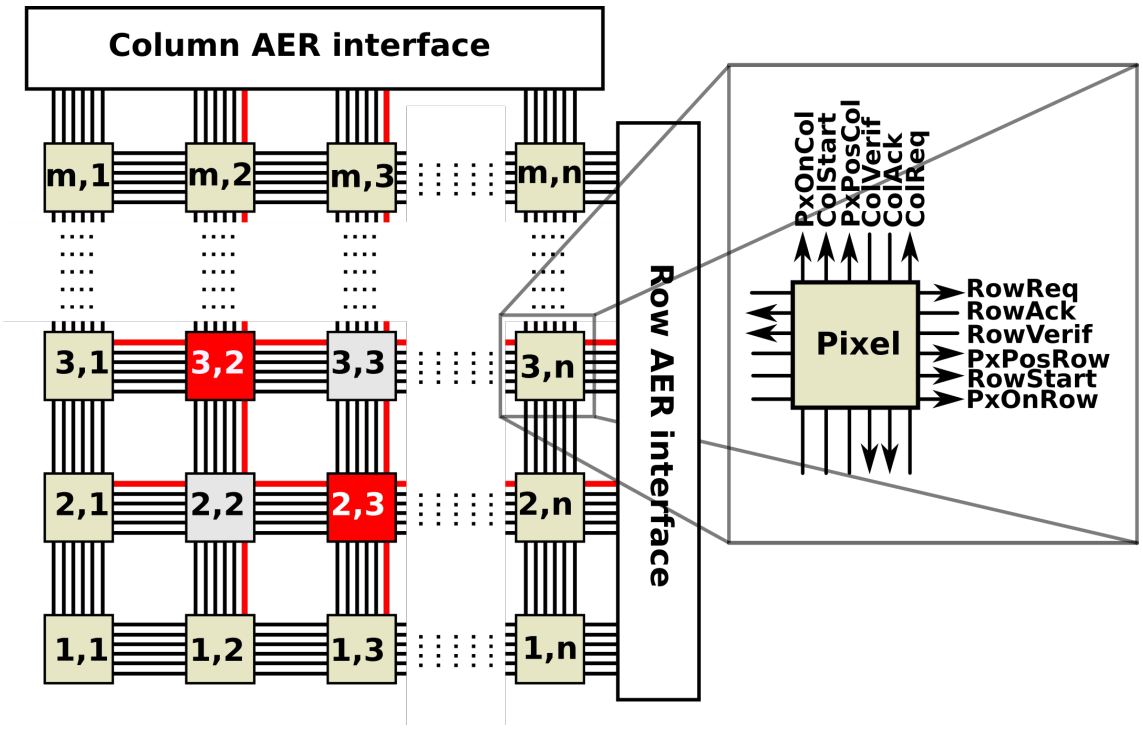


Figure 4.6: Pixel matrix view illustrating the need for verification

4.3 DVS and TFS pixel combination design

In this Section, we present the combination of the two pixels to form the kernels (f) and (b), that have been presented in the previous chapter. Only Kernel (f) is presented because as kernel (b) is similar to kernel(f) but without the averaging circuit. The two kenels are refered to as averaging (kernel (f)) and non-averaging (kernel (b)) kernels. The circuitry connecting the two pixels, the averaging circuit and the pixel components is detailled in this section.

4.3.1 Averaging pixel architecture

The Averaging pixel architecture includes a DVS pixel carefully placed and routed in the middle of four TFS pixels. Its particularity is that this DVS pixel has no photodiode. This latter has been advantageously replaced by the average photo-current of the 4 surrounding TFS photodiodes. Figure 4.7 displays the block schematic of the circuit. The circuit contains 4 TFS pixels with their photodiodes, a DVS pixel and asynchronous logic circuitry. Each component is detailed in the following separate subsections.

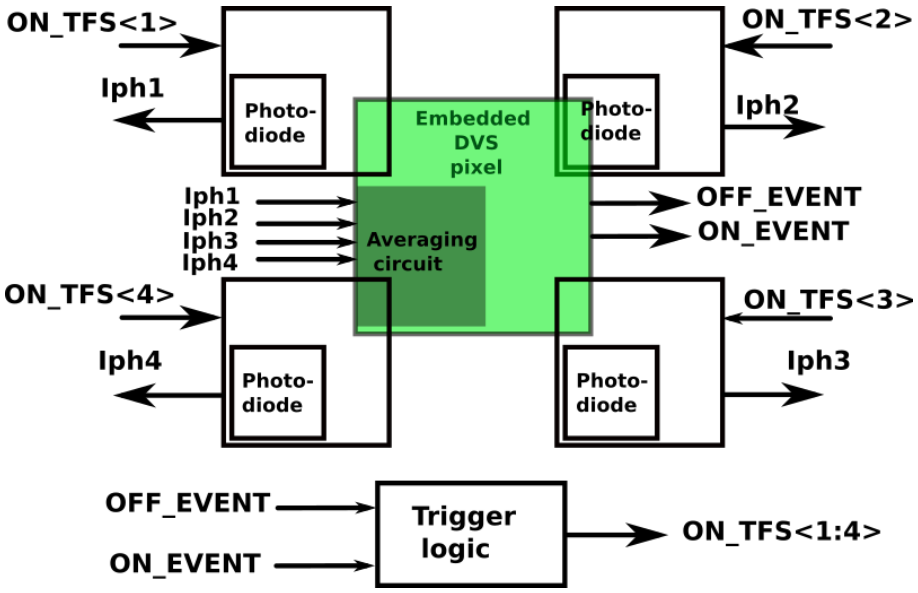


Figure 4.7: Abstracted full circuit schematic

4.3.2 the averaging circuit

The averaged photo-currents are copied using 4 current-mirrors on each TFS pixel photodiode. The sum of the currents is then fed to the DVS pixel, thanks to a final current-mirror circuit. All current-mirrors Widths are designed to ensure that photo-current sum is multiplied by a gain α of 1/4.

$$I_{dvs} = \alpha \sum_{n=1}^{4} I_{ph_n} \tag{4.1}$$

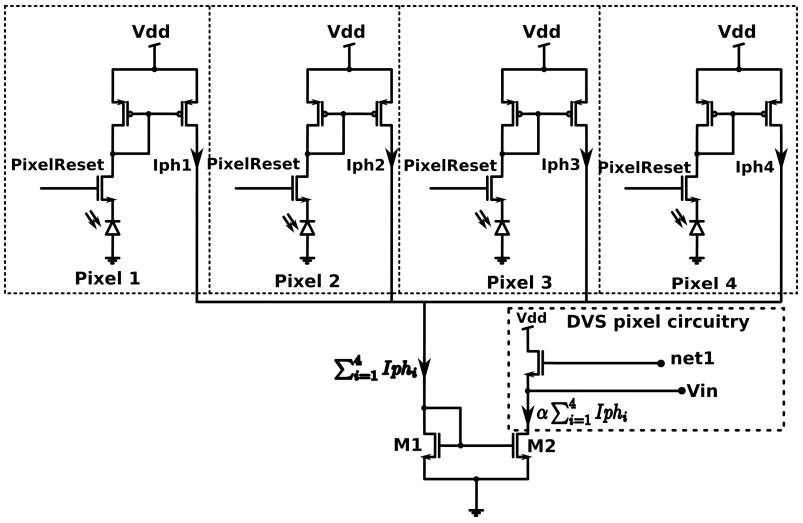


Figure 4.8: Averaging circuit(net1 and Vin link to the DVS pixel)

Figure 4.8 shows how the current mirrors are linked to the photodiode of the pixels. In each TFS pixel, the current-mirrors are PMOS transistors, in order to be inserted on top of the reset transistors. On the other hand, the DVS pixel receives the sum of the photo-currents using an NMOS current-mirror replacing its "own" photodiode.

4.3.3 The DVS pixel part

The circuitry of the pixel is exactly the same pixel as in [37]. Figure 4.9 below presents the full schematic of the pixel.

- Transistors M1 and M2 constitute the last stage of the current mirror, form the averaging circuit from the past sub-section.

- M4, M5 and M6 form an inverting amplifier(withing the logarithmic photo receptor) with external polarization Vpol2 for testing purposes, which feeds back to the logarithmic compressor M3.

- M7 and M8 form a source follower isolating the previous stages of the circuit from the switching of the next stage (differentiator circuit).

- Transistors M9, M10, M11 and capacitors C1 and C2 constitute the switch capacitor differentiator circuit performing the luminance change detection.

- The ON and OFF comparators composed of M12, M13 and M14, M15 output the ON and OFF events, by comparing Vdiff against the global thresholds *Voff* and *Von*.

In the pixel, the amplification gain is given by the capacitor ratio C_1/C_2. At the differentiator output, the luminance is expressed as follows:

$$Vdiff \sim \frac{C_1}{C_2}.\Delta(\log(I_{ph})) \tag{4.2}$$

Note that the gain introduced prior to the logarithmic compressor stage is cancelled by the differentiator circuit.

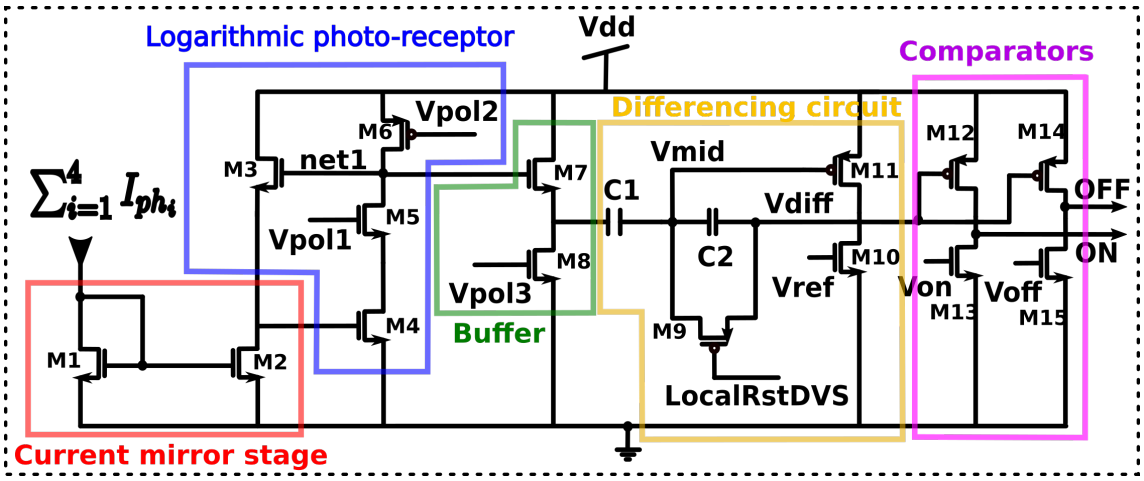


Figure 4.9: DVS pixel schematic

4.3.4 The 4 TFS pixels

The schematic of figure 4.10 is divided into three sections. The first one is the analog part of the pixel, the second is the digital part, and the last represents the bus connections.
In the first part:

- The first transistor M1 resets the photodiode whenever it receives the *PixelReset* signal from the digital part of the pixel.
- Transistors M2 and M3 constitute the current-mirror copying the photo-current from the photodiode to the averaging circuit.
- M4, M5, M6 and M7 constitute the comparator circuit. It detects the threshold crossing *VComplvl* after the photo-integration starts (once started, the photodiode voltage decreases).
- M8, M9, M10, M11, M12 control the reading and writing of the local request memory, through the *CompOut*, acknowledgement and verification signals.
- M13, M14, M15 and M16 compose the local request memory, which is no more than two feedback inverters.
- Mp is used to bias the comparator through an external voltage Vpol.
- Mr resets the memory when a reset signal is emitted by the digital pixel part.

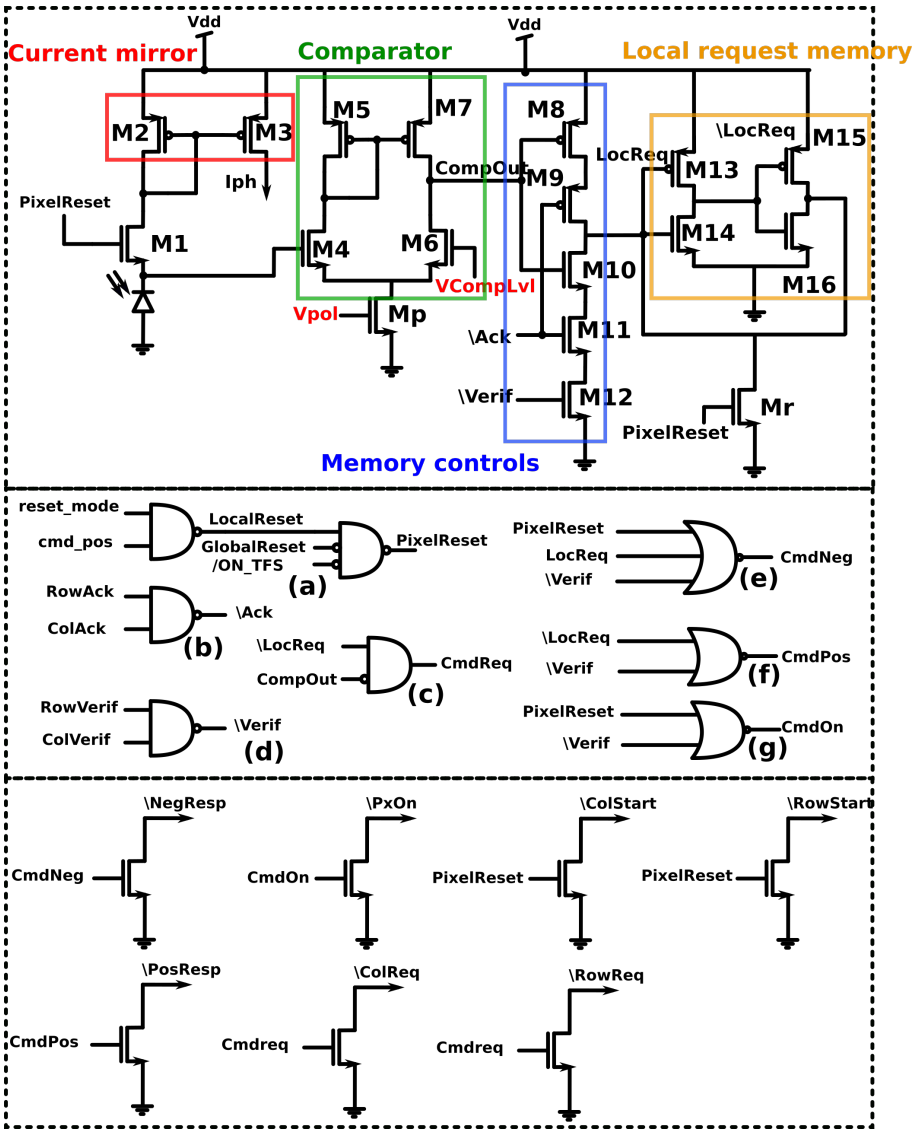


Figure 4.10: TFS pixel schematic

The second part of figure 4.10 is the digital circuitry of the pixel. The NAND gates (b) and (d) synchronize both column and row signals, for verification and acknowledge, by selecting the pixel. The logic gates (c),(e), (f) and (g) send the command signals (*CmdReq*,*CmdNeg*, *CmdPos* and *CmdOn*) to the bus transistors, in order to pull-down the adequate buses. The last 2-gate circuit (a) combines all the signals triggering the *PixelReset* signal. *ResetMode* selects the operating mode of the pixel, either a local reset mode where the pixel resets itself using the *LocalReset* signal when it fires, or a global reset mode, where all the pixels are externally reset through the *GlobalReset* signal. The *ON_TFS* signal removes the *PixelReset* when a request is coming from the DVS pixel to start the photo-integration.

Finally, the third part of Figure 4.10 shows all the TFS pixel NMOS transistors pulling down the buses when a pixel needs to send a request to the readout system. At the bottom of every pixel matrix row and column, there is a PMOS transistor for each bus. This latter pulls up the the buses when there is no pixel request in the rows and columns.

