



**HAL**  
open science

# Nouveaux paradigmes de capture d'images et traitements associés pour futurs SoC en nœuds CMOS nanométriques

Camille Dupoiron

► **To cite this version:**

Camille Dupoiron. Nouveaux paradigmes de capture d'images et traitements associés pour futurs SoC en nœuds CMOS nanométriques. Micro et nanotechnologies/Microélectronique. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAT100 . tel-01801134

**HAL Id: tel-01801134**

**<https://theses.hal.science/tel-01801134v1>**

Submitted on 28 May 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES**

Spécialité : **Nano-Electronique et Nano-Technologies**

Arrêté ministériel : 25 mai 2016

Présentée par

**Camille DUPOIRON**

Thèse dirigée par **Gilles SICARD**  
et co-encadrée par **Arnaud VERDANT**

préparée au sein du **Laboratoire d'Electronique et des Technologies  
de l'information, CEA-LETI**  
dans l'**École Doctorale EEATS**

## **Nouveaux paradigmes de capture d'images et traitements associés pour futurs SoC en nœuds CMOS nanométriques**

Thèse soutenue publiquement le **4 décembre 2017**,  
devant le jury composé de :

**Mme. Alice CAPLIER**

Professeur des universités, CNRS, GIPSA-Lab, Grenoble INP,  
Présidente

**M. Wilfried UHRING**

Professeur des universités, CNRS, ICube-SMH, Université de  
Strasbourg, Rapporteur

**M. Dominique GINHAC**

Professeur des universités, CNRS, Le2i, Université de Bourgogne,  
Rapporteur

**M. François BERRY**

Professeur des universités, CNRS, Institut Pascal, Université Clermont  
Auvergne, Examineur

**M. Gilles SICARD**

Ingénieur de recherche, HDR, CEA-LETI, Université Grenoble Alpes,  
Directeur de thèse.

**M. Arnaud VERDANT**

Ingénieur de recherche, CEA-LETI, Université Grenoble Alpes, Membre





# Remerciement

Je remercie très sincèrement Gilles Sicard et Arnaud Verdant pour leur encadrement éclairé, tous leurs conseils et toutes leurs remarques toujours constructives ainsi que leur accompagnement dans mes travaux.

Je remercie très chaleureusement Madame Alice Caplier et Messieurs Wilfried Uhring et Dominique Ginhac pour m'avoir fait l'honneur de respectivement, présider ce jury et évaluer ce travail de thèse. Je remercie également Monsieur François Berry pour avoir participé à ce jury de thèse.

Je remercie Michael Tchagaspanian et Fabrice Guellec de m'avoir accueilli au sein du laboratoire L3i.

Je remercie également les membres du laboratoire. Margaux Vigier, Bertrand Dupont, Laurent Millet, Thomas Pilloix et Amaury Nogier pour leurs aide et explications pédagogiques tout au long de ces trois ans, ainsi que Benjamin Caillat. Nicolas Monnier, Wissam Benjilali, Fabrice Guellec, Laurent Alacoque, William Guicquero, Jean-Alain Nicolas, Arnaud Peizerat, Jean-Pierre Rostaing, Kinia Barjean, Nils Margotat, Yoann Seauve, Guillaume Renaud, et Simon Emberger pour leur bonne humeur, la vie de labo agréable et les échanges nombreux tout au long de la thèse.

Je remercie mes amis et amies, plus ou moins loin, pour leur soutien et leur amitié incessants.

Claire, Samuel, Arielle, Agathe, Luce, Oriane, Adrien, et Guillaume.

Merci à Simon.

Merci à ma famille.

Et merci à celles et ceux que j'ai oublié de citer ainsi qu'à celles et ceux qui m'ont permis d'arriver jusqu'ici.



# Table des matières

<b>Publications et brevets</b>	7
<b>Introduction</b>	9
<b>1 Chapitre I : Acquisition d'image</b>	13
1.1 Introduction .....	13
1.2 Les capteurs d'images usuels : généralités .....	13
1.2.1 La chaine d'acquisition : du photon à l'information numérique .....	13
1.2.2 Caractéristiques principales des capteurs .....	19
1.2.3 Architectures pixel .....	20
1.2.4 Problématique du traitement d'images dans les capteurs de vision .....	26
1.2.5 Avantages et inconvénients des différentes architectures pixels usuelles .....	27
1.3 Les capteurs d'images événementiels.....	28
1.3.1 Le domaine temporel.....	28
1.3.2 Lecture des événements.....	31
1.3.3 Traitement des événements.....	35
1.3.1 Avantages et inconvénients des capteurs d'images événementiels .....	36
I.4 Conclusion de l'état de l'art, positionnement et justification des contributions de la thèse .....	37
<b>2 Chapitre II : Les technologies avancées pour l'imagerie</b>	40
2.1 Introduction .....	40
2.2 Matériaux photosensibles .....	40
2.3 Collecte des électrons .....	41
2.4 Technologies avancées.....	42
2.4.1 Evolution de la taille de la lithographie et conséquences .....	42
2.4.2 Technologies avancées pour l'imagerie .....	43
2.5 FD-SOI 28nm.....	43
2.5.1 Principe et avantages.....	43
2.5.2 Fuites de courant et conséquences pour l'acquisition d'image .....	45
2.6 Technologies émergentes au LETI .....	48
2.6.1 Etendre les fonctionnalités des transistors FDSOI en les rendant sensibles à la lumière et exploiter ces nouvelles fonctionnalités.....	48
2.6.2 Implémentation 3D .....	53
2.7 Travaux de test réalisés pendant cette thèse.....	55
2.7.1 Vérification de la sensibilité à la lumière du FD-SOI 28nm.....	55
2.7.2 L'approche envisagée de l'architecture par rapport aux avantages et inconvénients technologiques et des acquisitions des imageurs.....	61

2.8	Conclusion.....	62
<b>3</b>	<b>Chapitre III : Mécanismes d'acquisitions d'images pour technologie avancée</b>	<b>64</b>
3.1	Architecture TTFS avec mécanisme d'inhibition: Time to first spike – Inhibition Mechanism: IM-TTFS.....	64
3.1.1	Le domaine temporel.....	64
3.1.2	Time to first spike et les problèmes associés .....	66
3.1.3	Mécanisme d'inhibition .....	68
3.2	Transmission de données lissée dans le temps et utilisation de bus multiples .....	78
3.2.1	Principe de fonctionnement .....	79
3.2.2	Réduction de donnée par voisin direct .....	81
3.2.3	Résultats de simulation .....	82
3.2.4	Architecture développée.....	83
3.2.5	Paramètres ajustables dans l'architecture et simulations MATLAB....	85
3.3	Comparaison des deux architectures.....	88
3.4	Conclusion.....	89
<b>4</b>	<b>Chapitre IV : Co-design acquisition et traitement</b>	<b>90</b>
4.1	Descripteurs : principe et applications.....	90
4.1.1	Carte binaire.....	91
4.1.2	Histogramme des gradients orientés : HoG .....	96
4.2	Architecture .....	105
4.2.1	Pixel.....	105
4.2.2	Système d'inhibition.....	109
4.2.3	Système de lecture synchrone .....	113
4.2.4	Simulation Cadence et résultats .....	113
4.2.5	Discussions sur la consommation du système.....	114
4.3	Les apports de la solution .....	115
4.3.1	Par rapport à une solution non intégrée .....	115
4.3.2	Comparaison à l'état de l'art.....	116
4.4	Conclusion.....	118
	<b>Conclusion et perspectives</b>	<b>120</b>
	<b>Références</b>	<b>122</b>
A.	Annexe 1 : Architecture IM-TTFS .....	132
	Logigramme du pixel :.....	132
	Architecture du Handshake.....	132
	Architecture de l'inhibition .....	133

---

Résumé.....134

# Publications et brevets

## 3 publications publiées :

- C. Dupoirion, A. Verdant, and G. Sicard, "**Trade-off between the number of bits per pixel and motion detection quality for a low power image sensor**" *Electronic Imaging*, San Francisco, 2016, pp. 1-6.
- C. Dupoirion, A. Verdant and G. Sicard, "**Smart pixel architecture for low power CMOS image sensor: Time-to-first spike with inhibition mechanism**," *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, Strasbourg, 2017, pp. 49-52. <http://ieeexplore.ieee.org/document/8010102/>
- C. Dupoirion, A. Verdant, and G. Sicard, "**An Event-Based, Frame-Based Image Acquisition Mechanism for CMOS Image Sensors**," *2017 New Generation of CAS (NGCAS)*, Genova, 2017, pp. 221-224. <http://ieeexplore.ieee.org/document/8052309/>

## 2 brevets en cours de dépôt :

- Camille DUPOIRON, Gilles SICARD, Arnaud VERDANT, "**Capteur matriciel à codage temporel sans arbitrage**", N° 17 53032, 07/04/2017
- Camille DUPOIRON, William GUICQUERO, Gilles SICARD, Arnaud VERDANT "**Procédé de traitement d'une image**" N° 17 56555, 11/07/2017



# Introduction

L'internet des objets se définit comme un ensemble d'objets connectés pouvant partager des informations sur l'environnement dans lequel ils sont placés. Les principales applications concernent les domaines de la maison intelligente, des technologies portables, de la ville intelligente, de l'industrie 4.0 et de la voiture connectée. Les systèmes IOT, composés d'une multitude d'éléments autonomes et interconnectés appelés nœuds IOT, observent l'environnement dans lequel ils sont placés, envoient et reçoivent certaines informations par le biais d'internet, Figure 0-1-a. Selon Business Insider, la prévision du nombre de systèmes IOT connectés à internet s'élève à 24 milliards en 2020, Figure 0-1-b. Ils sont à différencier des objets connectés tels que les téléphones et montres portables, les tablettes et les ordinateurs qui eux, s'élèveront jusqu'au nombre de 10 milliards en 2020.

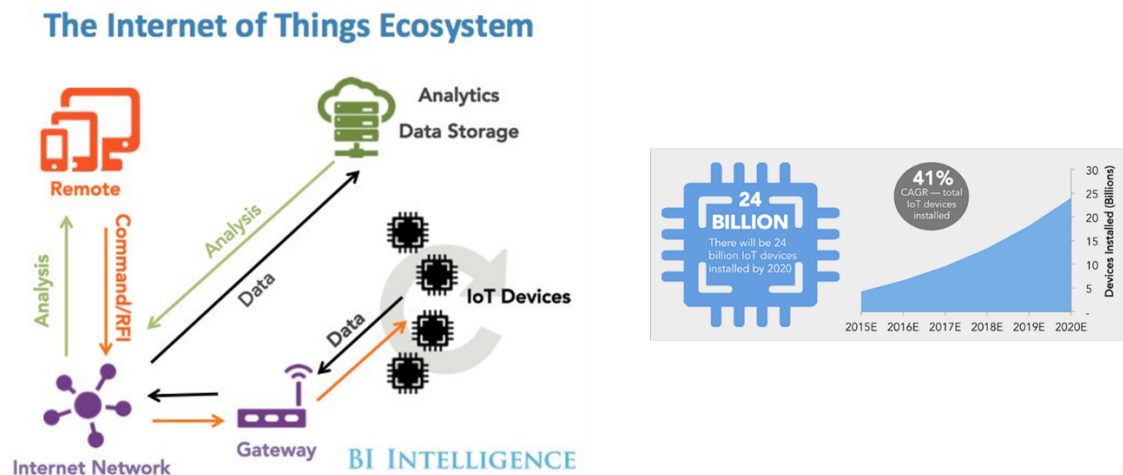


Figure 0-1: a) Ecosystème de l'internet des objets b) Evolution du marché de 2015 jusqu'en 2020

Chaque nœud IOT peut être composé de différents capteurs, d'un microcontrôleur, d'un module RF permettant de transmettre et de recevoir des informations et d'un module de gestion de l'alimentation et de l'énergie. Ils peuvent également avoir la capacité de gérer des signaux analogiques et mixtes. Afin d'optimiser la nécessairement très faible consommation d'un nœud IOT, l'intégration de ces différentes fonctions de manière monolithique dans un seul et même circuit est une approche séduisante car minimisant les interconnexions entre ces différentes fonctions. L'approche SoC (system on chip ou système sur puce) hétérogène permet effectivement d'optimiser le système et d'obtenir de faibles consommations d'énergie au niveau des IP ou du système global. C'est dans ce cadre que s'inscrit le projet L-IOT au CEA-LETI, permettant d'avoir un nœud polyvalent qui peut répondre à différents besoins applicatifs. Ce nœud comporte une partie toujours disponible (always-on) et une partie disponible sur demande (on-demand) dans lesquelles les différentes fonctions sont retrouvées comme illustré sur la Figure 0-2.

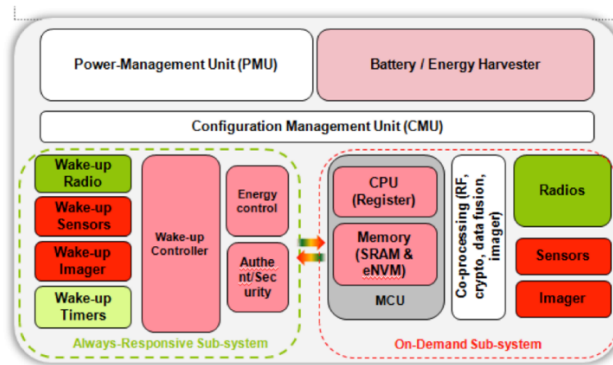


Figure 0-2: Nœud L-IOT : plateforme adaptative [LIOT2015]

La principale problématique dans la conception d'un système sur puce hétérogène est celle de devoir utiliser une seule et même technologie pour l'ensemble des fonctions à implémenter. Usuellement, les capteurs et systèmes analogiques / RF performants utilisent des nœuds technologiques moins avancés que les systèmes digitaux. Cela implique donc d'être théoriquement moins performants dans tels ou tels domaines en fonction de la technologie choisie.

L'apparition de technologies très avancées telles que la FDSOI 28nm a cependant changé la donne. Cette technologie propose, grâce à ses caractéristiques intrinsèques, la possibilité d'implémenter des fonctions numériques avec un rapport performance de calcul / consommation exceptionnel [Beigne2015a]. Cette technologie permet également d'implémenter des fonctions analogiques / RF très basse consommation avec des performances également exceptionnelles [Beigne2015b]. Cette technologie est donc une très bonne candidate pour implémenter un SoC IoT pour peu que l'on puisse implémenter avec des capteurs suffisamment performants. C'est ce que propose d'étudier cette thèse dans le domaine de l'imagerie.

Un exemple de système sur puce intégrant un capteur d'image serait [Schmitz2017]. L'architecture présentée consomme 31mW pour un capteur de résolution spatiale de 80x64 et permet de sortir une image statique, de réaliser du filtrage d'image ainsi que de l'extraction de contour. Cependant, l'architecture est effectuée en 180µm et n'a pas de capacité de transmission de donnée en RF, la puce doit donc être associée à d'autres puces afin de constituer un système IOT complet.

Cette thèse s'inscrit plus particulièrement dans la partie « always-on » de ce système. L'imageur aura donc à charge de basiquement détecter un évènement dans la scène filmée et de déclencher le réveil de la partie « on-demand » quand il sera nécessaire. Il sera donc ici question d'un détecteur plus que d'un imageur classique compte tenu de la contrainte always-on où la problématique de consommation d'énergie est extrêmement forte. Les caractéristiques primaires de ce capteur d'images se dessinent ici : très faible consommation, faible résolution, prétraitement d'images embarqué (détection). Une caractéristique intéressante sur le plan de la simplification matérielle du capteur donc de sa consommation et la non-nécessité forte d'avoir une image en sortie du capteur.

La première problématique de cette thèse sera d'évaluer les contraintes technologiques imposées par le FDSOI 28nm puis d'évaluer la faisabilité d'un tel détecteur et sous quelles conditions. La seconde problématique sera d'évaluer les

contraintes systèmes. En effet, la consommation énergétique de ce capteur doit être faible puisqu'il est destiné à être intégré dans la partie always-on du nœud IOT. Enfin, il faudra prendre en compte la problématique concernant l'identification des pré-traitements à intégrer au sein du capteur. Ces pré-traitements doivent simplifier les algorithmes de détections à effectuer en pied de colonne de la matrice ou même hors de la matrice.

Les contributions de la thèse sont les suivantes. Tout d'abord une étude a été menée sur l'impact des non-idéalités de la technologie pour une acquisition d'image en tension comparé à une acquisition d'image dans le domaine temporel. Puis un test chip a été réalisé pour évaluer la réponse électro-optique de pixels 3T en technologie FDSOI 28nm. Ces pixels ont différents modèles de photodiode. Puis un circuit contenant dix matrices de 3 par 3 pixels a été réalisé. Ce circuit utilise les possibilités de design offertes par une implémentation 3D parallèle CoolCube en FDSOI 28nm. Egalement, l'étude d'une architecture d'acquisition asynchrone utilisant la densité de métaux disponible en FDSOI 28nm, a été réalisée. Elle a été comparée à une précédente architecture étudiée : l'IM-TTFS. Enfin, l'étude d'une architecture robuste pour une implémentation en technologie nanométrique de type FDSOI 28nm et intégrant des primitives de traitements dans le plan focal a été menée.

Dans le chapitre 1, les différents modes d'acquisitions d'une image dans un imageur seront expliqués ainsi que les différents types de traitements d'image intégrés possibles pour chacun. Les modes d'acquisitions usuels correspondent à une lecture en tension ou en courant suivie d'un convertisseur analogique numérique (analog to digital convertor : ADC). L'ADC peut être global, en pied de colonne ou intégré dans le pixel. Puis les modes d'acquisitions événementiels seront abordés avec l'explication du domaine temporel et des différents mécanismes de lecture des événements. Une conclusion de cet état de l'art sera faite pour établir le positionnement ainsi que la justification des choix techniques faits durant cette thèse.

Dans le chapitre 2, les avantages et les inconvénients de la technologie FDSOI 28nm pour l'imagerie sont étudiés. Puis différentes technologies émergentes du LETI autour du FDSOI 28nm sont exploitées pour l'imagerie. L'utilisation de la photodiode sous le BOX d'un transistor permet de décaler la tension de seuil d'un inverseur en fonction de la lumière reçue. Cet inverseur peut être exploité afin de réaliser un pixel événementiel. La technologie 3D stack CoolCube permet de réaliser un imageur avec une couche photodiode et une couche transistor afin de maximiser la densité du pixel. A la fin du chapitre, et aux vues des différents arguments présentés sur la technologie, une acquisition dans le domaine temporel est justifiée.

Dans le chapitre 3, deux architectures événementielles sont détaillées. La première appelée IM-TTFS (Time-To-First-Spike with Inhibition Mechanism) a commencé à être étudiée lors d'un stage effectué en amont de cette thèse. La modélisation et les simulations de l'architecture complète ont été réalisées afin de pouvoir la comparer à une seconde architecture étudiée durant cette thèse. La seconde architecture permet de lisser temporellement l'histogramme des événements de la matrice.

Dans le chapitre 4, une architecture permettant d'extraire des descripteurs de l'image, est étudiée. Elle est réalisée en exploitant l'architecture IM-TTFS. Des simulations MATLAB sont d'abord réalisées afin de vérifier les traitements implémentés puis l'architecture est étudiée sous Cadence et comparée à l'état de l'art.





# 1 Chapitre I : Acquisition d'image

## 1.1 Introduction

Les capteurs d'images à base de circuits intégrés sont commercialisés depuis la fin des années 60, et ont remplacé petit à petit les pellicules argentiques dans les appareils photos à partir des années 90. Avec l'arrivée des smartphones, des tablettes et de l'internet des objets, les capteurs d'images en lumière visible, correspondant à la lumière de longueur d'onde entre 400nm et 800nm, ont pris et prennent encore une part de plus en plus importante du marché de l'électronique.

En 1969, le capteur à dispositif de transfert de charge dit CCD pour « charge couple device » est inventé dans les laboratoires Bell. Il permet de transférer les charges récoltées ligne par ligne grâce à des puits de potentiels successifs. Le CCD a pour avantage d'avoir un faible niveau de bruit, une grande dynamique, ainsi qu'un grand facteur de remplissage. Néanmoins, la technologie CCD ne permettant pas l'intégration de fonctions analogiques nécessaires à la numérisation du signal, un deuxième circuit externe en technologie CMOS par exemple est nécessaire pour assurer la lecture du signal en sortie du plan focal. Dans les années 1980, le développement des technologies CMOS a permis d'intégrer petit à petit des matrices de photodiodes. Comparé à la technologie CCD, les technologies CMOS ont apporté plusieurs avantages tels que la réduction de la tension d'alimentation, et une consommation d'énergie plus faible. Les technologies CMOS permettent également d'intégrer certains traitements de l'information lumineuse, intégrer sur la même puce un ADC (analog to digital convertor) ainsi que des commandes de signaux logiques. Enfin de permettre l'accès aléatoire aux données comme une RAM (random access memory). De plus, en 20 ans, les capteurs CMOS ont atteint les performances des capteurs CCD, et c'est pourquoi, les capteurs CMOS se sont imposés sur le marché. Actuellement ils sont utilisés dans la vaste majorité des applications. Dans cette thèse nous ne parlerons donc que de capteurs CMOS.

Dans ce chapitre, les différents modes d'acquisitions de l'image seront expliqués ainsi que les différents traitements d'image intégrés possibles pour chacun. Les modes d'acquisitions usuels correspondent à une lecture en tension ou en courant suivie d'un convertisseur analogique numérique. L'ADC peut être global, en pied de colonne ou intégré dans le pixel. Puis les modes d'acquisitions événementiels seront abordés avec l'explication du domaine temporel et des différents mécanismes de lecture des événements. Une conclusion de cet état de l'art sera faite pour établir le positionnement ainsi que la justification des choix faits durant cette thèse.

## 1.2 Les capteurs d'images usuels : généralités

### 1.2.1 La chaîne d'acquisition : du photon à l'information numérique

La chaîne d'acquisition complète est présentée sur la Figure 1-1. Le rayonnement incident, émis ou réfléchi par un objet distant de la scène observée, est tout d'abord focalisé sur la surface photosensible à l'aide d'un système d'optique. Le capteur permet de transformer la lumière incidente en un signal électrique

analogique. Un filtre de couleur appelé filtre de Bayer est placé en amont du capteur afin d'obtenir l'information de couleur de l'image. Chaque pixel produit un signal correspondant à une des trois couleurs (rouge, vert, bleu) en fonction de la couleur du filtre placé devant lui. Le signal analogique des pixels est lu puis numérisé grâce à un convertisseur. La couleur est ensuite reconstruite par interpolation spatiale dans l'unité de traitement aval. Pour des capteurs d'image destinés à produire des images agréables à l'œil, d'autres traitements relatifs à la qualité de l'image peuvent être effectués comme la balance des blancs, une correction de couleur ou la correction de défauts dû aux imperfections de l'optique et de l'imageur. Enfin, l'image est compressée puis stockée.

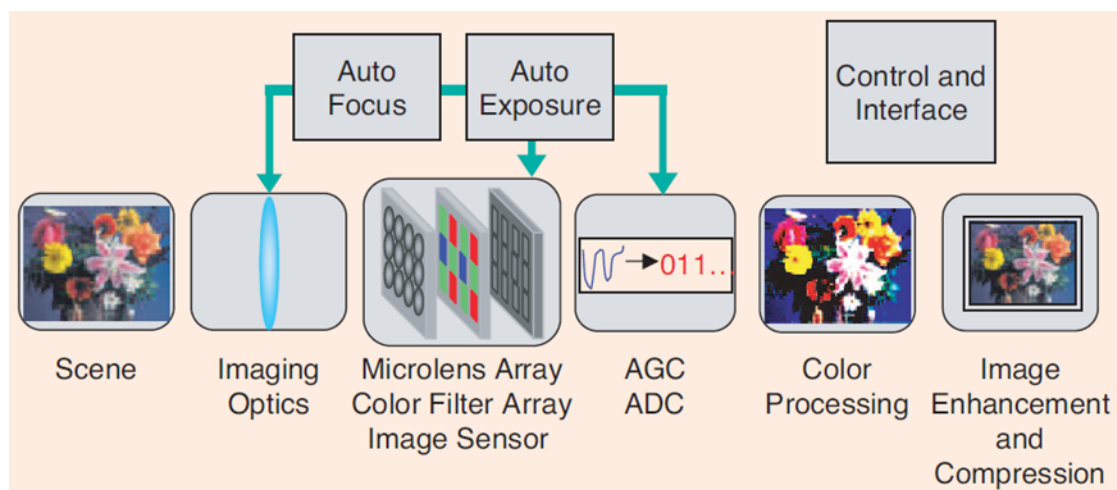


Figure 1-1: Chaîne complète d'acquisition d'image

Dans cette thèse, nous nous intéresserons à la partie capteur sans notion de couleurs et sans traitement d'image afin de rendre l'image agréable à l'œil mais avec des traitements pour extraire des informations descriptives de la scène observée telles que du mouvement ou de la reconnaissance de forme. La chaîne de l'image, du pixel à l'ADC est présentée dans la Figure 1-2. Le capteur est organisé suivant une matrice de « picture elements » dits pixels. Ces pixels sont de forme carrée, et non rectangulaire afin que l'image ne soit pas déformée. Les pixels sont contrôlés par le décodeur de ligne permettant de lire une image ligne par ligne, en activant le signal « sélection ligne » ou « row select » pour toute une ligne. Les informations des pixels dont la ligne est active sont transférées en bout de colonne, appelé « pied de colonne » grâce aux bus de colonne. Puis les signaux analogiques sont mis en forme par des amplificateurs, ce qui permet, notamment, de supprimer les imperfections liées à l'architecture du pixel. Enfin les signaux sont numérisés en mots de  $n$  bits par les convertisseurs analogique-numérique et bufférisés vers la sortie.

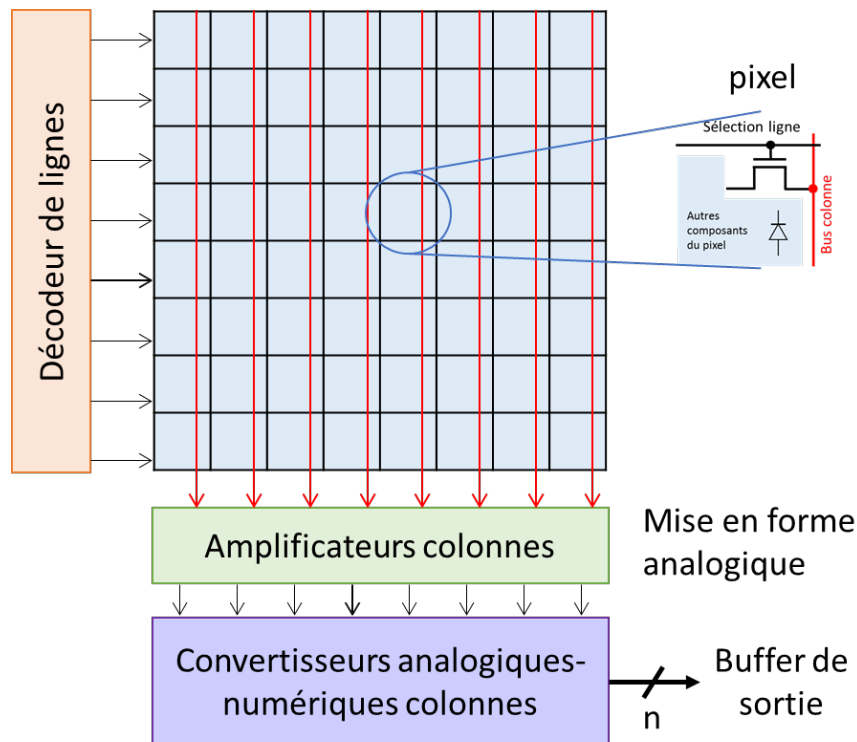


Figure 1-2: Chaîne de l'image du pixel à l'ADC

### 1.2.1.1 Lecture des informations

Il existe différents mécanismes de lecture de la matrice une fois l'information lumineuse récupérée par les pixels.

- Rolling shutter

Le rolling shutter est le mécanisme de lecture le plus classique. Le rolling shutter consiste à lire une image ligne par ligne, en activant, grâce au décodeur d'adresse, le signal de sélection de ligne pour toute une ligne. Cependant, afin que le temps d'intégration soit équivalent sur chaque ligne, le signal reset est décalé sur chaque ligne d'un temps de lecture, Figure 1-3.a. Cela peut créer des défauts, différents du flou de mouvement, dit « motion blur », lorsqu'un objet bouge à une vitesse élevée dans une image. L'exemple d'une hélice en train de tourner peut être cité Figure 1-3.b.

- Global shutter

Le global shutter, quant à lui, permet d'éviter les artéfacts du rolling shutter dans les images, Figure 1-3.d. Pour cela le reset se fait de manière globale dans la matrice, puis les valeurs des pixels sont mémorisées dans le pixel et enfin évacuées ligne par ligne, Figure 1-3.c.

Il est à noter aussi qu'en pied de colonne se trouvent généralement un amplificateur et un ADC. L'ADC peut s'implémenter à différents endroits comme il sera discuté dans la partie 1.2.1.2. A la sortie du système de lecture l'information est analogique, il faut donc la convertir en numérique.

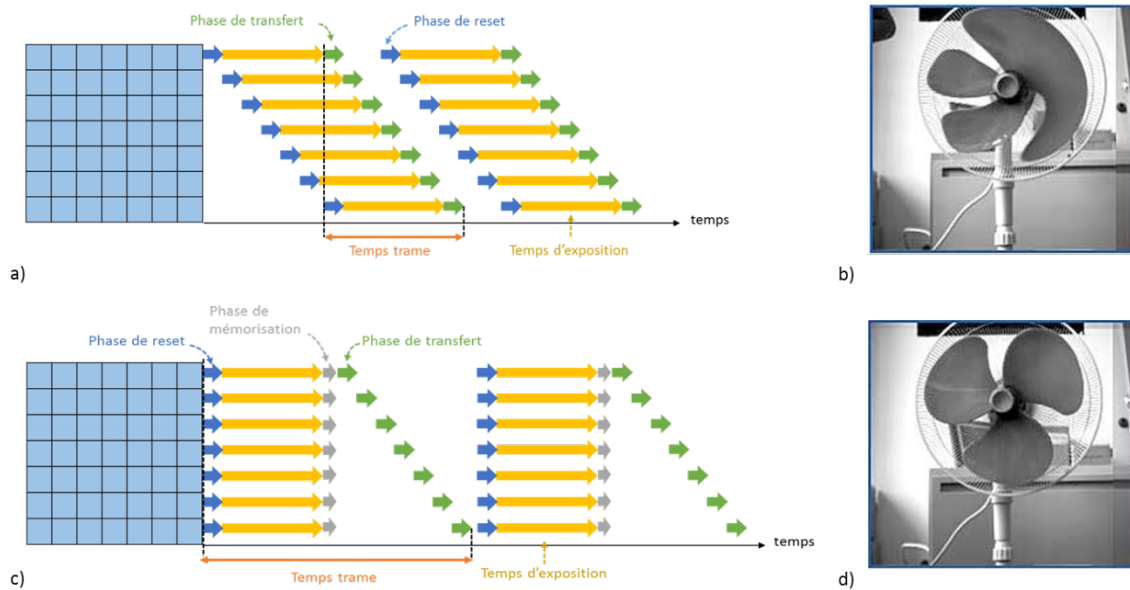


Figure 1-3: a) Rolling shutter, le reset est décalé d'une ligne à une autre d'un temps de lecture b) effet du rolling shutter sur un objet en mouvement c) global shutter, le reset est simultané pour toute la matrice, puis les valeurs sont mémorisées et enfin évacuées ligne par ligne d) L'intégration en global shutter n'apporte pas de déformation sur les objets en mouvement contrairement au rolling shutter. Cependant le motion blur sera à prendre en compte dans les deux modes d'intégration. Les images proviennent du site : <https://ieba.wordpress.com/2007/11/01/rollshut-2/>

### 1.2.1.2 Problématique de la conversion analogique numérique

L'ADC (analog to digital converter) ou CAN (convertisseur analogique numérique), permet de transformer une grandeur électrique (tension ou courant) analogique, en mot binaire sur  $n$  bits. Dans le cas de l'imagerie, on souhaite numériser les valeurs électriques sortant de chaque pixel, qui traduisent l'intensité lumineuse de la scène. Les ADC peuvent être intégrés à la matrice de trois manières différentes : un ADC global pour toute la matrice, un ADC placé en pied de chaque colonne ou un ADC par pixel comme montré sur la Figure 1-4.

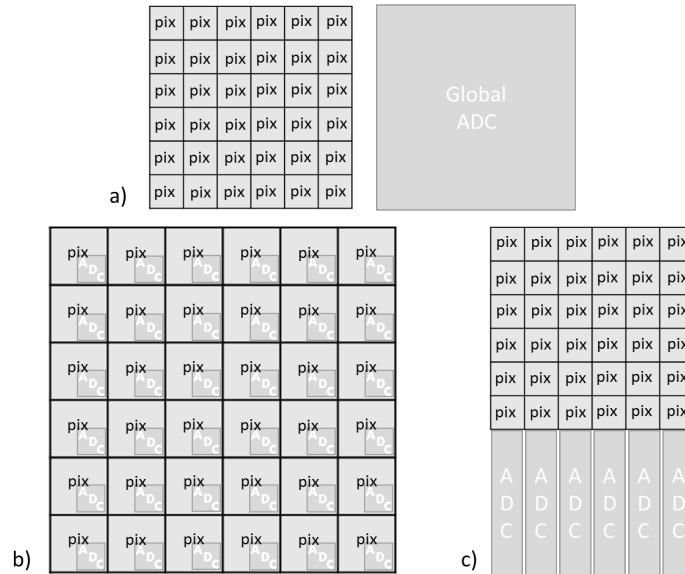


Figure 1-4: ADCs dans les matrices de pixels a) ADC global pour l'intégralité de la matrice b) ADC intégré dans chaque pixel c) ADC en pied de colonne

Un ADC global pour l'intégralité de la matrice permet d'avoir une surface de silicium réduite, les données doivent alors être traitées une à une, ce qui implique que l'ADC doit avoir une vitesse de conversion très rapide, c'est ce qui limitera la vitesse de lecture de l'imageur ou la taille de sa matrice. Les ADC implémentés en pied de colonnes permettent d'avoir une meilleure résolution ainsi qu'une vitesse de lecture plus importante, c'est la technique la plus utilisée actuellement. En revanche l'implémentation physique est contrainte par le pas de pixel. En effet puisqu'il y a un ADC au pied de chaque colonne, la largeur de l'ADC doit correspondre au pas du pixel. De plus, les données arrivant parallèlement il faut prévoir le matériel supplémentaire afin d'acheminer et d'enregistrer les données simultanément. Enfin la troisième possibilité serait d'implémenter un ADC par pixel et donc dans le plan focal directement, c'est le principe des pixels digitaux (DPS) évoqué en partie I.A.1.c. Dans ce cas, la vitesse de conversion est bien plus rapide que dans les autres cas [LeneroBardallo2014], mais cela implique de nombreux inconvénients tels qu'un facteur de remplissage assez bas, et donc une sensibilité faible, ainsi qu'une complexité accrue de l'architecture pixel et de son pilotage. Les avancées en architecture 3D pourront potentiellement permettre de résoudre ces problèmes, puisque les ADCs seront implémentés dans une autre couche que celle sur laquelle sera placés les photodiodes.

Les ADC pour imageurs doivent répondre à certaines contraintes [LeneroBardallo2014] notamment sur la résolution, le bruit aléatoire, l'espace utilisé pour l'implémentation, la vitesse de conversion ainsi que la consommation d'énergie. En effet, la résolution a un impact sur les différentes applications pour lesquelles un imageur serait dédié. Les imageurs habituellement utilisés pour la photographie sont sur 10 bits effectifs (ENOB : effective number of bits). Au contraire lorsque les images sont dédiées à être utilisées pour du machine learning, un nombre effectif de bits plus important, 14 bits ou plus, sera utilisé pour l'ADC. Un bruit de quantification apparaîtra mais il peut être réduit par l'exploitation du shot noise du pixel [LeneroBardallo2014]. L'ADC induit du FPN (voir paragraphe 1.2.2.4) dû aux problèmes d'appariement – dit mismatch – des transistors d'un ADC à un autre.

L'espace utilisé pour l'implémentation dépend de la façon dont l'ADC est utilisé, de manière globale, en pied de colonne ou dans le pixel. La vitesse de conversion est un compromis entre la vitesse de lecture, la résolution de conversion et la surface allouée à l'ADC.

Pour les capteurs d'image industriels actuels, la résolution spatiale ainsi que le nombre d'image acquise par seconde tendent à être maximisées, ce qui implique une vitesse de conversion plus élevée. Cependant, la consommation d'énergie des ADC augmente globalement linéairement avec la résolution de l'ADC et sa vitesse de conversion, par type d'ADC donné [LeneroBardallo2014], comme le montre le Table 1-1.

Si la consommation du capteur est prise au niveau système, la consommation de l'ADC représente plus de 40% de la consommation d'énergie totale [Choi2015], sur la Figure 1-5 la consommation de l'ADC est une grande part de la consommation de l'analogique ainsi qu'une partie du digital. Avec l'émergence des objets connectés portables, la consommation devient un élément clef, c'est pourquoi la réduction de la consommation de l'ADC, voire même sa suppression devient une question cruciale.

	Type d'ADC	Résolution	Consommation d'énergie	Vitesse de conversion
Jeon2017	Extended counting	12bits	22 $\mu$ W	185kS/s
Tang2016	Cyclic	12bits	120 $\mu$ W	N/A
Tsai2012	SAR	10bits	35.46 $\mu$ W	240kS/s
Matsuo2008	SAR	14bits	41 $\mu$ W	3,5MS/s
Lim2010	S. Slope	12bits	100 $\mu$ W	31,7kS/s
Chae2011	Sigma Delta	12bits	148 $\mu$ W	0,14MS/s
Park2009	Cyclic	13bits	300 $\mu$ W	19,2MS/s

Table 1-1: Table de comparaison de consommation d'énergie pour différents types d'ADC avec leurs résolutions respectives

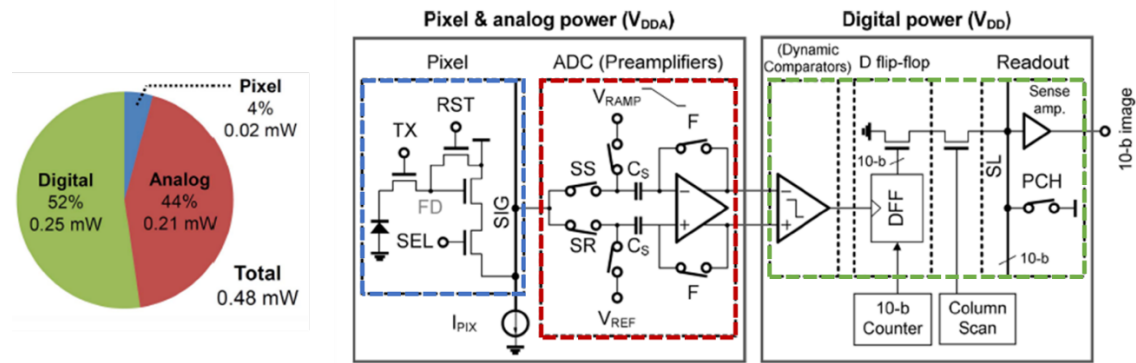


Figure 1-5: Consommation énergétique pour les imageurs conventionnel de type 4T détaillée par bloc de l'imageur: le pixel, la partie analogique étant majoritairement l'ADC, et la partie digitale dont une partie étant l'ADC [Choi2015]

## 1.2.2 Caractéristiques principales des capteurs

### 1.2.2.1 Taille du pixel

Plus la surface du pixel est petite et plus la matrice de pixels est petite. Cela permet de limiter les coûts de fabrication. Les pixels les plus petits sont de l'ordre du  $\mu\text{m}$  et la taille classique est de 3 à  $5\mu\text{m}$ .