4.3.5 The trigger

This circuit (see Fig.4.11) connecting the DVS pixel and the 4 TFS pixels includes 4 C-elements, one for each TFS pixel (a). This asynchronous logic circuit memorizes the DVS trigger signal, means that when en event is detected by the DVS pixel, this circuitry allows TFS pixels to start their photo-integration. The trigger also comprise a circuit that controls the DVS pixel reset (b). Sub-figure (c) details the signals sequence of a trigger:

- The circuit starts operating when both *LocalResetDVS* and *ResetDVS* (the Reset-DVS is the global reset signal for all DVS pixels in the pixel matrix) are inactive (high). Once the DVS detects an event, ON or OFF signals start to output events, then the signal *ON_TFSi* becomes low through the C element, and all four TFS pixel starts photo-integration.

- The first TFS pixel to successfully integrate and save an event is verified by the readout system. Once verified, the signal *CmdPos* goes up to confirm the presence of a local request. Following this, the signal *DVSLocalReset* is pulled down and the TFS *LocalReset* goes up, so that both the DVS and the firing pixel are reset.

- The 3 other TFS pixels continue integrating until they fire. Then they follow the same process described above.

By switching the *Reset_mode* (view the logic gates in subfigure 4.10.a), the pixels work in global reset mode. They follow the same trigger and integration process, except that the TFS pixels are prevented from auto-reseting, and wait for the global synchronization signal *GlobalReset* to start another cycle.

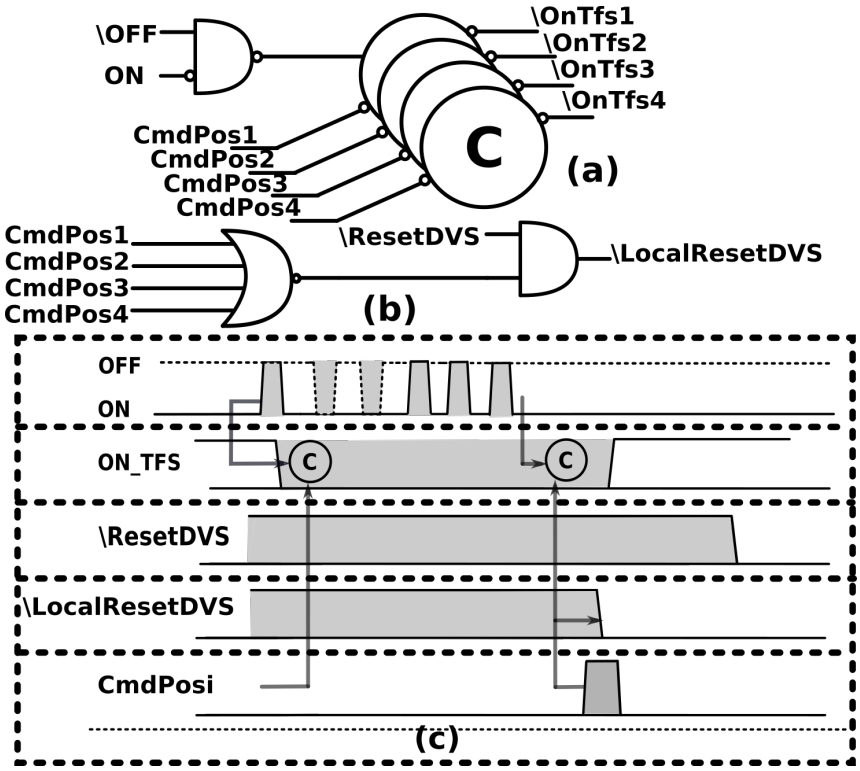


Figure 4.11: Trigger logic circuit and signals diagram

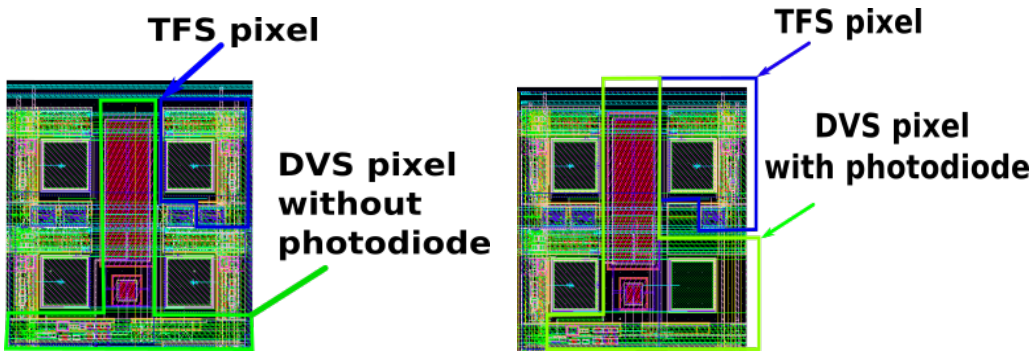
4.4 Circuit design and power estimation

The design of the pixel architectures described in the previous sections have been done in 28 nm FDSOI technology from STMicroelectronics (for both the averaging and non-averaging architectures). Here, we present the physical implementation of our pixel architectures, and a high level simulation of the implementation to approximately deduce the power consumption of our pixel architecture, since the pixel power measurement is quite complex to extract from the fabricated testchip.

4.4.1 Pixels and circuits design

In order to validate the advantages of the average trigger pixel architecture, we designed the non-averaging architecture from the past chapter(kernel (b)). In the non averaging architecture only 1 DVS pixel and 3 TFS pixels are routable in the same area of the averaging architecture. Figure 4.12 displays the layouts of both averaging and non-averaging

architectures. We can see that on the left, the averaging architecture enables the insertion of one more TFS pixel, compared to the non-averaging architecture, which increases the fill factor (table 4.1) for the averaging architecture. Looking at table 4.1, the photodiode area of the whole kernel is the some of all the photodiode areas it contains, hence, the kernel fill factor is the ratio of the photodiode area in kernel on the whole kernel area.



(a) Averaging architecture physical layout *(b) Non averaging architecture physical layout*

Figure 4.12: The two designed pixel matrices

	Averaging	Non averaging
Photo-diode area in the kernel(μm^2)	3x3x4	3x3x3
Kernel Area(μm^2)	16.5x15	16.5x15
Fill factor of the kernel (%)	14.5	10.4

Tableau 4.1: Averaging and non averaging physical implementation comparison

In the sequel, we will refer to the averaging pixel architecture as (Av) and to the non averaging architecture as (NAv).

4.4.2 Image Sensor Power Estimation

One of the event based image sensor characteristics during its operation is the event rate. It is defined as the total number of events generated by the image sensor or the pixel per second. From our simulation results in 3.6, we deduced the average event rate of a TFS pixel in each of the two architectures, this will be useful to compute the average power

consumption of each architecture next. By simply dividing the total event rates in 3.6 of each architecture by the resolution, and by the duration of stimuli, we obtain the following numbers in the table 4.13:

Test case	Road		Parking		Both cases
Architecture	Nav	Av	Nav	Av	Full TFS
Events/pixel/second	6.4	6	0.2	0.18	29

Figure 4.13: The average event rate per pixel per second of the TFS pixel in each architecture

We would like to remind that, we also added the full TFS pixels architecture for the sake of comparison, with an event based image sensor architecture that doesn't use any trigger. In the case of the Non-averaging and averaging architectures, the event rates presented in the table are the same for the DVS pixels in both architectures, since for each triggered TFS pixel, there was a trigger from a DVS pixel. Now the next step, was to go back to our post layout simulation and try to record the power consumption for each pixel type, in each test case. This was achieved by stimulating the pixels in the post layout simulation at the event rates in the table above. Therefore, for the TFS pixel in each test case and architecture, we record the power consumed by the pixel in the post layout simulation based on the event rate described in (table 4.14). Similarly for the DVS pixel, the average power consumption per pixel in each architecture and test case is in the table 4.15, note that in a full TFS pixel matrix architecture there are no DVS pixels. In this table 4.15, and unlike the TFS pixel, the post layout simulation results for the DVS pixel, revealed that most power consumed at low event rates is due to static power consumption, since the DVS pixels are larger, contain two capacitors and have more circuitry.

Test case	Road		Parking		Both cases
Architecture	Nav	Av	Nav	Av	Full TFS
Power Consumption / pixel(uw)	7	6.8	5.2	5	12.8

Figure 4.14: The average power consumption consumed by the TFS pixel in each architecture and test case

Test case	Road		Parking	
Architecture	Nav	Av	Nav	Av
Power Consumption / pixel(uw)	11	11	10.9	10.9

Figure 4.15: The average power consumption consumed by the DVS pixel in each architecture and test case

Estimating the pixel matrix power consumption is now made possible through the tables presented above. To estimate the power consumption of the two pixel matrices we multiply the average power consumption of both DVS and TFS pixels by their numbers in the pixel matrix in each architecture. Table 4.2 summarizes the power consumption of the pixels and pixel matrices. Analysing these results concludes the following, even though the Av architecture is more complex in terms of circuitry (1 DVS pixel + 4 TFS pixels + an averaging block and a trigger) compared to the NAv architecture (that only contains 1 DVS pixel + 3 TFS pixels and the trigger), especially having 1 more TFS pixel (33% more circuitry), the table 4.2 displays that the Av pixel architecture only increases its power consumption by 18 % in the high activity test case, and by 14 % in the low activity test case. We can also notice that both our triggered architecture consume around 37 % and 50 % less power than a full TFS pixel matrix sensor with no trigger, in high and low activity test cases. Furthermore, in the high level matlab simulation, the Av architecture has a PSNR (Peak Signal-to-Noise Ratio) of 20 dB, while the NAv pixel architecture has a PSNR of 11 dB. This can be mostly attributed to the difference of the photo-sensitive area and to the averaging.

	Nav		Av		Full TFS	
	Road	Parking	Road	Parking	Road	Parking
DVS (μ W)	11	11	10.9	10.9	–	–
TFS (μ W)	7	5.2	6.8	5	12.9	12.9
Total pixel matrix power consumption (mW)	33	27	39	31	53	53

Tableau 4.2: Post layout power consumption results

4.5 Conclusion

The presented hybrid event-based architecture is based on the combination of two event-based pixels (TFS and DVS). This architecture has proven limiting the firing rate of the pixels, since it only measures luminance when there is a relevant information in the scene. The Averaging architecture enhances the fill factor of the pixel matrix because the full photo-sensitive area is used for measuring luminance compared to the Non-averaging pixel architecture, which only uses 3 of the 4 photodiodes for luminance acquisition. Despite the averaging architecture having one more TFS pixel (33% more circuitry), it only increased its power consumption by 18% at high activity, and only increases its power consumption by 14% in low activity. This is due to the reduced number of events per pixel produced by this architecture. Both triggered architectures have 37% and 50% less power consumption than a non triggered TFS based image sensor architecture. The image quality is clearly increased because there is 33% more photodiodes in the averaging matrix for the same area with no geometrical distortion, thanks to a regular padding.

Two image sensors, one with the hybrid averaging architecture and the other one with the non-averaging architecture have been designed in FDSOI 28 nm from STMicroelectrtonics. The testchip has been fabricated and has been integrated in a camera test board for measurements.

5

The asynchronous readout design:

Here, we present the architecture of the image sensor readout, its components and how it connects to the image sensor

Sommaire

5.0.1 The Readout System 67

5.1 The readout system components 69

5.1.1 The Processing Element 69

5.1.2 The Time-to-Digital Converter 71

5.1.3 Global reset block 72

5.2 Readout system simulation 72

5.2.1 Simulation results 74

5.3 Conclusion 78

In this chapter, we present the architecture of our camera. The camera displayed in figure 5.1 is composed of a 28 nm microchip containing two pixel matrices based on kernels (b) and (f). The matrices have multiplexed inputs and outputs so that only one is active at a time. Outside of the microchip, we implemented our readout system in an FPGA for flexibility and testing purposes. The readout system contains a matrix of processing elements, which concurrently process pixel blocks in the matrix and a TDC to timestamp the arrival instants of the events from the AER interfaces. Thanks to the request timestamping, the system generates the data encoding the pixel luminance. These data are stored in a built-in memory embedded in the processing elements. Thus, each processing element stores a small part of the global image. Finally, the readout system includes a reset block for resetting the pixel matrix. The FPGA will embed VGA controller in order to display the images on a VGA monitor. Therefore, the FPGA processes in real time the data stream.

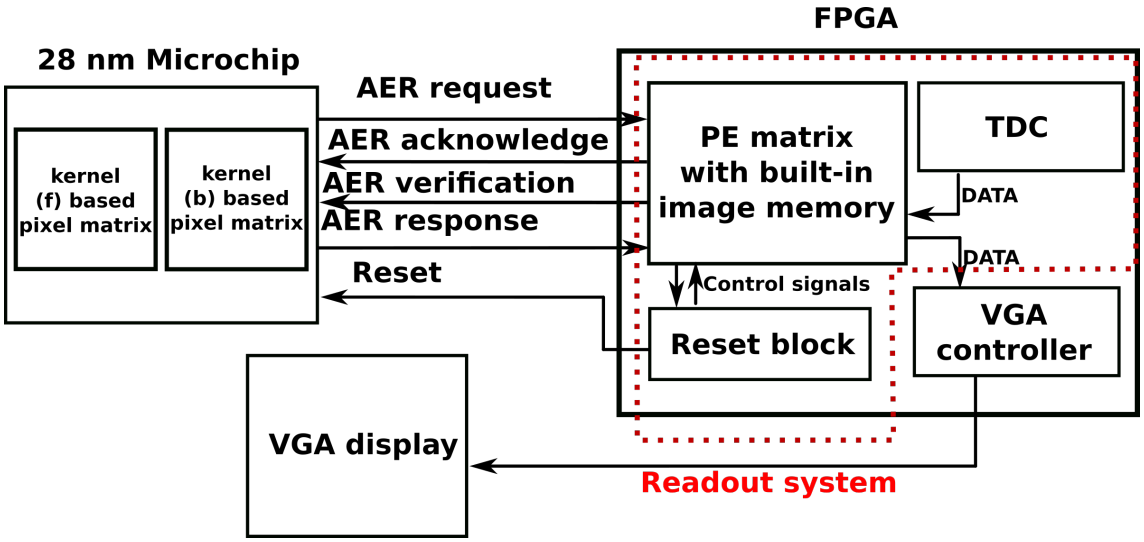


Figure 5.1: The event based camera block diagram

5.0.1 The Readout System

In order to understand how the images are extracted from the event-based image sensor and how they can be displayed, the details of the readout system (RS) are shown in Figure 5.2. The RS components and the signals they exchange are shown to ease the understanding. The processing elements (PE) are connected to the row and column AER interfaces, while the TDC is connected to both the PEs and the AER memory interfaces.

The memory AER interfaces access the PE local memory by selecting its position in the PE matrix (A PE with its connections is shown on the right side of the Figure 5.2). Each PE reads a small block of the pixel matrix. This allows to concurrently access the data or to serialize the reading according to the designer choice. For instance, reading the AER events produced by a matrix of *MxN = 1024x1024* pixels can be done by reading 1024 blocks of 32x32 pixels in parallel thanks to a *IxJ=32x32* matrix of processing elements. A PE handles in parallel the communication signals with the *IxJ=32x32* block of pixels. Working with small blocks also limits the drive on the pixel signals and make easier the image sensor scalability. Each PE communicates sub-AER addresses corresponding to I rows and J columns. The PEs share the row and column buses where the request, acknowledgement, verification and memory signals are embed. Notice that the input data bus from the PE matrix to the TDC is shared between all the PEs as well as the control signal sig_wait, frame_rst_enable and IS_reset_done. The PE, the TDC and the reset block are detailed in the sequel.

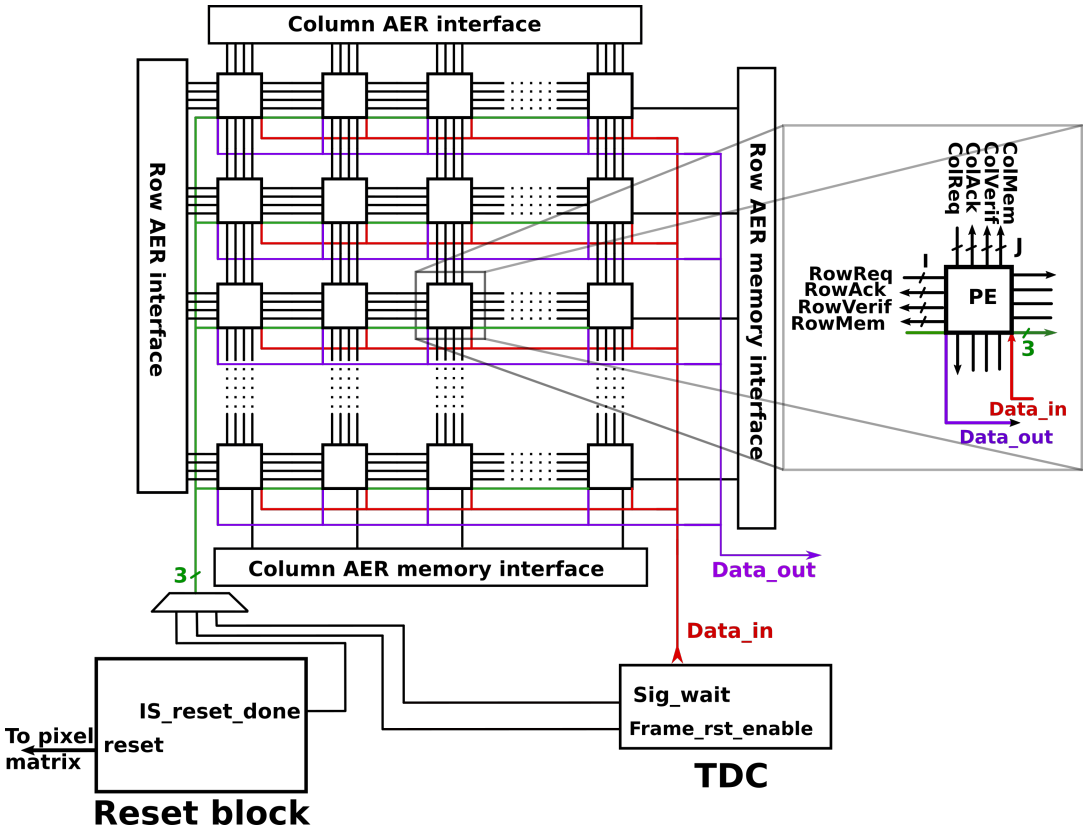


Figure 5.2: Block schematic of the readout system

5.1 The readout system components

The PE, the TDC and the reset block are detailed in the sequel.

5.1.1 The Processing Element

The PE also has inner components that are detailed in figure 5.3. The PE has a finite state machine sequencing the transactions between its internal components (AER reader, Local memory, LIFO memories) and the external block of pixels.

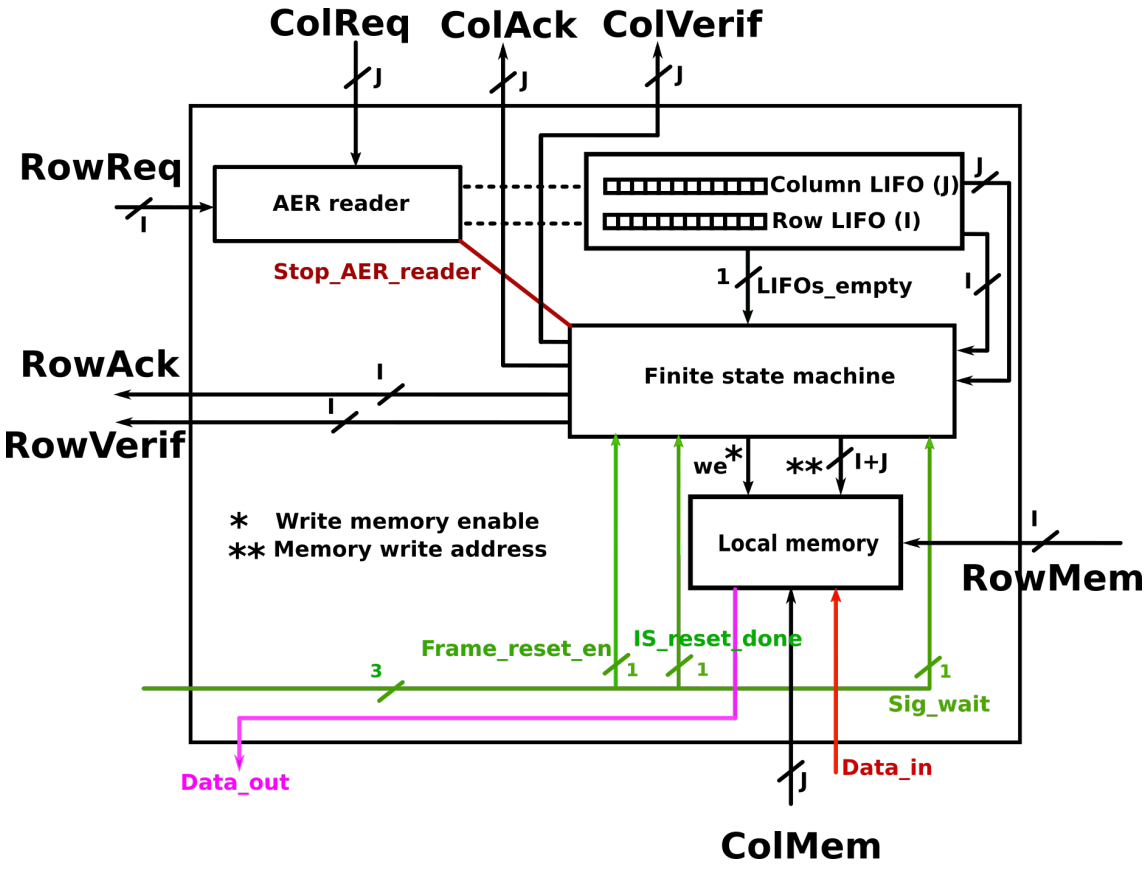


Figure 5.3: Block schematic of the processing element

The AER Reader

The AER reader circuit collects the events with the AER interfaces and transmits the AER addresses to the LIFO memories. The requests can be directly sampled but in our testchip, due to a limited number of pads, we decided to sequentially extract the data. The circuit is then an address counter sequentially checking the events on the request rows and columns. When a request is set high, the read address is one-hot encoded and stored in

the LIFO.

The LIFO Memories

These memories have a depth corresponding to the number of rows or columns, which are used in each pixel block. The addresses are one hot encoded, so that the reading of the LIFOs directly gives the row and column position of the request. This PE sub-component saves the AER addresses in the LIFO for generating the signals required during the verification process.

The Finite State Machine

The FSM has 9 states for controlling the data exchanges between the PE components as well as the pixel block signals. It is presented in the Figure 5.4.

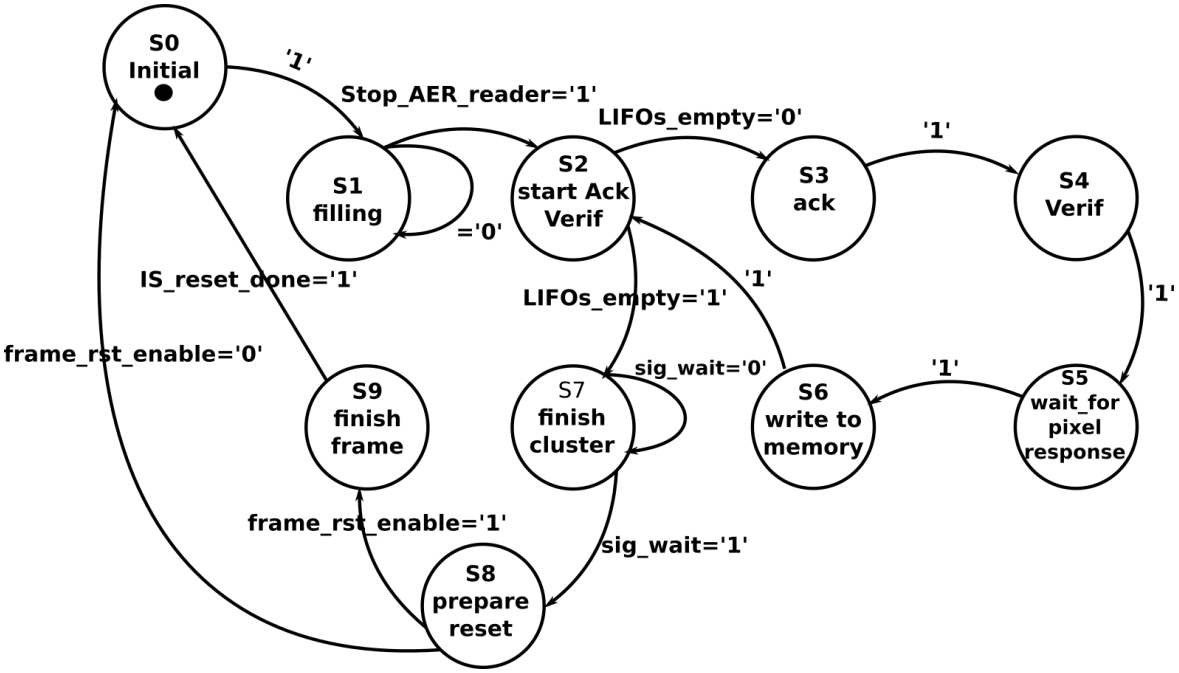


Figure 5.4: The FSM controlling the processing element

The FSM states are shortly described below:

State 0: The initial state, which resets the matrix.

State 1: The AER addresses are saved into the LIFO memories.

State 2: The FSM starts the event acknowledgements and the verification process.

State 3: The PE acknowledges the address found on the top of the LIFO.

State 4: The PE uses the same address and runs the verification process.

State 5: The readout waits for the pixel response.

State 6: The pixel sends the verification result and the decision to write (or not) the data into the local memory is taken accordingly.

State 7: Once all the addresses stored in the LIFO have been checked, the TDC (Time to Digital converter) returns a value coding the luminance.

State 8: When the TDC reaches its lowest value (corresponding to the longest integration time), the FSM emits a reset pulse to perform a global reset of the pixel matrix.

State 9: Once the global reset is done, the FSM enters in its initial state and start a new cycle.

5.1.2 The Time-to-Digital Converter

The time to digital converter (TDC) timestamps the arrival instants of events. The timestamped values simply encode the pixel luminance. The TDC clusters the integration times of several events arriving in a short interval. For example, the Figure 5.5 shows a small pixel matrix of 3x3 and how the TDC is used to encode and build an image. The TDC encodes here 4 different gray levels (255,191,127,63) as shown in the matrix with the numerical values. The small matrix shows the same but displays directly the pixel luminance. In practice, the brighter pixels fire first and the darker later. In the figure 5.5, we can see 4 intervals corresponding to 4 integration times. Each time the TDC receives an event during a time interval, it codes the corresponding pixels with the adequate gray levels according to the arrival event instants.

The pixel (1,1) is the faster to integrate and generates the first event. The latter is acquired by the PE, acknowledged and verified. Then the TDC writes in the local memory at the pixel address the luminance (255). The FSM state moves from *"State 6"* to *"State 2"*. Since no other event arrives in the first interval, the FSM moves to *state 7* and waits for the *sig_wait* TDC signal to go up in order to start another cycle of event capture. At the same time, the TDC decreases its output to the next gray level and this continues until the TDC reaches its smallest value. The FSM proceeds to states 8 and 9 to perform a global reset. Once the reset is done, the FSM starts again a new cycle from its initial state.

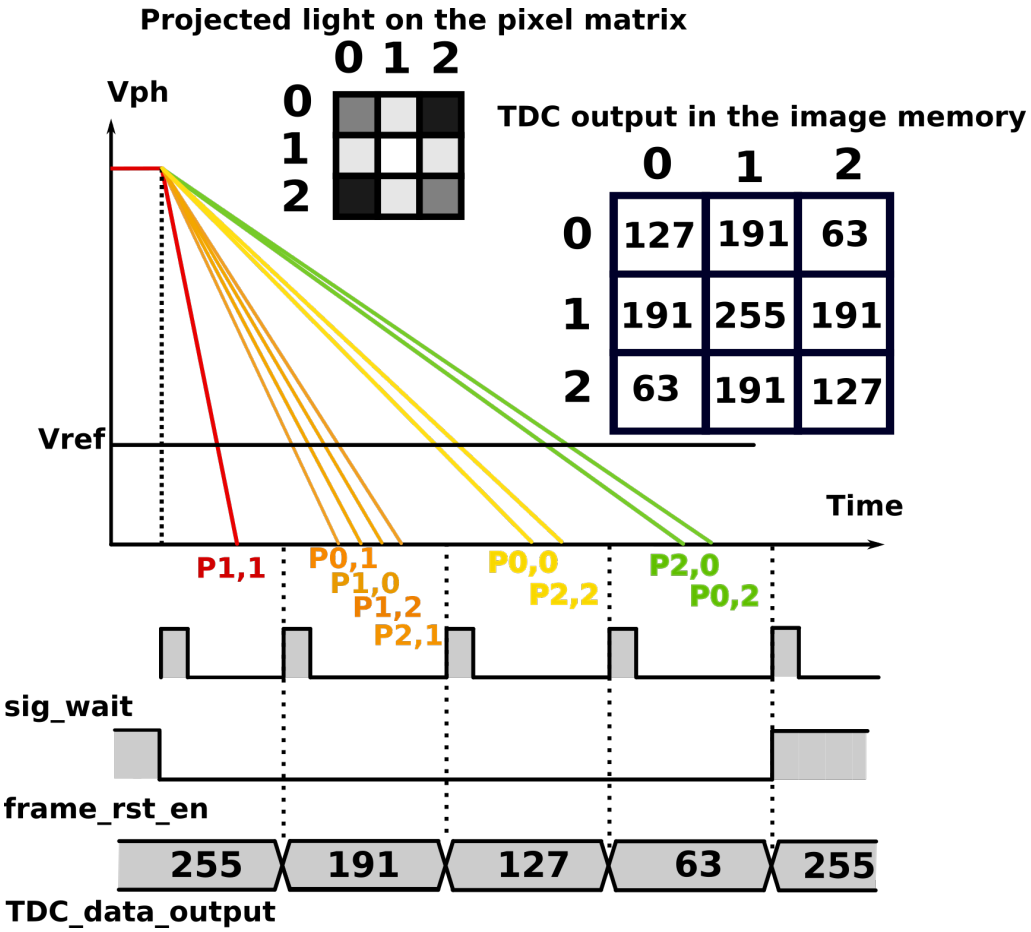


Figure 5.5: TDC time stamping example with spatial redundancy suppression of 4 grey levels or Tmax/4

5.1.3 Global reset block

This digital circuit, when receiving the FSM signal *frame_rst_enable*, starts the pixel matrix reset generating a reset pulse. When the reset is done, the signal *IS_reset_done* goes high, signaling to the FSM the beginning of a new cycle of event capture.

5.2 Readout system simulation

To validate our concept of the readout system in the previous section, we run a Modelsim simulation, in which we assess the accuracy of the readout, in translating the events into image data. The simulation takes in a **PNG** compression image and, thanks to a

Matlab script, this latter is turned into a flow of events. The brightest the pixels in the original image, the earlier the events. The script separates the events into Row and Column events, similarly to event based sensors, the two, go into separate text files, **Row_requests.txt** and **Col_requests.txt**. It also generates a verification response (see chapter 3) file **Verif_responses** that contains all proper pixel responses to permit the readout to verify the events. Reading these files sequentially provides the output of an event-based image sensor. When viewing the image from this bitstream, we find the original image. Therefore, the external environment of our readout is correctly emulated. Figure 5.6 bellow, illustrates the different stages, input, output and intermediate steps of the simulation.

The simulation goes in the following order, but not in Realtime: an image is read using the "Gen.m" Matlab script, to generate the image sensor simulator files. These latter are then read by a VHDL test bench that wraps the RTL of our readout, which is made of one processing element capable of reading a 32x32 block of pixels, hence the size of our input and output images. Consequently the readout generates two text files filled with the image memory content. Finally another Matlab script "display.m" reconstructs the full raw image.