### 1.2.2.2 Résolution

La résolution du capteur peut être définie selon plusieurs critères. Tout d'abord par la résolution spatiale, la taille de la matrice de pixel (array size) est exprimée en nombre de lignes et colonnes de pixels. La résolution est également définie par la fréquence d'image dit « frame rate » définie en trames par seconde (frame per second : fps) c'est la résolution temporelle.

### 1.2.2.3 Dynamic range (Dynamique de sensibilité à la lumière)

La dynamique de fonctionnement (dynamic range : DR) permet de mesurer la qualité de l'image obtenue. En effet, elle détermine la capacité du pixel à détecter diverses luminosités sans perdre de détail. Elle sera déclinée en nombre de niveaux de gris disponibles et donc codée sur un nombre de bits définis. Pour les capteurs d'images, elle est définie, en décibels, comme le rapport entre le signal d'entrée maximal non saturé sur la plus petite valeur d'entrée détectable. Il sera noté que l'œil humain a une dynamique supérieure à 140dB.

### 1.2.2.4 Sensibilité et rapport signal / bruit

La sensibilité du capteur prend en compte plusieurs caractéristiques. L'efficacité quantique (quantum efficiency : QE) de la photodiode, correspondant au nombre d'électrons générés par photon reçu.

De plus, le bruit a une influence sur la sensibilité du capteur. Il existe plusieurs sortes de bruits dans un capteur d'images : le bruit temporel et le bruit spatial fixe (fixed



pattern noise FPN). Le bruit temporel peut être décomposé en trois catégories : le bruit de scintillement (flicker noise ou bruit en  $1/f$ ), le bruit de grenaille (shot noise) et le bruit thermique. Le bruit en  $1/f$  est causé par des impuretés dans le matériau, créant des puits de potentiel, ou au contraire des charges libres en plus. Le bruit de grenaille vient de la nature granulaire de l'électricité ce qui crée une fluctuation du nombre de porteurs en fonction du champ électrique, créant des perturbations de courant. Enfin, le bruit thermique correspond à l'agitation thermique des porteurs créant une variation de potentiel aux bornes du conducteur, créant une légère fluctuation de la tension. De même, le bruit spatial fixe peut être divisé en deux parties, celui prenant place dans le pixel et celui dans les transistors placés en pied de colonne. Ce bruit correspond aux dispersions technologiques (mismatch). Le bruit de courant d'obscurité (dark current) correspondant au courant généré par la photodiode alors qu'aucune lumière n'arrive sur le capteur est contenu dans le FPN du pixel. La sensibilité du capteur est mesurée avec le ratio signal sur bruit (signal to noise ratio SNR) en décibel.

### 1.2.2.5 Facteur de remplissage

Enfin, la complexité du circuit peut être étudiée, ainsi que la complexité du pixel, et son facteur de remplissage (fill factor : FF) correspondant à l'aire de la partie photosensible du pixel sur l'aire totale du pixel. Un pixel ayant un facteur de remplissage élevé verra son efficacité quantique augmenter. Egalement, plus la complexité du circuit sera faible, plus la surface de silicium nécessaire à la réalisation du capteur sera faible, limitant ainsi le coût de fabrication.

### 1.2.2.6 Consommation énergétique

Afin de pouvoir comparer les circuits et les types d'architectures, une figure de mérite (figure of merits : FOM) a été retenue : l'énergie consommée par trame par pixel, exprimée en Joules par trames pixel ( $J/frame.pix$  ou  $J/fr.pix$ ).

Dans les tableaux de comparaisons, la technologie utilisée (process), le facteur de remplissage (fill factor), la plage de dynamique (DR) seront chiffrées, ainsi qu'une appréciation qualitative de la complexité du pixel et du système.

## 1.2.3 Architectures pixel

Le pixel est l'élément de base des capteurs d'images, il est composé d'une photodiode ainsi que de plusieurs transistors afin de permettre la lecture de l'intensité de la lumière reçue sur la photodiode. Deux méthodes se distinguent pour les méthodes conventionnelles de lecture CMOS, la lecture en tension et la lecture en courant.

La lecture en tension est décrite dans la Figure 1.6.a, la photodiode est polarisée en inverse et initialisée à une tension haute VDD, puis sa tension décroît linéairement avec la lumière incidente. Après un certain temps appelé temps d'intégration, la tension de la photodiode est lue.

La lecture en courant est décrite dans la Figure 1.6.b, les photons incidents pénétrant dans la zone de charge espace de la photodiode forment des paires électrons-trous. Les charges sont accumulées dans la photodiode et le courant résultant sera d'autant plus élevé que la lumière incidente sera forte. Cependant il est limité par la capacité de la

photodiode. Après le temps d'intégration, les charges sont transférées à l'extérieur de la matrice de pixel.

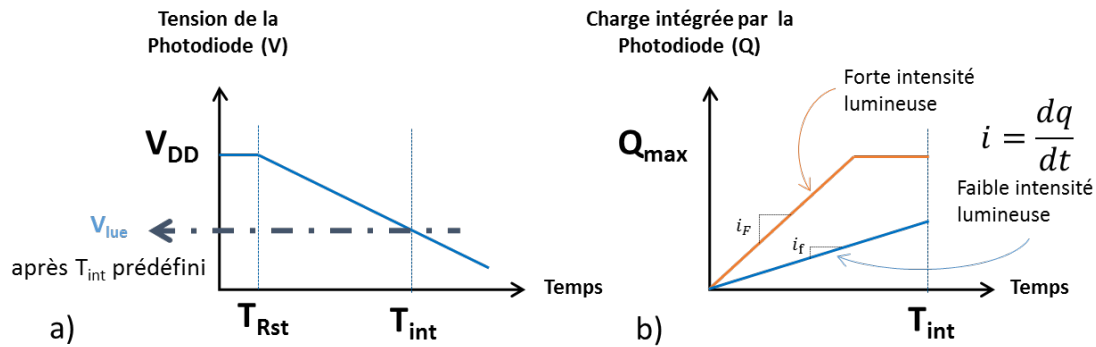


Figure 1.6 : a) Principe de la lecture en tension d'un pixel. La tension aux bornes de la photodiode polarisée à  $V_{DD}$  à  $T_{Rst}$  et dont la valeur est lue à  $T_{int}$  b) Principe de la lecture en courant d'un pixel, les charges sont accumulées dans la photodiode,  $i_F$  correspond à un fort courant dû à une forte intensité lumineuse, et  $i_f$  correspond à un courant plus faible dû à une faible intensité lumineuse.

### 1.2.3.1 Pixel passif PPS

La plus ancienne méthode de lecture en technologie CMOS est le pixel passif (PPS : passive pixel sensor), Figure 1-7 dont le pixel ne contient que la photodiode et un transistor de sélection dans le pixel. Lorsque le temps d'intégration prend fin, et lorsque le transistor de sélection devient passant, les charges accumulées pendant le temps d'intégration sont envoyés sur le bus de lecture de la colonne. Au pied de chaque colonne se trouve un transistor de reset ainsi qu'un amplificateur intégrateur afin de convertir les charges en tension et de permettre une meilleure lecture du signal. Cette méthode a l'avantage de permettre un facteur de remplissage élevé puisque la surface de photodiode est largement supérieure à celle du transistor. Cela permet également d'avoir une bonne efficacité quantique. Cependant la photodiode étant reliée directement au bus (donc à sa capacité parasite), la capacité de charge est fortement augmentée. De ce fait la constante de temps RC de charge du bus est elle aussi augmentée, limitant la vitesse de lecture et rendant cette méthode très sensible au bruit et donc peu robuste. C'est pourquoi les pixels actifs ont été mis en place et préférés. En lecture en tension, plusieurs architectures ont été développées : 3T, 4T, logarithmiques. En lecture de courant plusieurs architectures ont été développées, elles ne seront pas toutes développées ici, seulement une architecture classique d'un pixel actif avec lecture en courant sera présentée puisqu'elle représente la majeure partie des pixels en lecture en courant. En effet un pixel actif permet de lire la donnée sans la détruire.

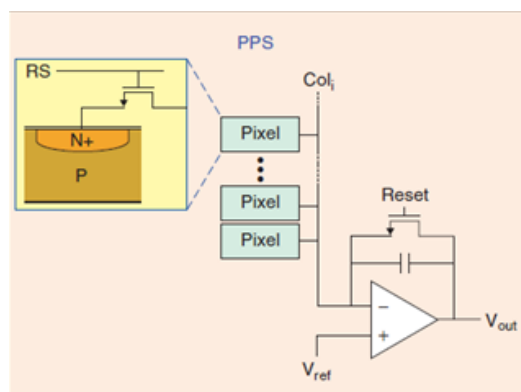


Figure 1-7: Pixel passif PPS et système de lecture avec amplificateur intégrateur, [ElGamal2005]

### 1.2.3.2 Pixel actifs (APS 3T 4T)

Les pixels actifs APS (Active Pixel Sensor) se déclinent en 2 types de lecture, la lecture en courant et la lecture en tension.

- Lecture en tension

Les pixels actifs avec une lecture en tension sont séparés en 2 catégories : les pixels ayant une réponse linéaire à la lumière et les pixels dit logarithmique ayant une réponse logarithmique à la lumière permettant une meilleure dynamique d'image.

Les pixels actifs linéaires ont des architectures dites 3T ou celles dites 4T. Le pixel 3T comprend deux transistors en régime ohmique et un transistor en régime saturé, comme indiqué sur la Figure 1-8.a. Les transistors en régime ohmique sont respectivement contrôlés par le signal de reset (RST) et le signal de row select (RS) permettant de lire les pixels un à un. Le transistor en régime saturé fonctionne en suiveur ou source follower (SF). La photodiode est polarisée à un potentiel  $V_{DD}$  pendant un temps de reset où le transistor RST est passant, puis l'intégration commence lorsque ce transistor se bloque. La capacité du nœud de la photodiode se décharge alors linéairement en fonction de l'intensité de la lumière incidente. Puis la tension aux bornes de la photodiode est lue à travers le transistor suiveur, lorsque le transistor de sélection devient passant. Le signal accumulé n'est pas détruit lors de la lecture, contrairement au pixel passif, il est donc possible de faire plusieurs lectures. Cette architecture est néanmoins sujette au bruit thermique qu'il est difficile de supprimer. La photodiode étant le lieu de photo-détection et de photo-conversion, cela oblige à un compromis au niveau du design de la photodiode. La capacité maximale de la photodiode définit la plage dynamique du capteur : plus la photodiode sera grande plus la plage dynamique sera large. Cependant, le gain de conversion est inversement proportionnel à la capacité de la photodiode, ce qui définit la sensibilité du pixel. Il faut donc trouver un compromis, au niveau du design de la photodiode, entre la plage de dynamique et le gain de conversion. Pour éviter la contrainte design de ce compromis, l'architecture 4T a été développée [Guidash1997].

L'architecture 4T est donc l'amélioration du 3T. En effet l'architecture 4T a été inventée afin de séparer la photo-détection de la photo-conversion, Figure 1-8.b. Cette architecture permet également de réaliser intrinsèquement la technique CDS (correlated double sampling) qui supprime le FPN dû au pixel, le signal est

échantillonné juste après le reset ainsi qu'à la fin du temps d'intégration. La soustraction des deux valeurs permet d'avoir la valeur du signal sans le bruit dû au transistor de reset, qui varie pour chaque pixel en fonction des dispersions. La charge accumulée pendant le temps d'intégration dans la photodiode est transférée dans la diffusion flottante (FD : floating diffusion) lorsque le transistor TX devient passant, la valeur est ensuite lue de la même manière que pour une architecture 3T. Cette architecture a donc un transistor supplémentaire par rapport à l'architecture 3T ce qui réduit son facteur de remplissage. Elle est également sensible au bruit et à cause du transfert nécessaire à FD, il peut y avoir un effet de rétention d'image (image lag). Cependant, et grâce à ses performances générales, le pixel 4T est actuellement le pixel utilisé par les grands industriels de l'imagerie.

L'architecture du pixel logarithmique, décrite dans la Figure 1-8.c, comporte comme l'architecture 3T un transistor suiveur (M2) et un transistor de sélection de ligne (M3). Le troisième transistor (M1) a sa grille reliée à son drain, il est dit « monté en diode » et opère en régime de faible inversion. Cette architecture permet d'avoir le courant de drain du transistor égal au photo-courant. Ainsi pour cette architecture, il n'y a pas de phase d'initialisation. La tension de la photodiode est lue en sortie du pixel à travers le transistor suiveur, selon une réponse logarithmique comme le montre l'équation (1.1). Cette architecture présente l'avantage d'avoir une dynamique en luminance plus large, jusqu'à 120dB. Néanmoins, le contraste s'en trouve réduit et la sensibilité pour les faibles luminosités est faible. De plus comparé aux autres architectures, l'architecture du pixel logarithmique présente plus de FPN [Ohta2008].

$$V_{ph} = V_{DD} - K \ln \left( \frac{I_{ph}}{I_0} \right) \quad (1.1)$$

Dans la Table 1-2 différentes consommations d'énergie pour différents pixels basse consommation sont répertoriées.

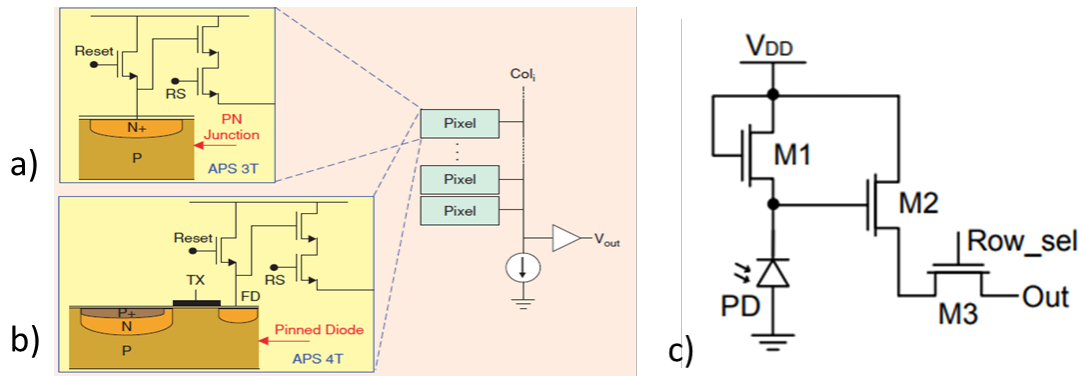


Figure 1-8: Les différentes architectures des pixels actifs a) le pixel 3T et b) le pixel 4T [ElGamal2005] ont le même principe de lecture en pied de colonne, un amplificateur et une polarisation. c) le pixel logarithmique [Brunetti2016]

	Process	Fill Factor	Pixel type	Pixel pitch	Array size	Power consumption	Frame rate	FOM (pJ/frame.pixel)	Dynamic Range
Choi2015	0.18μm CIS	45.5%	4T	5,6μm	336 x 256	19.9 μW	15fps	16.7 pJ/fr.pixel	57.1 dB
Liu2014	0.5μm CMOS	26.8%	3T	27,15μm	80 x 80	17 μW	30 fps	88.5 pJ/fr.pixel	48.6 dB
Choi2016	0.11μm CIS	N/A	4T	5μm	340x 240	45.5 μW	15 fps	34.3 pJ/fr.pixel	50 dB
Chou2012	0.18μm CMOS	32.5%	LOG (5T)	6μm	100 x 100	1.88 mW	50 fps	3760 pJ/fr.pixel	143.9 dB

Table 1-2: Table de comparaison pour des architectures de types lecture en tension. La figure de mérite décrit la consommation d'énergie pour les architectures de lecture en tension.

- Lecture en courant

La lecture en courant [Ohta2008] permet de lire le courant photo-généré dans la photodiode, il est lu à travers un transistor suiveur et un transistor de sélection comme indiqué sur la Figure 1-9. Dans un premier temps de reset, la photodiode est polarisée au potentiel  $V_{ref}$  décrit dans l'équation (1.2) puis le potentiel décroît linéairement avec le courant photo-généré pendant un temps d'intégration  $T_{int}$  (1.3). La capacité  $C$  est la combinaison de la capacité de grille du transistor suiveur et de la capacité de la zone de charge de la photodiode. Le contrôle de la grille du transistor de sélection permet d'activer la lecture du courant. Le courant du pixel est soustrait au courant de référence (1.5) ce qui permet de s'affranchir du FPN dû au mismatch de  $V_t$  du transistor suiveur, d'un pixel à l'autre.

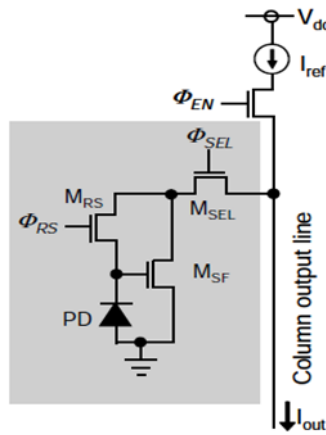


Figure 1-9.: Pixel actif, lecture en courant. Accumulation des charges et lecture de la différence entre l'intensité de référence et le courant photogénéré dans la photodiode et lu à travers un transistor suiveur [Ohta2008]

$$V_{ref} = \sqrt{\frac{2L}{K_n W} I_{ref}} + V_{th} \quad (1.2)$$

$$V_{PD} = V_{ref} - \frac{I_{ph} T_{int}}{C_{PD}} \quad (1.3)$$

$$I_{out} = I_{ref} - I_{pix} \quad (1.4)$$

$$I_{out} = \sqrt{2\mu_n C_{ox} \frac{W_g}{L_g} I_{ref} \frac{I_{ph} T_{int}}{C_{PD}}} - \frac{1}{2} \mu_n C_{ox} \frac{W_g}{L_g} \left( \frac{I_{ph} T_{int}}{C_{PD}} \right)^2 \quad (1.5)$$

Les pixels actifs avec une lecture en courant ont l'avantage de permettre simplement des opérations d'additions et de soustractions grâce aux lois de Kirchoff, ainsi que des multiplications grâce à des miroirs de courant. Néanmoins, le mismatch ne permet pas d'avoir une précision suffisante pour ces opérations, et la consommation d'énergie pour ces architectures est importante. Dans la Table 1-3, différentes consommations pour différents pixels sont répertoriées.

	Process	Fill Factor	Array size	Pixel Pitch & transistors /pixels	Power consumption	Frame rate	SNR (dB)	DR (dB)	FOM (nJ/frame.pixel)
Njuguna14	0.18 $\mu\text{m}$ CMOS	68 %	54 x 60	25 $\mu\text{m}$ 3T	270 mW	100 fps	58 dB	52 dB	833 nJ/fr.pixel
Njuguna12	0.6 $\mu\text{m}$ CMOS	30 %	128 x 109	10 $\mu\text{m}$ 3T	50 mW	50 fps	44 dB	55 dB	71 nJ/fr.pixel
Guev10	0.5 $\mu\text{m}$ Nwell CMOS	44 %	50 x 128	18 $\mu\text{m}$ 2T	5 mW	30 fps	43.3 dB	N/A	26 nJ/fr.pixel
Phillip07	0.35 $\mu\text{m}$ CMOS	26 %	128 x 128	10 $\mu\text{m}$ 3T	100 mW	40 fps	42 dB	64 dB	153 nJ/fr.pixel
Nakamura97	2 $\mu\text{m}$	N/A	1000 x 1000	10 $\mu\text{m}$ 3T	115 mW	1 fps	N/A	N/A	115 nJ/fr.pixel

Table 1-3: Table de comparaison de différentes architectures de types « lecture en courant ». La figure de mérite décrit la consommation par frame par pixel de chaque architecture.

### 1.2.3.3 Pixel « digitaux » DPS

L'architecture des pixels digitaux, permet de transformer l'intensité lumineuse en une information digitale. En effet un ADC ainsi qu'une mémoire sont intégrés dans le pixel, comme le montre la Figure 1-10, ce qui réduit drastiquement le facteur de remplissage. L'information sortie étant directement digitalisée, cette solution est robuste aux fuites potentielles de signal. Dans la Table 1-4 différentes consommations d'énergie pour différents pixels sont répertoriées.

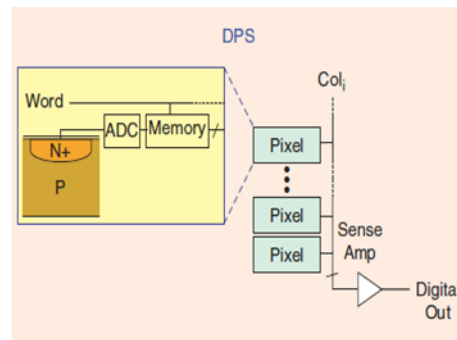


Figure 1-10: Architecture du pixel digital [ElGamal2008]

	Process	Pixel type	Pixel Pitch & transistors /pixels	Array size	Power consumption	Frame rate	FOM (nJ/frame.pixel)
Zhu2014	65nm CMOS	DPS	10,8µm 46T	100 x 100	9 mW	40 fps	22.5 nJ/fr.pix

Table 1-4: Table de comparaison pour des architectures de types DPS. La figure de mérite décrit la consommation d'énergie pour les architectures de type DPS

1.2.4 Problématique du traitement d'images dans les capteurs de vision  
 L'information lumineuse peut être traitée de deux façons différentes, au niveau du signal analogique ou bien au niveau du signal digital, c'est-à-dire avant ou après l'ADC. On parle ici des traitements intégrés au capteur et non réalisés par une unité de traitement externe au circuit ou même un autre circuit. Le principal avantage des traitements analogiques est qu'ils consomment moins d'énergie comparés aux traitements digitaux, qui eux ont l'avantage d'être plus précis et plus facilement programmables [Ohta2008]. Ces traitements sont implémentés dans des « processing elements » (PE) qui peuvent être placés, comme montré sur la Figure 1-11, au niveau du pixel [Massari2005], en pied de colonne [Elouardi2007] ou un PE pour toute la matrice [Panovic2016]. La Figure 1-12 donne une liste non exhaustive des différents prétraitements possibles. Usuellement, seuls des prétraitements sont effectués dans le capteur et des traitements plus complexes sont réalisés en dehors du capteur.

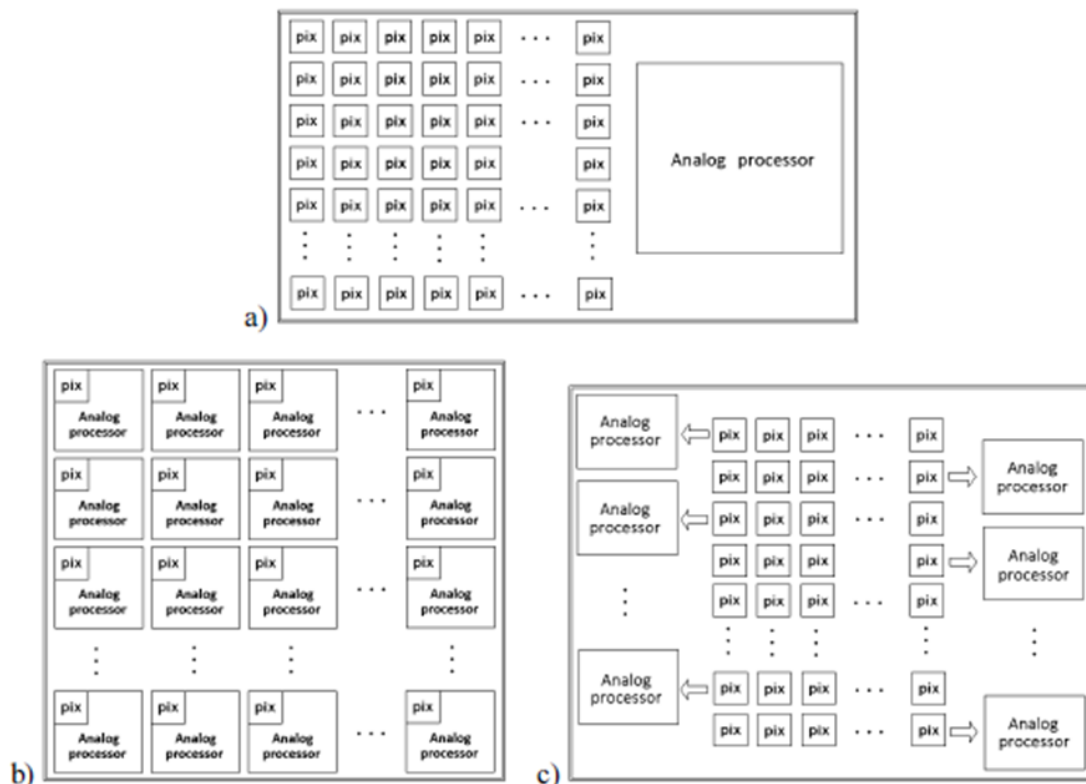


Figure 1-11: différentes manière d'intégrer les processing elements a) PE global b) PE intégré dans le pixel c) un PE par ligne

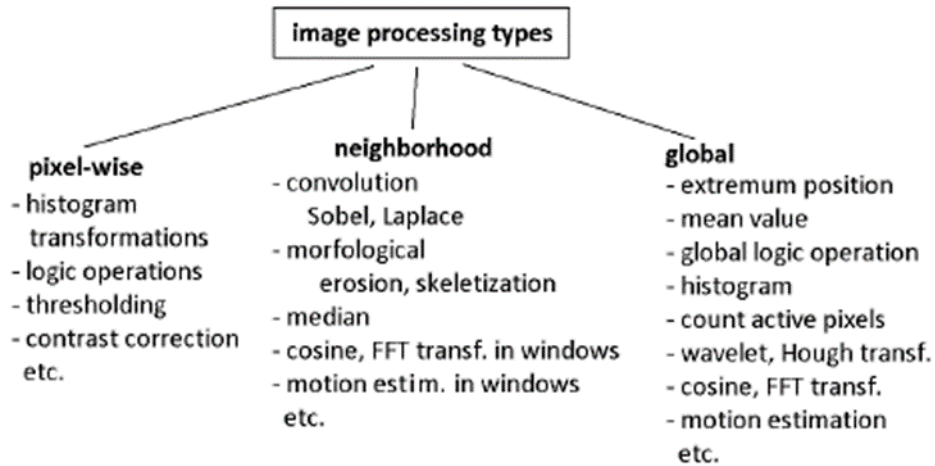


Figure 1-12: Types de traitement intégrés possibles [Jendernalik2013]

### 1.2.5 Avantages et inconvénients des différentes architectures pixels usuelles

Dans la Table 1-5, les différents avantages et inconvénients de chacune des architectures présentées dans ce chapitre sont récapitulés. Cette analyse porte sur des critères de robustesse pour une implémentation en technologie avancée, de compatibilité avec une faible consommation énergétique ainsi qu'un coût de fabrication optimisé. Ces architectures ont également l'avantage de pouvoir intégrer beaucoup de traitements existants (partie I.A.3.), en particulier les architecture APS 3T et 4T. Néanmoins, elles ne sont pas suffisamment robustes au bruit pour des implémentations en technologies avancées, pour des raisons qui seront expliquées dans le chapitre 2. De même ces architectures ne permettent pas une acquisition ultra-low power pour des applications de type Internet des Objets (Internet Of Things ou IOT). C'est pour ces deux raisons principales que les réflexions pour cette thèse se sont tournées vers l'acquisition événementielle dans le domaine temporel.

Type d'architecture	inconvénients	avantages
PPS	Sensible au bruit	Facteur de remplissage élevé
APS courant	Fort courant de fuite Peu de précision dans les traitements intégrés à cause du mismatch Forte consommation d'énergie	Intégration de traitements facilement grâce aux lois de kirchoff
APS tension 3T	Sensible au bruit thermique Contrainte design sur la photodiode	Nombreux traitements existants et intégrables



APS tension 4T	Facteur de remplissage moins élevé que le 3T	Suppression de FPN car CDS intrinsèque
	Sensible au bruit à cause du transistor de transfert	Nombreux traitements disponibles
Logarithmique	Non adapté aux faibles luminosités	Large dynamique & fonctionnement continu
	FPN plus important	
DPS	Facteur de remplissage faible	Robuste aux fuites grâce à la numérisation rapide des données

Table 1-5: avantages et inconvénients des types d'architectures présentés dans la partie I.A.

### 1.3 Les capteurs d'images événementiels

A l'inverse des capteurs conventionnels, l'information dans les capteurs événementiels n'est pas une tension ou un courant lu à la fin du temps d'intégration. L'information est représentée par le chronométrage ou le front d'impulsion d'un ou plusieurs signaux binaires générés, correspondant à la réalisation d'un événement.

#### 1.3.1 Le domaine temporel

Dans le cas des capteurs d'image événementiels, l'événement le plus courant est le passage d'un seuil prédéfini par la tension de la photodiode comme indiqué sur la Figure 1-13. Pour ce faire dans chaque pixel se trouve un comparateur. La gestion de cet événement peut-être diverse. Les principales méthodes sont le pulse width modulation (PWM) et le pulse frequency modulation (PFM). Le PWM prend en compte le temps auquel le pixel a généré son signal et le convertit directement en intensité lumineuse, alors que le PFM définit l'intensité lumineuse incidente par la fréquence des signaux générés par le pixel. Cela permet au pixel de « choisir » son temps d'intégration optimal. Ainsi la dynamique n'est plus restreinte par le niveau de tension d'alimentation ce qui permet une intégration plus robuste en nœuds technologiques avancés, d'autant plus qu'un signal de type binaire est généré et propagé sur le bus colonne.

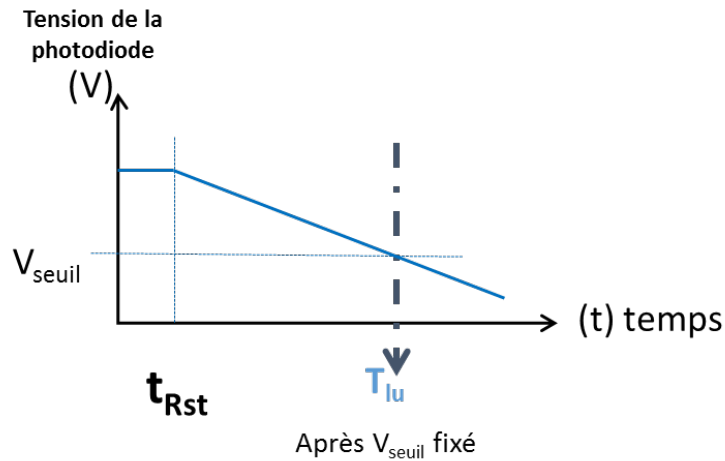


Figure 1-13: Lecture de l'information lumineuse dans le domaine temporel

### 1.3.1.1 Pulse Frequency Modulation : PFM

L'information lumineuse correspond à la fréquence des événements générés par le pixel comme indiqué sur la Figure 1-14. La photodiode est polarisée en inverse à VDD puis avec la lumière incidente, la tension aux bornes de la photodiode décroît et lorsqu'elle dépasse le seuil  $V_{ref}$  du comparateur, un signal binaire est créé. Après un court délai (longueur de pulse nécessaire), c'est ce même signal qui permet de rendre passant le transistor de reset et donc de commencer une nouvelle intégration. L'événement n'est donc qu'une impulsion, dite spike. Plus la fréquence des signaux générés, appelés événements ou spike, est élevée plus l'intensité lumineuse est élevée. En pratique, le nombre d'événement est compté pendant une période de temps donné, ce nombre est proportionnel à l'intensité lumineuse. Le compteur peut être implémenté à différents niveaux, dans le pixel [Andoh2000], en pied de colonne ou en dehors de la matrice [Yang1994]. La méthode de lecture pour le PFM peut-être soit synchrone soit asynchrone comme détaillé dans les parties suivantes. La consommation d'énergie par pixel est donc dépendante de la moyenne de l'intensité lumineuse reçue, plus les pixels seront éclairés plus la consommation sera forte. Les architectures PFM sont le plus souvent utilisées dans des circuits biomimétiques tels que les rétines.

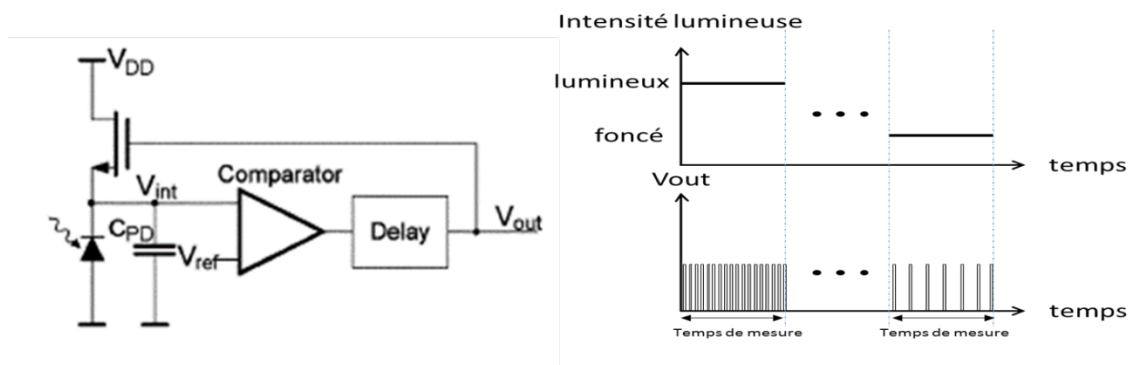


Figure 1-14: Architecture d'un pixel pulse frequency modulation et correspondance entre l'intensité de la lumière et la fréquence en sortie du pixel [Chen2011]

1.3.1.2 Pulse Width Modulation : PWM

Les architectures en pulse width modulation mettent directement en relation le chronométrage de l'événement généré avec l'intensité lumineuse incidente. La différence temporelle entre le signal de reset et la génération de l'événement lorsque le comparateur du pixel bascule, est mesurée par un compteur situé soit dans le pixel soit hors de la matrice. Plus le temps entre le reset et l'envoi de l'événement est long moins le pixel sera lumineux, Figure 1-15. Le système d'horloge (clock) gérant les compteurs est quant à lui commun à toute la matrice. La précision de l'horloge et donc du compteur, influe sur la dynamique. En effet, puisque l'information est temporelle, plus l'horloge est précise plus la dynamique sera importante. La dynamique est également corrélée à la vitesse de lecture. De plus, contrairement au PFM, la luminosité incidente n'influe pas sur la consommation d'énergie puisqu'importe le niveau de luminosité, le pixel n'enverra qu'un seul événement pour une luminosité ou trame donnée. Ces événements peuvent se lire de manière synchrone ou asynchrone. Les méthodes de lectures sont détaillées dans la partie 1.3.2.

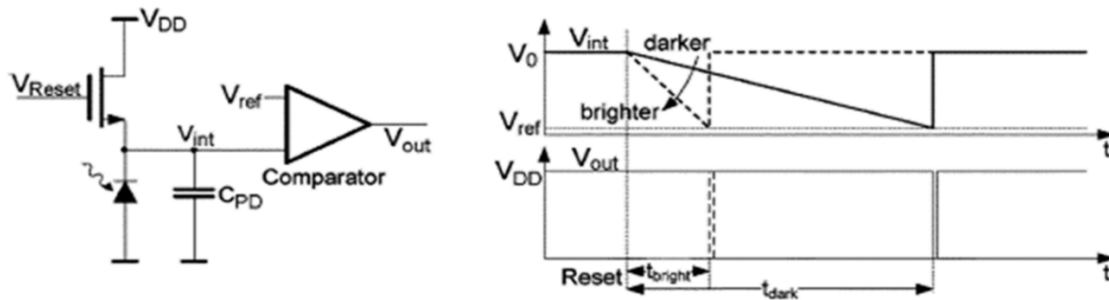


Figure 1-15: Architecture d'un pixel pulse width modulation et correspondance entre l'intensité de la lumière et le temps auquel l'impulsion apparaît en sortie du pixel [Chen2011]

1.3.1.3 Les autres événements possible

Que ce soit le PFM ou le PWM, le signal est généré lorsque l'événement : « la tension aux bornes de la photodiode passe le seuil prédéfini » mais évidemment d'autres événements sont envisageables. Comme des événements de contrastes tels que « un pixel est plus éclairé que son voisin ». [Berkovich2015] [LeneroBardallo2016] [Yin2016]. Ainsi l'information lumineuse est prétraitée, la Figure 1-16 montre un exemple d'une telle architecture avec les images extraites, elles permettent d'avoir les informations sur les contrastes de l'image. Chaque couleur correspond à une orientation de gradient.

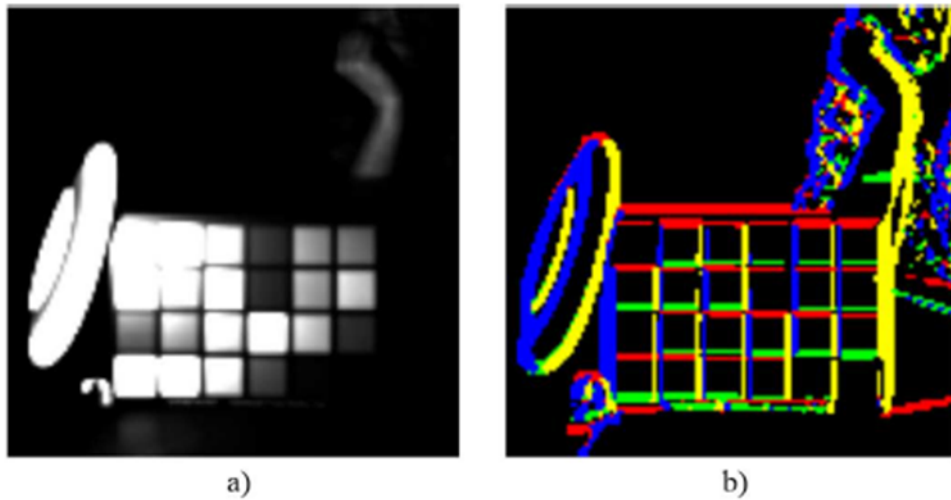


Figure 1-16: Exemple d'information sur le contraste d'une image en sortie de capteur [Berkovich2015]

Certaines architectures parlent également de contrastes temporels et n'envoient donc d'événements que lorsque la luminosité diffère d'avant. C'est le cas du pixel ATIS [Posch2011] qui utilise deux photodiodes par pixels, une pour détecter le changement temporel de luminosité et une pour actualiser la valeur mesurée et stocker l'intensité lumineuse. La Figure 1-17 explique le fonctionnement de ce pixel. Des événements de changement de luminosité sont générés, les ON events et les OFF events. Chacun de ces événements déclenche l'intégration sur la deuxième photodiode qui se comporte en PWM. Ainsi lorsque l'activité de la scène est nulle, le capteur consomme très peu.

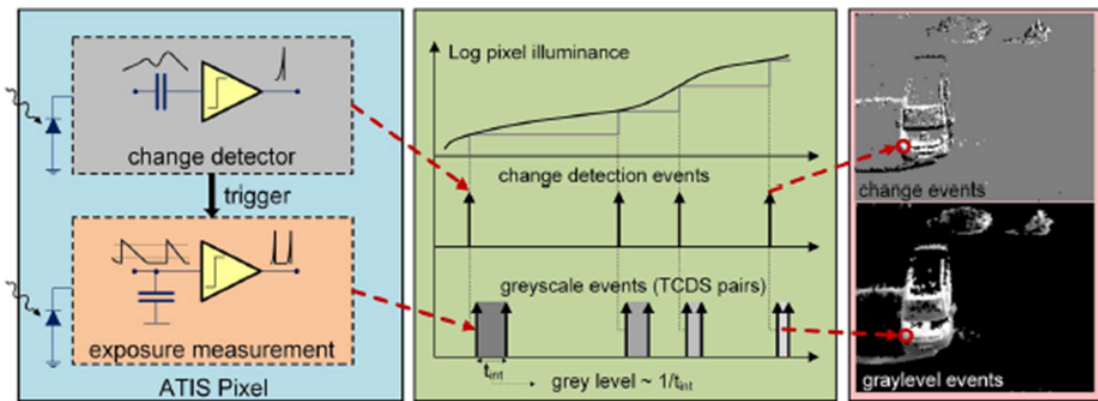


Figure 1-17: Architecture du pixel ATIS architecture globale détecteur de changement d'illumination d) impulsion émise par le détecteur de changement permettant de reconstruire la courbe d'évolution de l'intensité lumineuse e) signaux typiques entrant et sortant du pixel [Posch2011]

### 1.3.2 Lecture des événements

La lecture des événements peut prendre deux formes différentes : la lecture synchrone et la lecture asynchrone. Lorsqu'un pixel génère un événement, on dit que ce pixel flashe. La lecture synchrone (Figure 1-18.a) permet de lire tous les pixels successivement, elle est liée à une horloge. La lecture asynchrone (Figure 1-18.b)

permet de communiquer l'activité du capteur sans avoir à lire toute la matrice, mais uniquement les pixels ayant flashé.

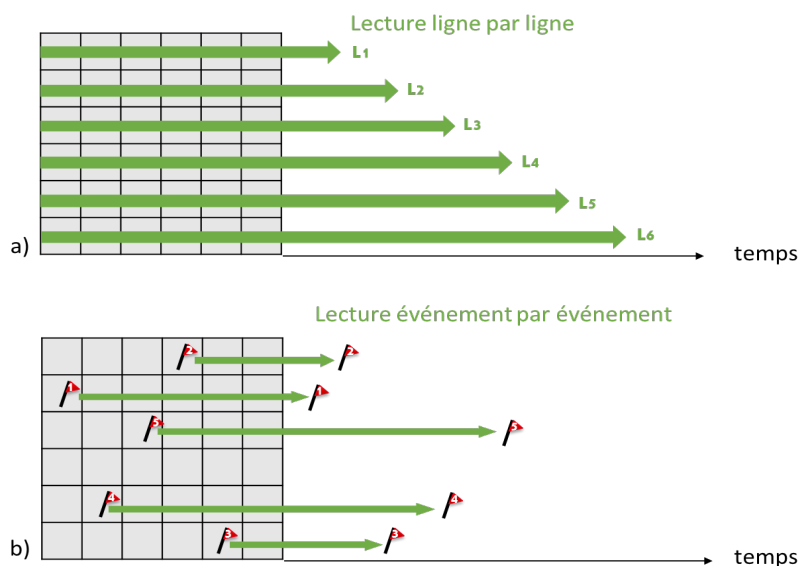


Figure 1-18: Système de lecture a) synchrone, tous les pixels sont lus ligne par ligne, cette lecture est liée à une horloge b) asynchrone, la lecture d'un pixel se fait lorsqu'un événement apparaît.

### 1.3.2.1 Lecture synchrone

La lecture synchrone consiste à lire une image ligne par ligne, en activant le signal de sélection de ligne contrôlé par le décodeur d'adresse, comme montré dans 1.2.1. Usuellement, la lecture peut s'effectuer en rolling shutter ou en global shutter comme expliqué dans 1.2.1.1. Dans le cas des capteurs dans le domaine temporel, la lecture synchrone reste la même, c'est-à-dire que les informations des pixels sont évacuées une à une dans le même ordre. Il peut cependant y avoir plusieurs approches : la lecture par trame, la lecture par événement, et la lecture en sigma delta.

La lecture synchrone par trame implique une mémorisation de l'information dans le pixel. Que ce soit en PFM [Shoshun2008] où le nombre d'intégrations est contenu dans des bascules et où le temps trame correspond ici à un temps défini pendant lequel sera évaluée la fréquence. Ou que ce soit en PWM [Kitchen2005] où la valeur temporelle de flash est stockée dans une mémoire implémentée dans le pixel. Le temps trame, au terme duquel sera lue cette valeur, fixe la valeur du pixel le plus sombre. Ces implémentations ont l'inconvénient de prendre de la place dans le pixel et donc de réduire le facteur de remplissage. Cependant, dans l'architecture PWM synchrone frame-based, cela peut permettre d'effectuer de la compression pendant la quantification [McIlrath2001] et donc d'y intégrer un prétraitement.

Une autre possibilité de lecture synchrone pour les PWM est de scanner rapidement tous les pixels un à un, leur valeur étant binaire [Crooks2009]. Ainsi lorsqu'un pixel flashe, donc crée un événement, la donnée binaire est lue rapidement. Puis en pied de colonne ou à l'extérieur de la matrice, un temps de lecture lui est attribué. Cette méthode permet de s'affranchir de la contrainte design, en effet il n'est pas nécessaire d'avoir une mémoire dans le pixel. Cependant elle ne permet pas d'avoir

une grande précision sur les événements et augmente drastiquement le nombre de données lues puisque chaque pixel est lu plusieurs fois par trame.

### 1.3.2.2 Lecture asynchrone

Pour la lecture asynchrone, il y a deux approches possibles également.

Dans l'approche asynchrone basée sur un événement (event-based), les pixels envoient leur information uniquement si besoin et se réinitialisent après. Le signal de reset n'est pas global mais indépendant pour chaque pixel en fonction de l'illumination respective des pixels. La lecture asynchrone event-based permet d'économiser de l'énergie et de réduire le flot de données puisque tous les pixels ne sont pas nécessairement lus. Par exemple, seuls les pixels qui donnent une information de changement d'illumination sont lus dans le cas du pixel ATIS [Posch2011]. Dans le cas du pixel issu de la thèse [Darwish2016], une lecture sert à évacuer tous les pixels actifs de la matrice à l'instant  $t$ . Cette lecture peut également être utilisée dans les capteurs de type PFM [Culurciello2003], les pixels flashant, envoient leurs adresses. Les pixels lumineux envoient plus souvent leur adresse que les pixels plus sombres. Néanmoins, ce mode de lecture asynchrone event-based, est contraignant au niveau du design. En effet, il faut rajouter dans le pixel et en pied de colonne, les transistors nécessaires pour la gestion de l'envoi des données et ceux nécessaires à l'arbitration pour éviter les conflits. L'arbitration peut être à un mécanisme AER (address event representation) ou un mécanisme de WTA (Winner Take All). Ces deux techniques sont présentées en détail dans la partie 1.3.2.3.

Dans l'approche lecture asynchrone frame-based, les pixels envoient nécessairement tous une information pendant le temps trame [Guo2007] et sont tous réinitialisés pour la trame suivante. C'est le principe du Time-to-First Spike où au début du temps trame toutes les photodiodes sont polarisée à une tension proche de VDD puis la tension décroît linéairement avec l'intensité lumineuse incidente. Une fois le seuil prédéfini passé, un événement est généré grâce à un comparateur. Le pixel envoie son adresse selon le protocole de communication AER. L'adresse du pixel est horodatée et c'est ainsi que l'intensité lumineuse est retrouvée. Si un pixel n'a pas envoyé son adresse à la fin du temps trame il est considéré comme ayant la valeur la plus sombre. Enfin toute la matrice est réinitialisée, une nouvelle acquisition commence jusqu'à la fin du temps trame. L'avantage des approches frame-based est que les algorithmes de traitement peuvent être directement appliqués. L'inconvénient est que le nombre de données envoyées n'est pas réduit.

### 1.3.2.3 Les mécanismes d'arbitrage

Lors de l'envoi des données, les mécanismes d'arbitrations permettent d'éviter les conflits et collisions résultant en la perte de données. Les principaux mécanismes d'arbitration de lecture asynchrone utilisés sont l'AER : « address event representation » et le WTA : « winner take all ». L'AER permet de n'envoyer comme information que l'adresse ligne et colonne du pixel, utilisant un arbre d'arbitration qui définit l'ordre des priorités, pour éviter les conflits. Le WTA est une autre méthode pour décider du pixel ayant la priorité.

- AER : Address Event Representation

Le protocole de communication AER était destiné en premier lieu à permettre aux réseaux de neurones de communiquer entre eux par trains d'impulsions. Dans le cas des imageurs, lorsqu'un pixel envoie une impulsion, elle est reçue par un encodeur d'adresse ligne et un encodeur d'adresse colonne. Ils transforment la donnée reçue en une adresse sur n bits. Lorsque deux pixels communiquent en même temps, les informations entrent en collision et une partie ou la totalité des informations sont perdues. C'est pourquoi deux arbitres sont implémentés autour de la matrice de pixels, Figure 1-19. Lorsqu'un pixel flashe, tout d'abord, il demande l'accès au bus sous forme de requêtes ligne et colonne envoyées aux arbitres. Ces arbitres communiquent par protocole de double poignée de main (handshake). Lorsque deux pixels flashent et envoient leurs requêtes en même temps, les arbitres décident du pixel gagnant la priorité en lui envoyant un signal d'acquiescement ligne et colonne. Ce pixel peut donc accéder au bus et envoyer un spike aux décodeurs d'adresse. Puis les signaux d'acquiescement sont relâchés et envoyés à un autre pixel demandant l'accès. Le protocole de handshake sera décrit en détail dans la partie 3.1.2). Ainsi, un spike contient deux informations, l'adresse du pixel et le temps de flash.

Les principaux inconvénients des circuits d'arbitrages sont la complexité de la circuiterie, la priorité fixe et les délais induits. La surface du circuit de l'arbitre augmente exponentiellement avec la résolution du capteur. Afin de résoudre le problème de la priorité fixe, un arbitre à priorité adaptative peut être implémenté [Shoshun2007]. Les délais induits à cause du circuit d'arbitrage endommagent l'information transmise dans les capteurs temporels puisque la donnée est codée en temps. Plusieurs solutions ont été développées pour y pallier, comme l'augmentation de la taille de la base de l'arbitre [Shoshun2007] ou l'envoi de la valeur du délai avec la donnée [Xu2013].

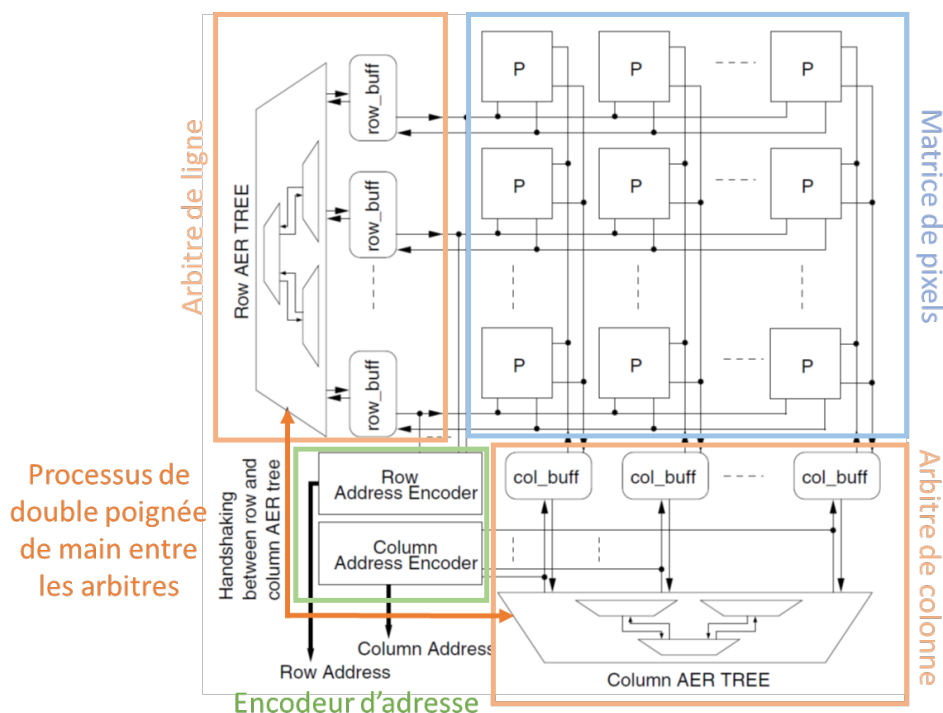


Figure 1-19: Exemple d'architecture d'un capteur avec système d'arbitration AER et encodeur d'adresse ligne et colonne [Linn2011]



- WTA : Winner Take All

Contrairement au protocole AER, le WTA permet de s'affranchir des encodeurs d'adresses lignes et colonnes ainsi que des arbres d'arbitrages (tree arbiters). Un système implémenté dans chacun des pixels permet de déterminer le pixel le plus illuminé et de les classer par ordre de traitement [Massari2007]. Les adresses lignes colonnes du pixel le plus illuminé sont envoyées sur deux bus séparés. Une fois l'adresse envoyée, un signal d'acquiescement est envoyé au pixel, ce qui permet au deuxième pixel le plus illuminé d'envoyer son adresse. L'architecture du système global est illustrée dans la Figure 1-20. L'inconvénient majeur de ces architectures est la réduction du facteur de remplissage.

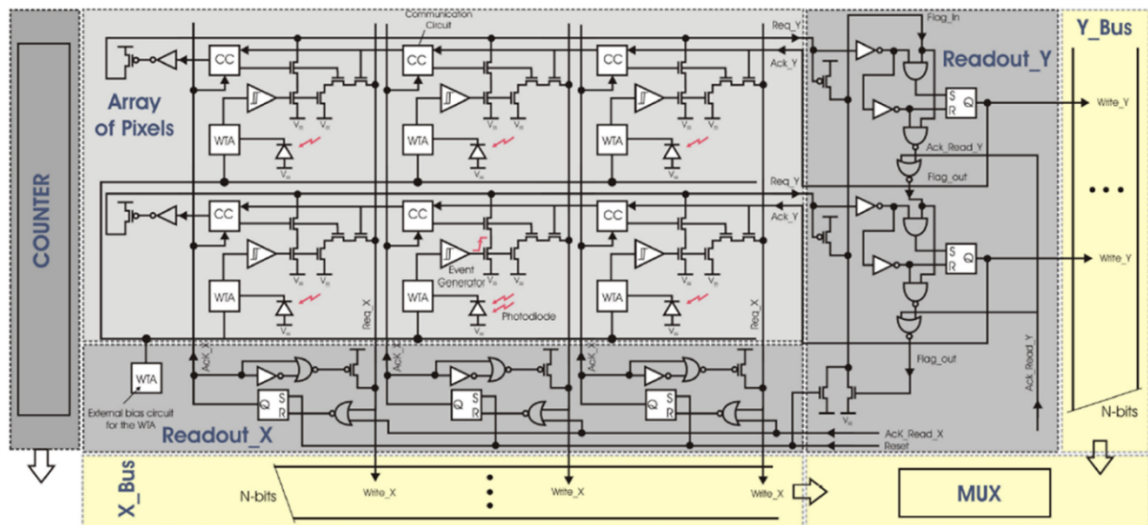


Figure 1-20: Exemple d'architecture implémentant un système de winner-take-all avec des briques dans le pixel et en pied de colonnes [Massari2007]

### 1.3.3 Traitement des événements

Les informations issues des capteurs événementiels peuvent être traitées afin d'en extraire d'autres informations. Un des principaux avantages de la lecture d'événements vient de la définition d'un événement. Un événement peut être défini comme le résultat d'une opération analogique, comme une comparaison spatiale ou temporelle, et de ce fait l'information lumineuse est déjà prétraitée.

Des traitements des informations lumineuses événementielles ont été développés. Par exemple, il est possible de faire de la reconstruction en 3D d'une scène [Rebecq2016] avec une caméra événementielle se déplaçant selon une trajectoire connue. La reconstitution se fait en projetant les événements émis par le capteur sur un espace virtuel dont la forme est déterminée par les mouvements du capteur.

Grâce à des événements ON/OFF de changement d'illumination, il est possible de suivre les objets en mouvement dans l'image. [Liu2016]



D'autres traitements sont possibles, néanmoins les solutions ne sont pas autant disponibles dans la littérature puisqu'ils sont encore en phase de développement comparés aux traitements envisageables pour les modes de lectures conventionnels. En effet puisque dans les lectures asynchrones les informations de l'image peuvent arriver avec parcimonie il faut adapter les algorithmes existants pour les rendre compatibles avec le mode asynchrone [Ieng2014]. Utiliser des filtres asynchrones permet de réduire le coût des calculs en ressources informatiques et d'augmenter la résolution temporelle. Les algorithmes décrits permettent de filtrer temporellement ou spatialement les images avec des filtres gaussiens ou des filtres bilatéraux ou d'effectuer une détection de contour avec un algorithme de type Canny.

### I.3.1 Avantages et inconvénients des capteurs d'images événementiels

La Table 1-6 donne une analyse qualitative des avantages et inconvénients des lectures synchrones et asynchrones pour des capteurs d'images événementiels. Trois types d'architectures sont étudiés. La première est une architecture event-based synchrone [Kitchen2005], il y a donc une mémoire dans le pixel afin de mémoriser le temps de flash du pixel. La valeur de l'intensité lumineuse est lue de manière synchrone puis elle est convertie grâce à un convertisseur temps-numérique (TDC : time to digital convertor). La seconde est le time to first spike : TTFS [Guo2007], où les données sont lues de manière asynchrone, mais il y a un reset global, ce qui permet d'avoir un temps trame fixe. La troisième architecture [Lichtsteiner2008] est une architecture event-based avec lecture asynchrone où l'événement défini est une augmentation ou diminution de delta dans l'information lumineuse, elle constitue la partie « change detector » du pixel ATIS [Posch2011]. Qualitativement, l'architecture TTFS présentée semble être la meilleure au niveau de l'économie d'énergie, et de la possibilité de traitement par trame. Cependant elle présente une complexité importante d'architecture notamment à cause de l'implémentation du protocole de communication AER. Et la Table 1-7 donne une analyse quantitative sur les données de consommation d'énergies de différentes architectures.

	<b>[Kitchen2005] Lecture synchrone</b>	<b>[Guo2007] TTFS, lecture asynchrone par trame</b>	<b>[Lichtsteiner2008] Lecture asynchrone par détection de delta.</b>
Economie d'énergie	+	++	++
Peu de FPN	--	+	+
Par trame pour faciliter les traitements par la suite	+	+	-
Réaction rapide au changement de luminosité	-	+	++
Mismatch faible	+	+	--
Architecture simple	+	-	-

Table 1-6: Analyse qualitatives des avantages et inconvénients pour trois types d'architecture. [Kitchen2005] est une architecture event-based avec une lecture synchrone et une mémoire dans chaque pixel. [Guo2007] est une architecture event-based avec une lecture asynchrone avec un temps trame fixe. Et [Lichtsteiner2008] est une architecture totalement asynchrone event-based par détection de delta.

	Process	Fill factor	Array size	Power consumption	Frame rate	FOM (nJ/frame.pixel)	Dynamic Range
Bol2014	65 nm	57 %	128 x 128	8.9 $\mu$ W	32fps	17pJ/fr.pix	40 dB
Hanson2010	0.13 $\mu$ m	31.6 %	128 x 128	1.19 $\mu$ W	8.5fps	8.54 pJ/fr.pix	N/A
Posch2010	0.18 $\mu$ m	30 %	304 x 240	50mW(static) 175mW (high activity)	30 fps	32 nJ/fr.pix	125 dB
Kim2010	0.18 $\mu$ m	42 %	128 x 128	1.4 mW	800 fps	91 nJ/fr.pix	N/A
Culurciello2003	0.6 $\mu$ m	14 %	80 x 60	7.5mW	30 fps	52 nJ/fr.pix	48.9 dB
Guo2007	0.5 $\mu$ m	2 %	32 x 32	3.1mW	30 fps	100 nJ/fr.pix	104 dB

Table 1-7: Table de comparaison des caractéristiques de différentes architectures événementielles. La figure de mérite décrit la consommation d'énergie pour différentes architectures event-based.

#### I.4 Conclusion de l'état de l'art, positionnement et justification des contributions de la thèse

Dans ce chapitre, les divers types de capteurs d'images et leurs principales caractéristiques ont été présentées, ainsi que les différents types d'acquisition d'images classiques. Les capteurs d'image CMOS traditionnels avec une lecture en charge, en tension ou en courant présentent des avantages au niveau des traitements intégrés connu dans la littérature mais ils ne sont pas compatibles avec une implémentation en nœuds technologiques avancés. En effet, comme nous allons le voir dans le chapitre II, les fuites de courant dans ces technologies sont trop importantes comparé au signal généré par la photodiode sous l'effet de la lumière incidente. La donnée a donc besoin d'être rapidement digitalisée. Le pixel digital présente cet avantage puisqu'un ADC est implémenté dans chacun des pixels, malheureusement cela diminue drastiquement le facteur de remplissage et ne serait envisageable que pour une implémentation 3D du capteur. La contrainte supplémentaire imposée pour l'architecture visée dans cette thèse, est une faible consommation énergétique. L'ADC d'un imageur représentant plus de 40% de la consommation énergétique, comme expliqué dans la partie I.A.2., les choix de systèmes d'acquisition se sont tournés vers les capteurs événementiels. Le domaine temporel permet de soit envoyer un train d'impulsions dont la fréquence déterminera l'intensité lumineuse du pixel, soit d'horodater les impulsions uniques des pixels et ainsi de retrouver leurs intensités lumineuses. La lecture asynchrone implique cependant de devoir rajouter des transistors dans le pixel et en bordure pour gérer l'envoi des informations ainsi que les conflits que cela peut engendrer. La lecture synchrone quant à elle nécessite la présence d'une mémoire dans le pixel ce qui dégrade, là aussi, le facteur de remplissage du pixel. Notons qu'une implémentation en 3D permettrait de s'affranchir de cette contrainte design.

Le capteur d'image étudié pendant cette thèse est un capteur événementiel avec traitement intégré, l'information sortie est ainsi une information binaire avec une lecture synchrone. Les informations extraites de la scène et pouvant être transmises

à d'autres capteurs ou à des unités de traitements sont suffisantes pour faire certains traitements tels que de la classification. Cette acquisition est compatible avec une implémentation en nœuds technologiques avancés et a une consommation énergétique réduite, puisqu'il n'y a pas d'ADC ni de transmission superflue à un capteur extérieur ou unité de traitement.

Le chapitre II présentera les technologies avancées pour l'imagerie, en particulier la technologie FDSOI 28nm.



## 2 Chapitre II : Les technologies avancées pour l'imagerie

### 2.1 Introduction

L'objectif de cette thèse est de définir une architecture d'acquisition d'image pouvant intégrer des traitements d'image et pouvant être intégrée au sein d'un nœud de capteur. Dans un premier temps, le mécanisme de génération d'électrons sous l'effet de la lumière dans un matériau photosensible sera explicité. Puis, la technologie FD-SOI 28nm sera décrite en soulignant les avantages de l'utilisation d'une telle technologie pour des applications de type nœud de capteur pour l'IOT. Cependant cette technologie présente des non idéalités pour la capture d'image. En effet, les fuites de courant des transistors seront explicitées et reliées aux contraintes imposées sur le type d'acquisition. Par la suite, les technologies émergentes au LETI seront explicitées. Tout d'abord l'utilisation d'une photodiode sous la couche d'isolant du FD-SOI comme l'a montré l'article [Kadura2016]. Puis le développement de la technologie 3D qui permet d'implémenter des traitements d'images au-dessus de la photodiode et donc d'avoir un facteur de remplissage approchant les 100%. Enfin les travaux de cette thèse seront décrits. En premier lieu, la sensibilité à la lumière de la technologie FD-SOI 28nm sera vérifiée et quantifiée. Puis, l'approche envisagée pour l'architecture d'un capteur d'image en technologie avancée tenant compte de ces contraintes sera discutée. Et enfin, un second circuit sera présenté, il permettra d'évaluer les potentialités d'une implémentation d'un pixel en 3D séquentiel, en termes de performance et de compacité.

### 2.2 Matériaux photosensibles

Certains matériaux tels que le germanium (Ge), le silicium (Si), ou l'arséniure de galium (GaAs), sont photosensibles. Les photons sont absorbés par le semi-conducteur et ceux dont l'énergie est supérieure à  $E_g$  permettent de former une paire électron-trou.  $E_g$  correspond à l'énergie nécessaire aux électrons pour passer de la bande de valence à la bande de conduction, elle diffère selon le matériau. Le coefficient d'absorption des photons par un semi-conducteur varie en fonction de la longueur d'onde. Cette variation est propre à chaque type de matériaux comme le montre la Figure 2-1, [Yadid-Pecht2004]. Le silicium est propice à la collecte des électrons des longueurs d'onde du visible. En effet, son coefficient varie graduellement de 400nm à 1050nm contrairement au GaAs dont le coefficient chute abruptement à 900nm. Le silicium est donc largement utilisé comme semi-conducteur pour fabriquer les capteurs d'image dans le visible.

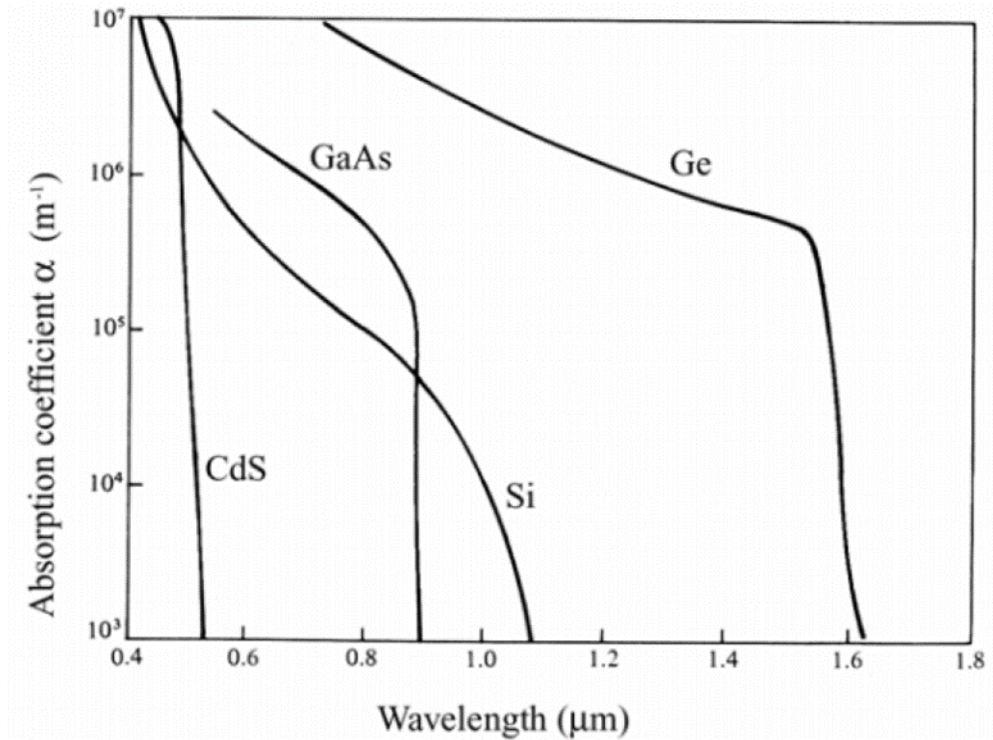


Figure 2-1: Coefficient d'absorption de la lumière en fonction de la longueur d'onde et pour plusieurs matériaux [Yadid-Pecht2004]

### 2.3 Collecte des électrons

Il existe plusieurs manières de collecter les électrons générés par les photons absorbés, comme les photodiodes et les phototransistors. Dans cette thèse, seules les photodiodes ont été étudiées, il n'y aura donc que celles-ci qui seront décrites ici.

La lumière incidente sur le matériau photosensible crée des charges mobiles comme vu dans II.2.2. Afin de collecter les charges, la photodiode est polarisée en inverse, Figure 2-2-a. Un champ électrique se crée aux bornes de la jonction permettant la collection des charges photo-générées dans la zone de charge espace avant qu'elles ne se recombinent, Figure 2-2-b. Le flux de charges collectées correspond au courant photonique d'intensité proportionnelle à la puissance lumineuse incidente comme montré sur la Figure 2-2-a.

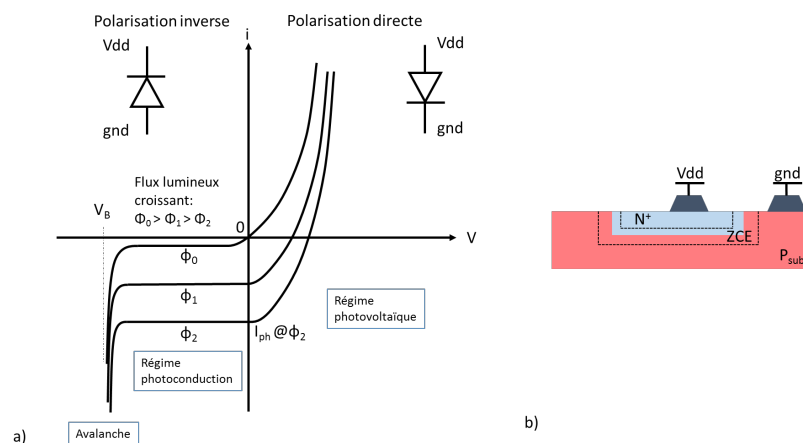


Figure 2-2: a) Caractéristique I-V d'une diode avec les modes photodiodes (photoconduction) et photovoltaïque b) implémentation d'une diode et création de la zone de charge espace (ZCE) sous l'effet du champ électrique induit par la polarisation en inverse.

## 2.4 Technologies avancées

### 2.4.1 Evolution de la taille de la lithographie et conséquences

Depuis les années 2000, la taille minimum possible en lithographie ne cesse de diminuer passant de  $20\mu\text{m}$  à  $14\text{nm}$  ou même  $7\text{nm}$  aujourd'hui, permettant comme le prédisait la loi de Moore d'intégrer de plus en plus de transistors dans un espace défini de silicium. Les technologies habituellement utilisées pour les capteurs d'image dans le domaine visible sont les technologies CMOS de taille  $0.35\mu\text{m}$ ,  $0.18\mu\text{m}$ ,  $0.13\mu\text{m}$ ,  $90\text{nm}$  et  $65\text{nm}$ . La taille du pixel varie linéairement avec la taille minimum de la lithographie [Yadid-Pecht2004]. Cependant la mobilité et le temps de vie des porteurs décroissent avec la taille minimum de la lithographie à cause de l'augmentation du dopage du substrat, Table 2-1, ce qui demande un design plus fin et plus rapide afin de ne pas perdre de l'information lumineuse. Le nombre de niveau de métaux augmente également ce qui permet de faire des designs plus complexes et plus compacts mais ce qui réduit la réponse électro-optique des capteurs. En effet, l'empilement des métaux autour de la photodiode empêchent la lumière d'atteindre les bords de la photodiode.

Process	$0.8\mu\text{m}$	$0.5\mu\text{m}$	$0.35\mu\text{m}$	$0.25\mu\text{m}$	$0.18\mu\text{m}$	$0.1\mu\text{m}$
Supply voltage (V)	5-3.3	5-3.3	5-3.3	2.5-1.8	1.8-1.5	1.2-0.9
Interconnect levels	2	2-3	4-5	5-6	6-7	8-9
Substrate doping ( $\text{cm}^{-3}$ )	$8 \times 10^{16}$	$1.2 \times 10^{17}$	$2.5 \times 10^{17}$	$3.4 \times 10^{17}$	$5 \times 10^{17}$	$1 \times 10^{18}$
Junction depth $d/S$ (nm)	350-450	300-400	200-300	50-100	36-72	20-40
Depletion region ( $\mu\text{m}$ )	0.71	0.57	0.39	0.24	0.19	0.1
Mobility ( $\text{cm}^2 \cdot \text{V}^{-1} \cdot \text{s}^{-1}$ )	825	715	550	485	425	345
Lifetime ( $\mu\text{s}$ )	3.6	2.3	1.1	0.8	0.6	0.3
Diffusion length ( $\mu\text{m}$ )	88	68	41	33	25	15

Table 2-1: Paramètres technologiques pour différents process. [Yadid-Pecht2004]

## 2.4.2 Technologies avancées pour l'imagerie

Les technologies dites avancées pour l'imagerie sont les technologies dont la taille de gravure est en dessous de 50nm. L'attrait pour ces technologies vient de la mise à disposition de blocs de traitement périphériques très performants et à consommation réduite.

L'utilisation des technologies 180nm à 65nm permet un très faible niveau de fuite de courant ce qui est vital pour des applications telles que l'internet des objets où les systèmes sont le plus souvent en veille et doivent donc consommer très peu d'énergie. Cependant l'utilisation de ces technologies limite la vitesse de ces systèmes. Par exemple, le temps de commutation d'un inverseur est de 7.23ns dans la technologie 130nm bulk alors qu'il est de 104ps dans la technologie FDSOI 28nm. C'est pourquoi la technologie FD-SOI 28nm émerge dans ces applications. En effet, la technologie permet d'avoir accès à la fois à un mode de haute performance en « forward body biasing » ou à un mode à très faible niveau de fuite en « reverse body biasing » [Beigne2015a]. Le forward body biasing augmente la largeur du canal permettant aux charges de transiter plus rapidement lorsque le transistor est passant. A l'inverse, le reverse body biasing réduit la largeur du canal, ainsi lorsque le transistor est bloqué, les fuites sont réduites, limitant ainsi la consommation d'énergie. Les performances pour des systèmes de radiofréquences [Martineau2016], du power management ou même des unités CPU [Beigne2015b] ont été démontrées. L'intégration de toutes ces briques d'applications sur une même puce permettrait d'avoir un système complet basse consommation pour une application telle que l'IOT. L'ajout d'une brique imageur dans un tel système monolithique complèterait d'autant plus ce système. Pour ces raisons, c'est la technologie FDSOI 28nm qui a été retenue dans le cadre de cette thèse, ainsi, seules les technologies SOI seront étudiées ici.

Des capteurs d'image en technologies SOI ont déjà été développés et ont été comparés à des imageurs en technologies bulk, [Brouk2007]. Cependant la taille minimum de lithographie était de 0.35 $\mu$ m. A ce jour, il n'y a pas eu de capteurs d'image complet (comprenant l'acquisition et le traitement sur la même puce) dans le domaine visible en FDSOI 28nm.

## 2.5 FD-SOI 28nm

### 2.5.1 Principe et avantages

Les wafer de silicon-on-insulator dit SOI, ont été inventés dans les années 1970. Contrairement aux technologies bulk, les technologies SOI ont une couche d'isolant entre deux couches de silicium, Figure 2-3, cela permet un meilleur contrôle du flux d'électrons. En effet, le substrat (body) peut être polarisé (body biasing ou back biasing), qui permet de créer une grille arrière (back gate) régulant le flux d'électron avec plus de précision et moins de fuites.



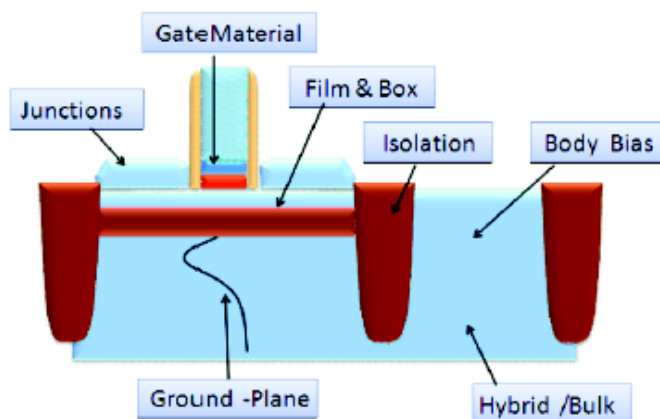


Figure 2-3: vue en coupe d'un transistor en technologie FDSOI [Hartmann2012]

Les avantages des technologies avancées en FD-SOI sont nombreux par rapport à une technologie bulk de même taille de lithographie. En effet, en technologie bulk, la diminution de la taille des grilles entraîne la diminution de la mobilité des porteurs à cause des impuretés du matériau [Yadid-Pecht2004], et l'augmentation du courant de fuite par la grille. De plus, les jonctions p-n sont de moins en moins profondes engendrant donc un courant de fuites de plus en plus important [Sakurai2006]. Le FDSOI permet de palier à ces défauts grâce à la couche d'isolant. Ainsi les fuites des jonctions sont diminuées, le latch-up ne se produit plus et les porteurs sont mieux contrôlés grâce à la polarisation du substrat : le back biasing. De plus, le FDSOI est moins sensible aux hautes températures et aux radiations [Sakurai2006]. Enfin, les capacités drain source étant réduites, les circuits peuvent fonctionner à une vitesse 30% supérieure à une technologie bulk équivalente [Hartmann2012]. La Figure 2-4 présente la comparaison des performances et de la consommation en énergie entre les technologies FDSOI 28nm et bulk 28nm. A une tension d'alimentation de 0.45V, la technologie FDSOI 28nm consomme 50% moins que la technologie bulk avec des performances de 20% (DMPS) contre 7% (DMPS). On remarque également que les performances du FDSOI sont globalement 30% plus élevée que celle de la technologie bulk.

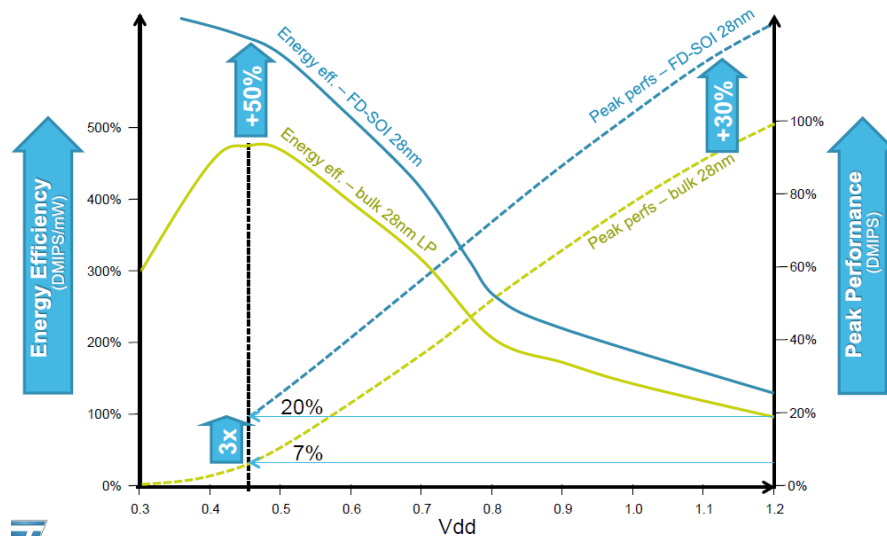


Figure 2-4: Comparaison des performances et consommation d'énergie entre les technologies FDSOI 28nm et bulk 28nm [Hartmann2012]

De plus, les performances atteintes associées à la faible consommation, mènent à penser que cette technologie est le parfait candidat pour des applications dans l'internet des objets [Beigne2015a]. En effet, l'intégration sur une même puce de plusieurs fonctions comme un processeur de traitement digital, un transmetteur et récepteur RF, et l'intégration de capteur embarqué, permettrait de réaliser un système de surveillance complet performant et à faible consommation potentielle, [Beigne2015a].

### 2.5.2 Fuites de courant et conséquences pour l'acquisition d'image

Néanmoins, afin d'intégrer de manière monolithique une brique imageur à ce système, plusieurs contraintes sont soulevées, notamment des contraintes design concernant les niveaux de fuites des transistors.