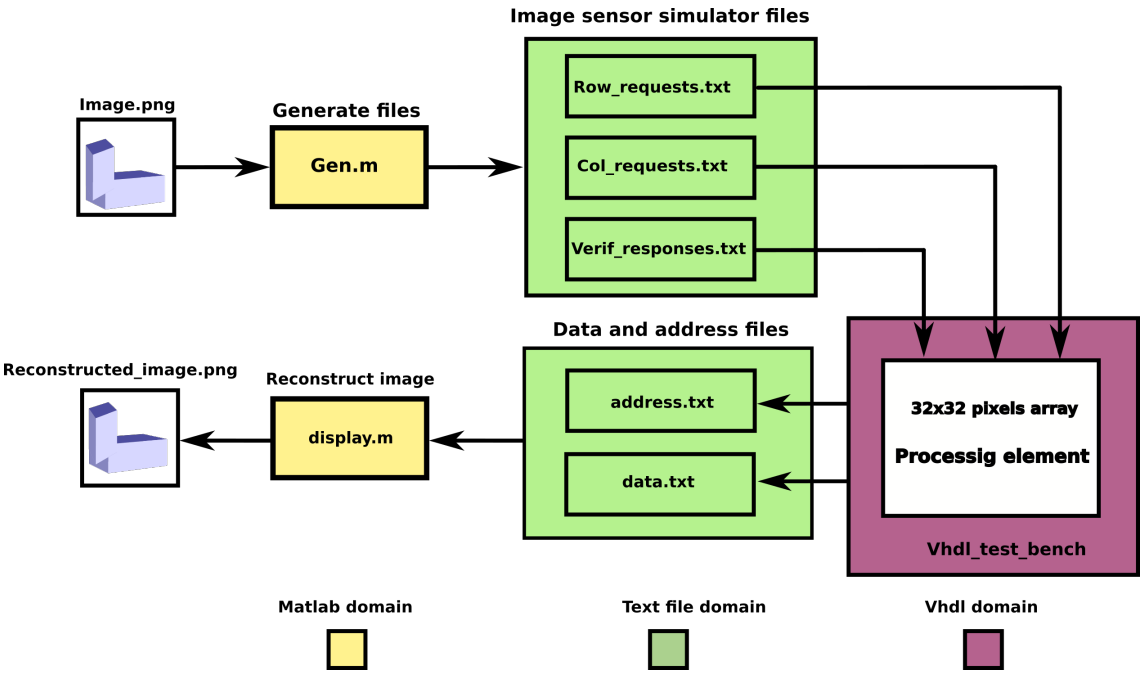


Figure 5.6: The readout simulation

As input, we entered two images with the goal of extracting their output for multiple

instances of the spacial redundancy suppression parameter α = (256, 64, 32, 16, 8, 4, 2), which also designates the number of grey levels the resulting image will have. Higher spacial redundancy suppression means less grey levels in the resulting image. Equation bellow:

$$\Delta t = \frac{T_{max}}{\alpha} \tag{5.1}$$

T_{max} is the maximal integration time corresponding to the exposure and processing duration necessary for the capture of events that constitutes a full frame, in analogy with standard CMOS imagers. As a result, Δt is the elementary exposure and processing duration needed to capture and store one cluster of close events in time, constituting a single grey level data in the resulting image. For our simulation, the maximal integration time was set to 32 ms, meaning that for 256 grey levels the minimum integration time is 125 us.

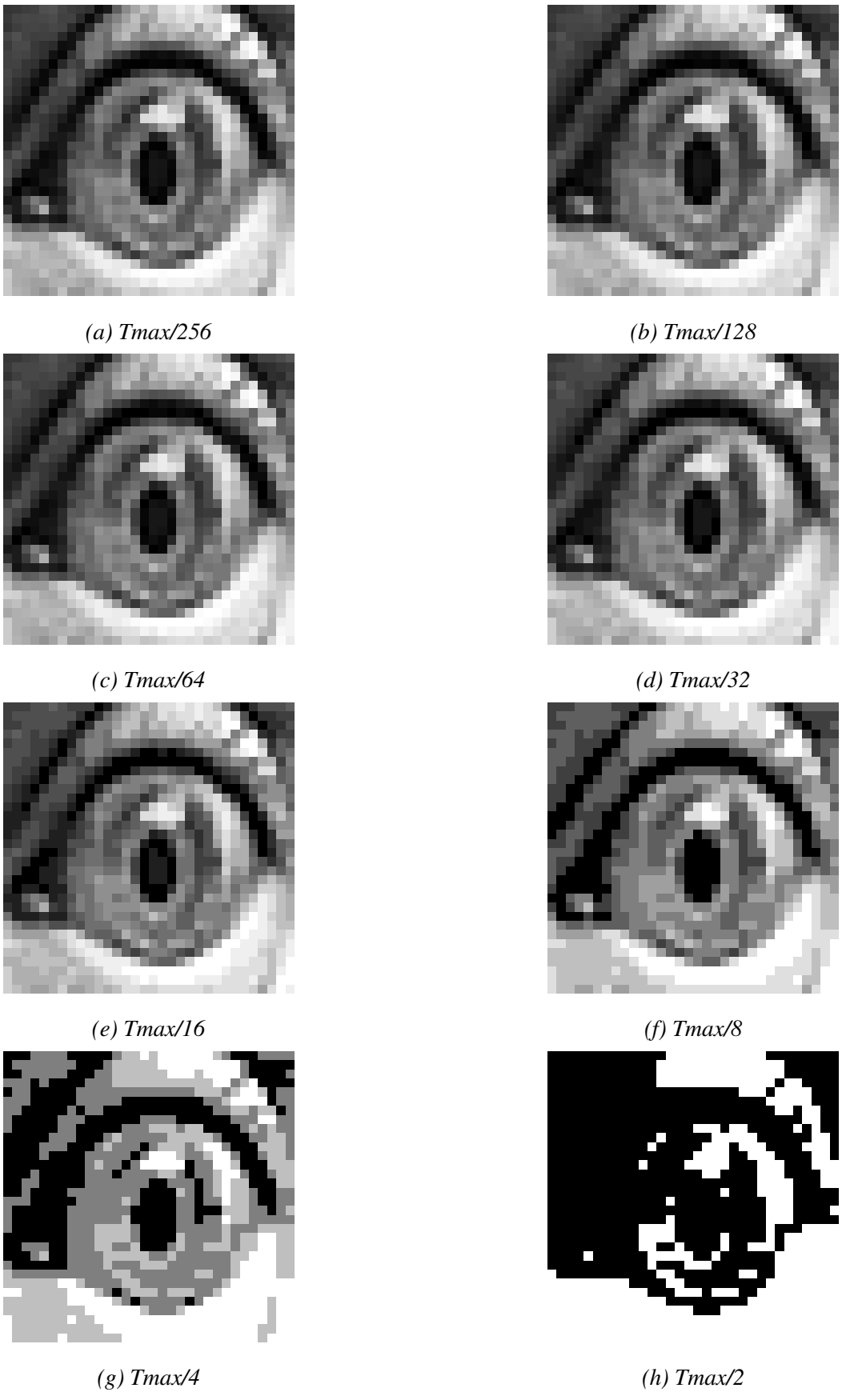
5.2.1 Simulation results

The readout manages to perfectly reproduce the original images, from the flow of the input events. Figures 5.7 and 5.8 display the output images for different values of Δ. At the first glance, we notice that the spacial redundancy suppression factor Δ does affect the number of verification, or the number of times the readout communicates with the image sensor. This number includes the effective events as well as the false events due to collision. However, what was not expected is that the number of verification tends to fall off significantly, at the smaller values of Δ. This is mainly linked to the fact that, at smaller values of Δ, there are less events in each cluster and less collisions, hence less verifications, and this may depend on the image. In order to be sure of this conclusion we need a statistical approach through a large number of input images, which unfortunately has not been accomplished.

Regarding the images quality, the results fit the expectation, which have predicted that a higher spatial redundancy suppression (smaller α) degrades the resulting image quality.

	Delta							
	$\frac{Tmax}{2}$	$\frac{Tmax}{4}$	$\frac{Tmax}{8}$	$\frac{Tmax}{16}$	$\frac{Tmax}{32}$	$\frac{Tmax}{64}$	$\frac{Tmax}{128}$	$\frac{Tmax}{256}$
EYE								
SSIM	0.3897	0.7747	0.9308	0.9808	0.9957	0.999	0.9998	1
Verifs number	1024	2676	5357	8662	10111	8801	6516	4235
Monkey								
SSIM	0.3421	0.6962	0.9158	0.9774	0.9941	0.9987	0.9997	1
Verifs number	928	2128	4292	6494	8784	9576	8270	5749

Tableau 5.1: Output images quality and verification numbers



(a) Tmax/256

(b) Tmax/128

(c) Tmax/64

(d) Tmax/32

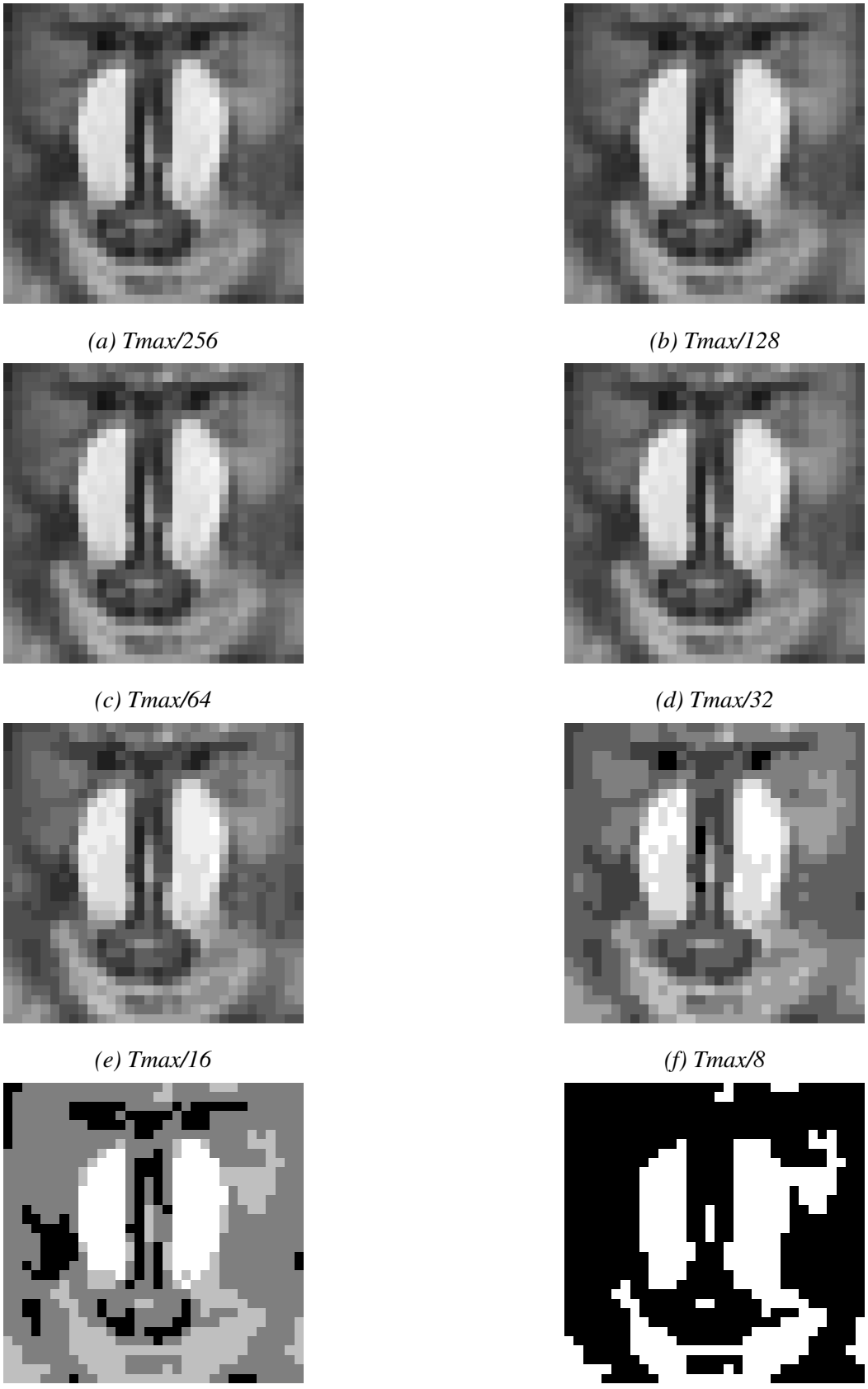
(e) Tmax/16

(f) Tmax/8

(g) Tmax/4

(h) Tmax/2

Figure 5.7: Simulation output images for different Δt values (the eye)



(a) Tmax/256

(b) Tmax/128

(c) Tmax/64

(d) Tmax/32

(e) Tmax/16

(f) Tmax/8

(g) Tmax/4

(h) Tmax/2

Figure 5.8: Simulation output images for different Δt values (the monkey)

5.3 Conclusion

The readout block has been tested and validated through RTL simulations. It can generate multiple outputs to fit data throughput requirements, at the expense of the image quality in the case of high spacial redundancy suppression. Also, the verification overhead tends to fall at a lower spatial compression, which makes our readout well suited for both quality and low power imaging to a certain extent. In the next chapter, this readout is used to test both our image sensor and a previous model of our image sensor containing only TFS pixels. The main use case of our readout is low power imaging, but not exclusively, as many other future specific use cases can unfold.

6

Test chip

Here we present our testing procedure and testing results

Sommaire

6.1 Event based image sensor (EBIS) 83

6.1.1 testing setup 83

6.1.2 testing methodology 85

6.2 Results 86

6.3 The hybrid event based image sensor testing 88

6.3.1 The testing setup 88

6.3.2 The testing methodology 89

6.3.3 The testing results 91

6.4 Conclusion 94

In this chapter, we will discuss the testing setup utilized for the two chips. The initial test involved a previous version of the event-based image sensor, which was implemented by a former PhD student of the team. This earlier sensor featured a 32x32 matrix of TFS pixels and was read utilizing the readout system detailed in the previous chapter. The second test focused on testing the image sensor chip, presented in the chapter 4. The two tests equipment are similar: a PCB board, an FPGA and an optical bench. The main difference is the adjustment of the photo-sensitive area of each sensor, since their implementation technology is different. The TFS matrix has been design in AMS 0.35 µm, and our the hybrid matrix is based on a ST FDSOI 28 nm technology. And in the next chapter, we present an analog arbitration circuit with three inputs, that we implemented alongside the pixel matrix in the sensor chip. The goal is to evaluate a kind of arbiter that could be of interest for these image sensors.

6.1 Event based image sensor (EBIS)

The TFS sensor was designed during the Post doctorate of Amani Darwish, after her PhD [38] in 2015. One of the goals of this sensor is to test the readout technique, that was developed during her PhD. Unfortunately, this sensor was not tested during that time, since its manufacturing took a significant time, and had been tested during this PhD. This sensor was implemented in AMS 0.35 µm process. It contains a 32x32 pixel matrix of TFS pixels and did not have row and column arbiters, since no arbitration has been designed. Its internal readout system only had row and column shift registers.