Par exemple, la simulation d'un transistor NMOS ( $W/L=300n/150nm$ ) alimenté en 1V qui serait utilisé comme transistor suiveur dans une architecture 3T, donne un courant de fuite de à sa grille de 800fA avec un back biasing à 0V, Figure 2-5. Ce courant de fuite diminue jusqu'à 220fA pour un back biasing de 2V. Cette valeur est la valeur de back biasing maximale applicable de 2V, le courant de fuite diminue linéairement lorsque la polarisation du substrat augmente (back biasing), Figure 2-6. Le courant de fuite de grille peut être négligeable en utilisant des transistors GO2 ayant un oxyde plus épais.

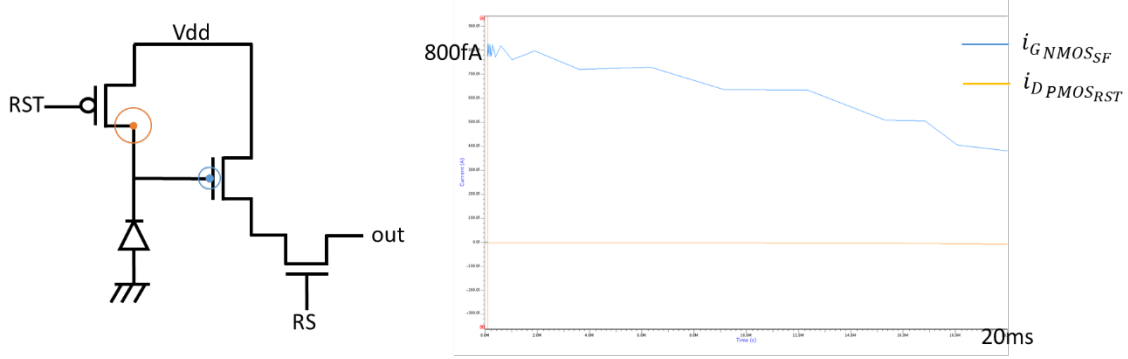


Figure 2-5: Fuites de courant des transistors dans une architecture de pixel 3T et évolution de ce courant dans le temps.

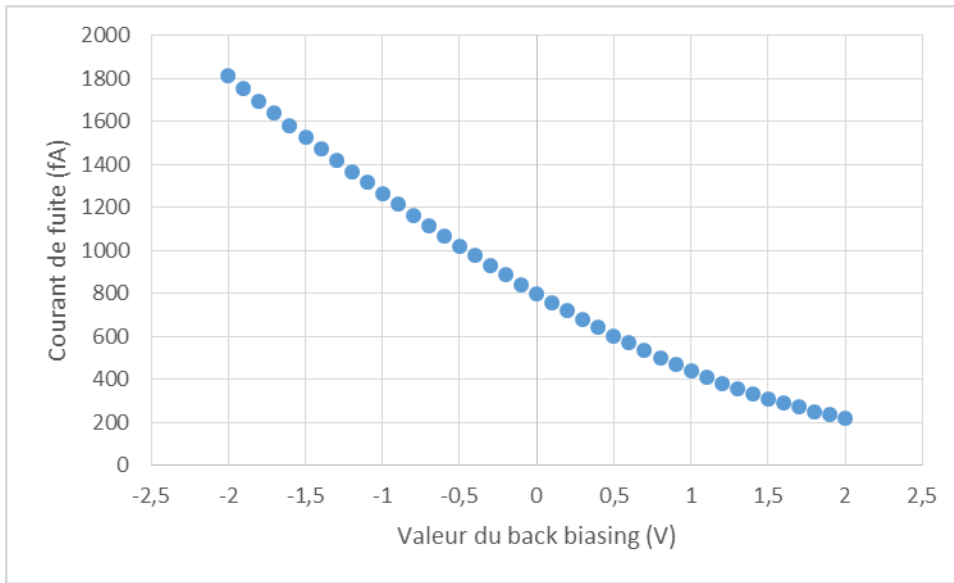


Figure 2-6: Fuites de courant de grille du suiveur dans architecture 3T avec  $i_{ph}=0$  en fonction du back biasing (ou polarisation du substrat) appliqué.

Le courant de fuite de grille peut être assimilé à du bruit de grenaille [Lee2003], [Manghisoni2007], son écart type se calcule donc comme la racine du nombre d'événement. Pour une acquisition dans le domaine temporel, un tel courant de fuite fixe la plus faible luminosité détectable, Figure 2-7. Le temps lu pour un tel courant de fuite dans une acquisition temporelle est de 23ms (2-1). Comme démontré ci-dessous, il est possible de déterminer le SNR obtenu.

$$i_{ph} = 0pA \quad i_f = 0.8pA \quad i = i_{ph} + i_f \text{ et } i = C \frac{\Delta V}{\Delta T}$$

$$C = 25fF \quad \Delta V = 0.75V \text{ donc } \Delta T = 23ms \quad (2-1)$$

$$\text{Nombre d'électron de fuite : } Ne^- = i * \frac{\Delta T}{\text{energie electron}} = 115ke^-$$

$$\text{Ecart type : } \sqrt{Ne^-} = 339e^-$$

$$\text{d'où l'écart type en courant } i_E = \frac{\sqrt{Ne^-} * \text{energie electron}}{\Delta T} = 2,4fA$$

$$t^+ = C \frac{\Delta V}{i - \frac{i_E}{2}} = 23,51ms \quad \text{et } t^- = C \frac{\Delta V}{i + \frac{i_E}{2}} = 23,37ms$$

$$\Delta t = t^+ - t^- = 70\mu s \text{ ainsi } SNR_{PWM} = \frac{23 \cdot 10^{-3}}{70 \cdot 10^{-6}} = 330 \quad (2-2)$$

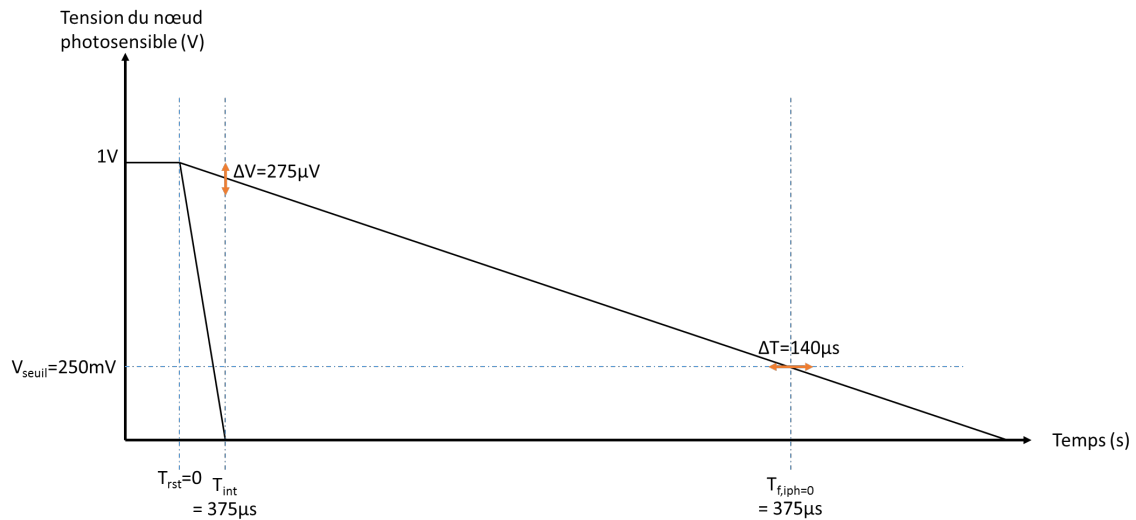


Figure 2-7: Evolution de la tension du nœud photosensible dans le temps.

Pour l'acquisition en tension avec un pixel 3T, afin de pouvoir lire le signal dans les fortes luminosités sans que celui-ci ne soit saturé, le temps d'intégration doit être fixé à  $375\mu s$  correspondant à un courant total de  $50pA$  (courant photo généré et les fuites de courant de grille). Ainsi à dynamique égale entre les deux modes d'acquisition, il est possible de comparer les SNR pour un courant photo-généré nul. Ainsi avec une lecture en temps, le SNR est de 330 alors qu'il n'est que de 44 pour une lecture en tension. Ce SNR a été obtenue de la même manière que pour l'acquisition temporelle. La Figure 2-8 montre l'évolution du SNR en fonction du courant photo-généré pour les deux types d'acquisition. Le temps d'intégration est fixe pour l'acquisition en tension alors qu'il varie en fonction du courant photo-généré pour l'acquisition temporelle. La différence de SNR vient du fait que l'intégration dure plus longtemps dans le cas de l'acquisition temporelle, de ce fait le ratio des charges intégrées sur les charges correspondantes au courant de fuites ( $\sqrt{Ne^-}$ ) est croissant.

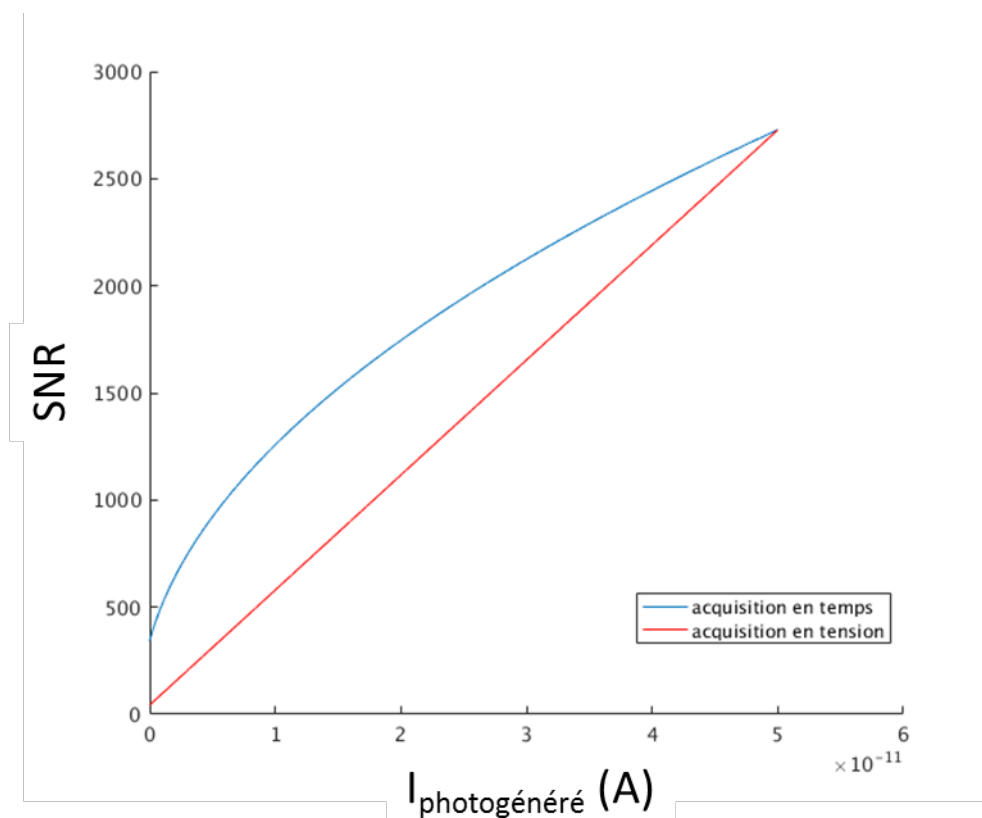


Figure 2-8: SNR en fonction du courant photo-généré pour une acquisition en temps ou une acquisition en tension

Une acquisition temporelle permet d'acquérir un signal moins dégradé qu'une acquisition en tension pour la même dynamique considérée. Cependant, il serait possible de faire plusieurs acquisitions avec un pixel 3T et un temps d'intégration variable. Puis il serait possible de reconstituer une image faisant la moyenne de ces acquisitions. L'image reconstruite aurait un SNR équivalent à celle obtenue par acquisition temporelle. Néanmoins, la consommation d'énergie nécessaire pour réaliser ces différentes acquisitions serait nettement supérieure à la consommation d'énergie nécessaire à une acquisition temporelle.

## 2.6 Technologies émergentes au LETI

### 2.6.1 Etendre les fonctionnalités des transistors FDSOI en les rendant sensibles à la lumière et exploiter ces nouvelles fonctionnalités.

#### 2.6.1.1 Sensibilité du FDSOI étendue grâce à une diode implémentée sous le box du transistor [Kadura2016]

Dans la publication de décembre 2016 présentée à la conférence IEDM (International Electron Device Meeting) [Kadura2016], une nouvelle approche sur la technologie FDSOI 28nm permettant d'implémenter de nouvelles fonctions avec les transistors est présentée. Une diode est implémentée monolithiquement sous le BOX des transistors, Figure 2-9.a. Elle est réalisée par implémentation ionique avant la gravure des grilles,

rendant ainsi les transistors sensibles à la lumière et n'entraînant aucun coût supplémentaire comparé à une implémentation ionique standard. La partie basse de la diode est relié à la masse tandis que l'autre partie est laissée flottante, ces parties peuvent être l'anode ou la cathode de la diode. La Figure 2-9.b montre les effets de la lumière sur la caractéristique  $I_{DS}/V_{GS}$  d'un transistor PFET avec une diode PN polarisée en mode direct et celle d'un transistor NFET avec une diode PN polarisée en inverse sous le BOX. En fonction de la polarisation de la diode, le  $V_T$  varie vers une tension de seuil plus basse en mode direct et plus haute pour une polarisation inverse. Il est à noter qu'il n'est pas nécessaire de relier la diode au transistor par une liaison électrique pour observer ces effets et qu'en l'absence de ces diodes, les effets de glissement de  $V_T$  grâce à la lumière incidente ne sont pas observables.

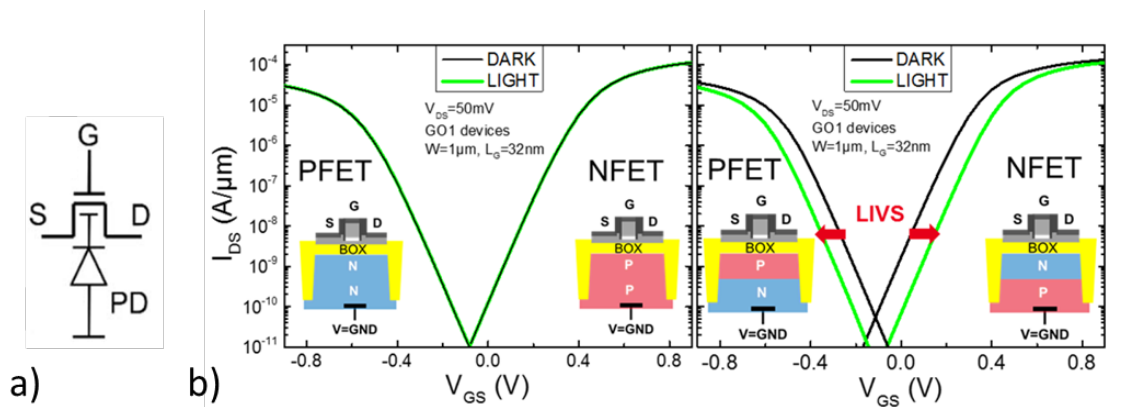


Figure 2-9: a) schéma de l'implémentation d'une diode sous le BOX d'un transistor b) caractéristiques des transistors sans et avec des photodiodes implémentées sous le box de chaque transistor. Le LIVS (Light-Induced  $V_T$  Shift) est visible. [Kadura2016]

La Figure 2-10, montre les effets de la lumière sur la caractéristique  $I_{ds}/V_{ds}$  d'un transistor NFET ayant une photodiode polarisée en direct puis en inverse. Une diode polarisée en direct soumise à un éclairage constant, augmente le courant drain-source pour un  $V_{ds}$  donné. Cet effet est appelé « forward optical back biasing » noté FOBB. Alors qu'une diode polarisée en inverse sous le BOX d'un NFET produit l'effet inverse, le courant produit diminue. Cet effet est appelé « reverse optical back biasing », noté ROBB. Les effets complémentaires sont observés sur un transistor PFET. Les résultats sont récapitulés dans la Figure 2-11.

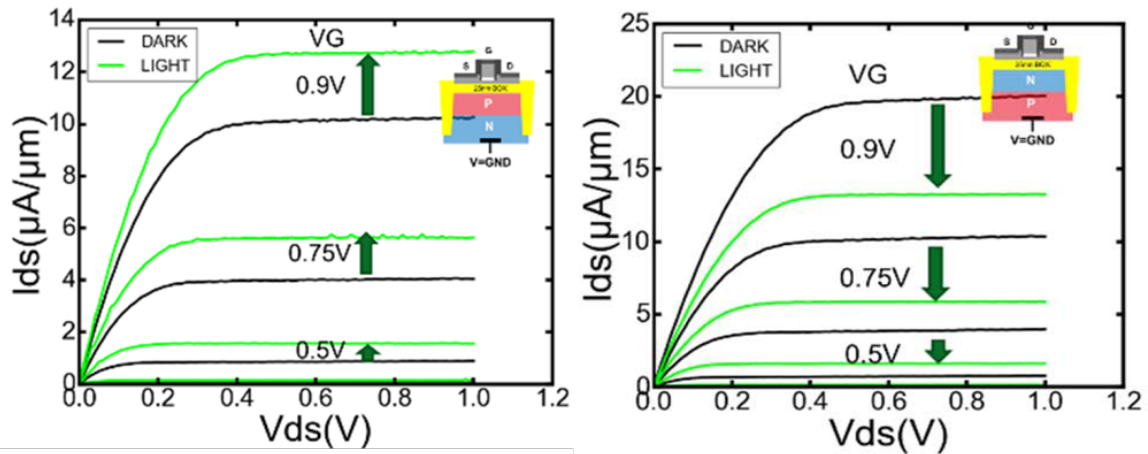


Figure 2-10: Caractéristiques  $I_{ds}$  en fonction de  $V_{ds}$  de deux transistors NFET. A gauche la diode sous le box est polarisée en direct ayant pour effet avec la lumière, un « forward optical back biasing » FOBB rehaussant l'intensité. A droite l'effet inverse est observé « reverse optical back biasing » avec une diode polarisée en inverse. [Kadura2016]

FET type	PFET				NFET			
Configuration below the BOX								
Light effect	negative back bias	positive back bias	none	none	negative back bias	positive back bias	none	none
FET Light-Induced $V_T$ -Shift (LIVS)	forward BB	reverse BB	none	none	reverse BB	forward BB	none	none
LIVS amplitude	Increases with body factor	Increases with body factor	none	none	Increases with body factor	Increases with body factor	none	none

Figure 2-11: Récapitulatif des effets produits par la lumière sur les caractéristiques des transistors PFET ou NFET avec une diode implémentée sous le box polarisée en direct ou en inverse. [Kadura2016]

La publication met également en avant l'architecture d'un inverseur avec deux diodes sous chacun des transistors. Les diodes peuvent être soit polarisée en inverse soit en direct mais les polarisations pour les deux diodes sous les box sont identiques. Dans le cas d'une polarisation en inverse, la tension de seuil de l'inverseur diminue lorsque la luminosité augmente et dans le cas où la diode est polarisée en direct, elle augmente lorsque la luminosité augmente, Figure 2-12.

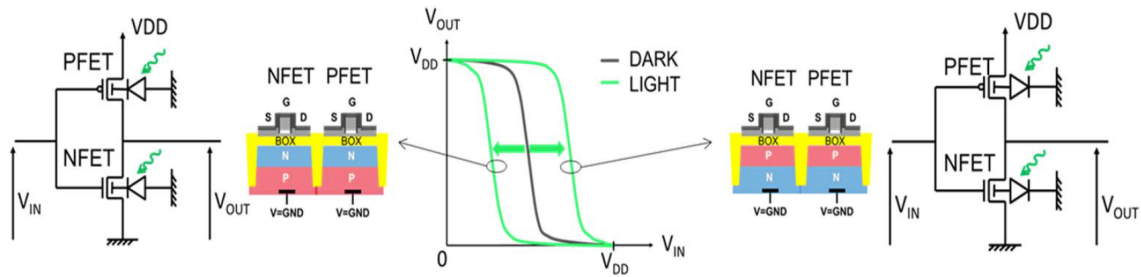


Figure 2-12: Inverseur avec des diodes implémentées sous le box, polarisées en inverse (à gauche) ou en direct (à droite). Au centre, la caractéristique de l'inverseur  $V_{out}$  en fonction de  $V_{in}$ , montrant l'influence de la lumière sur l'évolution de la tension de seuil.

### 2.6.1.2 Architectures proposées pour exploiter ces nouvelles fonctionnalités

Sur la base de ces travaux, des architectures ont été imaginées en utilisant la possibilité d'avoir une diode sensible à la lumière sous le BOX des transistors affectant la tension de seuil en fonction de la luminosité. La diode sous le BOX fonctionne en photovoltaïque. Les architectures décrites dans la partie ci-dessous n'ont pas pu être testées ou simulées, en attente d'un modèle électronique exploitable. Ainsi, seules leurs limites théoriques seront exposées ici.

#### 2.6.1.2.1 Pixel 2T et architectures associées

La première architecture imaginée tire avantage du fait que les photodiodes en mode photo- intégration ont une réponse à la lumière plus rapide que les photodiodes en mode photovoltaïque pour une même intensité lumineuse reçue [Kadura2016]. Une photodiode classique ( $P_{pix}$ ) est intégrée dans le pixel dont la cathode est reliée à la grille d'un NFET monté en suiveur. Le transistor NFET a une photodiode ( $P_{Photov}$ ) polarisée également en inverse sous le BOX, Figure 2-13. Lorsqu'un changement d'intensité se produit dans la luminosité, la photodiode  $P_{pix}$  a une réponse plus rapide que  $P_{Photov}$ . La différence des tensions  $V_0$  et  $V_t$  est lue, ce qui signifie que lorsqu'un changement de luminosité se produit, la dérivée de  $V_0-V_t$  varie. En quantifiant cette dérivée un spike peut être généré. Cette architecture pourrait donc être très utile dans le cas d'une architecture event-based.

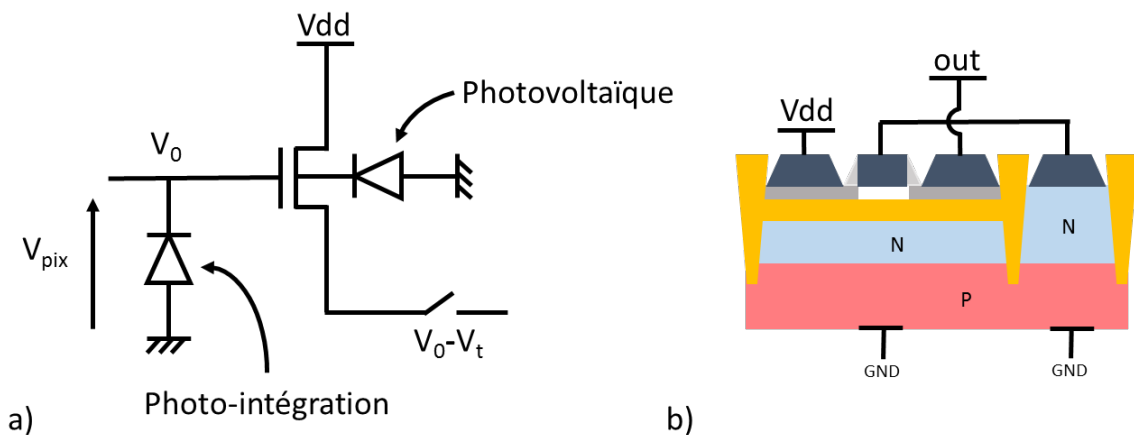


Figure 2-13: a) schéma de l'architecture d'un pixel avec 2 photodiodes et 2 transistors b) vue en coupe de cette architecture.



- Pixel dans le domaine temporel

Les pixels dans le domaine temporel permettent de s'affranchir du temps de lecture fixe afin d'avoir une dynamique plus large. En effet, la dynamique n'est plus limitée par la tension d'alimentation et le pixel fixe son propre temps d'intégration en fonction de l'intensité lumineuse reçue. La donnée lue est le moment où la tension de la photodiode passe le seuil prédéfini, généralement le seuil du comparateur. A ce moment un événement est généré. Lorsqu'il est évacué par le système de lecture, il est horodaté ce qui permet de retrouver la donnée lumineuse grâce à l'équation (2-3). L'inconvénient de ce mode d'acquisition est décrit dans l'équation. En effet le temps de flash étant inversement proportionnel à l'intensité lumineuse reçue, la densité des événements est plus élevée pour les luminosités plus fortes que pour les luminosités plus faibles. De ce fait, un goulet d'étranglement apparait lors de l'évacuation des données des pixels les plus lumineux, ce qui dégrade les informations et crée donc des erreurs. Par exemple un temps de trame de 20ms correspond à un temps de flash égal à 20ms pour une intensité lumineuse reçue nulle. Dans ce cas, un delta temps de 10µs comprendra plus de valeurs de luminosité correspondante, dans pour les courts temps de flash comparé aux temps de flash plus long. C'est-à-dire que 0µs et 10µs séparent 29 valeurs de luminosité codées sur 8 bits soit entre 255 et 226 alors que 350µs et 360µs n'en séparent que 3 à savoir 54 à 56.

$$I = C_{phD} \frac{V_{DD} - V_{seuil}}{T_{lu} - T_{rst}} \quad 2-3)$$

En utilisant un inverseur avec une photodiode polarisée en inverse sous le BOX comme comparateur, Figure 2-14, cela permet de baisser le seuil du comparateur sous l'effet de la lumière [Kadura2016]. Ainsi la densité des événements devient moins élevée pour les hautes luminosités, Figure 2-15. De plus cela permet d'avoir une plus grande dynamique dans les faibles luminosités pour un temps de trame donné, Figure 2-16. En effet, le temps de trame est fixé et limite donc la dynamique dans une architecture TTFS (time to first spike) classique.

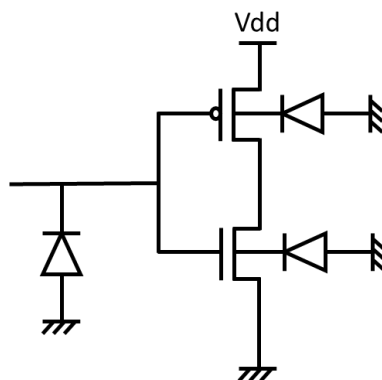


Figure 2-14: Schéma d'un pixel dans le domaine temporel avec inverseur dont le seuil est commandé par une photodiode sous le BOX comme comparateur.

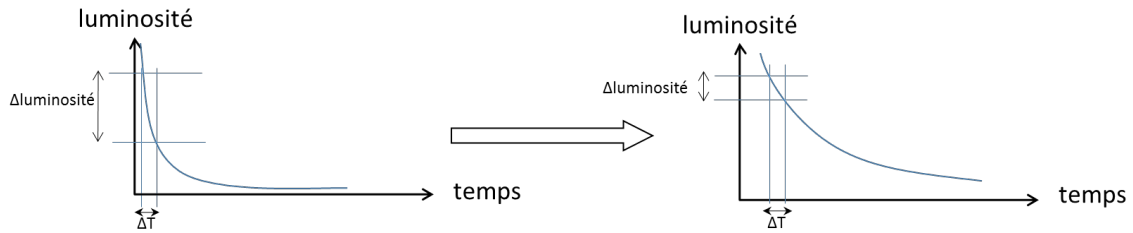


Figure 2-15: Représentation de la densité des événements, pour un intervalle de temps de flash donné, l'intervalle de luminosité est plus grand dans une architecture classique de TTFS [Guo2007] (graphe de gauche) que dans une architecture avec un comparateur dont le seuil est commandé par la luminosité (graphe à droite).

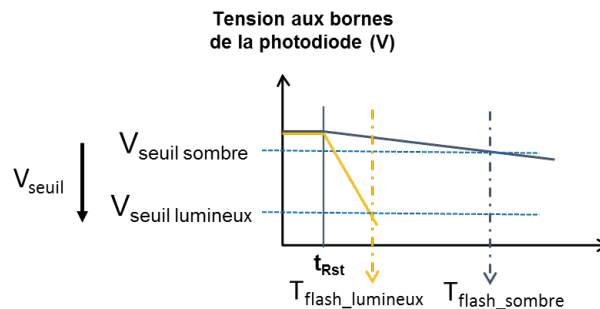


Figure 2-16: Déplacement du seuil du comparateur avec l'intensité.

Néanmoins tant qu'un modèle électronique de la réponse de la photodiode sous le BOX n'est pas disponible, il n'est pas possible de vérifier la viabilité de ces approches.

## 2.6.2 Implémentation 3D

### 2.6.2.1 Implémentation en 3D stack : principe et avantage

La taille de la lithographie est de plus en plus fine afin d'avoir des transistors dont les performances augmentent pour un coût de production plus faible. En effet, la vitesse des transistors augmente alors que leur consommation dynamique diminue pour une densité plus élevée sur une puce de même taille pour deux tailles de lithographie données. Les coûts de production et en consommation d'énergie sont donc réduits. L'implémentation 3D permet d'améliorer ces performances [Sukegawa2013] là où arrivent les limites de la réduction de la taille de la lithographie. Si la taille minimale possible en lithographie diminue rendant de ce fait les transistors plus rapides, les fils permettant de relier les transistors ne diminuent pas de tailles et limitent la rapidité du système [Sukegawa2013]. Ainsi en réduisant la taille de ces fils, le couplage capacitif entre les fils et leur environnement sera diminué et la vitesse pourra de nouveau augmenter. Les fils sont placés à la verticale reliant ainsi différentes puces ayant chacune une fonction particulière, Figure 2-17. De plus si le couplage capacitif est réduit, la consommation énergétique qu'il faut pour charger et décharger ces fils sera moindre. Le gain de densité d'une telle architecture apparait avec évidence puisque sur l'espace d'une seule matrice, les fonctions analogiques et digitales sont également implémentées. De plus, lorsque toutes les fonctions sont implémentées sur la même puce certaines étapes de fabrications sont appliquées à l'intégralité du circuit alors que seules certaines parties le nécessitent. En séparant les fonctions sur

différentes puces, cela optimise les étapes de fabrication, réduisant d'avantage les coûts de production. Actuellement, il n'y a pas d'imageur reliant grâce à un TSV (through-silicon-via) chacun des pixels à un ADC puis cet ADC à une unité de traitement. [Yamazaki2017] a une grande quantité de TSV mais uniquement sur un bord de la puce, les traitements se font en colonnes. L'architecture présentée dans [Takahashi2017] gère des régions d'intérêt de 10 par 16 pixels en les reliant à un ADC, ce qui permet d'avoir une lecture rapide et de gagner en consommation. Des problèmes d'alignement des différentes puces se posent également. Ces imageurs sont éclairés en face arrière (BSI : back side illumination). Les photons réagissent avec le substrat à quelques nanomètres de profondeur, le substrat est donc aminci afin que la lumière atteigne la zone de charge espace de la photodiode et que les charges soient collectées.

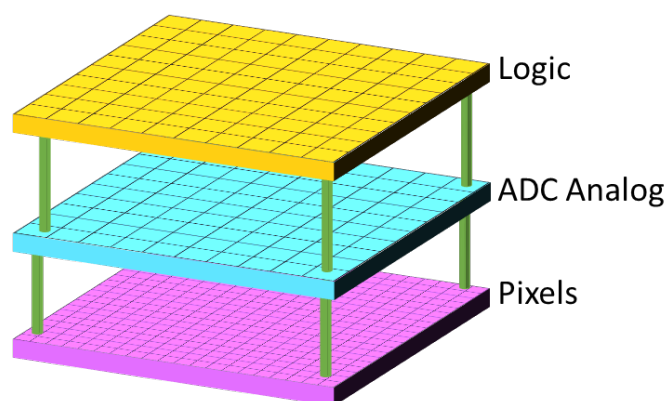


Figure 2-17: Illustration d'une implémentation 3D de 3 couches de circuit avec 4 TSV aux coins de la puce uniquement.

### 2.6.2.2 Implémentation en 3D séquentiel : principe, avantages et contribution de la thèse

Une autre méthode développée en 3D est la 3D parallèle en FD-SOI 28nm. Cela consiste à refaire une autre couche de transistors par-dessus la première couche de transistors FD-SOI classique grâce à un procédé à plus basse température non destructif pour les couches inférieures (CoolCube™). Des contacts permettent de relier les transistors de la couche inférieure à ceux de la couche supérieure. Les contacts sont encore plus courts que ceux d'une implémentation 3D-stacked, permettant de consommer encore moins. De plus, les problèmes d'alignements sont écartés.

Pendant la thèse, un circuit présentant 10 groupes de 3 par 3 pixels ont été envoyés en fabrication suivant ce procédé. Cette technologie évoluera en BSI grâce à un amincissement du substrat mais pour l'instant, les architectures se font en front side illumination. Néanmoins le design est totalement transposable. Les dix architectures réalisées correspondent à différentes configurations décrites dans la Figure 2-18, différentes configurations dans les transistors des pixels afin d'évaluer les performances des PMOS et des NMOS sur les couches de dessus et dessous, et des différents modèles de photodiodes : diode réalisée à partir d'un transistor bipolaire dite « diode bipolaire » et diode de modèle NP. Les modèles de photodiodes ainsi que le choix d'implémentation des photodiodes dans ce circuit seront expliqués dans la partie suivante. Différents layout sont présentés sur la Figure 2-19, elle présente le layout du design complet et certaines de parties du circuit.

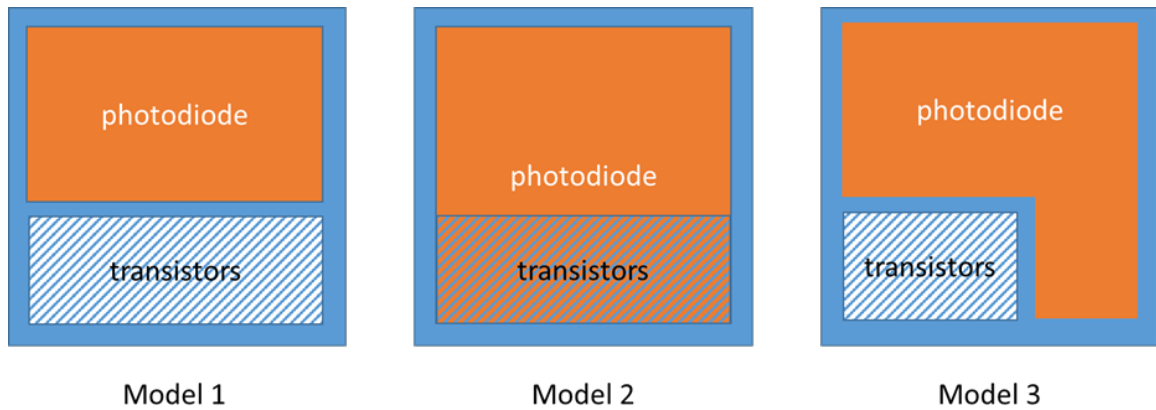


Figure 2-18: Configuration des pixels entre transistors et photodiode. Les transistors sont représentés par le rectangle hachuré, ils sont donc sur la couche CoolCube™. La photodiode est représentée par le polygone orange et est implémenté sur le wafer lui-même. Les résultats des différents seront comparés.

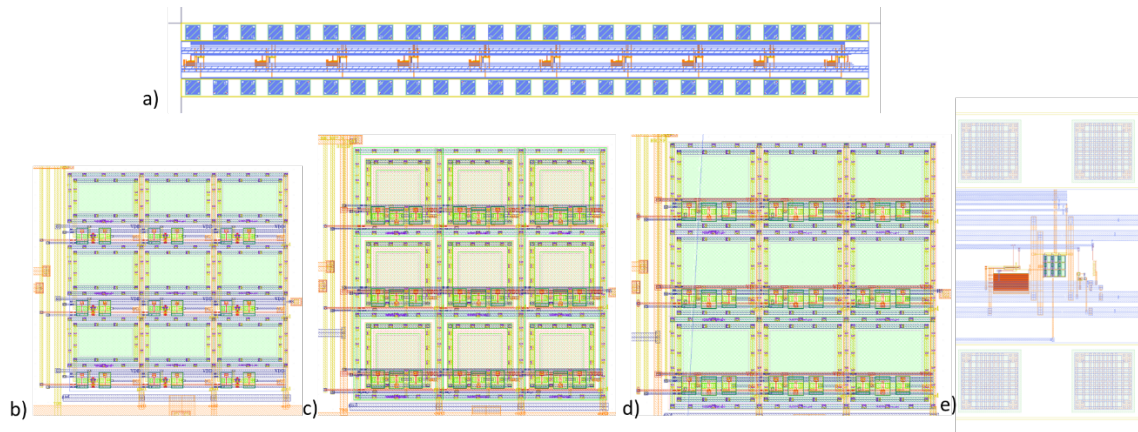


Figure 2-19: a) Layout de la contribution en entier avec les 2 rails de 25 pads chacun b) c) et d) exemple de matrice de 3 par 3 pixels avec différents modèles : Model 1 avec photodiode NP (b) Model 2 avec photodiode PNP (c) et Model 2 avec Photodiode NP (d). e) Matrice de 3 par 3 pixels entre les pads avec capacité visible.

## 2.7 Travaux de test réalisés pendant cette thèse

### 2.7.1 Vérification de la sensibilité à la lumière du FD-SOI 28nm

Afin de connaître la réponse électro-optique de la technologie FD-SOI 28nm et ainsi pouvoir designer une architecture compatible avec le niveau de signal reçu, un ensemble de pixel a été réalisé et testé.

Le circuit réalisé comprend 14 pixels 3T placé en série pour des raisons de quantité restreintes de plots de pointes afin de lire et envoyer les signaux. Les pixels sont lus un à un grâce à un système de bascule chaînées dont la sortie de chacune des bascules commande le transistor de lecture du pixel, Figure 2-20. Un jeton D est généré au début et il est propagé de bascule en bascule grâce au signal d'horloge (CLK), Figure 2-21. Le circuit final se présente comme sur la Figure 2-22, il y a donc 8 plots en haut et 6 plots en bas afin de venir poser les pointes. Et un exemple de chronogramme de lecture pour les 3 premiers pixels est décrit sur la Figure 2-23.

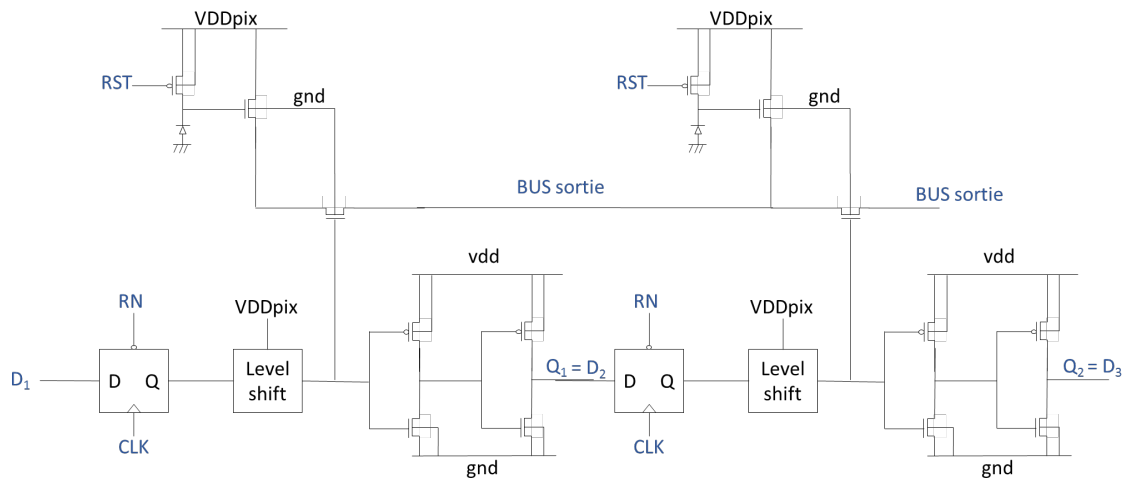


Figure 2-20: Schématique de deux pixels en série avec leurs bascules associés. Le bus de sortie est commun à tous les pixels.

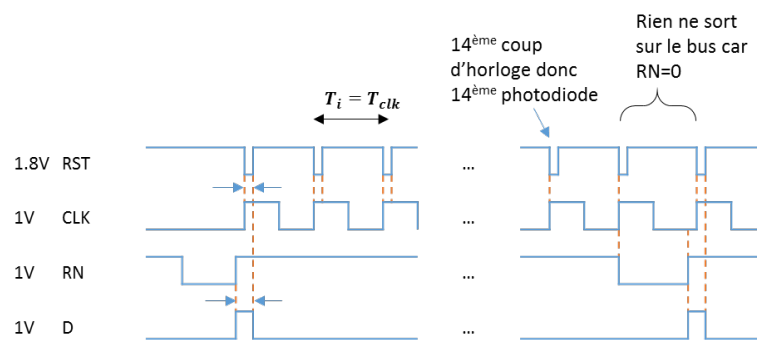


Figure 2-21: Chronogramme des signaux de contrôle pour le circuit

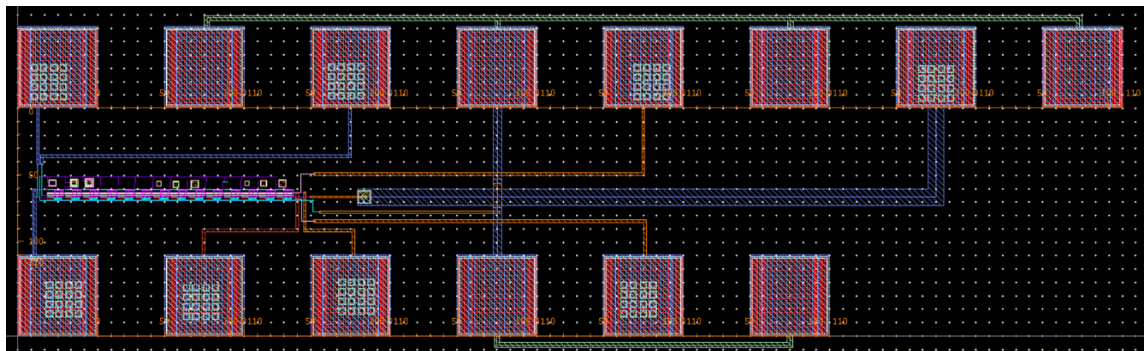


Figure 2-22: Circuit de test des 14 pixels 3T en série avec profils de photodiodes différents



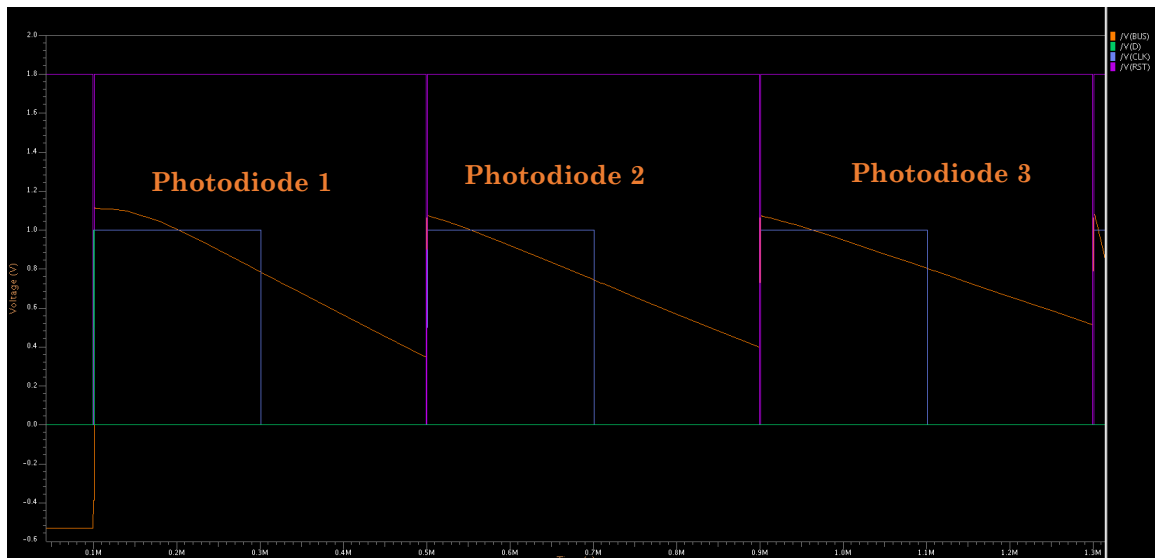


Figure 2-23: Chronogramme pour 3 photodiodes successives avec un reset à chaque coup d'horloge, CLK (bleu), RST (violet), Data D (vert), BUS (orange) (ici temps d'intégration 400 $\mu$ s)

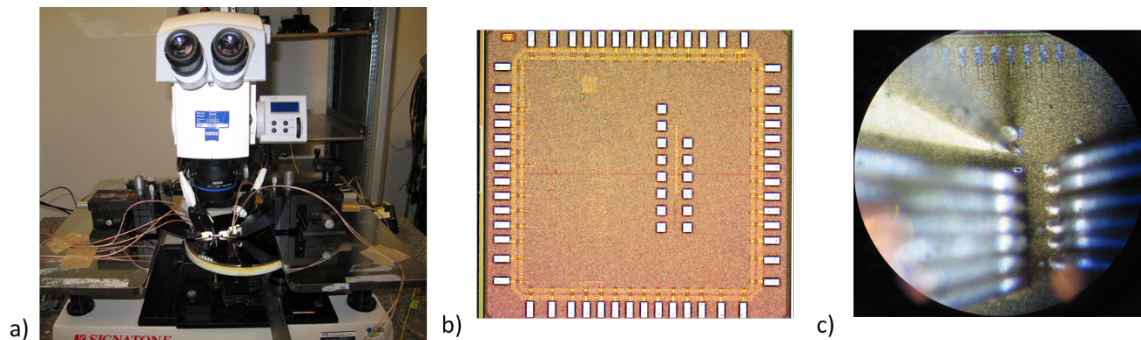


Figure 2-24: a) Photo du test sous pointe du circuit b) puce complète avec les rails de pad du circuit visible au milieu c) vision dans la binoculaire du circuit avec les pointes posées sur les pads.

### 2.7.1.1 Différents modèles de photodiodes et justifications

Plusieurs modèles de photodiodes ont été imaginés, en tirant parti de la technologie et en essayant d'optimiser la partie photosensible. En effet, la technologie FD-SOI 28nm peut avoir jusqu'à 10 niveaux de métaux, ce qui multiplie les risques de réfractions liés aux changements d'indices des couches de matériaux présents dans la couche de back end, et augmente les zones d'ombres, du fait de la hauteur des empilements de métaux. Une particularité de la technologie FDSOI-28nm est également d'avoir la possibilité de supprimer la couche d'isolant pour avoir accès à une technologie bulk classique. En effet, les photons du spectre de lumière visible, réagissent avec le silicium à une certaine profondeur, il faut donc que la zone de charge d'espace soit suffisamment profonde afin de pouvoir collecter correctement les porteurs.

Le premier modèle de photodiode imaginé est une liaison N<sup>+</sup>P classique, Figure 2-25. Il a été possible de réaliser ce modèle dans 6 tailles différentes, la plus petite étant

une  $0.5$  par  $0.5\mu\text{m}^2$  et la plus grande de  $5$  par  $5\mu\text{m}^2$ . Les tailles seront explicitées en détail dans la partie 2.7.1.2.

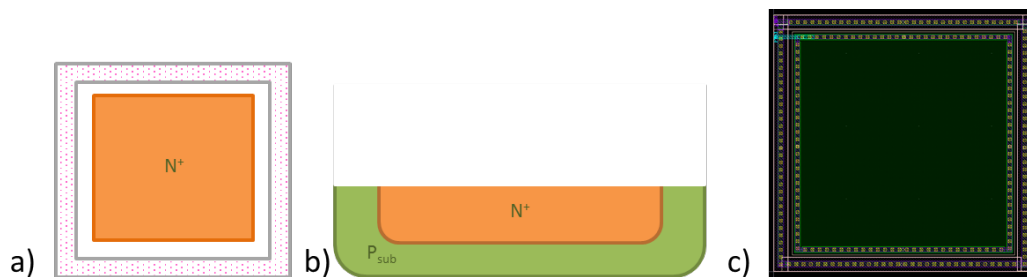


Figure 2-25 : Photodiode de modèle  $N^+P$  a) schéma layout vu de dessus b) vue en coupe c) layout

Puis, pour être capable de mesurer l'effet de la concentration des porteurs dans les zones dopées sur la réponse opto-électronique d'une photodiode, un deuxième modèle de photodiode a été imaginé. C'est une jonction  $N_{\text{well}}P$  comme montré sur la Figure 2-26. Seules 5 tailles ont pu être réalisées, la taille  $0.5$  par  $0.5\mu\text{m}^2$  ne permettant pas de respecter les règles de DRC.



Figure 2-26: Photodiode de modèle  $N_{\text{well}}P$  a) schéma layout vu de dessus b) vue en coupe c) layout

Enfin, à l'aide d'un transistor bipolaire, un troisième modèle de photodiode a été mis en œuvre. En reliant l'émetteur et le collecteur à la masse cela permet d'avoir deux liaisons PNP à différentes profondeurs dans le silicium. Les schémas et layout sont montrés sur la Figure 2-27, et la Figure 2-27.d représente le schéma équivalent de la photodiode sur modèle bipolaire. Seules 3 tailles ont pu être réalisées, les tailles  $0.5$  par  $0.5\mu\text{m}^2$ ,  $1$  par  $1\mu\text{m}^2$  et  $2$  par  $2\mu\text{m}^2$ , ne permettant pas de respecter les règles de DRC.

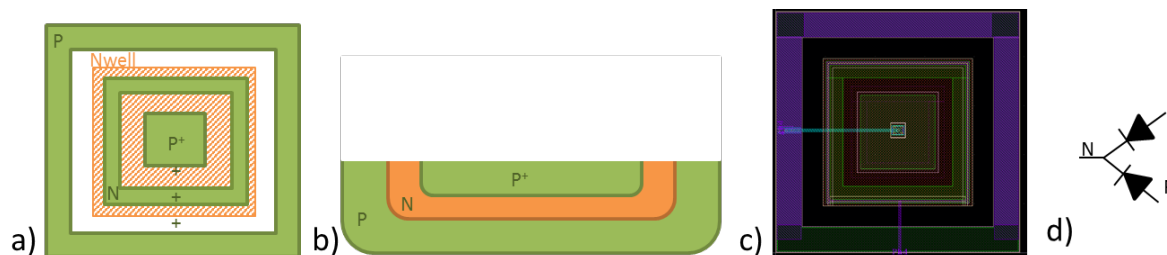


Figure 2-27: Photodiode de modèle bipolaire PNP a) schéma layout vu de dessus b) vue en coupe c) layout d) schéma équivalent

2.7.1.2 *Expérience et résultats*

Afin de tester la réponse électro-optique de ces photodiodes, des pixels ont été conçus puis testés sous pointes. Dans ce circuit les transistors MOS utilisés sont des transistors GO2 afin de minimiser l'impact des fuites de courant de grille (partie 2.5.2) dans les mesures de sensibilités. Une simulation « extract » a permis de connaître les capacités parasites dû aux métaux des différents nœuds photosensibles. Les valeurs de capacités ont été mesurées grâce à des simulations temporelles sous Cadence en mesurant la pente de la décharge de la tension de la photodiode polarisée à 1V à laquelle un courant de 100pA est appliqué. Des simulations TCAD ont également confirmé que la capacité d'une jonction N<sup>+</sup>P avec un profil d'implantation Gaussien et une concentration d'environ 1<sup>e</sup>18 at/cm<sup>3</sup>, était de 1fF/μm<sup>2</sup> à 0V. Pour une photodiode dont la surface est de 16μm<sup>2</sup>, la capacité est donc bien d'environ 16fF. La taille mesurée pour les photodiodes est la taille de la zone dopée N puisque la zone de charge espace où les photons seront générés dépend en majeure partie de la zone dopée N. Ainsi les capacités réelles des nœuds photosensibles peuvent être déterminées, elles évoluent entre 0.275f F et 25.6f F, Figure 2-28.

type	taille de la zone N (μm)	capacité réelle (F)	capacité parasite (F)	capacité de la photodiode à 0V (F)
PNP	3x3	9,43E-15	4,27E-16	9,00E-15
	4x4	1,66E-14	5,73E-16	1,60E-14
	5x5	2,56E-14	5,66E-16	2,50E-14
N <sup>+</sup> P	0,5x0,5	2,75E-16	2,53E-17	2,50E-16
	1x1	1,03E-15	2,85E-17	1,00E-15
	2x2	4,05E-15	5,20E-17	4,00E-15
	3x3	9,04E-15	3,50E-17	9,00E-15
	4x4	1,60E-14	4,18E-17	1,60E-14
	5x5	2,50E-14	4,50E-17	2,50E-14
N <sub>well</sub> P	1x1	1,05E-15	5,39E-17	1,00E-15
	2x2	4,24E-15	2,43E-16	4,00E-15
	3x3	9,37E-15	3,73E-16	9,00E-15
	4x4	1,65E-14	4,90E-16	1,60E-14
	5x5	2,56E-14	6,12E-16	2,50E-14

Figure 2-28: capacité du nœud photosensible du pixel en fonction du type de photodiode et de la taille de la zone N.

Suite aux tests sous pointes des puces revenues de fonderie, le courant photo-généré a pu être mesuré en utilisant (2-4).

$$i_{ph} = C_{pd} \frac{\Delta V}{\Delta T} \quad (2-4)$$

La Figure 2-29 permet d'avoir un aperçu d'une mesure réalisée sous un éclairage de 7000lux sur le pixel n°2 du circuit.



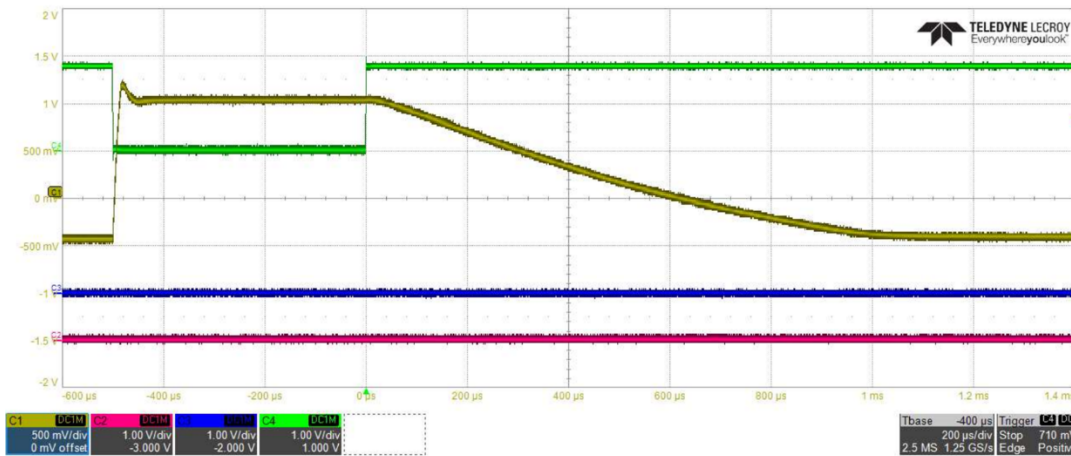


Figure 2-29: Mesure de la décroissance de la tension en sortie d'un pixel à l'oscilloscope. La tension du bus de sortie est en jaune avec 500mV/div. Le signal vert correspond au reset de la matrice avec 1V/div. Le signal bleu est la clock et le signal rose le signal D, tous deux à 1V/div. Ils sont à 0 puisque l'on pointe sur le pixel 2.

L'éclairage sur la puce a été mesuré à environ 7000lux avec un luxmètre, la lumière est principalement envoyée de la lampe de la binoculaire. La plage de tension a été mesurée ( $\Delta V$ ) ainsi que la plage de temps correspondante ( $\Delta T$ ). Le courant photo-généré déduit, varie donc entre 2,12fA et 50,1pA, Figure 2-30. Les photodiodes de type  $N_{well}P$  n'ont pas pu être testées. Les photodiodes sont placées en série et un système de contrôle de lecture permet d'autoriser l'accès au bus de sortie pour les pixels un à un. Les pixels avec une photodiode de type  $N^+P$  sont en position 10 à 14 et les signaux de sélections ne se propageaient pas plus loin que la photodiode n°9.

type	taille de la zone N ( $\mu m$ )	capacité du nœud (F)	courant photogénéré (A)
PNP	3x3	9,43f	13,2p
	4x4	16,6f	27,6p
	5x5	25,6f	N/A
$N^+P$	0,5x0,5	0,275f	2,12f
	1x1	1,03f	103f
	2x2	4,05f	4,34p
	3x3	9,04f	19,9p
	4x4	16f	29,2p
	5x5	25f	50,1p

Figure 2-30: Courant intégré mesuré en fonction du type de photodiode et de la capacité du nœud photosensible. Le design utilise des transistors GO2 afin de pouvoir négliger l'impact des fuites de courant. Ainsi le courant intégré correspond au courant photo-généré.

Une estimation du courant maximal pouvant être obtenu peut être calculée. 7000lux équivaut à une puissance d'éclairage de 10,25W/m<sup>2</sup>. La convention utilisée est 1W équivaut à 683 lumen pour une longueur d'onde de 550nm, les photons verts. En considérant l'énergie d'un photon vert à 3.6e<sup>-19</sup> J, l'énergie de l'électron à 1,6e<sup>-19</sup>J et pour une photodiode de surface 16 $\mu m^2$ , le courant photo-généré pour un rendement électro-optique de 100% (1 photon produit 1 électron) est de 65pA (2-6). Le rendement

électro-optique avec la photodiode PNP de  $16\mu\text{m}^2$  a donc un rendement de 43% alors que celui avec la photodiode N<sup>+</sup>P de  $16\mu\text{m}^2$  est de 45%. De plus les contraintes DRC sont plus relâchées sur le model N<sup>+</sup>P.

$$i_{ph} = \frac{\text{énergie électrique reçue}}{\text{énergie photonique}} * \text{taille photodiode} * \text{énergie électron} \quad (2-5)$$

$$i_{ph} = \frac{10,25}{3,61.10^{-19}} * 16.10^{-12} * 1,6.10^{-19}$$

$$i_{ph} = 65.10^{-12} A \quad (2-6)$$

Le courant photo-généré avec un rendement électro-optique de 45% et une photodiode de  $25\mu\text{m}^2$  soit 25fF est récapitulé dans la Table 2-2 pour différentes luminosités. Une salle de travail éclairée aura un éclairage entre 200 lux et 4000 lux alors qu'une scène extérieure par ciel dégagé aura un éclairage entre 50 000 lux et 100 000 lux.

éclairage (lux)	courant iph (A)
0,5	3,65f
20	0,146p
70	0,511p
100	0,730p
200	1,46p
400	2,92p
500	3,65p
3000	21,9p
7000	51,1p
15000	110p
25000	183p
50000	365p
100000	730p

Table 2-2: Courant photo généré pour différents éclairage pour une photodiode de  $25\mu\text{m}^2$  et un rendement de 45%

Lors de la réalisation du circuit en 3D séquentiel, les deux types de photodiodes PNP et N<sup>+</sup>P ont été implémentés. A la lumière des résultats présentés en amont, les matrices de pixels ayant été implémentées avec des photodiodes de type N<sup>+</sup>P devraient avoir un photo-courant plus élevé que celles implémentées avec une photodiode de type PNP.

### 2.7.2 L'approche envisagée de l'architecture par rapport aux avantages et inconvénients technologiques et des acquisitions des imageurs.

Les parties précédentes ont soulignés les avantages et les inconvénients de la technologie FDSOI-28nm. En effet, le courant mesuré serait donc avec la technologie FDSOI 28nm jusqu'à 50pA pour une photodiode de  $5 \times 5\mu\text{m}^2$  sous un éclairage de 7000lux. Le design utilisé pour mesurer ce courant a été implémenté en utilisant des transistors GO2 permettant de négliger les fuites de courant de grille des transistors. Néanmoins lors de l'utilisation de transistors GO1, ces fuites ne sont pas négligeables. En effet, cela représente 800fA de courant de fuite de grille pour un transistor de taille W/L=300n/150n. Ce courant peut être réduit grâce aux techniques de polarisation par le substrat disponible en technologie FDSOI 28nm. Dans l'optique de rendre

l'architecture finale de cette thèse compatible avec une implémentation 3D séquentielle, seuls les transistors GO1 peuvent être utilisés. Ainsi afin de garantir un SNR optimal en une acquisition dans le domaine temporel semble préférable. De plus nous verrons qu'effectuer des traitements sur le plan focal et transformer la donnée d'intensité lumineuse en métadonnée représentant l'image semble être une possibilité viable en vue des contraintes de la technologie et des applications visées (voir chapitre 4).

## 2.8 Conclusion

L'objectif de cette thèse est de définir une architecture intégrant des traitements dans l'imageur pour avoir un system on chip (SoC) performant énergétiquement. La technologie est donc intéressante car elle présente des possibilités d'intégration dense ainsi que des ressources de calcul conséquentes disponibles. En revanche la technologie peut présenter des non idéalités, en termes de niveaux de fuites, notamment par la grille des transistors, ainsi qu'en termes de performances électro-optiques. C'est pourquoi la thèse s'oriente vers une architecture de type TTFS. En effet, ce type d'architecture est robuste par rapport aux fuites et sera également intéressant d'un point de vue génération des opérateurs de calcul (ordonnancement intrinsèque des valeurs de pixels) présenté dans le chapitre IV.

De plus, un circuit de test de pixels comportant différents modèles de photodiodes a été réalisé et testé en partie. Le test chip a permis de dégager la réponse électro-optique avec des transistors en G02 (qui permettent de limiter l'impact des fuites) et valide donc la possibilité d'acquérir du signal.

La technologie FDSOI 28nm se développe également pour des architectures 3D, que ce soit du 3D dit 3D stack ou du 3D dit 3D séquentiel avec la méthode de fabrication CoolCube™. Cela permettrait de réduire d'autant plus la consommation d'énergie et d'augmenter drastiquement la densité des transistors. Un circuit de dix matrices de 3x3 pixels a été réalisé pendant la thèse, il permettra ultérieurement à la thèse d'avoir des informations sur l'efficacité et la pertinence d'un tel circuit, notamment en terme de diffusion thermique. Cependant seuls les transistors GO1 sont compatibles avec l'implémentation 3D stack CoolCube. Les architectures développées par la suite seront développées en GO1 en TTFS, type d'architecture sélectionné pour sa robustesse et son intérêt pour le processing intégré.

A noter, que le développement de la technologie FDSOI 28nm au LETI permet d'intégrer de nouvelles fonctions grâce à une photodiode implémentée sous le BOX des transistors ce qui a pour effet de déplacer la tension de seuil du transistor sous l'effet de la lumière incidente. Des architectures intégrant cette nouvelle fonction ont été imaginées, elles seraient utiles pour des capteurs événementiels ou des capteurs dans le domaine temporel. Ces architectures ont été imaginées durant la thèse mais ni simulées ni testées. Elles n'ont pas été retenues pour la suite des travaux de cette thèse puisque le modèle de la photodiode sous le BOX des transistors n'est pas encore disponible.



## 3 Chapitre III : Mécanismes d'acquisitions d'images pour technologie avancée

Les mécanismes d'acquisition d'images pour une implémentation en technologie avancée doivent prendre en compte les différents avantages et limites de la technologie explicités dans le chapitre 2 (paragraphe 2.5). En effet, afin de ne pas dégrader voire perdre la donnée lumineuse, les architectures envisagées doivent être robustes aux fuites et, pour ce faire, permettre une digitalisation rapide de la donnée.

Dans ce chapitre, deux architectures seront présentées : l'IM-TTFS et la transmission sur bus multiples avec un système de réinitialisation des pixels. L'IM-TTFS permet d'adresser les problèmes soulevés par une architecture classique TTFS. En effet, dû à une densité plus importante, dans le domaine temporel, des événements correspondant à de fortes luminosités, il peut y avoir un goulet d'étranglement de données en sortie résultant en une diminution du PSNR et donc de la qualité de l'image. L'architecture « transmission sur bus multiples avec un système de réinitialisation des pixels » permet d'utiliser les avantages de densité d'intégration offerts par la technologie FDSOI 28nm. Les deux architectures seront ensuite comparées afin de pouvoir choisir quelle architecture adopter pour l'intégration des traitements d'images dans le capteur par la suite.

### 3.1 Architecture TTFS avec mécanisme d'inhibition: Time to first spike – Inhibition Mechanism: IM-TTFS

#### 3.1.1 Le domaine temporel

Le domaine temporel permet de représenter la donnée lumineuse non plus en tension ni en courant mais en temps. Comme expliqué dans les chapitres précédents (partie 1.3.1.2), la photodiode est initialisée à une tension haute puis sous l'incidence des photons, une paire électron-trou se forme créant un courant déchargeant la capacité de la photodiode. Lors d'une acquisition classique, l'intensité lumineuse reçue par le pixel est déterminée par la tension obtenue aux bornes de la photodiode au bout d'un temps d'intégration  $T_{int}$  fixé. On déduit la valeur de l'intensité lumineuse en fonction de la valeur de la tension lue à travers les transistors suiveur et de sélection pour un pixel standard, Figure 3-1.

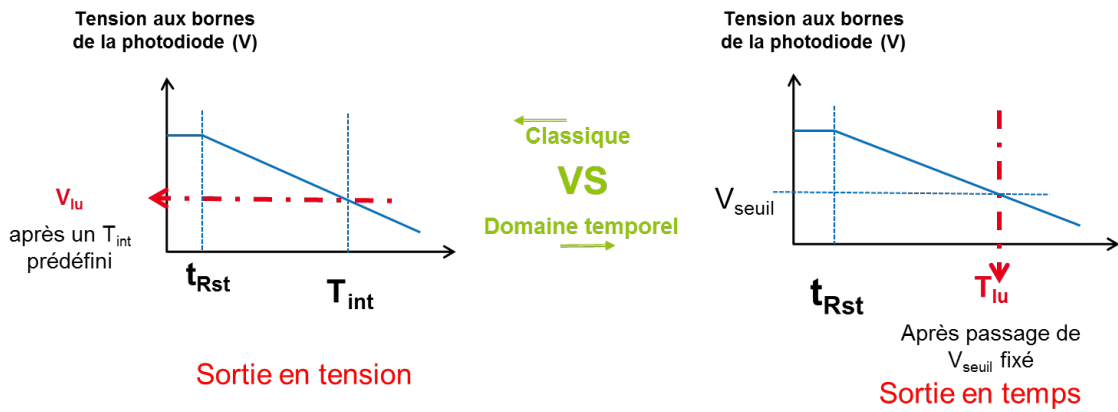


Figure 3-1: Différence entre une acquisition dite classique et une acquisition dans le domaine temporel

Lors d'une acquisition dans le domaine temporel, le temps d'intégration n'est pas fixé, ce qui permet à chacun des pixels de définir son propre temps d'intégration et d'obtenir ainsi une plus grande dynamique d'image vers les faibles intensités. La valeur de tension de seuil est, quant à elle, fixée. Lorsque la tension aux bornes de la photodiode dépasse le seuil prédéfini, le temps auquel cet événement se produit est lu et/ou sauvegardé. Le temps correspondant à l'intervalle entre le temps de reset et le temps lu est appelé le temps de flash,  $T_{flash}$ , il est inversement proportionnel à l'intensité lumineuse. L'intensité lumineuse estimée dépend également de la capacité de la photodiode et de l'intervalle entre la tension de polarisation de la photodiode communément appelée  $V_{DD}$  et la tension de seuil  $V_{seuil}$ , 3-1). Quantitativement, plus l'intensité lumineuse reçue est importante et plus la pente de décharge de la photodiode est raide, Figure 3-2.

$$I = C_{phD} \frac{V_{DD} - V_{seuil}}{T_{lu} - T_{rst}} = C_{phD} \frac{V_{DD} - V_{seuil}}{T_{flash}} \quad 3-1)$$

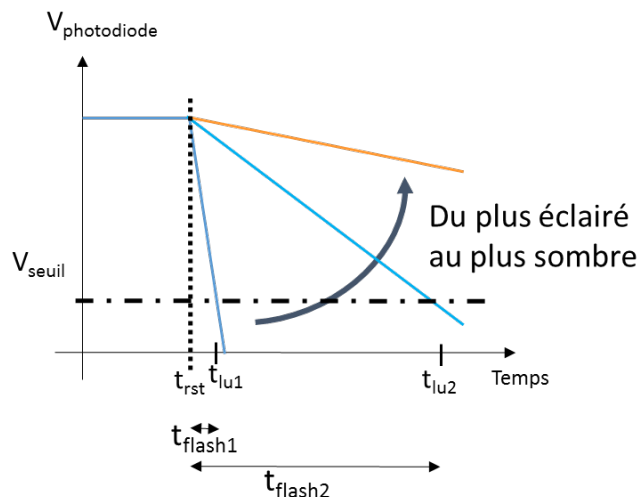


Figure 3-2: Illustration des intensités reçues par des pixels et de l'impact sur la pente de la décharge de la photodiode.

Il existe plusieurs méthodes pour traiter les données acquises dans le domaine temporel comme expliqué dans le chapitre I. Ce chapitre sera dédié à la description détaillée de la méthode appelée le time to first spike, TTFS, ou time to saturation.

### 3.1.2 Time to first spike et les problèmes associés

La méthode « time to first spike », [Guo2007], permet de traiter les données temporelles des pixels en associant l'adresse du pixel à l'horodatage du moment où la tension de la photodiode est passée sous le seuil prédéfini.

Tout d'abord un événement est généré dans le pixel grâce à un comparateur qui commute lorsque la tension du nœud photosensible passe sous le seuil fixé, Figure 3-3. L'événement généré est un signal digital. Puis, le pixel envoie une requête d'accès ligne qui est traitée par une communication spécifique entre le pixel et le système de lecture : un « double handshake ».

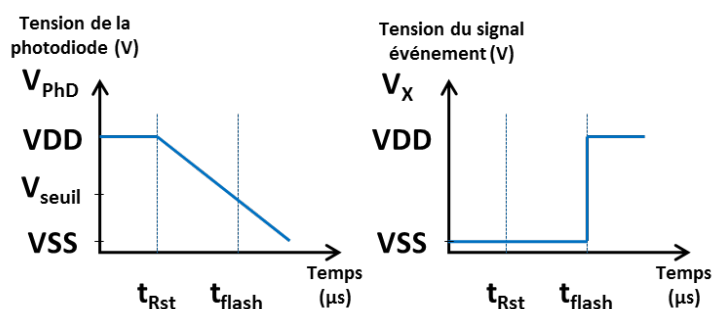


Figure 3-3: Génération d'un événement lors de la décharge d'un événement

Le double handshake est une pratique commune en logique asynchrone qui permet d'éviter les collisions lorsque deux pixels flashent en même temps. La collision d'événements entraîne la perte d'information. Le pixel envoie sa requête « require » au système AER. Lorsque le système est disponible, une réponse « acknowledge » est envoyée au pixel. Celui-ci peut maintenant transmettre sa donnée, ici son adresse, sans se soucier d'une potentielle collision. A la fin de la transmission, le pixel éteint la requête « require » ce qui relâche le signal « acknowledge ». Le système est de nouveau disponible. Afin de faciliter le routage, les signaux sont mis en commun par lignes et par colonnes. De plus dans le cas d'un double handshake, il y a une requête ligne et une requête colonne. Cela signifie qu'il y a également un signal acknowledge ligne et un signal acknowledge colonne. Le fonctionnement est le même que décrit précédemment et est montré sur la Figure 3-4.

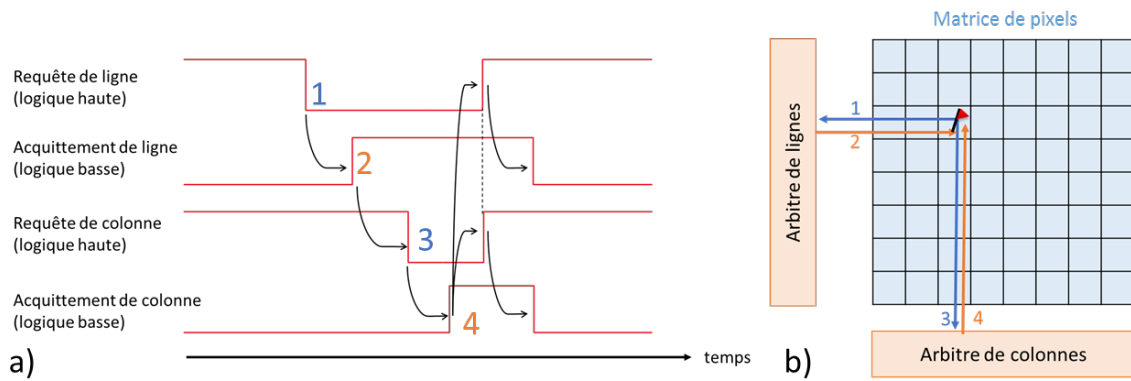


Figure 3-4: a) Mécanisme du « double handshake » b) systèmes des arbitres et signaux entrant et sortant pour réaliser le handshake dans le cas de cette architecture.

Une fois que le pixel a envoyé son adresse, il est possible de retrouver l'information lumineuse grâce à l'horodatage associé à l'adresse du pixel, 2-3).

La méthode TTFS apporte plusieurs avantages associés à la lecture asynchrone, [Sparso2001]. En effet, la consommation d'énergie est diminuée, la vitesse de traitement augmentée et le processus est plus robuste qu'une acquisition classique (voir partie 2.6). De plus, le temps d'intégration est adapté pour chaque pixel, ce qui permet d'avoir des images avec une grande dynamique. Cependant cette dynamique est limitée par la précision du compteur ainsi que par la profondeur de comptage (voir partie 1.3.1.2).

Lors du passage dans le domaine temporel, la densité d'événements est plus importante pour les pixels les plus lumineux que pour les pixels plus sombres. En effet, un même contraste dans des conditions lumineuses générales différentes (faibles et fortes) ne présentera pas la même différence de valeurs temporelles. Ce phénomène est illustré dans la Figure 3-5.a, la différence des temps de flash correspondant respectivement aux intensités 200 et 150 est plus faible que la différence de temps de flash correspondant respectivement aux intensités 100 et 50. Le système de lecture reçoit donc plus de requêtes la première milliseconde que dans le reste du temps trame, pour une image donnée et un temps trame de 20ms, Figure 3-5-b&c. Ce phénomène provoque un goulet d'étranglement pour l'évacuation des données. De ce fait, le système de handshake induit des délais. L'information étant codée en temps, un délai crée une erreur dans la reconstruction de l'intensité lumineuse. Il faut donc imaginer des mécanismes de réduction de données permettant de réduire le flot de données sans pour autant dégrader la qualité de l'image acquise.



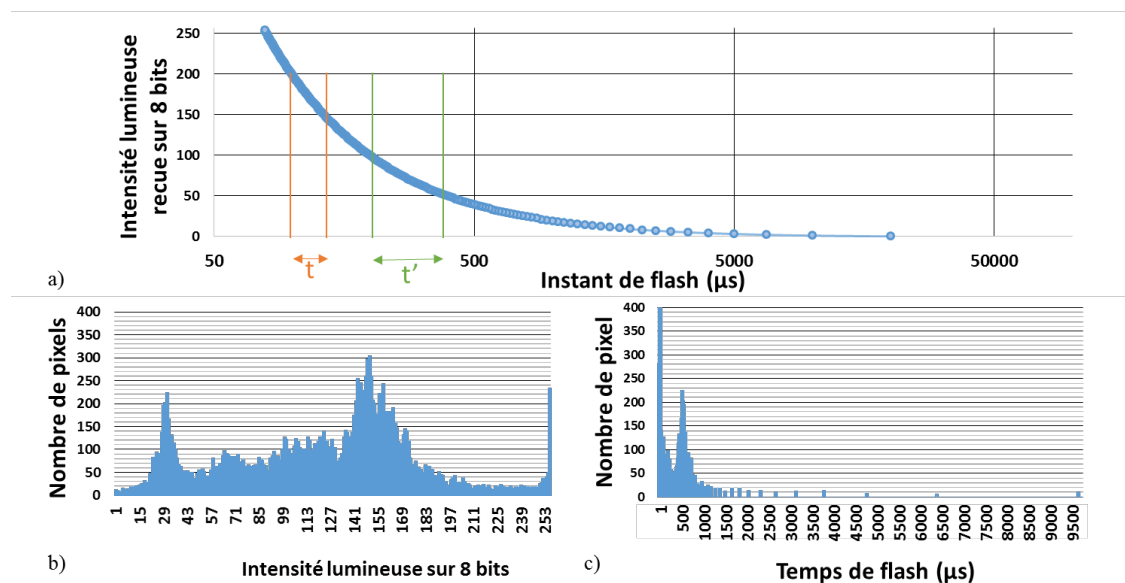


Figure 3-5: a) Intensité lumineuse reçue sur 8 bits en fonction du temps de flash b) Histogramme d'une image sur 8 bits c) Histogramme de la même image dans le domaine temporel.

### 3.1.3 Mécanisme d'inhibition

Afin de réduire le flux de données, un mécanisme d'inhibition des pixels a été imaginé dans un stage effectué en amont de la thèse par David Dehaene. Au cours de cette thèse, la modélisation de l'architecture complète a été mise en place sous MATLAB afin de pouvoir effectuer des simulations sur une image et pouvoir ainsi, exploiter et comparer les résultats obtenus (3.1.3.2). Ce mécanisme permet au sein d'un bloc défini, de regrouper les pixels de même intensité et de les traiter en même temps. Ce mécanisme sera tout d'abord expliqué puis les résultats des simulations MATLAB, effectuées afin de prouver l'efficacité, seront explicités et enfin l'architecture du pixel sera développée.

#### 3.1.3.1 Principe de fonctionnement

Réduire les redondances spatiales peut permettre de réduire le flux de données ce qui minimise les risques de collisions, la matrice est divisée en blocs de  $N$  par  $N$  pixels. L'initialisation, ou reset, de la matrice est globale. Au sein d'un même bloc, les pixels ayant une information similaire au pixel le plus éclairé, sont regroupés sous une seule adresse transmise. Pour ce faire, un signal d'inhibition est généré lorsque le premier pixel du bloc flashe, i.e. lorsque la tension de la photodiode dudit pixel passe en dessous de la tension de seuil préalablement fixée. Le temps pendant lequel ce signal est actif, est appelé le temps d'inhibition. Un pixel qui flashe pendant ce temps d'inhibition est donc inhibé, il n'envoie pas de requête de transmission d'adresse et s'éteint. Après le temps d'inhibition, les pixels qui flashent peuvent envoyer leurs requêtes normalement. Les pixels inhibés quant à eux restent éteints jusqu'au prochain signal de reset global qui vient également réinitialiser le système d'inhibition. Ce mécanisme est décrit dans la Figure 3-6. Au temps 1, moment où la tension de la photodiode du pixel le plus éclairé passe sous le seuil, le signal d'inhibition se déclenche et reste actif jusqu'à  $T_{flash} + T_{inhib}$ . Pendant la période 2 correspondant au temps d'inhibition, les

pixels qui flashent sont éteints et leurs requêtes inhibées. Le temps 3 correspond à un instant quelconque après le temps d'inhibition. Le pixel flashant à cet instant envoie donc sa requête normalement et s'éteindra lorsqu'il aura envoyé son adresse au système de lecture. Le pixel n'inhibe pas le bloc puisqu'il ne peut y avoir qu'une seule inhibition par bloc et par trame. L'instant 4 correspond à la fin du temps trame, la tension des photodiodes des pixels recevant trop peu de lumière n'a pas pu passer sous le seuil. Les pixels sont donc forcés à flasher et à envoyer leur adresse afin de ne pas les confondre avec des pixels inhibés au niveau de la mémoire. En effet, lorsque la première adresse du bloc est reçue par la mémoire avec l'horodatage correspondant, la valeur de l'intensité déduite est inscrite pour tous les pixels du bloc. Lorsque d'autres adresses du bloc arriveront alors, cette valeur sera actualisée pour l'adresse du pixel reçue. A la fin du temps trame, seuls les pixels inhibés n'auront pas actualisé leur valeur sur 8bits et garderont donc la même que celle du pixel le plus éclairé du bloc, comme montré sur la Figure 3-7.