6.1.1 testing setup

Significant time was taken for testing this sensor, as it was an important step towards the test of the hybrid sensor. The first step was the verification of the PCB after manufacturing, to ensure that all the connections were correct, especially the orientation of the packaging, as an image sensor chip has to not face the PCB inwards, making this mistake can potentially result in permanently damaging the chip, due to short-circuit risk. Setting up the FPGA interface using the dedicated software is the next step. Finally, we can

connect all equipment together. Figure 6.1, displays our setup for this test:

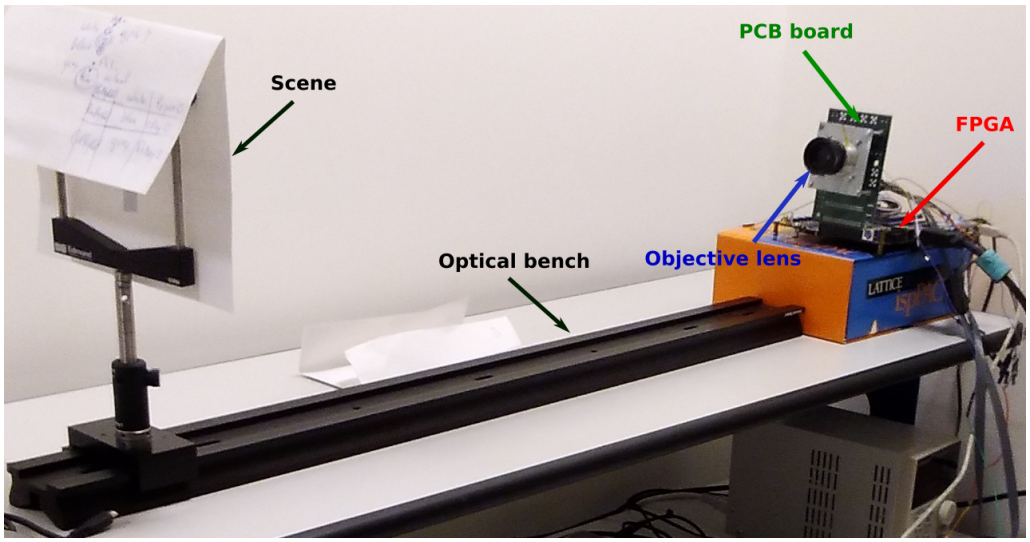
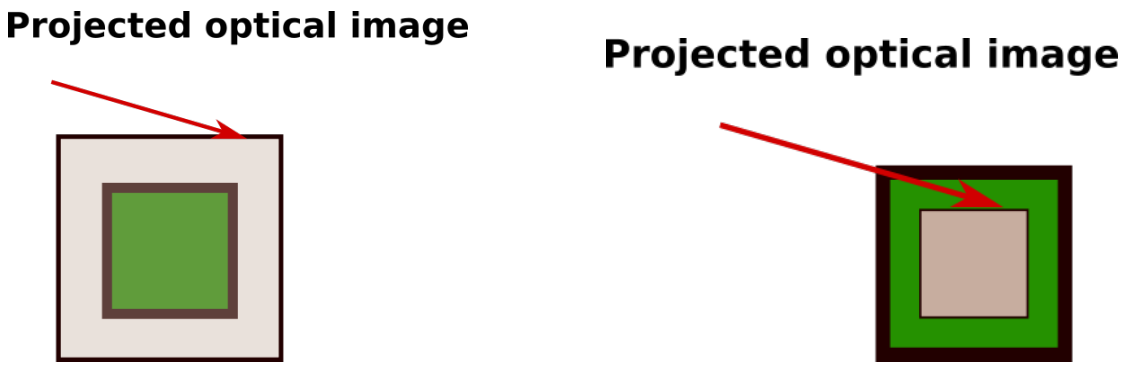


Figure 6.1: Test setup

When choosing an objective lens, the image sensor area is an important parameter that should be considered, since objective lens come with a predefined sensor area. This means that if the image sensor area is larger than the image area produced by the objective lens, only a part of the image sensor will view the whole scene 6.2b. If the image sensor area is too small, the whole image sensor will be viewing a part of the scene, resulting in a cropping of the image 6.2a. Also the box that holds the objective lens, should meet the focal length of the objective, which is the right distance between the objective and the image sensor. It is also worth mentioning that the working distance at which viewing will be done, the angle of lighting and many more are important factors that come into play when choosing an objective lens. For our test, we choose a general purpose objective lens with a fixed focal length of 18 mm, a viewing distance from 0 to ∞, and a sensor area of 2.5 cm^2, which is much larger than the image sensor area of 0.04 cm^2.



(a) The optical image projected by the objective lens is bigger than the image sensor area

(b) The optical image projected by the objective lens is smaller than the image sensor area

Figure 6.2: The importance of the sensor area and the objective lens projected image area: (a) result in image cropping (b) results in not using all of the pixel matrix efficiently

For our interface, we choose the altera evaluation board with its Cyclone II FPGA, thanks to its availability, experience using it, diverse peripherals and flexibility. We also used an oscilloscope and a VGA monitor, to visually inspect events and the contents of the image memory when debugging. In the FPGA, there is the readout, the image memory and the VGA controller. The image sensor chip digital IOs use 3.3 V, which is directly available on the FPGA without the need to any level shifters.

6.1.2 testing methodology

Here, we describe the steps we went through to carry out the testing, due to the lack of image sensor testing expertise and equipment, like a luxmeter, we need to find dedicated solutions. After all equipment have been set up correctly, and the power supply can be turned on. The testing of the sensor goes through several phases:

The first one is knowing the pixel sensitivity to light. The image sensor is exposed to different power levels of lighting, and we monitor the arrival of the events in time. For this, it is particularly interesting that to implement a test pixel alongside the pixel matrix, that has direct inputs and outputs to the IOs of the chip. This was indeed done for the image sensor being tested.

After measuring the average response time of the pixels to different light intensities, the next step is to configure the external TDC in the FPGA, to count the longest and

shortest integration time the pixels emit, meaning that, the TDC should have a precision corresponding to the shortest integration time, and reset at the longest integration time. Well trimming the TDC results in an accurate imaging. The worst case of the TDC configuration, can result in a saturation of the full image or parts of it, when the TDC is too slow, that it ends up catching most the events at a later time. In the worst case of a fast TDC, the events are not caught at the last integration time, which results in a partially or totally uncaptured image. Indeed, the TDC is resetting between the end of the integration of the pixels.

The pixel response uniformity should be verified, as it is mostly the result of the fabrication process variations. In critical imaging applications, variation process errors, require another layer of processing in the readout, to fill the missing pixels.

To recapitulate, before proceeding to scene imaging, these two steps must be carried out:

- Evaluating the pixels sensitivity to light and response uniformity if relevant.
- Setting the TDC timing configuration.

6.2 Results

Following the instructions above, we were able to obtain the results in the table below 6.1. We unfortunately did not own a luxmeter to verify the levels of lighting to use. However, we had in mind a scene of high contrast to photograph, which was a black duct tape line over a white paper. In order to calibrate the TDC for this scene at room daylight, we first filmed a white paper to capture the fast events, which arrived at an average time of 5 ms. Then, we used a paper with duct tape all over its area. The events in this case arrived on average at 100 ms. Therefore, we had the most probable dynamic range of the scene we were trying to film. Consequently, we were set for filming our duct tape line over a white paper scene. The figures bellow illustrate both the scene 6.3 and the resulting image 6.4 on the monitor. We note here that the readout produces a constant flow of events to the monitor, from which the image is monitored in real-time. The next step in the test was to try to film a bit more challenging shapes, like a circle, triangle, and

a star, to end up filming a real object. Unfortunately, filming complex shapes and objects did not produce a clear image in the sense that the borders of the shape were not perfectly recognizable, this is mainly due to the low sensor resolution and probably to our optic set up and readout. At this point, the proof of concept has been made and we had to give up and move preparing our new chip at the time of this test.

blank	Min integration time	Max integration time	TDC period
Room day lighting	5 ms	100 ms	296.875 us

Tableau 6.1: Integration time and TDC period for room lighting contrast

Figure 6.3: Scene

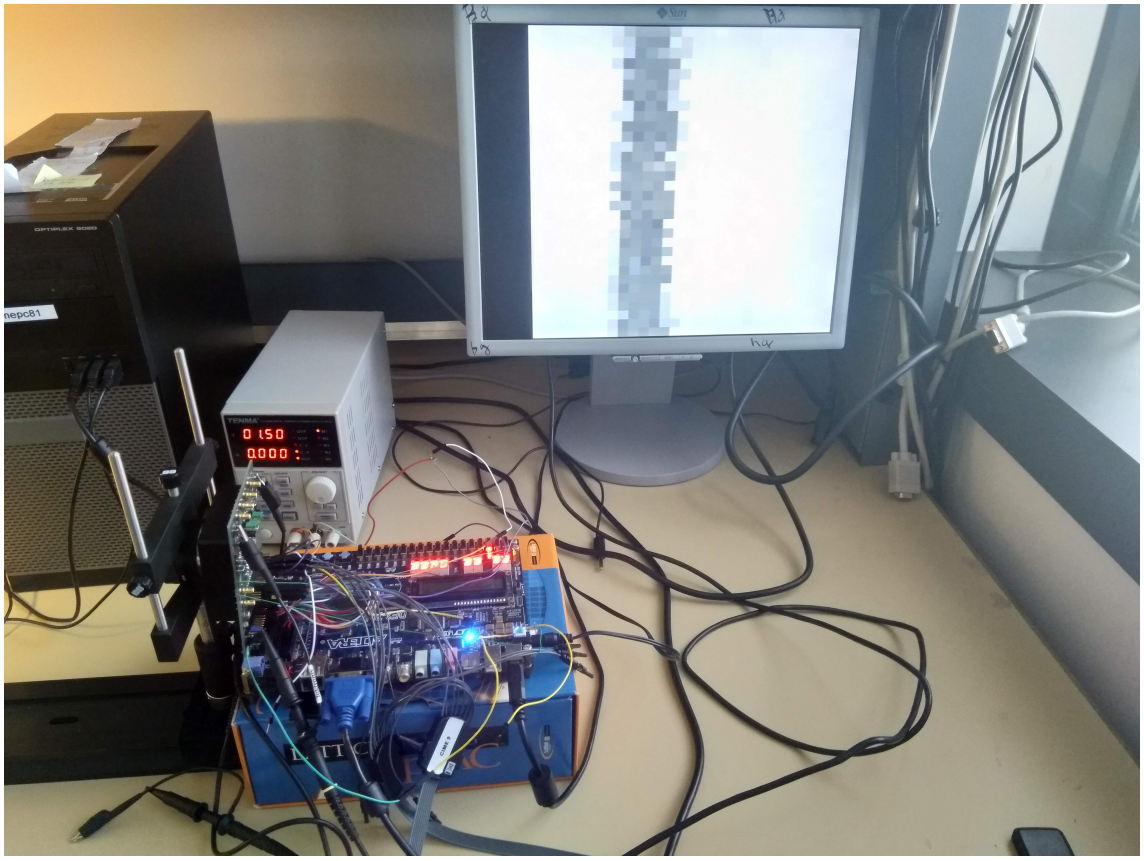


Figure 6.4: Live video seen at the monitor

6.3 The hybrid event based image sensor testing

Our hybrid image sensor chip has this time significant differences: It does not include a testing pixel or kernel due to a lack of IOs. Indeed, the die has been used for several IPs, which limited the available IOs for the image sensors.

We can also notice another difficulty, which is the image sensor area. This latter is able to view less than 10 % of the filmed of scene. Indeed, in 28 nm, the pixels are very small. This means that the objects to be filmed should be at the center of the field of view.

6.3.1 The testing setup

Testing our image sensor went through the same steps as described in the previous section, however, there are differences in the setup. For starting, the photosensitive area and the

pixel matrix in this chip is also small, around 0.23 mm². Consequently, there is a need for an objective that can focus the field of view into 0.23 mm². The most appropriate objective we could find has the following characteristics:

- Maximum area of the sensor 6.25 mm².
- Focal length of 1.9 mm.
- Viewing distance of 400 mm to ∞

Several changes were introduced to the PCB to accommodate the objective 6.5.

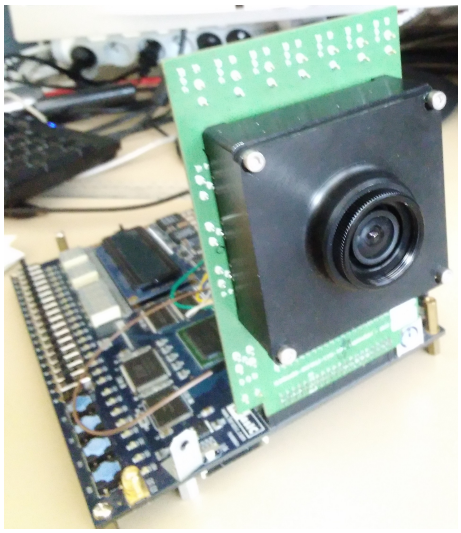


Figure 6.5: The hybrid image snesor camera

This time, the 28 nm chip IOs only support 1.8 Volts. Connecting these IOs to a 3.3 V IO FPGA, requires level-shifters. Luckily, the FPGA we choosed, is able to generate a multitude of IO voltages, including 1.8v. The rest of the equipment is the same as those used for the previous testchip. The scene photography is challenging this time, as the architecture of the pixels has to detect lighting changes in time. for this, we thought to simply duct tape objects onto a circular paper, attached to a motor, which has a controllable speed. This way the rotating object generates the desired stimuli, in a similar way to the tests done in previous event-based sensors in the art.

6.3.2 The testing methodology

Before talking about the testing approach, we would like to present the constraints that limited what we can test and not. We have implemented a feature that makes possible

to trigger all the TFS pixels in each pixel matrix, regardless of the DVS pixel state of trigger (ON, OFF). We have also put test pixels. Unfortunately, the test pixels were not connected to any external IOs, since our image sensor, shares the chip area and IOs with two other circuits, due to a tight budget. Figure 6.6 bellow, highlights one of the two other circuits, which is a CNN. The other circuit is an asynchronous demodulator, which is much smaller (this is why we do not see it). Nevertheless, it takes a significant number of IOs. Basically, the IO budget was limited in analog and digital, so we had to remove some of our debugging IOs. without forgetting that most our request, acknowledge and verification IOs, are sent and received through shift registers, like the request signals, and start integration signals that are sent in series, and need to be made parallel and separated before any processing , which is not the best for an asynchronous event based image sensor, all to save up the maximum number of pins, and this has limited a lot our testing capacity.

Two aspects of our image sensor should be taken into account during the test: the photo-measurement and the trigger, the TFS and the DVS groups of pixels. These two, work together to provide the desirable feature of the sensor. However, they need to be tested separately, to insure that each is functioning properly. Due to the limitations above, we could only test separately the TFS pixels, as they are the eyes of our sensor, we prioritised there debugging IOs over the DVS pixels.

Finally, only the TFS pixel matrices are available to debug, and even this task is made harder without the testing pixels, but it is still possible. After verifying that communication with the image sensor is functional, we will proceed to pixel response testing and the TDC configuration, correspondingly to the previous test.

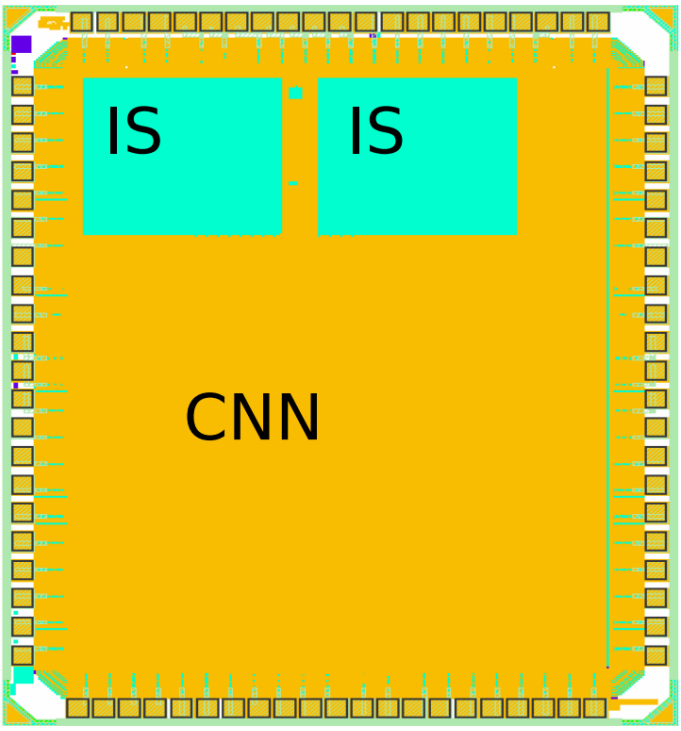


Figure 6.6: The pixel matrix schematic

6.3.3 The testing results

During our first checking steps, we detected our first issue. The pixels could not remove their requests after being acknowledged. This means that we could never proceed to verification process in order to save a correct image in the image memory, with no event collisions. After a careful observation and debugging, we discovered that most row and column buses do not go back to their reset state, which is a logical one, looking at figure 6.7. This is due to the PMOS transistors at the end of each bus that form a wired OR with the NMOS transistors in the pixels. After a number of pixels drive the bus down to 0, the reset PMOS transistors are incapable to drive theme back to their reset state. The only thing we could measure was the arrival time of the first events, before all the rows and columns request buses are saturated with events. This time was sparse and ranged from 1 ms to 10 ms.