A noter que, afin de simplifier le mécanisme AER, lorsque qu'une requête ligne est reçue et acquittée, i.e. le signal d'acquiescement pour la ligne est envoyé, alors l'intégralité de l'état de la ligne est mémorisée. Ainsi, tous les pixels actifs sur cette ligne appartenant à d'autres blocs envoient leurs adresses colonnes et sont acquittés un par un. Une fois que toutes les adresses colonnes sont envoyées, l'adresse ligne est envoyée, Figure 3-8. Cela permet de n'envoyer qu'une seule fois l'adresse de ligne et donc de réduire le flot de donnée davantage. Et puisque l'inhibition commence lorsque le signal d'acquiescement est envoyé, cela permet de commencer l'inhibition pour plusieurs blocs de la ligne.

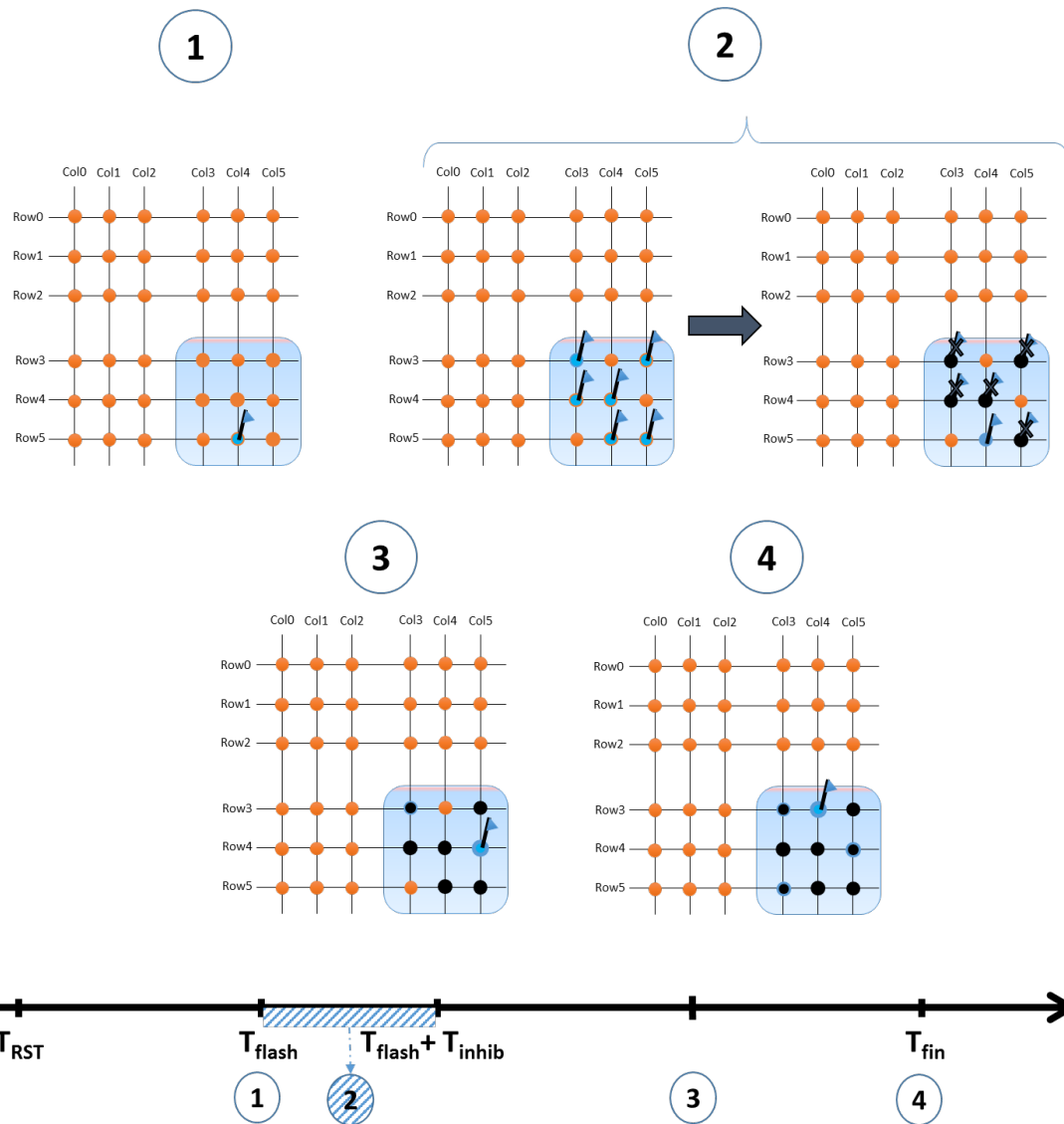


Figure 3-6: Mécanisme d'inhibition pour un bloc sur un temps frame

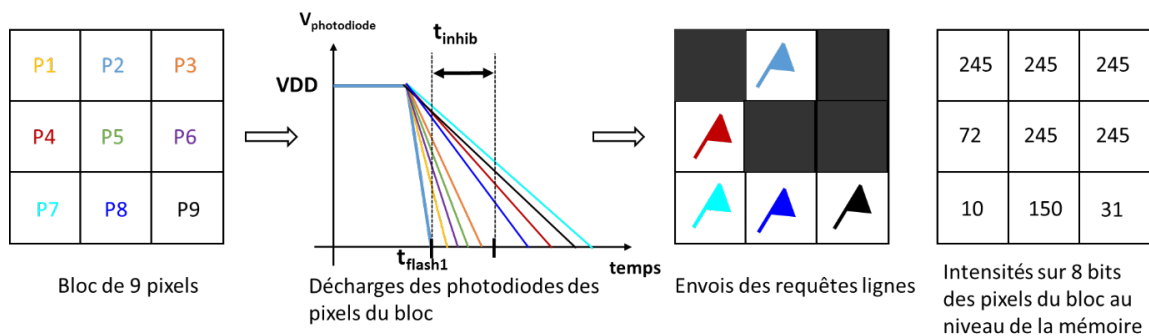


Figure 3-7: Schéma de l'acquisition avec le mécanisme d'inhibition et intensités au niveau de la mémoire, pour un bloc de 9 pixels.

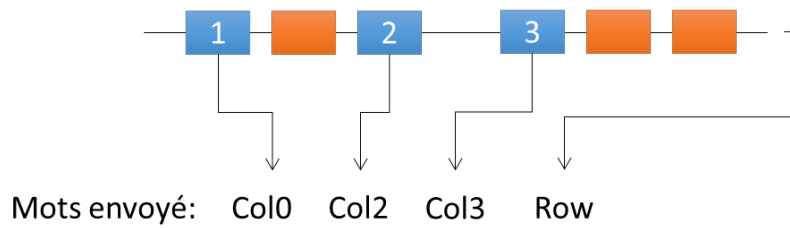


Figure 3-8: Ligne de pixel ayant reçu le signal d'acquittement. Les adresses des colonnes (Col) sont envoyées puis l'adresse de la ligne (Row).

### 3.1.3.2 Simulations MATLAB et résultats

Le concept a été tout d'abord validé par des simulations MATLAB. Pour ce faire, un set d'images sur 8 bits, Figure 3-9, a été transposé dans le domaine temporel avec un temps trame de 20 ms fixant les pixels les plus sombres. Cela équivaut à une dynamique de 86dB pour 50fps (3.2). Grâce à cette transposition, la séquence des requêtes lignes et colonnes des pixels a été générée. Il est donc possible de les traiter avec le mécanisme d'inhibition et d'AER. A la fin de la simulation, il est possible pour chaque image de calculer le pourcentage de réduction du flux de données ainsi que de montrer l'image reconstruite avec le PSNR associé. Ces simulations ont été réalisées pour différents temps d'inhibition et différentes tailles de blocs. Les résultats de la simulation montrent que le flux de données peut-être réduit entre 5% (pour l'image Figure 3-9-d) et 50% (pour l'image Figure 3-9-a) comparé à une approche TTFS traditionnelle où tous les pixels envoient leurs adresses, Table 3-1.

$$DR = 20 * \log \left( \frac{t_{int,max}}{t_{int,min}} \right)$$

$$DR = 20 * \log \left( \frac{20 * 10^{-3}}{1 * 10^{-6}} \right) = 86dB \quad (3.2)$$

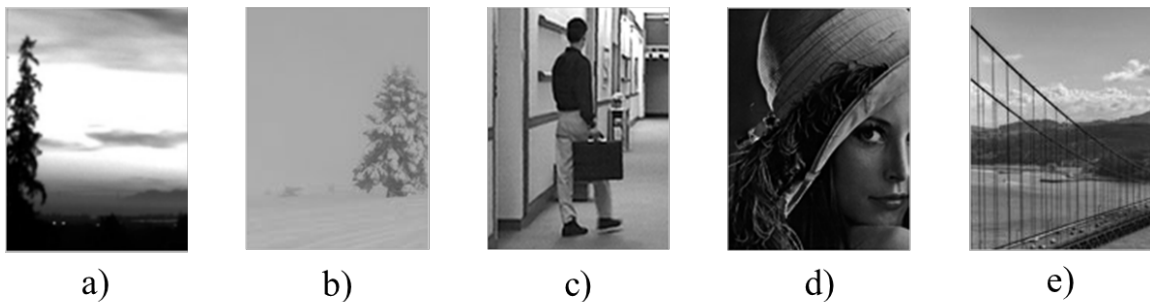


Figure 3-9: Set d'image utilisé pour les simulations MATLAB

Taille d'image 160x120	$T_{\text{inhibition}} = 4\mu\text{s}$ taille des blocs : 4x4 pixels	
	Data %	PSNR (dB)
Vine Sunset	51,43	41,74
Snow	25,51	44,54
Hall Monitor	77,73	43,93
Lena	96,65	55,43
Bridge	71,33	44,42

Table 3-1: Pourcentages des données envoyées et PSNR (dB) pour différentes images avec un temps d'inhibition de  $4\mu\text{s}$  et une taille de bloc de 4x4 pixels

La cartographie, Figure 3-10, permet de visualiser les données transmises (pixels blancs) et les données inhibées (pixel noirs). Lors de la simulation, seules les données transmises sont utilisées pour reconstruire l'image.

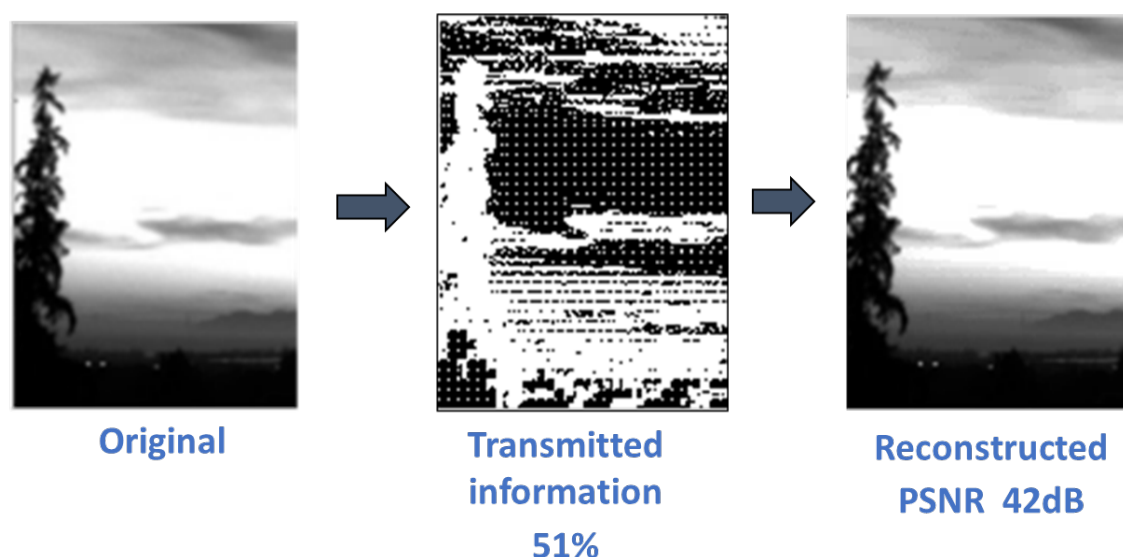


Figure 3-10: Résultat de la simulation pour l'image "vine sunset" avec des blocs de 4x4 et un temps d'inhibition de  $4\mu\text{s}$  pour un temps trame de 20ms

Cependant, des artéfacts sont observés, les blocs sont visibles sur la reconstruction de l'image, Figure 3-11. Les erreurs observées peuvent être expliquées par deux variables : le temps de lecture des données et la taille des blocs. En effet, le temps de lecture correspond à la capacité des bus à transmettre les adresses suffisamment rapidement afin de ne pas créer de délai ni de file d'attente d'évacuation des données. Afin d'éviter les collisions et les conflits, un arbitrage des requêtes est effectué grâce à un handshake et un arbre d'arbitrage, ils seront explicité en annexe. Une autre manière d'éviter le problème de délai peut être de réduire la résolution du capteur en fonction de l'application visée. La résolution du capteur et la taille des blocs sont corrélés à la surface de l'objet à détecter. La taille des blocs influence le flux de données ainsi que la qualité de l'image mesurée par la PSNR, Figure 3-12. Lorsque la taille des blocs augmente, le flux de donnée diminue mais le PSNR également. Dans le cadre des simulations faites pour des images QQVGA (160x120), un compromis acceptable entre la taille des blocs, la qualité de l'image et le pourcentage de données transmises

serait des blocs de 4 par 4 pixels et une inhibition de 4 $\mu$ s, le temps trame étant toujours fixé à 20ms.

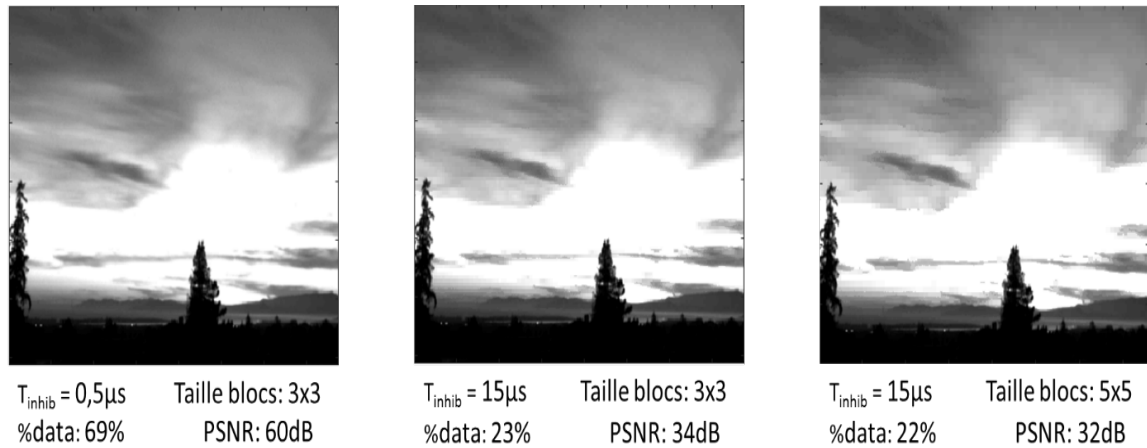


Figure 3-11: Effets du temps d'inhibition et de la taille des blocs sur la quantité de données transmises et la qualité de l'image.

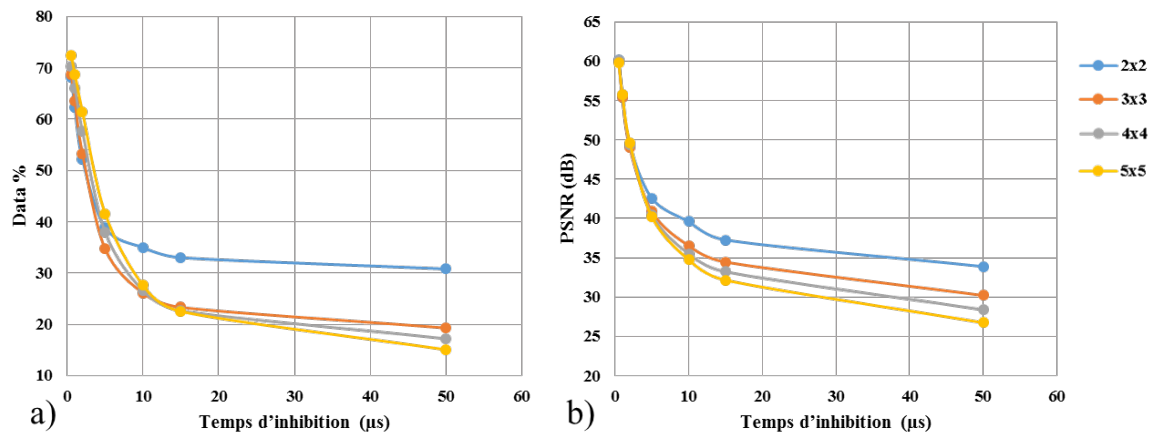


Figure 3-12: Graphes mettant en relief les effets du temps d'inhibition et de la taille des blocs sur le pourcentage de données transmises (data %) et la qualité de l'image (PSNR dB)

### 3.1.3.3 Architecture développée

L'architecture développée a fait l'objet d'un dépôt de brevet, [Dehaene2016]. En annexe, un schéma synoptique explique le fonctionnement du système de manière globale.

#### 3.1.3.3.1 Architecture du pixel

Le pixel est conçu en quatre parties entremêlées : la photodiode et le comparateur, les transistors d'initialisation et de signal de fin de trame, le mécanisme de handshake dans le pixel, et le mécanisme d'inhibition dans le pixel. Chacune de ces parties sera expliquée ci-dessous. L'architecture de ce pixel s'appuie sur celle d'un pixel classique TTFS [Shoshun2007].

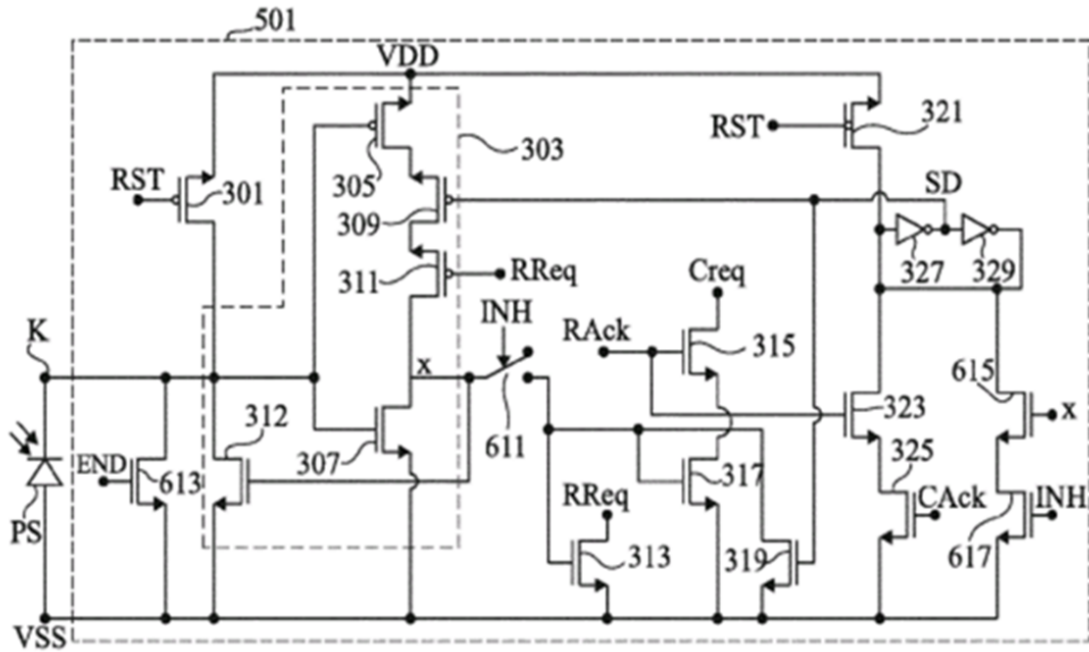


Figure 3-13: Architecture du pixel issu du brevet [Dehaene2016]

- Le comparateur

La photodiode du pixel, polarisée en inverse, est reliée à un comparateur qui a pour but de générer un événement digital lorsque la tension de la photodiode passe sous le seuil prédéfini. La génération d'événement doit également être rapide afin de limiter la consommation d'énergie. Différents types de comparateurs sont alors envisageables [Culurciello2003] : un simple inverseur, des inverseurs avec rétroaction capacitive ou un générateur d'événement avec rétroaction en courant. Un simple inverseur a pour avantage de prendre une surface réduite dans le pixel. Cependant la consommation d'un tel comparateur est trop élevée en zone de commutation. Les générateurs d'événements avec une rétroaction capacitive ne peuvent pas être envisagés pour une implémentation dans un pixel puisque la taille des capacités réduirait drastiquement le facteur de remplissage, Figure 3-14-a. Les générateurs d'événements avec rétroaction en courant ont l'avantage de réduire la consommation en ajoutant un miroir de courant sous un inverseur, Figure 3-14\_b. En effet, la rétroaction accélère la transition lorsque le courant du miroir du courant devient supérieur au courant photo-généré. Néanmoins, les miroirs de courant sont trop sujets au mismatch, ce qui pourrait créer des disparités trop importantes entre pixels. De plus les miroirs de courants sont réalisés avec des transistors MOS ayant un oxyde épais dit « GO2 » ayant des dimensionnements plus larges que des transistors (dit « GO1 ») en tailles minimales.

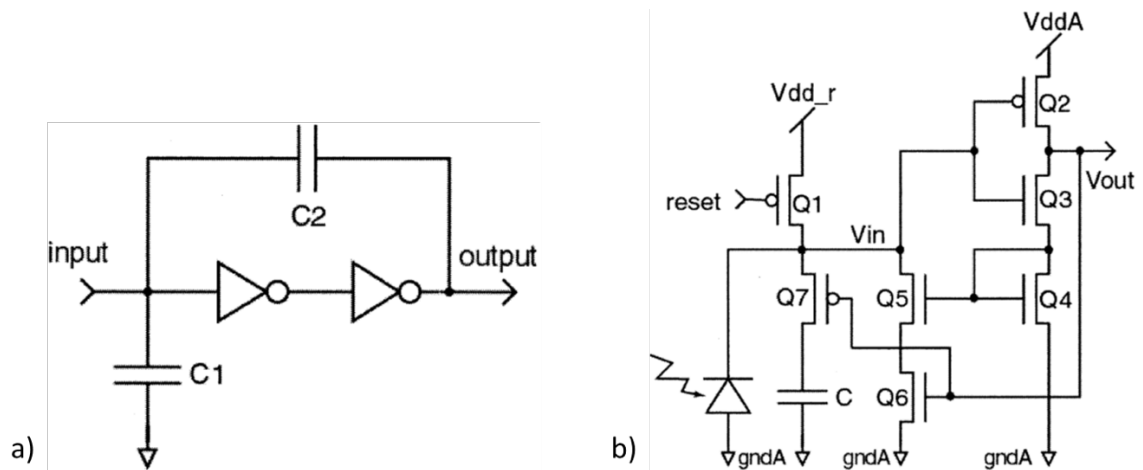


Figure 3-14: [Culurciello2003] Générateurs d'événements a) avec une rétroaction capacitive b) avec une rétroaction en courant

Le comparateur choisi a donc été un simple inverseur (bloc 303 sur la Figure 3-13) avec un transistor (312 sur la Figure 3-13) comme rétroaction. Ainsi, lorsque l'inverseur rentre en zone de conduction dynamique, la transition est brusquement accélérée. La tension du nœud photosensible décroît sous l'effet de l'incidence de la lumière sur la photodiode puis elle chute pour atteindre GND lorsque le seuil est dépassé, Figure 3-15.

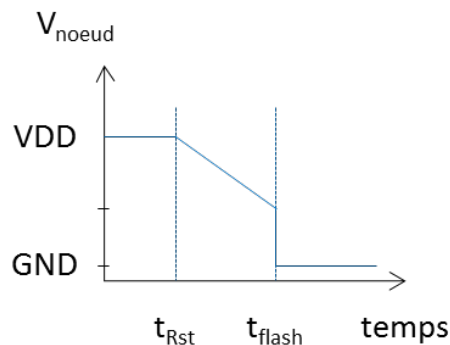


Figure 3-15: Evolution de la tension du nœud photosensible avec accélération brusque lors du passage du seuil grâce à une contre réaction sur le comparateur, afin de réduire la consommation dynamique du comparateur.

- Les transistors d'initialisation et de fin de trame

Le transistor d'initialisation de la photodiode (301 sur la Figure 3-13) dit transistor de reset est un PMOS en régime ohmique afin de préserver au maximum la dynamique. Il permet de polariser la photodiode (PS sur la Figure 3-13) à Vdd, pour ensuite laisser la tension du nœud photosensible décroître avec l'intensité lumineuse reçue.

Un autre transistor d'initialisation est implémenté (321 sur la Figure 3-13) afin d'initialiser tous les signaux digitaux ainsi que le point mémoire du pixel.

Le transistor de fin de trame (613 sur la Figure 3-13) permet de forcer les pixels trop peu éclairés à envoyer leurs adresses. Pour ce faire, le transistor NMOS est commandé par un signal nommé END, forçant la tension du nœud photosensible à GND.



- Handshake

Les signaux RAck et CAck (logique haute) ainsi que les signaux RReq et CReq (logique basse) sont partagés avec les lignes et les colonnes respectivement. Lorsqu'un événement est généré, le signal RReq (Row Request) passe à 0 grâce au transistor NMOS 313 sur la Figure 3-13, et une requête est envoyée au système de handshake. Lorsque le bus de communication est libre, le signal RAck (Row Acknowledge) provenant du système de handshake est reçu. Le transistor commandé par le signal RAck (315 sur la Figure 3-13) et le transistor commandé par la sortie du comparateur (317 sur la Figure 3-13) permet alors de générer le signal CReq en logique basse afin d'envoyer une requête colonne au système de handshake. Le signal CAck est alors reçu. Lorsque les deux signaux RAck et CAck (323 et 325 sur la Figure 3-13) sont actifs, le point mémoire (SD sur la Figure 3-13) est activé et le comparateur est éteint (309 sur la Figure 3-13). Par le biais du transistor 319 sur la Figure 3-13, le signal de sortie du comparateur est mis à 0 et le pixel est acquitté.

- Inhibition dans le pixel

Lorsqu'un signal RAck arrive pour la première fois dans le bloc, ce qui correspond au fait qu'un pixel a flashé et peut envoyer sa donnée, le signal d'inhibition est enclenché. L'interrupteur contrôlé par le signal INH (611 sur la Figure 3-13) permet d'empêcher les pixels de communiquer leurs données pendant le temps d'inhibition. De plus, si un événement est généré et que l'inhibition est activée, le point mémoire est mis à 1 par les transistors contrôlés par INH et X (617 et 615 sur la Figure 3-13) et le comparateur est éteint jusqu'à la prochaine initialisation.

### 3.1.3.3.2 Architecture extérieure au pixel

L'architecture extérieure au pixel est composée de deux mécanismes, Figure 3-16 : Le handshake (ou AER) qui gère de manière globale l'évacuation des données, et le mécanisme d'inhibition implémenté pour chaque bloc.

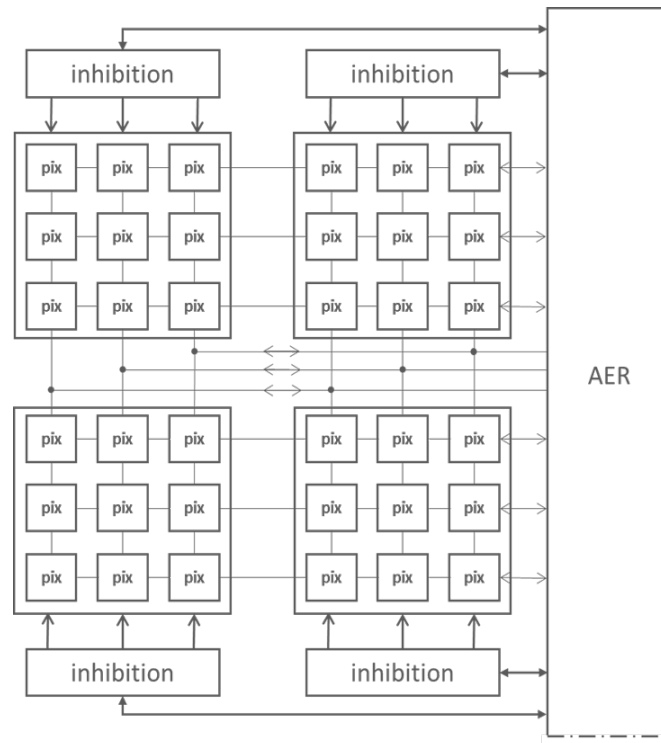


Figure 3-16: Architecture globale pour un capteur de 36 pixels regroupés en 4 blocs, avec les mécanismes d'inhibition pour chaque bloc et le système AER

- Mécanisme de handshake global

Le handshake gère la réception des signaux RReq et CReq ainsi que la génération des signaux RAck et CAck. Il gère également la réinitialisation des signaux de requête après avoir envoyé les signaux d'acquiescement.

Un arbre d'arbitrage permet de gérer quel requête doit être acquiescée en premier. Cette architecture est développée dans l'annexe.

- Mécanisme d'inhibition par bloc

L'inhibition doit être capable de générer un signal pendant un temps défini afin de stopper l'envoi des requêtes des autres pixels du bloc. Il faut également prévoir un système de mémorisation afin de ne lancer l'inhibition qu'une seule fois par trame et par bloc.

Dans l'implémentation effectuée pendant le stage, le système d'inhibition était sujet au mismatch des transistors. Le mécanisme tel qu'il était implémenté est décrit en annexe. Nous verrons dans le chapitre IV, une implémentation plus robuste aux variations de process.

#### 3.1.3.3.3 Résultats

Les simulations CADENCE ont été réalisées avec un capteur de 36 pixels séparés en 4 blocs de 9 pixels. Le temps d'acquisition par trame a été fixé à 20ms. La consommation mesurée est de 3nJ soit 0.15 $\mu$ W. Pour un capteur présumé de 128 par 128 pixels, la consommation peut être estimée à moins de 100 $\mu$ W, en admettant qu'un tiers de la consommation vient du système de lecture et que les deux tiers restant soient la consommation énergétique des pixels.

Cette architecture est comparée à d'autres architectures de type événementielles, Table 3-2. La consommation en termes de FOM (figure of merit) serait donc largement inférieure aux architectures antérieures pour un facteur de remplissage estimé plus important. Lorsque cette architecture est comparée à des capteurs d'acquisition en tension, le niveau de consommation est trop élevé. En effet, les consommations d'énergie pour de tels capteurs sont : [Choi2012] 15.4pJ/fr.pix, [Chung2012] 147pJ/fr.pix, [Bol2014] 17pJ/fr.pix. Néanmoins, une telle architecture permet de réduire le flux d'information à la source. Elle peut également être intégrée monolithiquement dans un system-on-chip et utiliser les ressources de calcul disponible afin de réaliser directement des traitements sur l'image.

	Power consumption	Pixel array	Frame rate (equivalent)	FOM	Pixel pitch (fill factor)
PFM-AER [Culurciello2003]	7.5mW	80x60	30fps	52nJ/fr.pix	31 $\mu$ m (14%)
ATIS [Posch2010]	50mW (static) 175mW (high activity)	304x240	30fps	32nJ/fr.pix	30 $\mu$ m (30%)
TTFS [Guo2007]	3.1mW	32x32	30fps	100nJ/fr.pix	17 $\mu$ m (1.9%)
IM-TTFS	Less than 100 $\mu$ W	128x128	50fps	<122pJ/fr.pix	~7 $\mu$ m (~40%)

Table 3-2: Comparaison des performances des différents capteurs. IM-TTFS correspond à l'architecture présentée ici.

### 3.2 Transmission de données lissée dans le temps et utilisation de bus multiples

Les capteurs événementiels utilisent traditionnellement un système d'AER classique avec un handshake et un arbre d'arbitration afin de gérer les collisions entraînant la perte d'information. Cependant ces systèmes sont très coûteux en termes de surface de silicium utilisée et ont une complexité système élevée. C'est pourquoi nous avons développé un mécanisme d'acquisition événementielle permettant de s'affranchir du handshake et de l'arbre d'arbitration afin d'optimiser la taille du capteur. Ce système s'appuie sur les avantages des technologies avancées : le nombre de métaux disponible et la densité. Habituellement pour des raisons d'efficacité électro-optique, tous les niveaux de métaux disponibles pour une technologie donnée ne sont pas utilisés. Néanmoins dans le cadre d'une application pour de la vision industrielle, la qualité de l'image n'est pas une priorité. C'est pourquoi nous avons choisi d'utiliser tous les niveaux de métaux (8 à 10 niveaux disponibles en FDSOI28) pour ce mécanisme. Cette architecture a fait l'objet d'un dépôt de brevet.

Ce mécanisme a pour but dans un premier temps de lisser l'histogramme de l'image dans le temps afin de réduire le phénomène de goulet d'étranglement. Dans un second temps, ce mécanisme a été pensé pour réduire le flot de données en réduisant les redondances spatiales dans les colonnes.

### 3.2.1 Principe de fonctionnement

Ce mécanisme repose sur le postulat que l'intensité lumineuse ne change pas au cours d'un temps trame qui est largement supérieur au temps de lecture d'une information. Les colonnes de la matrice sont indépendantes et ont chacune un ensemble de bus de sortie. Au pied de chaque colonne se situe un système de lecture, il est commun à tous les bus de la colonne. L'architecture est à temps trame fixe avec une lecture asynchrone et a une initialisation globale (global reset).

A la suite d'une phase d'initialisation, lorsqu'un pixel génère un événement au temps  $t_1$ , le pixel demande l'accès au premier bus de communication de sa colonne. Deux cas sont possibles. Soit le système de lecture est occupé, car il est en train de recevoir les informations d'un autre pixel, soit il est disponible. Si le système est occupé, alors le pixel annule sa demande d'accès et réinitialise sa photodiode. La photodiode intègre de nouveau. Le pixel générera un autre événement à  $2*t_1$  puisque l'intensité lumineuse reste la même sur un temps trame et le pixel demandera l'accès sur le second bus. Si le système de lecture est disponible, alors le pixel peut envoyer son information. Dans notre cas, le pixel envoie son adresse. Un conflit apparaît lorsque le dernier bus est requis et que le système n'est toujours pas disponible. Dans ce cas, le pixel ne réinitialise pas sa photodiode et n'annule pas sa requête, elle est placée en file d'attente jusqu'à ce que le système de lecture se libère. Avec ce système, l'erreur de quantification est réduite et limitée uniquement aux requêtes sur le dernier bus.

La Figure 3-17, décrit ce mécanisme d'acquisition pour une colonne. Lorsque deux pixels veulent communiquer au même moment, comme par exemple au temps  $T_1$  (Figure 3-17), un ordre de priorité a été décidé, la priorité allant au pixel au rang le plus faible. Ce qui signifie qu'au temps  $T_1$ , le pixel 1 a la priorité sur les pixels 2, 4 et 5. Le pixel 1 envoie donc son adresse. Lorsqu'elle est reçue par le système de lecture, elle est horodatée ce qui permet de connaître le temps entre la dernière phase globale d'initialisation et le temps de flash du pixel. Le numéro du bus sur lequel a été transmise l'adresse est également noté. Au temps  $T_2$ , les pixels 2, 3, 4 et 5 flashent. Puisque les pixels 2, 4 et 5 ont déjà tenté de communiquer leurs données, ils demandent l'accès au bus 2, alors que le pixel 3 qui flashe pour la première fois, ayant reçu moins d'intensité lumineuse, demande l'accès au bus 1. C'est le pixel 2 qui a la priorité, il envoie donc sa donnée sur le bus 2 et les pixels 3, 4 et 5 sont réinitialisés. En effet, puisqu'il n'y a qu'un seul système de lecture en pied de colonne, le pixel 3 ne peut pas transmettre sa donnée même s'il demande l'accès sur un bus différent que les autres pixels. Au temps  $T_3$ , les pixels 4 et 5 flashent pour la troisième fois, ils demandent donc l'accès sur le bus 3. Le pixel 4 a la priorité sur le pixel 5 et transmet donc sa donnée au système de lecture. Cependant, dans cet exemple, il n'y a que 3 bus. Le bus 3 étant le dernier bus, le pixel 5 ne se réinitialise pas et n'annule pas sa requête, il attend que le système de lecture soit libre afin de pouvoir transmettre sa donnée. Il y a donc un délai entre le temps de flash et le temps de lecture, ce qui induit une erreur dans la valeur de l'intensité mesurée pour le pixel 5. Pour retrouver l'intensité, l'horodatage est divisé par le numéro du bus sur lequel l'adresse du pixel a été transmise (3.3). En effet, le numéro du bus correspond au nombre de fois que le pixel a intégré l'intensité lumineuse reçue. Lorsque le pixel 3 flashe une seconde fois, sa donnée est évacuée sur le bus 2, il n'entre en conflit avec aucun autre pixel.

$$I = \frac{C(V_{dd} - V_{th})}{\frac{T_{flash} + T_{delai}}{N_{bus}}} \quad (3.3)$$

C est la capacité du nœud photosensible,  $(V_{dd}-V_{th})$  correspond à la plage de tension parcourue par la photodiode avant de générer un événement,  $T_{flash}$  est le temps auquel l'événement est généré et où la requête est envoyée, en considérant que  $T_{rst}=0$ .  $T_{flash}+T_{delai}$  correspond au temps auquel la donnée est réellement envoyée et  $N_{bus}$  est le numéro du bus sur lequel l'adresse a été envoyée.