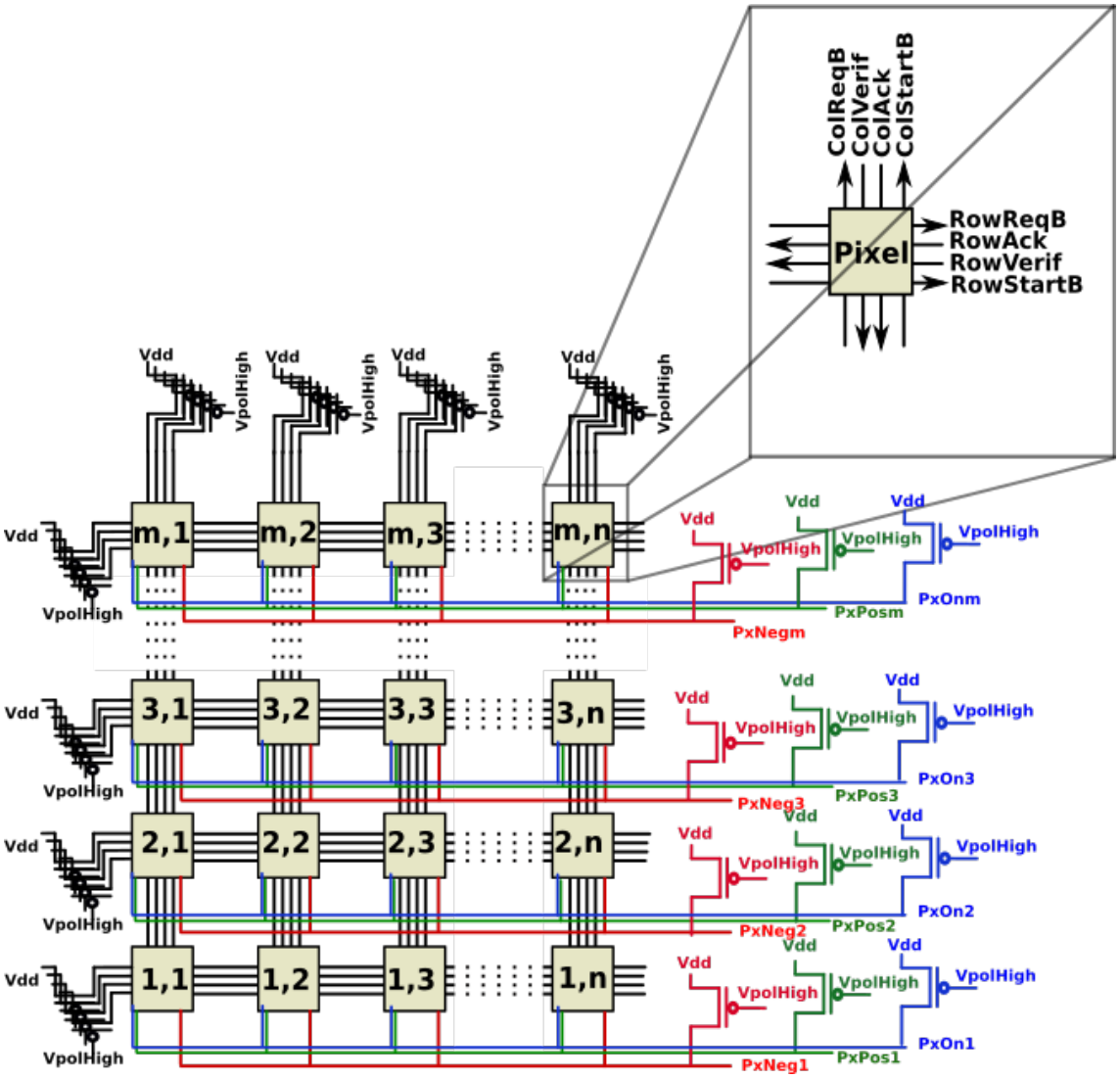
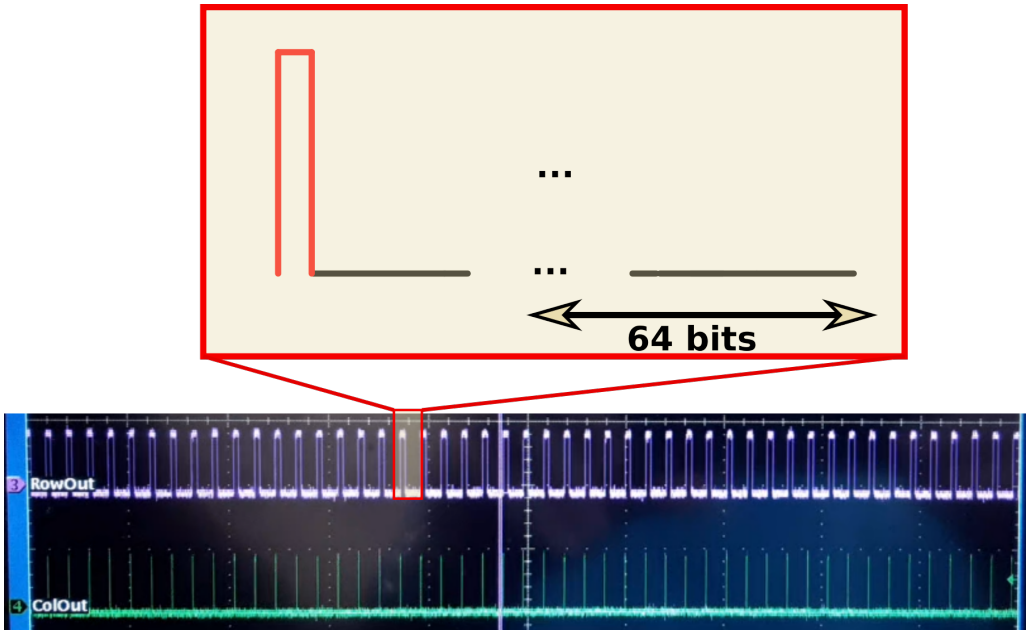
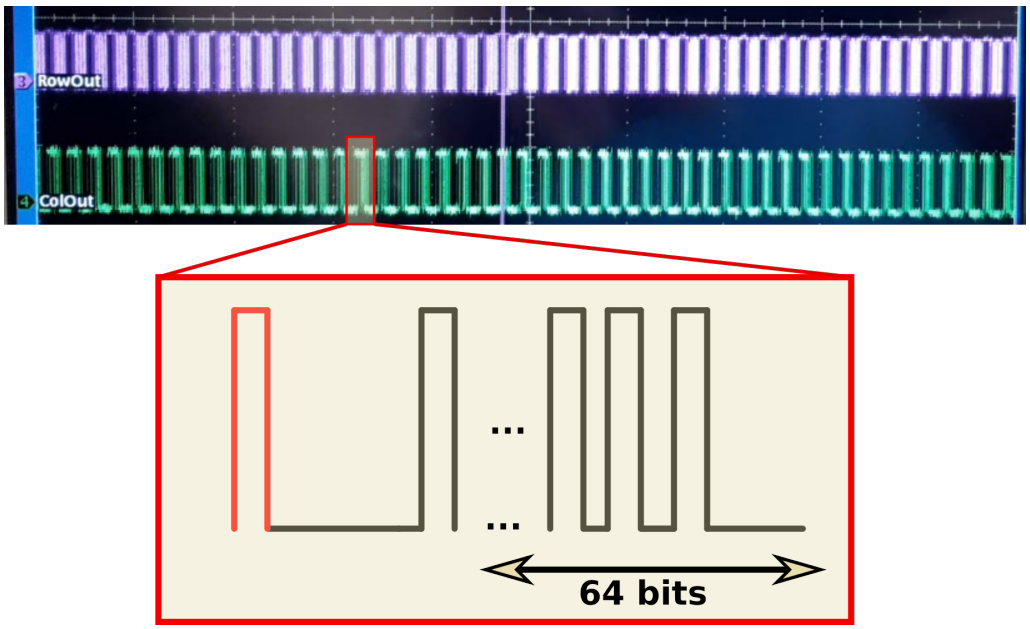


Figure 6.7: Chip layout

In order to read the data out of the image sensor, we have to constantly monitor the shift registers output at fixed period (short enough to not miss the shortest integration time), to detect any new events. The request signals are the last 64 bits of the shift register output. The row and column requests recorded on the scope are shown in the two figures bellow 6.8. We can see in the figure 6.8a above the complete lack of requests for the first 20 ms, when the image sensor is not exposed to light. When the image sensor is exposed to light we can see requests starting to appear 6.8b during the first 20 ms, waiting any longer with or without lighting will reveal requests, since TFS pixels always integrate, even with very low lighting, it takes however longer time.



(a) Under low lighting conditions



(b) With high lighting

Figure 6.8: Row and column requests activity recorded on the scope

6.4 Conclusion

In this chapter, we have summarized the accomplished two chip tests, one for a previous event based image sensor, and the second for our image sensor we designed in chapter 5. The first test enabled us to functionally validate our readout system, that was developed during the PhD, and to be used as the readout of our hybrid image sensor. The testing of the second image sensor, unfortunately provides us partial functional results to verify our design. However, the image sensor did produce events that reacted to light intensity change, but a design issue in the internal readout, prohibited us from continuing the readout protocol of events in the pixel matrix, as a result no images where extracted.

7

Winner Take All (WTA) arbiter

Here, we present our arbiter architecture, based on a simple winner take all circuit, the design and testing results.

Sommaire

7.1 WTA cells (Lazzaro cells) 99

7.2 Proposed architecture 100

7.2.1 scaling 103

7.3 Chip test results and comparison 103

7.4 Conclusion 106

In this chapter, we present an innovative approach for designing arbiters, which could be of interest for our event-based image sensor. Most arbiters in the state of art use mutual exclusion cells, as a solution to propagate their signals to the next stage. This approach is binary and can only process two inputs in each unit. This leads to binary tree, which are one of the limitation of the event-based sensors. Indeed, When scaling up the image sensor resolution, the arbiter tree is becoming wide and deep introducing delays and, so imprecision on the time stamping operations. moreover, the area increases very quickly.

Here, we explore the usage of multiple input arbiters, which does not require a binary tree. It is made possible thanks to the winner-take-all analog circuits. We first explain their functionality, their utility, and how they are adapted to our need. Then, our test setup and results are presented.

7.1 WTA cells (Lazzaro cells)

In general, Winner-take-all (WTA) electronic circuits are neural network circuits that select the input with the highest value, while suppressing or ignoring the other inputs. These circuits are commonly used in image processing, pattern recognition, and other applications that require the selection of a single winner from a group of inputs. There are different types of WTA circuits, such as analog and digital circuits, and they can be implemented using different technologies such as CMOS, memristors, or others [39]. For our arbiter we have chosen Lazarro cells as the building block of our arbitration unit, these cells are known to compute a high number of inputs with trade-offs. Figure 7.1 bellow illustrates Lazarro winner-take-all cells. The circuit contains several cells composed of two transistors (T_{1_k} and T_{2_k}). The cells share a wire with the potential V_c, that computes the inhibition [1] of all the cells. In operation, each cell in the circuit sinks a current through the transistor T_{2_k} to compute the global inhibition of the circuit, transistor T_{1_k} applies this inhibition locally to each cell, the winning cell is logarithmically encoded by its input

[1] Inhibition refers to a neurological or psychological process that reduces or prevents certain neural or behavioral responses. Inhibition can occur at various levels of the nervous system, including the synaptic level, where one neuron can inhibit the firing of another neuron.

current I_k. To summarize, the inputs of the circuit are the currents (I_1,..,I_k,...,I_n), and the output is the voltage V_k 7.1, which translates the logarithm of the maximum current I_k, which is sinking through T_{1_k} of the winning cell:

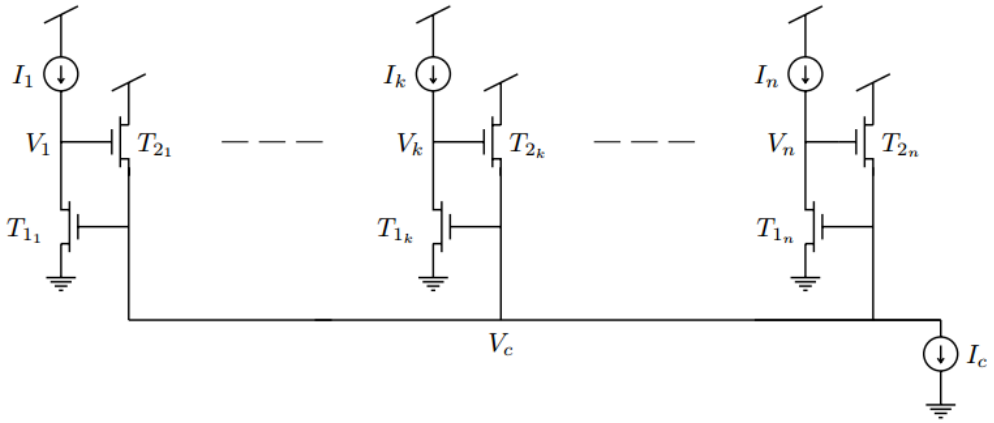


Figure 7.1: Lazarro winner take all circuit [39]

$$V_k(winner) = V_0 \log(\frac{I_k(winner)}{I_0}) + V_0 \log \frac{I_c}{2I_0}, V_k(losers) \approx 0. \tag{7.1}$$

where $V_0 = \frac{kT}{q}$ is the thermal voltage of a MOS transistor and I_0 is an intrinsic constant of the transistor. The mechanism behind the inhibition computation is the early effect.

7.2 Proposed architecture

The circuit presented in the previous section is an analog circuit, and in order to use it with a digital readout circuit, it needs an appropriate interface. We are looking for using this circuit as an arbiter that is able to take multiple inputs at the same time and guarantee the election of a winning cell in the shortest possible time. Indeed, the main constraint is the arrival time of the next serie of inputs. Insuring the processing time in Lazarro cells is challenging, it is observed that the common-signal node Vc slows the operation down dramatically, as the system complexity increase, that is the number of identical cells increase, due to the device mismatches, the resolution of the circuit also degrades. Another important design trade-off is the decreasing mismatch effects with the increasing transistor size at the expense of more area [40]. Indeed, electing a winning

cell is a slow process when two inputs with the same current arrive at the same time. This causes a slow inhibition. Therefore, it is necessary to avoid simultaneous activation of inputs with the same current sources. To counter this, we can use different current values, which introduce a form of a fixed priority arbitration. The worst case here is, if two neighbouring (neighbouring in priority) cells compete. Consequently they must use source currents, that are different enough to meet the time constraint above. The best case is when two cells at the extremities (far in priority) of the WTA compete. The winning cell is then elected in a minimum time. This thought process led to the circuit in the figure bellow 7.2. Here, we use current sources with increasing currents, and Mutex circuits to accelerate the inhibition in the worst case (two neighbouring cells competing). The Mutex circuits also solve our need for an interface with a digital circuitry. To sum up, our arbiter contains an analog and a digital domain.

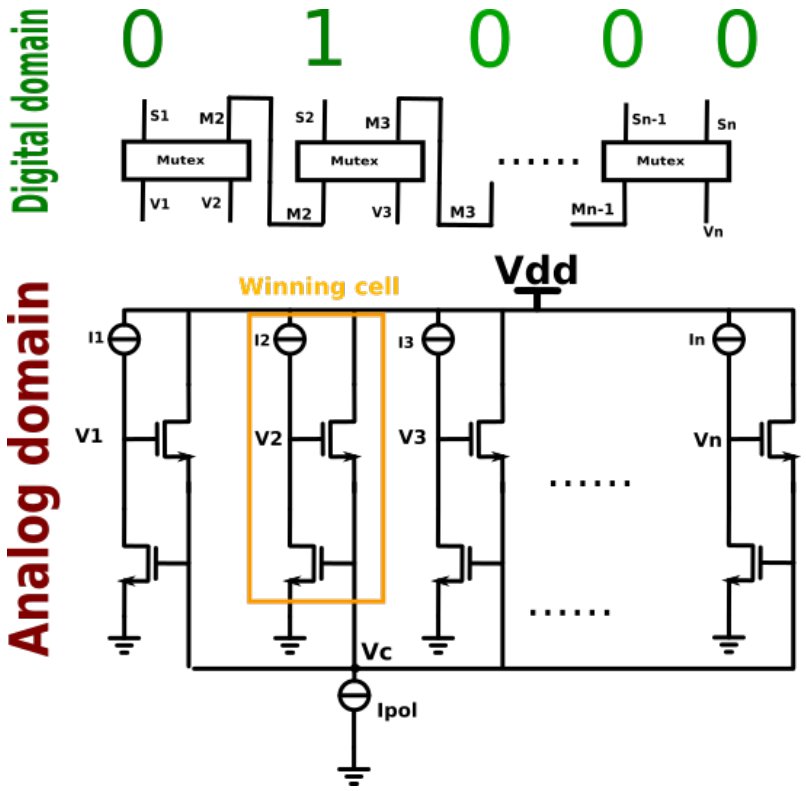


Figure 7.2: Winner take all arbiter

To test the proposed architecture, we implemented a three input arbiter in the 28 nm technology, in the same chip of our image sensor. The schematic of the global circuit is in figure 7.3. In this circuit, there are three Lazarro cells, each connected to a fixed current that increases from right to left to insure the fixed priority between the cells. All

cells input current sources are dependant on a reference cascode current source(on the right in the figure), which is a also controlled by an external voltage Vpol, to increase or decrease the current in the reference current source, our circuit also has an external voltage Vpol_s to control the sinking current from all the cells. The cell current sources are copied using current mirrors from the reference current source . Cell2 input signal is an external signal In2, when it is low it activates the current source of the cell, similarly Cell1 has In1 as the input signal that activates its current source, Celld does not take any input signal because this latter is active all the time. The reason is the necessity to have a default winning cell, because if we don't have a default active cell, when all the cells are inactive, the inhibition process will still elect a winning cell at some point, because of the leakage currents. Therefore, a default winning cell is mandatory. For example, if the input signals IN1 and IN2 (from cells 1 and 2 which are higher in priority than celld) are inactive, the output will be ((Out2,Out1,Outd) = (0,0,1)). All the transistors in this circuit have the same W and L and they have a width of 200 nm, except the transistors P0, P1 and P2, which have a Width of W, 2 W and 3 W respectively. This linearly increases the current source of each cell to insure the fixed priority between the cells.

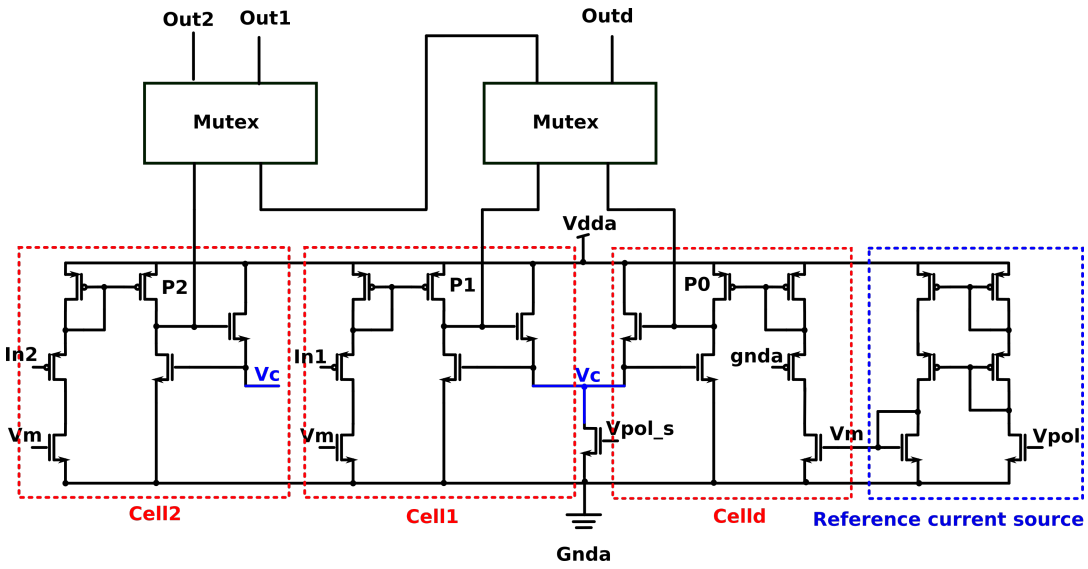


Figure 7.3: three cells, three output arbiter (28 nm, Vdd=1V)

For our reference current source, we planned to use an external current source for a better precision. However, we ended up using an internal current source due to the already mentioned pin shortage. Hence, our reference current source is voltage controlled by one of our image sensor bias voltages. In the figure, the voltage Vpol controls the current

source for all the cells. the inhibition is controlled thanks to Vpol_s, which is also an image sensor bias voltage in the chip. Therefore, for a correct functioning, we turn off our image sensor matrices.

7.2.1 scaling

Our circuit has been designed to be a proof of concept of a triple cell elementary block that can be used in a ternary arbitration tree, instead of the traditional binary tree. For example, processing a 128-bit input using a binary tree, will require 7 layers of binary arbiters stacked in a pyramid. Using our ternary arbiter, we can process up to 243 inputs in a ternary tree of 5 layers. Conceptually, our arbiter is better when scaling the number of inputs. Also, our arbiter unit can be designed to process more inputs by simply increasing the number of cells at the expense of latency increase, a trade-off that should be considered.

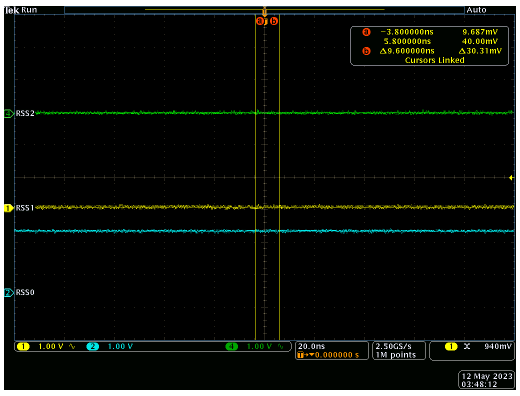
7.3 Chip test results and comparison

The following results where obtained after nine different measurements of the arbitration delay, with different values of Vpol the current source external control voltage and Vpol_s, the external control of the sink current, both displayed in figure 7.3. We simultaneously activate the inputs IN1 and IN2, guarantying the worst case for our arbiter. Table 7.1 lists the different arbitration latency for each pair of Vpol and Vpol_s, Our arbiter has yielded experimentally an optimal latency time of 4.6 ns, which equates to electing 217 million times per second. We notice, that increasing Vpol_s above 350 mv has very little effect on the latency, since the NMOS transistor sinking current enters saturation after 350 mv, and its drain current varies little after saturation, however it has to at least be higher than the Vth of the NMOS sink transistor, otherwise, the inhibition won't function properly as seen during tests. Vpol on the other hand, had a significant effect on the latency when Vpol_s is arround 350 mv , since it decides how much current the reference source current will be sourcing into the cells, what was not expected is the fact that Vpol didn't have any effect on the latency for the other values of Vpol_s, we

couldn't investigate carefully this, since we didn't have access to the currents each cell is sourcing inside the chip. During simulation the latency reached 2 ns at the max values of Vpol=750 mv and Vpol_s = 750 mv. Figure 7.4, provides the transient response as seen on the oscilloscope, before and after arbitration takes effect, on the left 7.4a we see the three outputs of the arbiter before arbitration, only the default cell is active, after 7.4b arbitration, we can see that the second cell is winning in 4.6 ns and forcing the other two cells into 0.

Tableau 7.1: The arbitration latency time vs different values of Vpol and Vpol_s (chip test results)

Vpol_s(mv)	350			500			750		
Vpol (mv)	350	500	750	350	500	750	350	500	750
Latency(ns)	7.6	5.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6



(a) Arbiter outputs state before any triggers of the two arbiter inputs IN1 and IN2 (circuit7.3), the default cell in this case is winning

(b) Simultaneously activating the arbiter inputs IN1 and IN2(circuit 7.3), yields Cell2 as the winning cell in no longer than 4.6ns

Figure 7.4: Scope screen captures of the arbiter output, before and after the trigger RSS0(outd), RSS1(out1), RSS2(out2) for (Vpol and Vpol_s at 500mv)

We unfortunately did not have access to the currents each cell was sourcing during its activation, however table 7.2 bellow lists their values during post layout simulations, and the power consumption of the whole circuit. By overlaying the results of the tables 7.2

and 7.1, both simulation and experimental data align, to confirm that increasing Vpol results in sourcing higher currents, and decreases latency, at the expense of a higher power consumption.

Tableau 7.2: The power cinsumption and the arbiter cell currents during activation(Simulation, Vpol_s = 350 mv)

Vpol (mV)	350	500	750
Current (Cell2,Cell1,Celld) nA	100, 250, 380	300 , 650, 1000	350 680 1000
Power consumption (uW)	1.22	2.7	2.78

After collecting simulation and experimental data from our chip test, we compare our results with other implementations of arbitration cells in the art, the comparison is summarized in this table 7.3, the 28 nm technology provided us with an advantage compared to other implementations of the arbitration circuits in older technologies(0.35 um, 2 um, 2.4 um, 1 um), like a smaller area, voltage supply and power consumption which is only 10% of the best power consumption amongst all the other WTA implementation [41]. Also, using a smaller number of inputs (3) compared to the other implementations(8 inputs and more) in the table, means that we can afford a higher difference between the current sources in each cell, resulting in a lower delay or latency. Our arbiter is designed to be a unit in an arbitration tree to propagate signals the fastest possible, like in [42]. In the table all the other WTA circuits, can be a possible improvement to our arbiter that uses the classical WTA lazarro cells, for example to have a higher number of inputs with smaller currents.

Tableau 7.3: Comparison of the experimental and simulation results of our three input arbitration circuit. (*) Obtained from simulation results

Parameter	This work	WTA after [41](High input currents)	WTA after [41](small input currents)	WTA after [43]	WTA after [44]	WTA after [45]
Measurement results	Experimental and simulation	Experimental	Only simulation	Only simulation	Experimental	Experimental
Input	voltage	current	current	current	current	current
Output	voltage	voltage	voltage	voltage	current	voltage
Range of input currents	0.1-1(uA)(*)	3-55 (uA)	0-50 (nA)	at least 60 (uA)	at least 110 (uA)	at least 1 (mA)
Technology	28 nm-FDSOI	0.35 um	0.35 um	2 um	2.4 um	1 um
Voltage supply	1 V	3.3 V	3.3 V	5 V	5 V	not reported
Power dissipation	1.22-2.78 uW (*)	87.5 uW	22.5 uW	not reported	100 uW	not reported
Delay	4.6-7.6 nsec	8-32 nsec	34 nsec	10 nsec	72 nsec for 8 inputs	57 nsec
Circuit Area	6x10 um^2	26.4x22.6 um^2	26.4x22.6 um^2	not reported	80x280 um^2	not reported

7.4 Conclusion

In this chapter, we have proposed a digital arbiter based on lazzarro cells. This arbiter can be a potential substitute for the binary mutual exclusion gates used in a binary tree to

randomly select one input out of n inputs. Our arbiter can be scaled to fit several input signals and to be efficiently used in n order-nary trees to reduce the latency required for crossing multiple arbitration level in a tree. Since, in an n order-nary tree, the higher the n the lesser levels are needed to reach the final output. Our arbiter with tree cells managed to converge an output at only 4.6 ns, which can be suitable for image sensor digital circuits, even with larger resolutions, than habitable 128x128 pixels in the art. This could be a good enhancement for event-based image sensor readout circuits.

8

Conclusion

Today, image sensors are really usual devices, which are used in many applications. These latter are not limited to cameras shooting images, but are largely deployed in robot vision, non-destructive inspections, healthcare or autonomous driving for instance. As the variety of applications is quite large, it exits various configurations, resolutions, speeds for the image sensors. Nevertheless, until the last years, there not too much effort done for lowering the power consumption of these sensors. Based on this observation, investigating the architecture of image sensors in order to decrease their consumption has opened the doors to a strong introspection on what was done in the past. Indeed, the frame structure of video is quite obvious for the engineers but results from the ancestors of CMOS imagers, the video camera tubes. The standard frame approach has been rethought since 20 years by the introduction of the first event-based image sensors. This PhD is focused on such an approach, which leads to frameless sensors, a reduced data throughput and thanks to this a lower power consumption.

After presenting the state of art of the event-based image sensors and the two main strategies for reducing the data throughput by canceling the spatial or the temporal redundancies, this work explores architectures able to combined them together. This first study analyzes a strategy exploiting DVS pixels triggering TFS pixels. This can be done by associating a DVS pixel to one or more TFS pixels (pixel kernel). Thanks to high level simulations and the choice of different kernels, we were able to analyze the impact of the kernels on the data throughput of an image sensors. We finally conclude that the kernel (f) with 4 photodiodes (the DVS function averages the photocurrent of the 4 surrounding TFS photodiodes) was a good trade-off for processing low- and high-activity scenes.

The next step has been to evaluate the feasibility and the performances of such event-based image sensors. Therefore, we designed two image sensors in FDSOI 28 nm from STMicroelectronics, one based of the kernel (f) (averaging kernel) and the other one on the kernel (b) (non averaging kernel), which also uses 4 photodiodes (1 for the DVS and 3 for the TFS). Thus, the design of this two kinds of pixels has been studied in order to assemble them into kernels (b) and (f). Then the complete matrices have been designed, evaluated by simulation and compared in terms of optical quality, data throughput and power consumption. We noticed that the kernel (f) was most appropriate because it has a better fill factor (14.5%), an almost equal power consumption to the kernel (b), and a reduced throughput compared to a full TFS pixels image sensor. It is important to remind

that such image sensors are able to reduce the amount of data by two orders of magnitude compared to a standard image sensors.

In order to give the full picture of the designed event-based sensors, the behavior of the readout system is detailed. Indeed, this latter is part of our strategy for canceling redundant data and, especially, spatial information. Moreover, a specific verification step has to be implemented to guaranty the correctness of the captured images. The readout is fully analyzed and simulation results show the impact of this verification process on the throughput on one side and on the image quality on this other side.

Then, our experimental setup is presented in order to show how such a testchip can be tested and evaluated. In order to develop the test method, we started on an existing image sensor designed im AMS 350 nm by a former PhD student. This was very helpful because this first event-based image sensor only emebedded TFS pixels. Once this task has been successful, we were ready to evaluate the 2 matrices (with kernels (b) and (f)). For this second test setup, we had to manage several issues such as the very small area of our matrices (designed in 28 nm technology), but also a limited number of IOs due to the integration of several IPs without no connection with our event-based hybrid image sensors. This has probably made the test of the fabricated sensors more complex and probably explains the partial results we obtained. Nevertheless, the kernels react well to luminance changes and generate the requests. After analyzing the issues we had, we discover that the row and column buses do not go back to their reset state, which is necessary for a correct behavior of the image sensors.

Inside the testchip, a winner-take-all arbiter has also been implemented in order to evaluate event-based image sensor solutions using an arbbiter tree. We achieved the design of a digital arbiter with tree inputs, as a building block for a ternary arbitration tree. It demonstrates an advantage over the binary tree, especially when the complexity grows. This arbiter shows a worst case latency of 4.6 ns when evaluating it on chip.

Overall, this PhD work has contributed to develop novel event based image sensor architectures and has introduced the idea that such sensors could benefit from the association of DVS and TFS pixels. This can be done by integrating these two kind of pixels into a kernel making "hybrid" the image sensor. Thanks to this approach, the data throughput is drastically reduced compared to standard image sensors. Indeed, such a sensor is able to remove spatial and temporal redundancies and, thus lowering the overall power

consumption by limiting the amount of stored and processed data.

This opens the way for future image sensor chips that will implement frameless strategies. In fact, this revolution has already started with new products using frameless architectures. We are just at the beginning because such approaches also impact the image processing methods on one hand and the way to display the images on the other hand.

When looking the design of our image sensor, we have no doubt that it could be enhanced. For instance, the verification process makes our design arbiterless, which is an advantage for the time stamping accuracy and the circuit complexity. Nevertheless, this requires a post-readout processing. A good trade-off is probably to use n-ary arbitration tree in order to reduce their depth, latency and time inaccuracy.

Once more, we should completely rethink our mindset when designing image sensors because the heritage of video camera tubes is always pregnant in our design way.