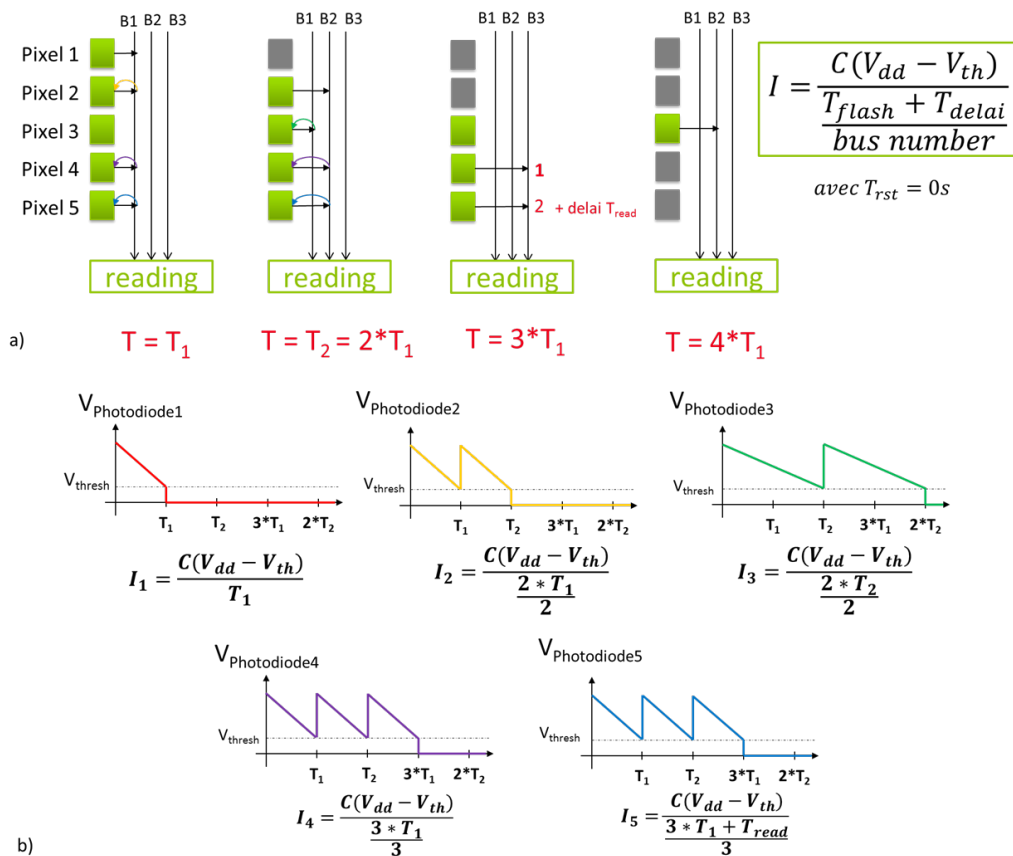


Figure 3-17: a) Chronogrammes décrivant le principe de transmission sur des bus multiples b) Décharges des différents noeuds photosensibles des pixels et leurs intensités correspondantes après acquisition.

Le fonctionnement niveau colonne est expliqué de manière synoptique, Figure 3-18.

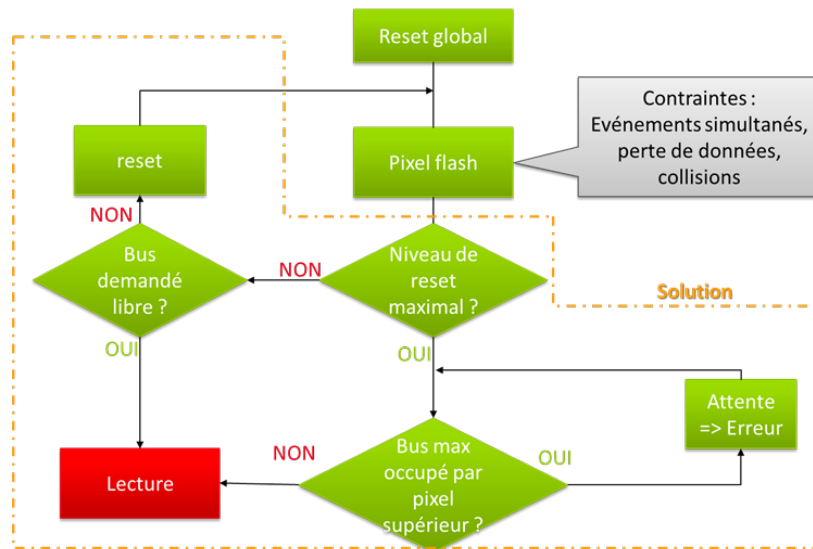


Figure 3-18: Schéma synoptique du mécanisme d'acquisition pour une colonne

### 3.2.2 Réduction de donnée par voisin direct

Ce mécanisme est complété par une méthode de réduction des redondances spatiales par colonne. Les pixels recevant la même information lumineuse et sous certaines conditions sont donc groupés.

La réduction est réalisée en groupant sous une même adresse, les pixels directement adjacents et qui flashent en même temps sur le même bus. Les pixels directement adjacents, dit aussi voisins directs, sont définis comme le pixel  $i$  et le pixel  $i+1$ . Par exemple, sur la Figure 3-19, les pixels 1 et 2 sont voisins directs mais 2 et 4 ne le sont pas. Cette définition s'étend également à plusieurs pixels adjacents, comme par exemple les pixels 4, 5 et 6 sur la Figure 3-19. Ainsi, si deux voisins directs génèrent un événement sur le même bus comme pixel 1 et 2, cela signifie qu'ils ont la même information lumineuse, alors seule l'adresse du pixel le plus au-dessus, ici le pixel 1, est transmise, le pixel 2 est éteint jusqu'à la prochaine phase globale d'initialisation afin d'économiser de l'énergie. Dans l'exemple de la Figure 3-19, les pixels 5 et 6 reçoivent la même intensité lumineuse que le pixel 4. Ils sont donc également éteints alors que le pixel 4 se réinitialise afin de recommencer une intégration puisqu'il ne peut pas communiquer son information pour le moment. Au temps  $T_2$ , le pixel 3 flashe en même temps que le pixel 4, ils sont voisins directs mais ne demandent pas l'accès sur le même bus. La priorité va au pixel au rang le plus faible. Le pixel 3 transmet donc son information au système de lecture alors que le pixel 4 se réinitialise et commence une nouvelle intégration. Le pixel 4 pourra envoyer son information au temps  $T_3$  sur le bus 3.

Comme pour l'IM-TTFS, à la fin du temps trame, certains pixels n'ont pas reçu suffisamment d'intensité lumineuse afin que la tension du nœud photosensible passe la valeur de seuil. Ils n'ont donc jamais flashé. Ainsi, ces pixels sont forcés à envoyer leur adresse et sont considérés comme des pixels de valeurs d'intensité 0. Au niveau de la mémoire, les pixels qui n'ont pas envoyé leurs informations, même à la fin du temps trame, correspondent aux pixels ayant été inhibés. Ils prennent la valeur de l'intensité du pixel au-dessus d'eux. Les pixels les plus sombres ont, eux, envoyés leurs

informations à la fin du temps trame afin qu'ils ne soient pas confondu avec des pixels inhibés.

Le temps de transmission de l'information par un bus ou plus communément appelé temps de lecture  $T_{read}$ , est un paramètre clef. En effet, plus ce temps de transmission est long plus le nombre de pixel générant un événement durant ce temps sera élevé. La réduction de données est d'autant plus effective que le temps de lecture du bus est long, c'est un temps d'inhibition implicite. Cependant si ce temps est trop long la résolution ainsi que la qualité de l'image seront moins élevés. En effet, si le temps de lecture est trop long, la file d'attente sur le dernier bus sera d'autant plus longue ce qui augmentera le nombre d'erreurs et réduira la qualité de l'image. La résolution est donc liée au temps de lecture et au temps trame. De plus, le temps de lecture est directement lié à la capacité du bus et à l'intensité nécessaire pour le charger, il influe donc sur la consommation d'énergie de la colonne.

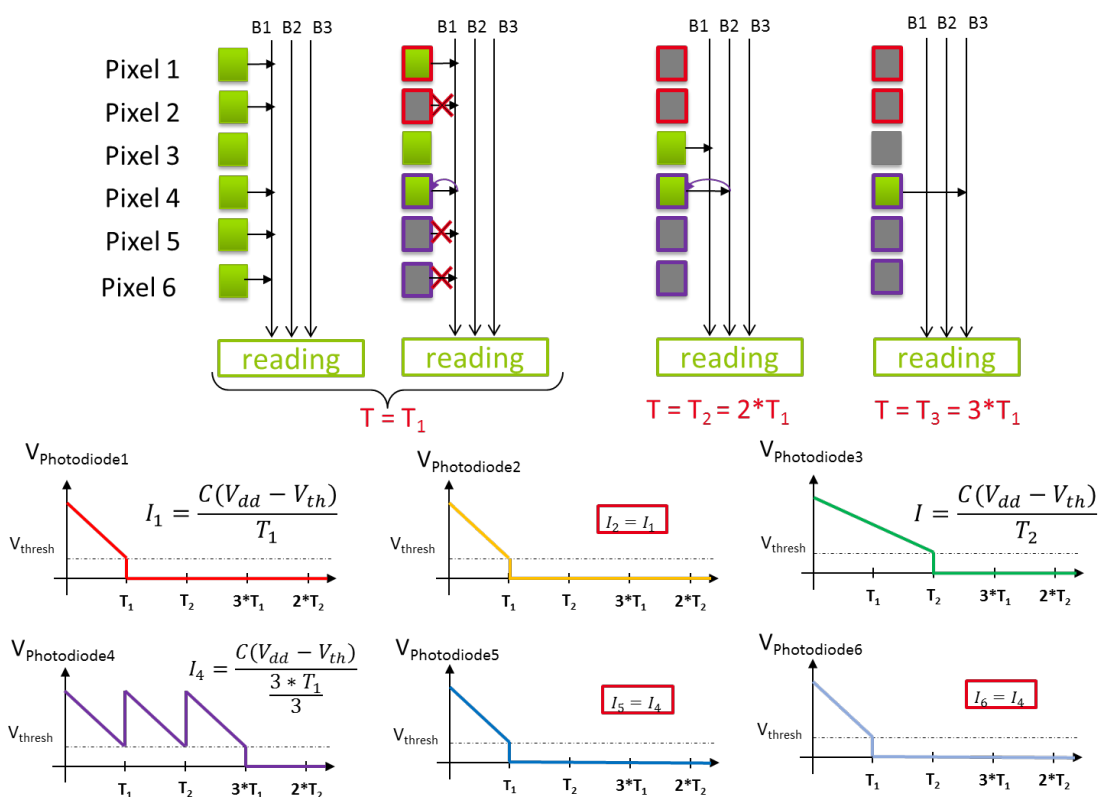


Figure 3-19: Chronogrammes décrivant le mécanisme de réduction de données par voisins directs et les graphes de tension en fonction du temps des nœuds photosensibles.

### 3.2.3 Résultats de simulation

Les simulations ont été effectuées dans les mêmes conditions que pour l'architecture IM-TTFS. Les résultats, Figure 3-20, montrent que plus il y a de bus implémentés plus la qualité de l'image augmente bien que le nombre de données envoyées reste constant : seulement 60%. Cependant, pour un SNR équivalent, il ne fallait que 23% de données envoyées pour l'architecture IM-TTFS. La réduction de donnée est donc beaucoup moins efficace pour cette architecture. De plus, des artefacts colonnes se

forment lorsque les données sont détériorées dû à une file d'attente sur le dernier bus colonne trop importante, comme pour 2 bus sur la Figure 3-20.

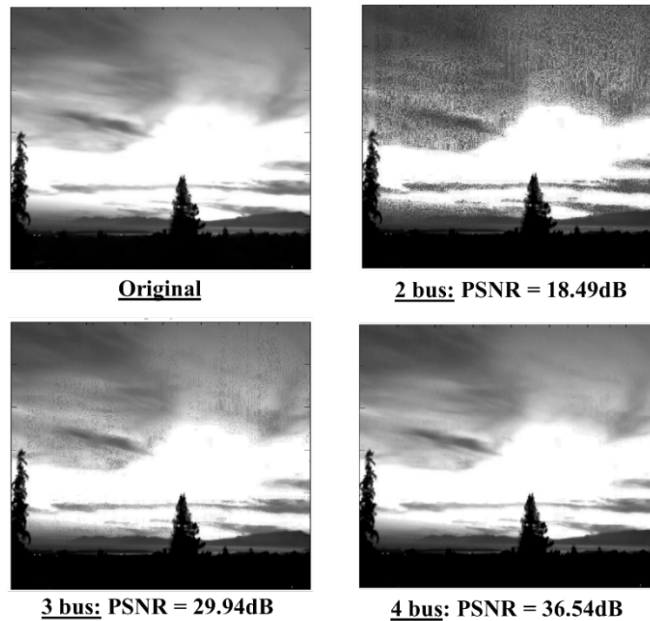


Figure 3-20: Résultats de simulation pour l'image Vine Sunset avec 2, 3 et 4 bus. L'image originale est en haut à gauche

### 3.2.4 Architecture développée

Les architectures des blocs développées ci-dessous n'ont pas fait l'objet d'une étude approfondie, de ce fait, seules des architectures fonctionnelles et non optimisées sont présentées ici.

#### 3.2.4.1 L'architecture globale du pixel

Bien que cette architecture ait un pixel assez complexe, son architecture globale reste simple et complètement digitale, mis à part la photodiode et un suiveur. Cette architecture fonctionne à temps de trame fixe, ce qui permet aussi d'implémenter facilement des traitements de l'image par la suite. Le pixel comprend une photodiode en mode intégration polarisée en inverse. Elle est reliée à un comparateur qui génère un événement, lorsque la tension du nœud photosensible passe sous le seuil prédéfini. Si le pixel ne peut accéder au bus de transmission, il se réinitialise, c'est le rôle du signal RST issu du bloc bus access, Figure 3-21. Le signal EXT1 issu du bloc « neighbor block » éteint le pixel jusqu'à la prochaine initialisation globale si le pixel est voisin direct d'un pixel supérieur à celui qui envoie sa requête sur le même bus que le pixel  $P_i$  au même moment. Le signal EXT2 permet d'éteindre le pixel lorsque celui-ci a transmis sa donnée. Il reste éteint jusqu'au prochain signal de reset. Les blocs seront décrits dans les parties suivantes.



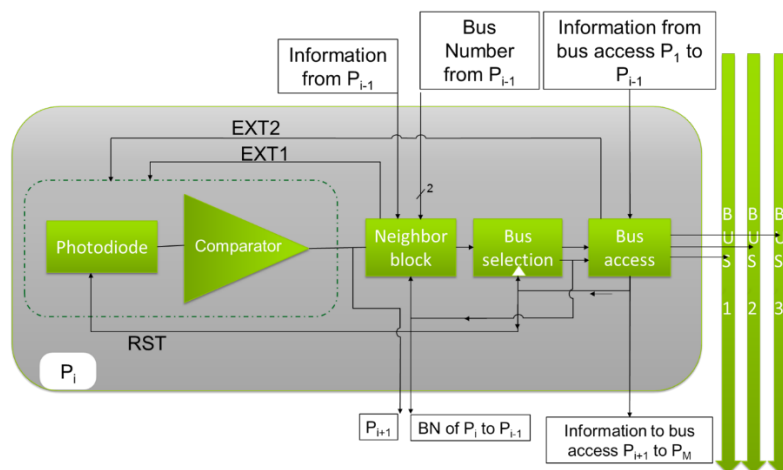


Figure 3-21: Schéma bloc fonctionnel du pixel  $P_i$ .

### 3.2.4.2 Le bloc d'accès au bus

Le bloc d'accès au bus, Figure 3-22, permet de savoir si le système de lecture est libre ou non et si le pixel peut donc envoyer sa donnée ou doit se réinitialiser. Si le système de lecture n'est pas occupé par un pixel du dessus ( $P_{i-1}$ ), le pixel peut donc envoyer la donnée sur le bus requis. Le bus est choisi grâce à un démultiplexeur qui prend comme entrée de sélection le numéro du bus codé sur  $n$  bits donné par le bloc « bus selection ». Le bloc « bus selection » s'incrémente à chaque fois qu'un événement se produit et il est réinitialisé par le signal de reset global.

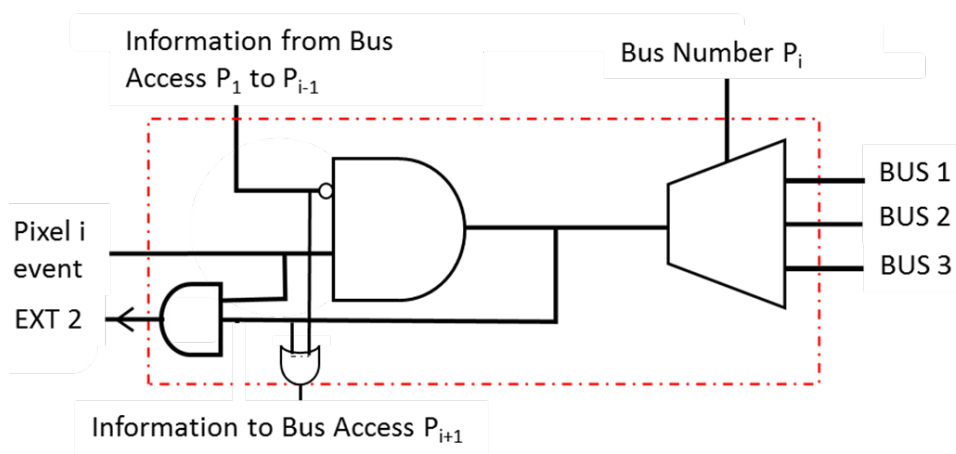


Figure 3-22: Schéma d'une architecture possible pour le bloc d'accès au bus.

### 3.2.4.3 Le bloc d'inhibition des voisins directs

Le bloc d'inhibition permet d'inhiber grâce au signal EXT1 le pixel. Cela se fait si deux événements de pixels voisins se produisent sur le même bus. Pour connaître si les informations sont transmises sur le même bus, les numéros du bus noté BN sont comparés par un opérateur XNOR, Figure 3-23 .

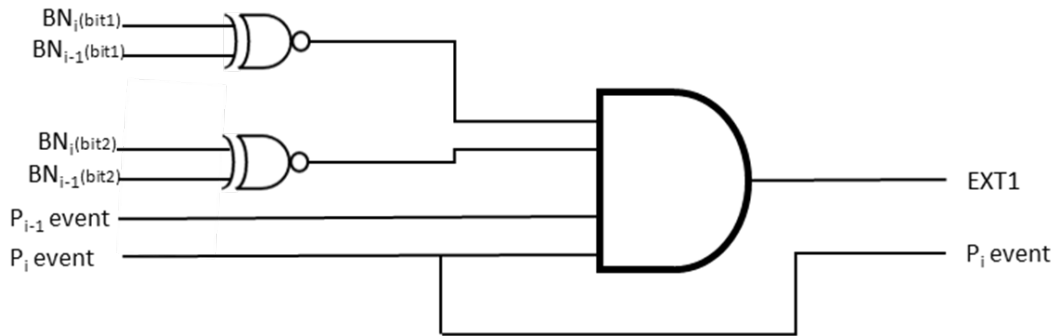


Figure 3-23: Schéma fonctionnel du bloc d'inhibition

### 3.2.4.4 Architecture globale

Mis à part la mémoire et le système de lecture en pied de colonne qui comprend un convertisseur temporel-numérique (TDC: time-to-digital convertor), il n'y a aucun autre bloc dans l'architecture. Cependant le pixel a une complexité assez élevée. En effet, si l'architecture devait être implémentée de cette manière sans optimisation il faudrait un peu moins de 80 transistors par pixels alors qu'il en faudrait un peu moins de 60 pour l'architecture du pixel ATIS [Posch2011] et un peu moins de 50 pour l'architecture [Darwish2015].

### 3.2.5 Paramètres ajustables dans l'architecture et simulations MATLAB

Plusieurs variantes ont été imaginées afin d'améliorer ce système de transmission à bus multiples. Elles sont récapitulées dans Figure 3-24.

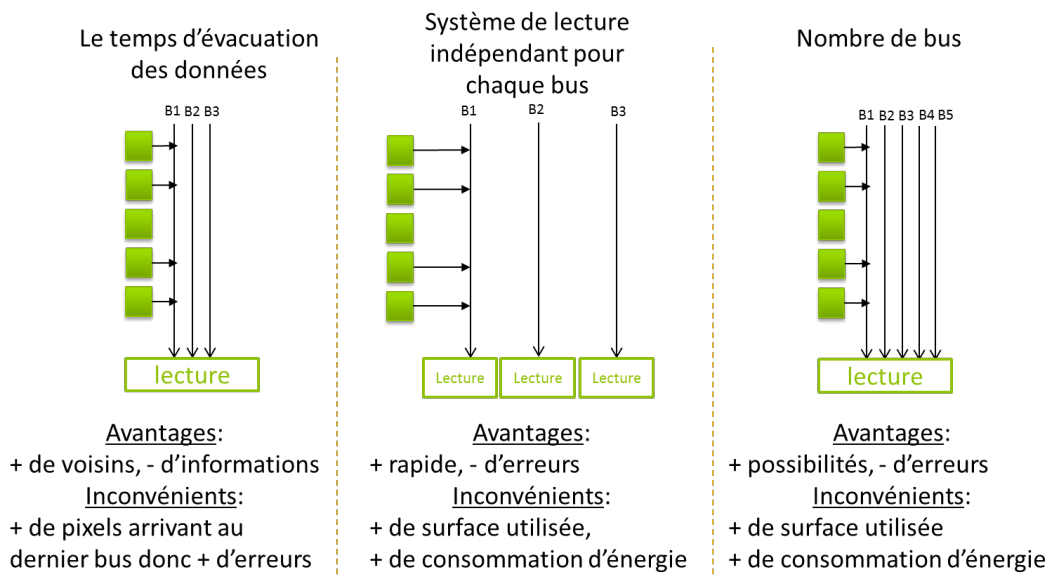


Figure 3-24: Récapitulatif des différentes variantes envisagées pour l'architecture avec leurs inconvénients et avantages.

### 3.2.5.1 Temps d'évacuation des données

Augmenter le temps d'évacuation des données, noté  $T_{read}$ , revient à augmenter le temps implicite d'inhibition et donc permet de générer plus de pixels voisins identiques. Cependant, cela générera aussi plus d'erreurs puisque les informations seront évacuées moins rapidement. Il y aura donc plus de pixels qui atteindront le dernier bus. Des tests ont été réalisés succinctement afin d'étudier ce phénomène avec 3 bus et un temps de lecture 10 fois inférieur. Et les hypothèses sont validées puisque les voisins (pixels bleu foncé sur les images Figure 3-25) sont plus nombreux pour un temps de lecture plus élevé (vérifié sur d'autres images également), 26% augmenté jusqu'à 28%. Et le nombre de pixels sur le bus 3 est également plus important, pour un temps d'inhibition plus long, 40% comparé à 29%. Cela augmente donc le nombre d'erreur et diminue de ce fait le PSNR mais le nombre de données envoyé est également réduit. Cela dépend donc de l'application visée pour cette architecture.

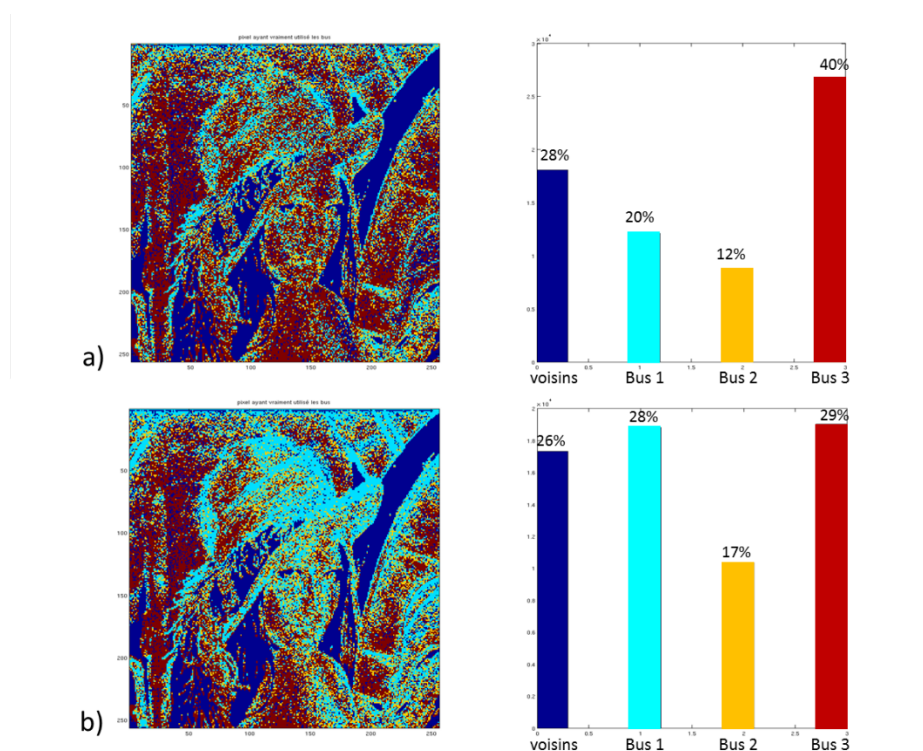


Figure 3-25: a) Cartographie de Lena avec un temps de lecture 10 fois supérieur à la cartographie b). Les histogrammes sont représentés en fonction des bus utilisés. Les pixels bleu foncé sont les pixels n'ayant pas communiqué leurs adresses nommés « voisins »

### 3.2.5.2 Systèmes de lecture

Il est également possible d'augmenter le nombre de systèmes de lecture en pied de colonne afin d'en mettre un sur chaque bus. De ce fait, plusieurs bus pourront transmettre des données en même temps et ainsi réduire la file d'attente du dernier bus. La Figure 3-26 permet de valider les hypothèses. En effet, le pourcentage de pixel ayant été lu sur le dernier bus est 40% lorsqu'il n'y a qu'un système de lecture par colonne. Ce pourcentage diminue jusqu'à 35% lorsque 3 systèmes de lectures sont

implémentés en pied de chaque colonne, un par bus. Cependant pour des raisons d'optimisation de surface, cette solution semble difficile à implémenter.

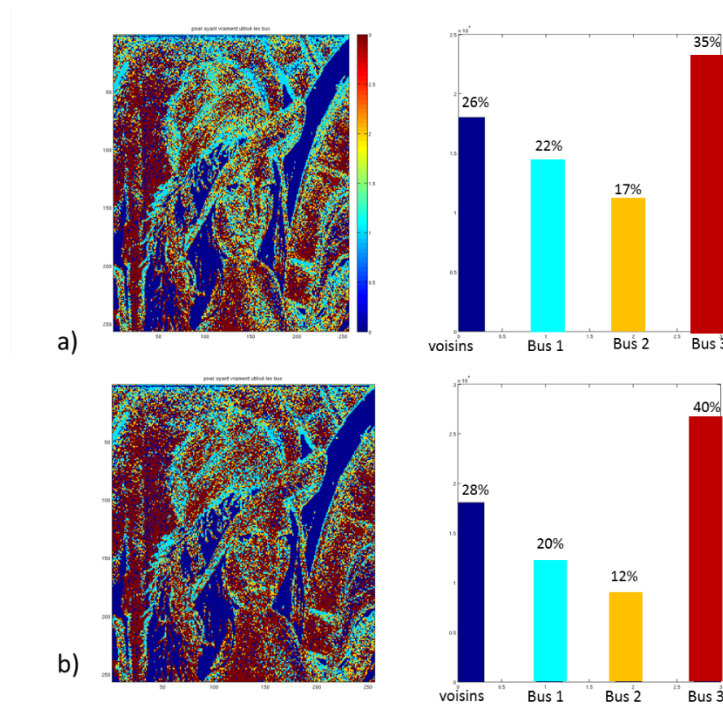


Figure 3-26: Cartographie de l'image Lena avec le nombre de pixel n'ayant pas transmis leurs données, et ceux ayant utilisés les bus 1, 2 ou 3. a) acquisition avec 3 systèmes de lecture par colonnes b) un système de lecture par colonne.

### 3.2.5.3 Nombres de bus disponibles par colonnes

Il est également possible d'augmenter le nombre de bus disponibles afin de désengorger la file d'attente sur le bus 3 et donc d'augmenter la qualité de l'image mesurée par le PSNR (dB). Figure 3-27 montre l'influence du nombre de bus disponible sur la qualité de l'image et sur la longueur maximale de la file d'attente sur le dernier bus. Pour 3 bus, le PSNR est proche de 30dB ce qui est suffisant comme qualité d'image. Les résultats sont meilleurs pour 4 bus mais il faut garder à l'esprit que l'espace de silicium doit être optimisé, il faut donc trouver les bons compromis entre chaque paramètre.

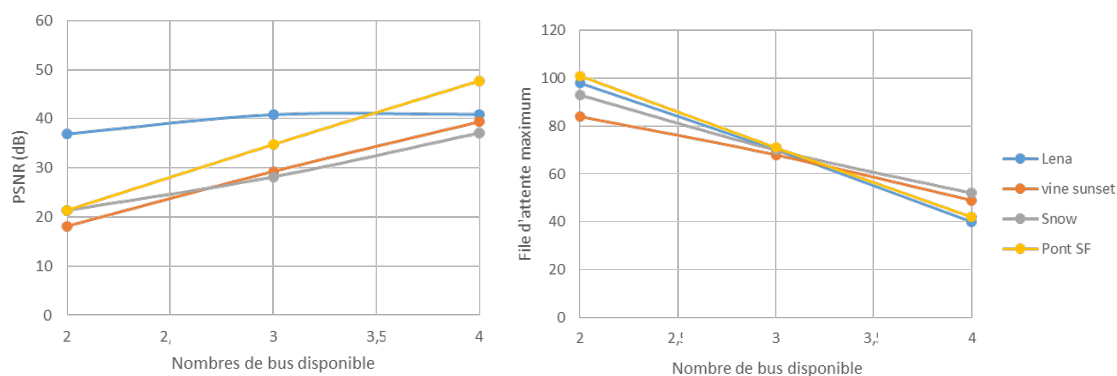


Figure 3-27: Influence du nombre de bus disponible sur a) la qualité de l'image (PSNR en dB) b) la longueur maximale de la file d'attente sur le dernier bus disponible pour différentes images.

### 3.3 Comparaison des deux architectures

Les deux architectures utilisent une acquisition time-to-first-spike (TTFS), la photodiode est polarisée en inverse en phase d'initialisation et la tension du nœud photosensible décroît avec l'intensité lumineuse reçue. Lorsque cette tension dépasse un seuil prédéfini, un événement est généré par un comparateur composé d'un inverseur et d'un transistor de contre réaction afin de diminuer la consommation d'énergie.

L'IM-TTFS est une architecture où les pixels sont regroupés par blocs au sein duquel les pixels ayant une information similaire au pixel le plus lumineux du pixel n'émettent qu'une seule adresse. Lorsqu'un événement apparaît des requêtes sont envoyées au système de handshake. Lors de la réception des signaux d'acquiescement, les pixels envoient leurs adresses au système de lecture. Puis ils s'éteignent jusqu'à la prochaine phase d'initialisation globale. Cette architecture a pour avantage de réduire considérablement le flot de données et de permettre des traitements d'images. La consommation pour un tel système est estimée à moins de 122pJ/fr.pix pour un capteur de 128x128 pixels à 50fps. Cette architecture nécessite un pixel assez dense (22 transistors) et de taille raisonnable avec un pas pixel inférieur à 10 $\mu$ m, ainsi que d'un système extérieur assez complexe avec un handshake fonctionnant avec un arbre d'arbitrage, un système de lecture et un système par bloc générant le signal d'inhibition.

L'architecture à transmission de données lissées dans le temps avec système de bus multiple, regroupe également les pixels de même luminosité mais par colonnes uniquement. De plus, cette acquisition permet de réduire le goulet d'étranglement qui se forme lors d'une acquisition événementielle avec un système d'address event representation, grâce au système de réinitialisation des pixels. En effet, les pixels se réinitialisant afin de refaire une intégration, la donnée n'est pas dégradée par un délai induit lors de l'évacuation des données. Néanmoins ce système ne semble pas optimal pour des traitements associés, du fait de la réduction de donnée en colonne et non par bloc. De plus la réduction de donnée est moins efficace pour cette architecture que pour l'IM-TTFS pour une même qualité d'image. Enfin, le pixel est très complexe (80 transistors sans optimisation) même si cela permet d'avoir un système global

fonctionnellement très simple. Aucune implémentation n'a été effectuée il n'est donc pas possible d'avoir de données sur la consommation d'énergie, la taille du pixel ou sur le facteur de remplissage.

### 3.4 Conclusion

Les problèmes de fuites de courant et d'intégrité des données induits par la technologie (paragraphe 2.5.2) orientent vers des solutions de conversion temporelles robustes dont plusieurs versions ont pu être présentées. Deux nouvelles architectures ont été étudiées ici : l'IM-TTFS et la transmission sur bus multiples avec un système de réinitialisation des pixels. Dans un premier temps, le fonctionnement de l'architecture IM-TTFS a été explicité. Le mécanisme d'inhibition permet de palier aux différents problèmes associés avec une architecture de type TTFS. Dans un second temps, le fonctionnement de l'architecture dite « multi-bus » a été expliqué, cette architecture tire profit de la densité qu'offre la technologie FDSOI 28nm. Puis le mécanisme de réduction de données ainsi que l'impact des différents paramètres ajustables de l'architecture ont été étudiés. Les deux architectures présentent une acquisition compatible avec une implémentation en technologie avancée telle que le FDSOI 28nm et des possibilités de réduction de données intégrées dans le système d'acquisition. Cependant, l'architecture IM-TTFS paraît plus pertinente pour la suite des travaux de la thèse puisque les traitements envisageables semblent plus cohérents. L'architecture IM-TTFS sera donc exploitée afin de construire des algorithmes de traitement (paragraphe 0).

## 4 Chapitre IV : Co-design acquisition et traitement

Les chapitres précédents ont souligné les avantages d'une acquisition dans le domaine temporel, permettant de maximiser le SNR tout en minimisant le besoin énergétique. De plus dans le contexte de l'internet des objets, le développement d'un SoC hétérogène est pertinent afin de pouvoir répondre à plusieurs applications et plusieurs besoin énergétique. Ainsi la technologie FDSOI 28nm permet d'avoir accès à des ressources de calcul digital performante et faible consommation. C'est pourquoi dans ce chapitre, nous allons chercher à exploiter la technologie FDSOI 28nm afin d'intégrer des traitements proche pixel. L'objectif est de réduire les contraintes le traitement externe des images. Cela sera effectué en extrayant des métadonnées de l'image comme par exemples des descripteurs.

Dans un premier temps, l'architecture IM-TTFS sera donc reprise et modifiée afin de simplifier la lecture des données et d'intégrer des traitements d'images. L'image extraite sera sous forme de carte binaire représentant les contrastes de l'image avec un seuil adaptatif par bloc. Après avoir simulé les traitements réalisables avec la carte binaire, à savoir de la détection de contours, de la détection de mouvement ainsi que de la classification, l'architecture du pixel a été étudiée. Le pixel comporte principalement une photodiode, un comparateur et un point mémoire. Le point mémoire est contrôlé par un signal d'inhibition généré par un système commun à un bloc de pixel de taille 3 par 3. Il est à noter qu'afin que l'architecture soit implantable en 3D séquentiel CoolCube, seul des transistors GO1 seront utilisés même si les transistors GO2 présentent moins de fuites de courant de grille. Les fuites de courant seront donc limitées grâce à la technique de polarisation par le substrat (back ou body biasing) [Pelloux-Prayer2015]. Enfin, les caractéristiques de l'architecture seront comparées à celles de l'état de l'art.

### 4.1 Descripteurs : principe et applications

L'IM-TTFS (time to first spike – inhibition mechanism) est une architecture qui a été développée afin de réduire le goulet d'étranglement qui se forme lors de l'évacuation des données. Cette réduction de données est particulièrement efficace pour les données des pixels les plus lumineux, du fait de la répartition des temps de flash dans le domaine temporel. Pour rappel, le temps de flash correspond au moment où la tension du nœud photosensible passe un seuil prédéfini, l'adresse du pixel est horodatée avec cette valeur temporelle qui permet de retrouver l'intensité lumineuse incidente du pixel. Dans cette architecture, pour chaque bloc défini, les pixels recevant une intensité similaire au pixel le plus éclairé du bloc considéré sont regroupés sous une seule adresse. Cela permet de réduire le nombre de données transmises tout en gardant une qualité d'image acceptable (supérieure à 30dB). Cependant, un des inconvénients de l'architecture IM-TTFS est l'importante surface occupée par tout le matériel nécessaire à la transmission des données, le système AER dans le pixel et en bordure de la matrice également. Ces conclusions ont été faites à partir de simulations

MATLAB prenant pour données d'entrées des images quantifiées sur 8bits. La limite de ces résultats réside dans la nature de ces données. En effet, le processus de quantification peut au final associer un code similaire à deux valeurs de signaux ayant une différence d'intensité plus faible que l'erreur de quantification intrinsèque au quantificateur utilisé. Il est donc possible de faire l'hypothèse que dans le cadre du capteur TTFS, les requêtes des pixels n'arrivent pas toutes en même temps mais les unes après les autres. Cela signifie qu'avec un système de lecture rapide, le goulet d'étranglement est limité. Cependant, une vitesse de lecture rapide n'est pas compatible avec des applications basse consommation d'énergie. De plus pour des applications de type internet des objets et un capteur dit *always-on*, les informations contenues dans l'image devront être extraites au plus tôt dans la chaîne de lecture. C'est pourquoi, dans ce chapitre, le capteur développé prétraite les images et ne sort qu'une image binaire. La méthode utilisée pour la binarisation s'appuie sur le système d'inhibition et permet d'utiliser un système de lecture classique *global shutter* pour sortir les informations binaires des pixels.

#### 4.1.1 Carte binaire

##### 4.1.1.1 Réalisation de la carte binaire

L'image sortie par le capteur est nommée « carte binaire ».

Pour rappel, un pixel « flashe » lorsque la tension du nœud photosensible passe un seuil prédéfini sous l'effet de l'intensité lumineuse reçue. L'inhibition est le fait de considérer un temps après le flash d'un premier pixel dans un bloc et de bloquer les requêtes sortantes des autres pixels pendant ce temps d'inhibition. Juste après cette inhibition, une carte binaire du bloc se forme, Figure 4-1 : les pixels ayant flashé durant le temps d'inhibition (1) et les pixels n'ayant pas flashé (0).

Il est à noter que cette carte binaire pourrait également s'effectuer dans un bloc de traitement externe à un capteur classique. Cependant implémenter un capteur sortant uniquement une carte binaire permet de baisser la consommation d'énergie, puisqu'il n'y a plus besoin d'ADC. De plus la consommation peut-être d'autant plus limitée que lorsque le temps d'inhibition est terminé tous les pixels du bloc peuvent être éteint qu'importe s'ils ont été inhibés ou non. Dans l'architecture IM-TTFS, le temps d'intégration maximal était fixé par le temps de trame, ici le temps d'intégration est implicitement fixé par bloc, par le temps de flash du pixel le plus lumineux du bloc plus le temps d'inhibition.



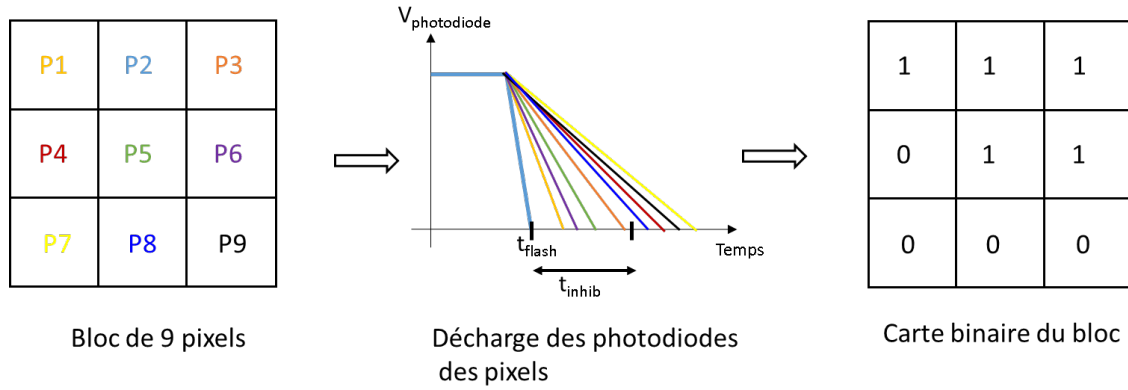


Figure 4-1: Mécanisme de formation de carte binaire pour un bloc d'un capteur événementiel avec inhibition,  $P_2$  est le pixel qui flashe en premier, pendant le temps d'inhibition le graphe nous indique que les pixels  $P_1$ ,  $P_6$ ,  $P_5$ , et  $P_3$  flashent, ils sont donc inhibés, ce qui donne la carte binaire du bloc à droite.