Publications

- Laurent Fesquet, Rosalie Tran, Xavier Lesage, Mohamed Akrarai, Stéphane Mancini, Gilles Sicard, “Low-Throughput Event-Based Image Sensors and Processing”, DATE 2023, Antwerp, Belgium, 17-19 April, 2023

- Mohamed Akrarai, Nils Margotat, Gilles Sicard, Laurent Fesquet, “A hybrid event-based pixel for low-power image Sensing”, 28th IEEE International Conference on Electronics Circuits and Systems (ICECS 2021), Sofitel Dubai The Obelisk, Dubai, UAE, 28th Nov – 1st Dec 2021, 10.1109/ICECS53924.2021.9665509

- Mohamed Akrarai, Nils Margotat, Gilles Sicard, Laurent Fesquet,, “An asynchronous hybrid pixel image sensor” , 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC 2021), hosted by Galois, Inc., September 7-10, 2021, pp 55-61, DOI 10.1109/ASYNC48570.2021.00016.

- Mohamed Akrarai, Nils Margotat, Gilles Sicard, Laurent Fesquet, “Arbiterless Event-Based Imager Architecture with temporal and spatial redundancies suppression”, 6th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP 2020) , Krakow, Poland, 23-25 September 2020, DOI: 10.1109/EBCCSP51266.2020.9291345

- Mohamed Akrarai, Nils Margotat, Gilles Sicard, Laurent Fesquet, "A Novel Event Based Image Sensor with Spatial and Temporal Redundancy Suppression", 18th IEEE International NEWCAS Conference, Montréal, Canada, June 16-19, 2020, DOI: 10.1109/NEWCAS49341.2020.9159847

- Laurent Fesquet, Mohamed Akrarai, Gilles Sicard, Mamadou Dialo, "Event-Based Image Sensors", 5th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP 2019), 27-29 May, 2019, Vienna, Austria

- Mohamed Akrarai, Gilles Sicard, Laurent Fesquet, " A Novel Event Based Image Sensor Architecture", IP-SoC 2019, 3-4 December, 2019, Grenoble, France

Bibliography

[1] IEA, "Digitalisation and energy," International Energy Agency, Paris, Tech. Rep. Pages 43-44, 2017. [Online]. Available: https://www.iea.org/reports/digitalisation-and-energy

[2] J. Mäkipää and O. Billoint, "Fdsoi versus bulk cmos at 28 nm node which technology for ultra-low power design?" in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 554–557.

[3] P. Horowitz and W. Hill, *The art of electronics; 3rd ed.* Cambridge: Cambridge University Press, 2015, no. Pages 1142. [Online]. Available: https://cds.cern.ch/record/1981307

[4] T. Kuroda, *Essential Principles of Image Sensors; 1st Edition*. Boca Raton: CRC Press, 2015, no. Pages 192. [Online]. Available: https://doi.org/10.1201/b17411

[5] E. R. Fossum and D. B. Hondongwa, "A review of the pinned photodiode for ccd and cmos image sensors," *IEEE Journal of the Electron Devices Society*, vol. 2, no. 3, pp. 33–43, 2014.

[6] W. S. Boyle and G. E. Smith, "Charge coupled semiconductor devices," *The Bell System Technical Journal*, vol. 49, no. 4, pp. 587–593, 1970.

[7] Omnivision. Ovb0b 200 megapixel product brief. [Online]. Available: http://ovt.wpengine.com/wp-content/uploads/2022/01/OVB0BH0-PB-v1.0-WEB.pdf

[8] L. Luo, "Principles of neurobiology: Liqun luo: Taylor & amp; francis group," p. 22, Sep 2020. [Online]. Available: https://doi.org/10.1201/9781003053972

[9] G. Young, "Early evolution of the vertebrate eye—fossil evidence," *Evolution: Education and Outreach*, vol. 1, pp. 427–438, 10 2008.

[10] M. Maher, S. Deweerth, M. Mahowald, and C. Mead, "Implementing neural architectures using analog vlsi circuits," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 5, pp. 643–652, 1989.

[11] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[12] M. M., "The silicon retina," *In: An Analog VLSI System for Stereoscopic Vision. The Springer International Series in Engineering and Computer Science (VLSI, Computer Architecture and Digital Signal Processing), vol*, vol. 265., 1994.

[13] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, 2014.

[14] G. Sicard, G. Bouvier, V. Fristot, and A. Lelah, "An adaptive bio-inspired analog silicon retina," in *Proceedings of the 25th European Solid-State Circuits Conference*, 1999, pp. 306–309.

[15] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor,," *Solid-State Circuits, IEEE Journal of*, vol. 43, pp. 566 – 576, 03 2008.

[16] B. Pain and E. R. Fossum, "Approaches and analysis for on-focal-plane analog-to-digital conversion," in *Infrared Readout Electronics II*, E. R. Fossum, Ed., vol. 2226, International Society for Optics and Photonics. SPIE, 1994, pp. 208 – 218. [Online]. Available: https://doi.org/10.1117/12.178483

[17] D. Yang, B. Fowler, and A. El Gamal, "A nyquist-rate pixel-level adc for cmos image sensors," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 348–356, 1999.

[18] A. Bermak, A. Bouzerdoum, and K. Eshraghian, "A vision sensor with on-pixel adc and in-built light adaptation mechanism," *Microelectronics Journal*, vol. 33, pp. 1091–1096, 12 2002.

[19] R. V. Rullen and S. J. Thorpe, “Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex,” *Neural Computation*, vol. 13, no. 6, pp. 1255–1283, 2001.

[20] C. Shoushun and A. Bermak, “A low power cmos imager based on time-to-first-spike encoding and fair aer,” in *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 5306–5309 Vol. 5.

[21] X. Qi, X. Guo, and J. Harris, “A time-to-first spike cmos imager,” in *2004 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 4, 2004, pp. IV–824.

[22] E. Culurciello, R. Etienne-Cummings, and K. Boahen, “A biomorphic digital image sensor,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, 2003.

[23] T. Nguyen, “Total number of synapses in the adult human neocortex,” *Undergraduate Journal of Mathematical Modeling: One + Two*, vol. 3, 05 2013.

[24] 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.

[25] M. Mahowald, “Vlsi analogs of neuronal visual processing: A synthesis of form and function,” May 1992. [Online]. Available: https://resolver.caltech.edu/CaltechCSTR:1992.cs-tr-92-15

[26] C. Posch, D. Matolin, and R. Wohlgenannt, “An asynchronous time-based image sensor,” in *2008 IEEE International Symposium on Circuits and Systems*, 2008, pp. 2130–2133.

[27] R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, “A 240×180 10mw 12us latency sparse-output vision sensor for mobile applications,” in *2013 Symposium on VLSI Circuits*, 2013, pp. C186–C187.

[28] T. Serrano-Gotarredona and B. Linares-Barranco, “A 128<formula formulatype="inline"><tex notation="tex">×</tex> </formula>128 1.5% contrast sensitivity 0.9% fpn 3 µs latency 4 mw asynchronous frame-free dynamic vision sensor

using transimpedance preamplifiers," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, 2013.

[29] M. Kwon, S. Lim, H. Lee, I.-S. Ha, M.-Y. Kim, I.-J. Seo, S. Lee, Y. Choi, K. Kim, H. Lee, W.-W. Kim, S. Park, K. Koh, J. Lee, and Y. Park, "A low-power 65/14nm stacked cmos image sensor," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–4.

[30] 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, "5.10 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," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 112–114.

[31] 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. Ovsiannikov, 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)*, 2017, pp. 66–67.

[32] C. Dupoiron, "Nouveaux paradigmes de capture d'images et traitements associés pour futurs soc en nœuds cmos nanométriques," Ph.D. dissertation, 12 2017.

[33] A. Darwish, L. M. G. Rocha, L. Fesquet, and G. Sicard, "Design of a fully asynchronous image sensor reading system," in *2015 Conference on Design of Circuits and Integrated Systems (DCIS)*, 2015, pp. 1–5.

[34] S. Makovejev, N. Planes, M. Haond, D. Flandre, J.-P. Raskin, and V. Kilchytska, "Self-heating in 28 nm bulk and fdsoi," in *EUROSOI-ULIS 2015: 2015 Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon*, 2015, pp. 41–44.

[35] J. Mäkipää and O. Billoint, "Fdsoi versus bulk cmos at 28 nm node which technology for ultra-low power design?" in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 554–557.

[36] L. Kadura, L. Grenouillet, O. Rozeau, A. Chelnokov, and M. Vinet, "1t pixel sensor based on fdsoi transistor optical back biasing," *IEEE Transactions on Electron Devices*, vol. 66, no. 5, pp. 2249–2255, 2019.

[37] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x 128 120 db 15 µs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[38] A. Darwish, "Capteur d'images événementiel, asynchrone à échantillonnage non-uniforme," Ph.D. dissertation, 06 2016.

[39] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of o(n) complexity," in *Proceedings of the 1st International Conference on Neural Information Processing Systems*, ser. NIPS'88. Cambridge, MA, USA: MIT Press, 1988, p. 703–711.

[40] Z. Gunay and E. Sanchez-Sinencio, "Cmos winner-take-all circuits: a detailed comparison," in *1997 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, 1997, pp. 41–44 vol.1.

[41] A. Fish, V. Milrud, and O. Yadid-Pecht, "High-speed and high-precision current winner-take-all circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 131–135, 2005.

[42] A. M. T. Linn, D. A. Tuan, C. Shoushun, and Y. K. Seng, "Adaptive priority toggle asynchronous tree arbiter for aer-based image sensor," in *2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip*, 2011, pp. 66–71.

[43] J. A. Starzyk and X. Fang, "Cmos current mode winner-take-all circuit with both excitatory and inhibitory feedback," *Electronics Letters*, vol. 29, pp. 908–910, 1993.

[44] A. Demosthenous, S. Smedley, and J. Taylor, "A cmos analog winner-take-all network for large-scale applications," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, no. 3, pp. 300–304, 1998.

[45] T. Serrano-Gotarredona and B. Linares-Barranco, "A high-precision current-mode wta-max circuit with multichip capability," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, pp. 280–286, 1998.

List of Figures

2.1 Individual pixels in an image 9
2.2 Charge Coupled Device operating principle 11
2.3 Schematic diagram of the most common CCD image sensor architecture [4] 12
2.4 3 transistor CMOS image sensor schematic 13
2.5 The main cells in the biological retina [13] 15
2.6 DVS pixel block schematic [13], [15] 16
2.7 (A) Block diagram of the proposed pixel based ADC (B) Voltages of the different nodes of the circuit [18] 17
2.8 The AER pulses from spiking neurons are transmitted serially by broadcasting addresses on a digital bus. Multiplexing is transparent if the encoding, transmission, and decoding processes cycle in less than $\Delta = n.s$, where Δ is the desired spike-timing precision and n is the maximum number of neurons that are active during this time [24] 19
2.9 Event based IS basic architecture: many detail communication signals in figure (a) were omitted for the sake of simplicity 21
2.10 ATIS readout output [26], [13] 22
2.11 DAVIS: APS pixel schematic and the DVS block schematic [27]. 23

3.1 Some kernels made of TFS and DVS pixels (DVS pixel colored in green and the TFS pixel colored in blue) 32
3.2 The high level simulation main steps 33
3.3 Illustration of the two test cases and the simulation parameters 34
3.4 Spatial redundancy suppression principle in [33] 36

3.5 Illustration of the verification problem 37
3.6 The total number of generated events through the whole duration of stimuli: (a) and (b) represent the total number of events generated in each test case per DVS threshold, (c) is a comparison of the different architectures with a standard image sensor (at 5% DVS threshold) 38
3.7 Verification scaling with the resolution, and Spatial redundancy suppression effect on verification 40
3.8 (a) The viewed scene (b) The resulting image generated by the pixel matrix of kernel (f) at 1.5% DVS threshold and no spatial redundancies suppression. (c) The flow of the new events updating the image memory . . . 41
3.9 Effect of applying spatial redundancy suppression, kernel (f) 42

4.1 Cross-section comparison of bulk and FDOSI MOS transistors 47
4.2 Front and back side illumination technology comparison 48
4.3 DVS pixel 49
4.4 TFS pixel block schematic 51
4.5 TFS pixel communication signals 52
4.6 Pixel matrix view illustrating the need for verification 53
4.7 Abstracted full circuit schematic 54
4.8 Averaging circuit(*net1* and *Vin* link to the DVS pixel) 54
4.9 DVS pixel schematic 56
4.10 TFS pixel schematic 57
4.11 Trigger logic circuit and signals diagram 59
4.12 The two designed pixel matrices 60
4.13 The average event rate per pixel per second of the TFS pixel in each architecture 61
4.14 The average power consumption consumed by the TFS pixel in each architecture and test case 62
4.15 The average power consumption consumed by the DVS pixel in each architecture and test case 62

5.1 The event based camera block diagram 67
5.2 Block schematic of the readout system 68

5.3 Block schematic of the processing element 69
5.4 The FSM controlling the processing element 70
5.5 TDC time stamping example with spatial redundancy suppression of 4 grey levels or Tmax/4 72
5.6 The readout simulation 73
5.7 Simulation output images for different Δt values (the eye) 76
5.8 Simulation output images for different Δt values (the monkey) 77

6.1 Test setup 84
6.2 The importance of the sensor area and the objective lens projected image area: (a) result in image cropping (b) results in not using all of the pixel matrix efficiently 85
6.3 Scene 87
6.4 Live video seen at the monitor 88
6.5 The hybrid image snesor camera 89
6.6 The pixel matrix schematic 91
6.7 Chip layout 92
6.8 Row and column requests activity recorded on the scope 93

7.1 Lazarro winner take all circuit [39] 100
7.2 Winner take all arbiter 101
7.3 three cells, three output arbiter (28 nm, Vdd=1V) 102
7.4 Scope screen captures of the arbiter output, before and after the trigger RSS0(outd), RSS1(out1), RSS2(out2) for (Vpol and Vpol_s at 500mv) . . 104

List of Tables

2.1 Comparison of CIS with event based image sensors in the art 24

2.2 Comparison of CIS with industry manufactured event based image sensors in the art 25

4.1 Averaging and non averaging physical implementation comparison 60

4.2 Post layout power consumption results 63

5.1 Output images quality and verification numbers 75

6.1 Integration time and TDC period for room lighting contrast 87

7.1 The arbitration latency time vs different values of Vpol and Vpol_s (chip test results) 104

7.2 The power cinsumption and the arbiter cell currents during activation(Simulation, Vpol_s = 350 mv) 105

7.3 Comparison of the experimental and simulation results of our three input arbitration circuit. (*) Obtained from simulation results 106

Résumé

Dans le cadre du projet européen OCEAN 12, cette thèse de doctorat a réalisé la conception, la mise en œuvre, les tests d'un capteur d'image basé sur les événements, ainsi que la publication de plusieurs articles scientifiques dans des conférences internationales, y compris des conférences renommées telles que le Symposium international sur les circuits et systèmes asynchrones (ASYNC). La conception de capteurs d'images basés sur les événements, qui sont sans trame, nécessite une architecture dédiée et une logique asynchrone réagissant aux événements. Tout d'abord, cette thèse donne un aperçu des architectures basées sur une matrice de pixels hybrides comprenant des pixels TFS et DVS. En effet, ces deux types de pixels sont capables de gérer respectivement la redondance spatiale et la redondance temporelle. L'un des principaux résultats de ce travail est de tirer parti de la présence des deux types de pixels dans un capteur d'image afin de réduire le débit de bits de sortie et la consommation d'énergie. Ensuite, la conception des pixels et de la lecture en technologie FDSOI 28 nm de STMicroelectronics est détaillée. Enfin, deux capteurs d'image ont été implémentés dans une puce de test et testés.

Mots-clés : Événementiel, Capteur d'image, Vision dynamique, Asynchrone

Abstract

In the framework of the OCEAN 12 European project, this PhD achieved the design, the implementation, the testing of an event based image sensor, and the publication of several scientific papers in international conferences, including renowned ones like the International Symposium on Asynchronous Circuits and Systems (ASYNC). The design of event-based image sensors, which are frameless, require a dedicated architecture and an asynchronous logic reacting to events. First, this PhD gives an overview of architectures based on a hybrid pixel matrix including TFS and DVS pixels. Indeed, this two kind of pixels are able to manage the spatial redundancy and the temporal redundancy respectively. One of the main achievement of this work is to take advantage of having both pixels inside an imager in order to reduce its output bitstream and its power consumption. Then, the design of the pixels and readout in FDSOI 28 nm technology from STMicroelectronics is detailed. Finally, two image sensors have been implemented in a testchip and tested.

Keywords : Event Based, Image sensor, Event driven, Dynamic vision, Asynchronous