De façon plus générale, l'image binaire peut être formée bloc par bloc en fonction du maximum de ce bloc et d'un intervalle, ou seuil. Le seuil est défini en amplitude dans une image classique, et en temps dans un imageur événementiel. Pour un bloc, tous les pixels dont la valeur est comprise dans l'intervalle entre la valeur de l'intensité maximale et cette même valeur moins un seuil, sont étiquetés avec un 1, (4-1) et (4-2). Pour une image quantifiée classiquement à l'issue d'une intégration de temps défini, le seuil est fixe. Mais si on considère une image dans le domaine temporel, une même différence de luminosité dans les fortes luminosités et dans les basses luminosités n'ont pas la même différence de temps de flash (voir paragraphe 3.1.2). Dans le domaine temporel, les valeurs d'une image sont moins espacées pour les fortes intensités lumineuses et plus espacées pour les faibles intensités lumineuses. C'est pourquoi le seuil d'inhibition est croissant, afin de séparer le même nombre de valeurs pour une image quantifiée sur 8bits, il est court pour les fortes valeurs d'intensités lumineuses reçues et plus long pour les faibles valeurs.

$$v_{\max} = \max_{\text{bloc}}(v_{\text{pix}}) \tag{4-1}$$

$$\begin{cases} \text{si } v_{\max} \leq v_{\text{pix}} \leq v_{\max} - \text{seuil}, & \text{pix} = 1 \\ \text{sinon} & \text{pix} = 0 \end{cases} \tag{4-2}$$

#### 4.1.1.2 Traitement direct des données de la carte binaire

Il est possible de traiter directement la carte binaire afin d'effectuer un premier niveau de réveil du capteur. Ces traitements ne doivent pas, par conséquent, demander des moyens de calcul trop importants afin de limiter la consommation d'énergie. En effet, ce premier réveil serait développé dans le but d'avoir un capteur « always-on » et qui une fois le réveil activé libérerait davantage de ressources de calcul. Différents traitements seront proposés dans les paragraphes suivants selon un degré de complexité croissant. Tout d'abord, la carte binaire extraite permet d'avoir directement les contours de la scène. De la détection de mouvement peut être réalisée également sur le plan focal en rajoutant quelques fonctions digitales et un point mémoire. Enfin des traitements plus complexes peuvent être réalisés en utilisant le

CPU on-demand du SoC. L'emplacement de la réalisation de ces traitements dans le SoC est illustré dans la Figure 4-2.

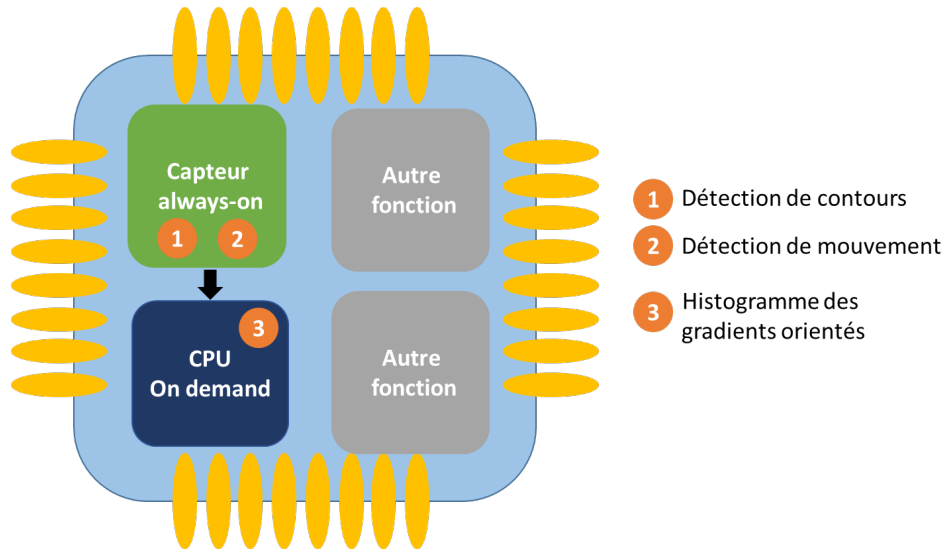


Figure 4-2: System on chip complet et emplacement de la réalisation des traitements d'images

#### 4.1.1.2.1 Détection de mouvement

Le premier traitement de bas niveau sur cette carte binaire a été de la détection de mouvement. La détection est réalisée sous MATLAB en comparant la carte binaire  $N$  à la carte  $N+1$ . Cependant, la carte binaire contient quelques pixels qui clignotent et faussent les résultats. Il est possible de réduire le nombre de ces clignotements en prenant la valeur médiane des pixels sur plusieurs trames. La détection de mouvement par la différence trames à trames de la carte binaire, Figure 4-4 est moins efficace que de la détection par différence de deux trames d'images d'origine, Figure 4-5. En effet, la Table 4-1 montre que le taux de détection est trop faible et le taux de fausse alarme trop élevé. Le taux de détection, 4-3), correspond aux pixels correctement détectés sur le nombre de pixels qui auraient dû être détectés soit les vrais positifs et les faux négatifs (TP : true positives et FN : false negative) [Lazaveric2006], Figure 4-3. Et le taux de fausse alarme, 4-4), correspond aux pixels ayant été détectés alors qu'ils n'auraient pas dû, c'est à dire le taux de pixels détectés en mouvement alors qu'ils appartiennent à l'arrière-plan statique. C'est donc le ratio des faux positifs (FP : false positives) sur le nombre de pixels détectés soit les vrais positifs et les faux positifs. Ce sont deux indicateurs qui varient de 0 à 1. Pour attester d'une bonne détection, le taux de détection doit être proche de 1 alors que le taux de fausse alarme doit être proche de 0.

$$d_r = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad 4-3)$$

$$f_a = \frac{N_{FP}}{N_{TP} + N_{FP}} \quad 4-4)$$

	Taux de détection	Taux de fausse alarme
Différence entre cartes binaires	0.2956	0.9666
Différence entre images 8bits	0.3081	0.8625

Table 4-1 : Résultats des simulations MATLAB pour de la détection de mouvement grâce à de la différence trame à trame sur la carte binaire en comparaison avec la différence trame à trame directement sur les images d'origine

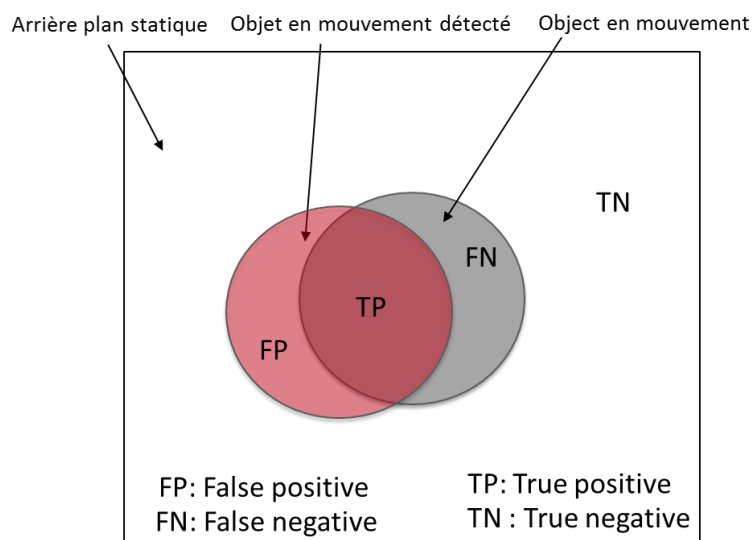


Figure 4-3: Illustration des éléments de calculs pour les taux de détection et de fausse alarme

Le taux de fausse alarme est élevé dans les deux cas. Cependant dans la carte de vérité utilisée, aucun contour de l'arrière-plan n'apparaît alors qu'ils font partie des principaux pixels qui clignotent dans les différences trames à trames.

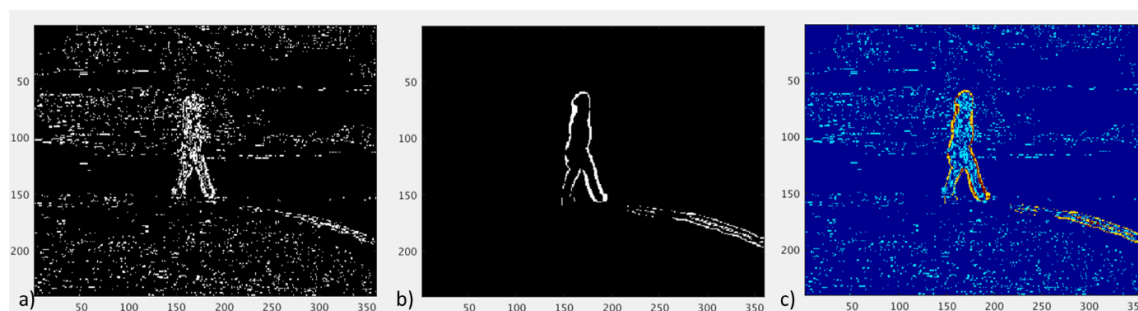


Figure 4-4 : a) différences de de 2 cartes binaires successives b) différences de 2 trames de la carte de vérité (Groundtruth) c) superposition des images a et b. Les pixels bleus cyan correspondent aux pixels blancs appartenant uniquement à la carte a). Les pixels jaunes correspondent aux pixels blancs appartenant uniquement à la carte b). Et les pixels rouges correspondent aux pixels blancs appartenant aux deux cartes. Les pixels bleus foncés sont les pixels noirs des cartes.

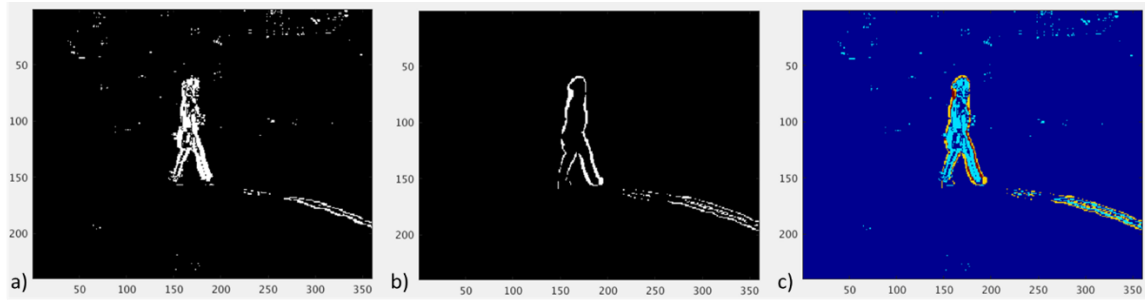


Figure 4-5: a) différences de 2 images successives et binarisation en fonction d'un seuil b) différences de 2 trames de la carte de vérité (Groundtruth) c) superposition des images a et b. Les pixels bleus cyan correspondent aux pixels blancs appartenant uniquement à la carte a). Les pixels jaunes correspondent aux pixels blancs appartenant uniquement à la carte b). Et les pixels rouges correspondent aux pixels blancs appartenant aux deux cartes. Les pixels bleus foncés sont les pixels noirs des cartes.

Une autre simulation a été effectuée en prenant comme carte de vérité de mouvement la différence entre deux images successives sur 8bits. Le taux de détection est maintenant de 0.4103 et le taux de fausse alarme de 0.8709. Les résultats sont montrés sur la Figure 4-6 .

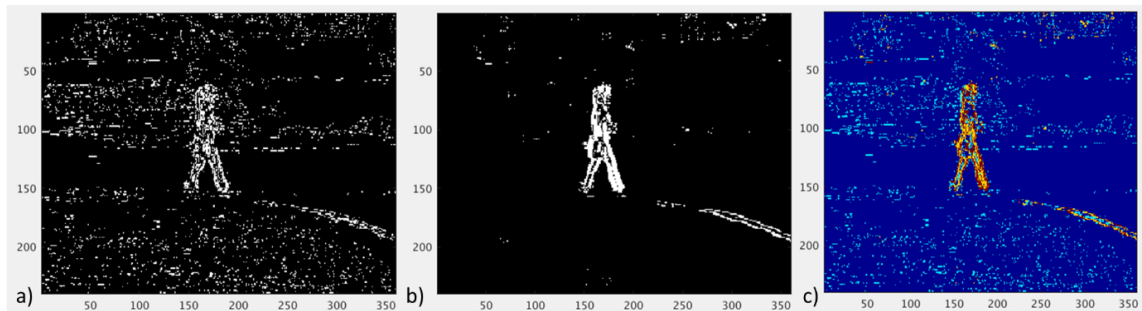


Figure 4-6: a) différence de 2 cartes binaires successives b) différences de 2 images successives et binarisation en fonction d'un seuil c) superposition des images a et b. Les pixels bleus cyan correspondent aux pixels blancs appartenant uniquement à la carte a). Les pixels jaunes correspondent aux pixels blancs appartenant uniquement à la carte b). Et les pixels rouges correspondent aux pixels blancs appartenant aux deux cartes. Les pixels bleus foncés sont les pixels noirs des cartes.

Une des trames de la séquence vidéo utilisée pour les tests est montrée Figure 4-7:



Figure 4-7: trame originale de la séquence vidéo utilisée.

## 4.1.1.2.2 Extraction de contour

La carte binaire telle qu'elle est construite permet d'avoir les informations de contours. L'avantage de cette carte est qu'elle est invariable aux changements lumineux puisque la décision est prise à l'intérieur du bloc, donc fonction de la luminosité locale. Ainsi un changement global de luminosité n'aura que peu d'effet sur la carte binaire. La carte a donc été superposée à une extraction de contour de Canny afin de comparer les résultats, Figure 4-8. [Canny1986]. Cependant l'extraction de contour de Canny ne s'implémente pas facilement contrairement à la carte binaire qui est l'image de sortie du capteur.

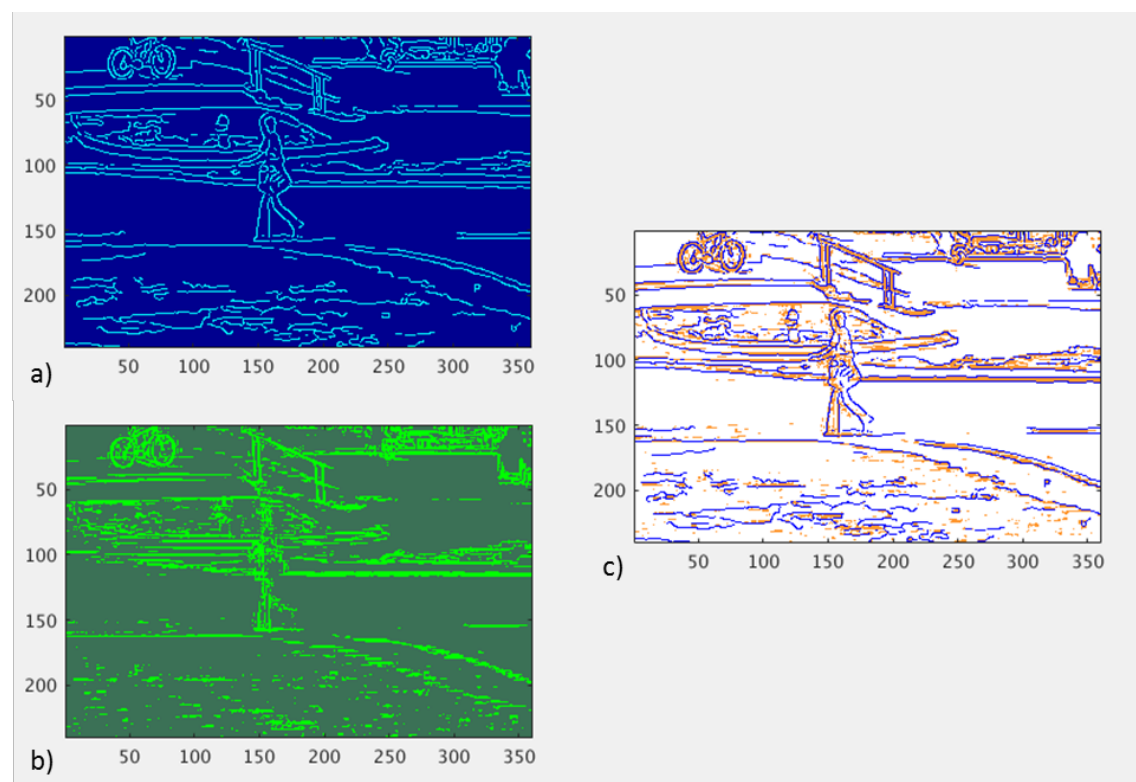


Figure 4-8: Résultats des simulations MATLAB sur l'extraction de contour. a) Extraction de contour de Canny sur l'image de 8 bits. b) Carte binaire c) Superposition des deux: en bleu les contours de Canny et en orange la carte binaire

## 4.1.2 Histogramme des gradients orientés : HoG

Après avoir créé la carte binaire, il est possible de l'utiliser afin de faire des traitements d'image plus complexes. Une implémentation monolithique en technologie avancée permet d'avoir accès à des ressources de calcul tels qu'un CPU, sans avoir besoin de transmettre les données via une connexion sans fil. Les traitements seront donc réalisés dans l'unité de traitement à la demande. Puisque la carte binaire permet d'avoir les contours, elle permet aussi d'avoir accès aux gradients orientés. C'est pourquoi l'étude s'est axée sur la méthode des histogrammes des gradients orientés.

#### 4.1.2.1 Description du principe des histogrammes des gradients orientés

L'histogramme des gradients orientés a été introduit par l'INRIA en 2005 pour la reconnaissance de personnes. La chaîne de traitement de l'image est décrite par la Figure 4-9.



Figure 4-9: chaîne de traitement du HOG [Dalal05]

Après avoir normalisé le gamma et la couleur sur toutes les images, les gradients sont calculés grâce au masque centré  $[-1 \ 0 \ 1]$ , qui après comparaisons [Dalal05] avec d'autres méthodes (masque non centré, taille différente) s'avère être le masque qui présente les meilleurs résultats de classification. Chaque pixel vote pour une orientation de contour avec un poids associé à ce vote en fonction de l'orientation de son gradient ainsi que de son amplitude. Pour une zone donnée appelée « cellule », l'histogramme des orientations est donc créé à partir de la sommation de ces votes sur la zone indiquée. Ces histogrammes sont normalisés par rapport à ceux de groupement de cellules, appelés « blocs ». Cela rend le descripteur plus robuste aux changements d'illumination. Cette normalisation se fait sur des blocs glissants. Ainsi chaque histogramme de cellule contribue sur plusieurs morceaux du descripteur final. Ce descripteur final est le vecteur de tous les composants des histogrammes normalisés par blocs glissants de la fenêtre de détection considérée. C'est donc ce vecteur qui est envoyé dans un algorithme de classification (ex. SVM (Support Vector Machine) linéaire, partie 4.1.2.2.3) afin d'effectuer la classification. La Figure 4-10 montre les différentes définitions des termes employés.

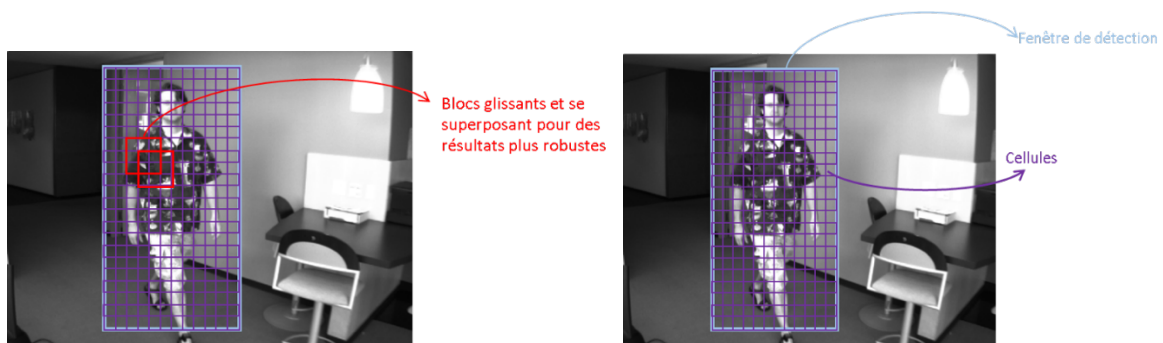


Figure 4-10: Définition des termes fenêtré de détection, cellules et blocs glissants

Pour résumer : pour chaque cellule, ou ensemble de pixels, il y a un histogramme d'orientation des gradients réalisé, ces histogrammes sont normalisés et regroupés par blocs. Ce sont donc ces histogrammes finaux qui constituent le vecteur du descripteur envoyé au SVM linéaire. Une implémentation basse consommation est décrite dans [Suleiman2016]



## 4.1.2.2 Exploitation de la carte binaire pour réaliser les HoG

Dans un premier temps, les traitements effectués à partir de la carte binaire ont été de la classification avec des histogrammes de gradient. Le mécanisme est décrit et comparé à celui d'un HoG classique dans la Figure 4-11. Les blocs changeants sont : la formation des histogrammes d'orientation ainsi que, la normalisation sur les blocs glissants. La carte binaire extraite grâce au capteur développé permet d'avoir accès à l'intensité et à l'orientation majoritaire du gradient. Pour cela, des masques angulaires sont utilisés. Ils permettent de savoir combien de pixels par blocs sont inhibés par orientations. Les orientations sont dénommées Nord, Sud, Est, Ouest et Nord-Est, Nord-Ouest, Sud-Ouest et Sud-Est (respectivement, N, S, E, O, NO, NE, SO, SE). En fonction du poids de ces orientations, l'orientation majoritaire ainsi que l'intensité du gradient sont définis, Figure 4-12. Le descripteur est défini comme un vecteur contenant les histogrammes des gradients orienté par macro-blocs comme expliqué dans la partie ci-dessous. Ces vecteurs sont ensuite traités par un SVM pour effectuer une classification.

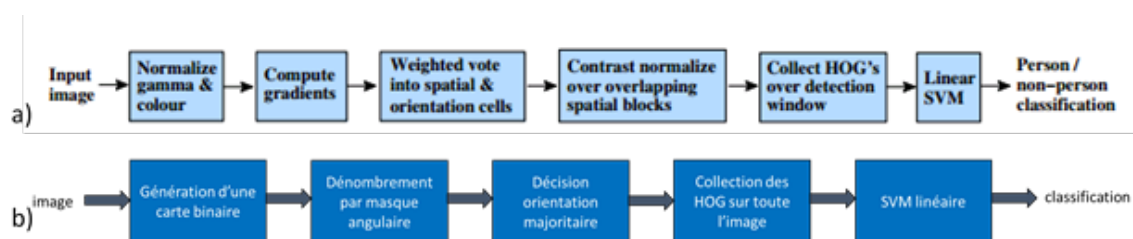


Figure 4-11: Mécanismes des HoG a) HoG classique b) HoG issu de la carte binaire extraite



Figure 4-12: de gauche à droite: image originale, image binaire, orientations et intensités des gradients calculés grâce à la carte binaire

## 4.1.2.2.1 Dénombrement par masque angulaire

Le gradient prépondérant, en orientation et en intensité, peut être déterminé à partir de l'image binaire générée. Précédemment, une carte binaire avait été créée pour chaque bloc. Ici, grâce à des masques angulaires, il est possible de déterminer l'intensité du gradient dans chaque orientation considérée. Cette opération consiste donc en une sommation des valeurs binaire par zones, tel qu'illustré sur la Figure 4-13-a et Figure 4-13-b. Il est possible de pondérer les valeurs binaires incluses dans chaque zone en fonction de leur localisation, tel qu'illustré en Figure 4-13-c et Figure 4-13-d. Cela peut-être fait, par exemple, en fonction du nombre de fois qu'un pixel sera considéré dans les différents calculs de l'intensité des orientations, comme c'est le cas

pour la technique appelée Soft-LBP décrite plus loin dans le chapitre. Cela permet une normalisation préétablie des futurs histogrammes. Comme montré sur la Figure 4-13, les masques peuvent être de tailles variées 3x3 ou 5x5 et il peut y avoir 4 ou 8 orientations considérées.

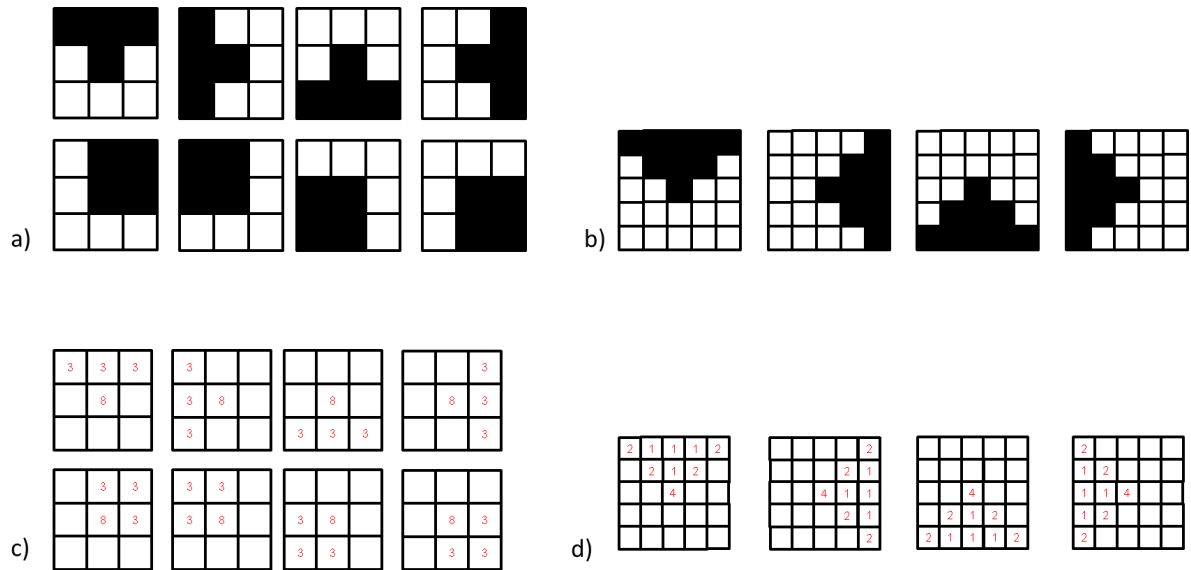


Figure 4-13: a) masques angulaires taille 3x3 pixels 8 orientations (de gauche à droite haut en bas : N, O, S, E, NE, NO, SO, SE) b) masques angulaires tailles 5x5 pixels 4 orientations c) et d) masques pondérés des masques présentés respectivement en a) et b). (Les coefficients appliqués sont les 1/x des chiffres affichés, x étant les chiffres affichés)

#### 4.1.2.2.2 Intensité et orientation majoritaire

Après avoir attribué à chaque orientation de gradient son intensité pour chaque bloc, il faut décider de l'orientation majoritaire du gradient. Une simple opération « max » ne peut suffire puisque des conflits peuvent apparaître comme sur la Figure 4-14.

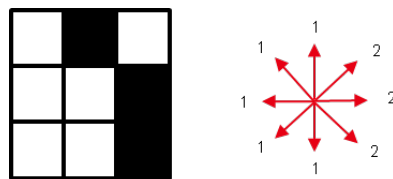


Figure 4-14: Exemple d'un conflit sur l'orientation majoritaire

Pour éviter ces conflits, une convolution avec un signal triangulaire de taille « nombre de filtre/2 + 1 » peut être effectuée avec le vecteur représentatif des intensités en fonction des masques angulaires adjacents, et ainsi l'orientation majoritaire du gradient est décidée à partir de ce type de filtrage, comme vu sur la Figure 4-15.



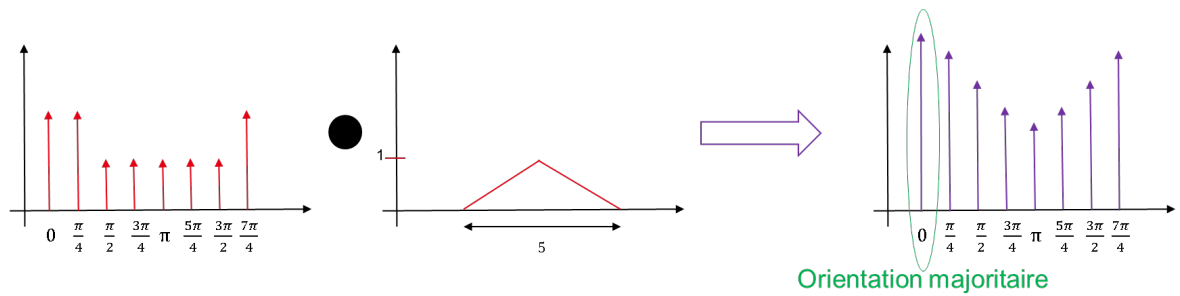


Figure 4-15: Convolution avec un signal triangulaire pour déterminer l'orientation majoritaire

#### 4.1.2.2.3 Classification

- Construction d'histogramme

La construction d'histogramme se fait classiquement par un CPU après avoir extrait la carte binaire. La Figure 4-16 montre un exemple des différents histogrammes créé sur des blocs glissants. Il y a un gradient (intensité et orientation) par blocs, les histogrammes sont réalisés pour un ensemble ou macro-blocs de 3 par 3 blocs. Puis l'histogramme suivant est réalisé avec un autre ensemble de blocs qui recouvre certains blocs du précédent macro-bloc. Puis ces histogrammes sont vectorisés et mis bout à bout ils permettent de créer le descripteur de l'image. En prenant un ensemble d'image appartenant à plusieurs classes d'objets, il est possible de les classifier grâce à une machine à vecteur de support (SVM).

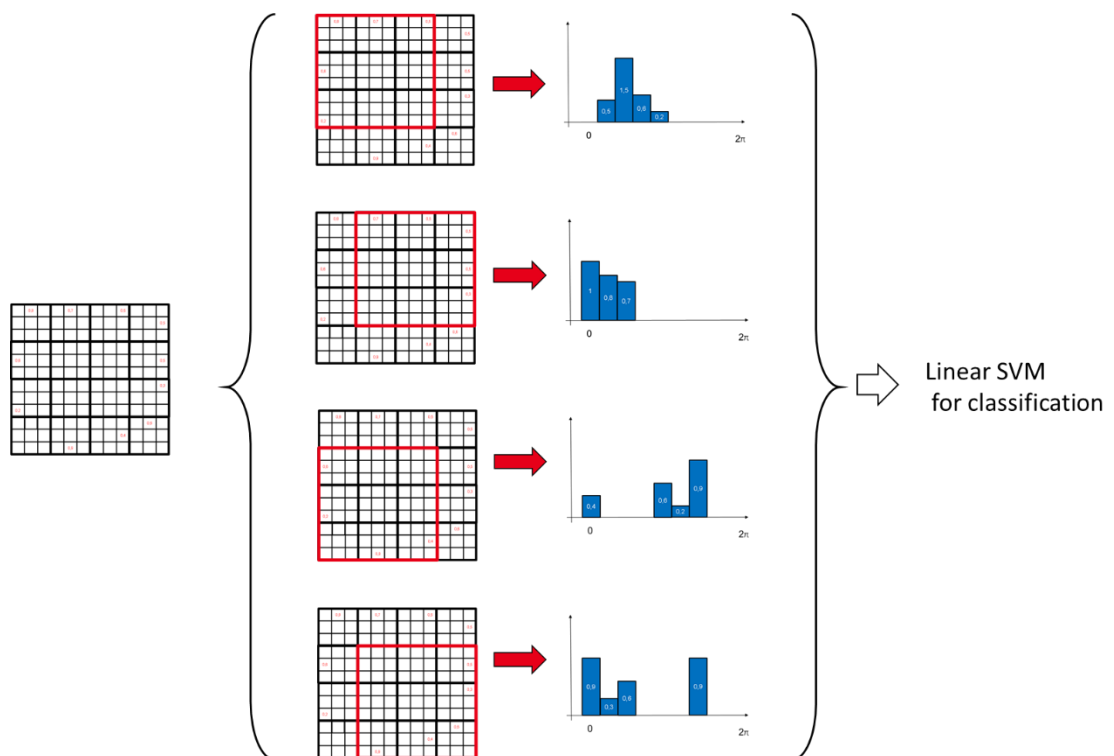


Figure 4-16: Création d'histogrammes

- Utilisation d'une machine à vecteur de support (SVM)

Une machine à vecteurs de support est une technique d'apprentissage supervisé qui permet de résoudre des problèmes de discrimination. Le but est de construire une fonction de décision telle que 4-5), afin de décider la classe d'appartenance d'un objet.

$$h(x) = \omega^T x + b \quad 4-5)$$

$x$  étant l'objet,  $\omega$  les vecteurs support et  $b$  un offset

Dans un cas de discrimination binaire, le problème à résoudre se pose de telle manière :

$$\begin{cases} x_i \text{ de classe 1 si } h(x) \geq 1 \\ x_i \text{ de classe 2 si } h(x) < 0 \end{cases} \quad 4-6)$$

Si les échantillons sont séparables linéairement, il existe une infinité de plans pouvant séparer les deux classes. La solution recherchée est celle qui maximise la marge entre le plan de séparation et les échantillons les plus proches. Ces échantillons sont appelés vecteurs support dans le sens où ils supportent l'hyperplan de discrimination, Figure 4-17. L'objectif de l'algorithme lors de la phase d'apprentissage dite de training est de trouver  $\omega$  et  $b$  tels que :

$$\forall (x_i, l_i) \begin{cases} \omega^T x_i + b \geq 1, & l_i = 1 \\ \omega^T x_i + b \leq -1, & l_i = -1 \end{cases} \Rightarrow l_i(\omega^T x_i + b) \geq 1 \quad 4-7)$$

$l_i$  étant la classe d'appartenance de  $x_i$ . L'équation à résoudre est donc la suivante :

$$\min_{\omega, b} \left\{ \frac{1}{2} \omega^T \omega \right\} \text{ avec } y_i(\omega^T x_i + b) \geq 1 \quad 4-8)$$

La solution est retournée sous forme de vecteurs supports  $\omega$  et de la valeur  $b$ , la valeur d'offset.

Dans le cas où les échantillons ne sont pas linéairement séparables, des variables d'écarts sont introduites. Le but étant de modéliser les erreurs de classification par des variables d'écart positives [Schölkopf2002].

Pour classifier des échantillons appartenant à un nombre de classes supérieur à 2, les vecteurs supports sont calculés pour chaque classe, de sorte qu'ils séparent la classe de toutes les autres classes. Cette méthode est appelée « one against all ».

Pour finir, afin d'évaluer la précision du modèle, les vecteurs sont testés pour classifier des échantillons test différents des échantillons de training. La classe de chacun des échantillons test est déterminée par 4-9). La classe d'un échantillon correspond donc à la marge la plus élevée. Puis le taux de réussite de classification est calculé.

$$C_i = \underset{c}{\operatorname{argmax}}(\omega^T x_i + b) \quad 4-9)$$

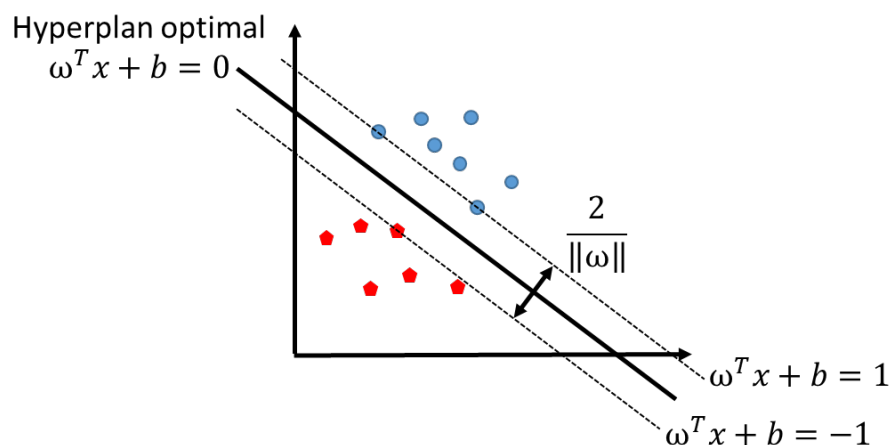


Figure 4-17: Définition de l'hyperplan optimal et des vecteurs supports.

#### 4.1.2.2.4 Simulation MATLAB et Résultats

Le processus de la création des histogrammes à partir de la carte binaire tel qu'il a été décrit dans les paragraphes précédents a été implémenté sur MATLAB. Les images utilisées pour cette classification sont les images du dataset « small NORB » qui consistent en 5 classes d'images : animaux, humains, voitures, camions, avions. Ces objets ont été pris en photos sous différents angles et différentes luminosités. L'arrière-plan est néanmoins uniformément gris. Les simulations ont été effectuées avec tout d'abord deux classes, puis trois et enfin les cinq classes. Les résultats sont notés Table 4-2 et ils sont comparés avec une classification des mêmes images mais dont les histogrammes ont été réalisés avec la méthode classique des histogrammes de gradients orientés (HoG). Pour les deux méthodes, une matrice est formée pour l'intégralité du set d'image. Elle contient par colonnes, le vecteur descripteur de l'image qui dépend de la méthode choisie. Le nombre de colonnes dépend donc du nombre d'images à classifier. Il est possible d'en conclure que pour 5 classes, la classification est efficace jusqu'à un seuil de 15, ces résultats ne sont pas éloignés d'une classification avec un HOG classique.

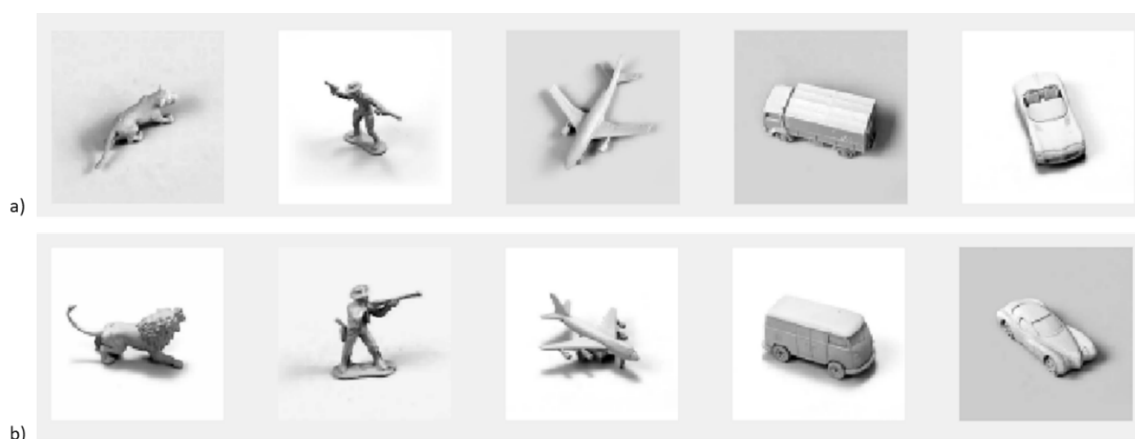


Figure 4-18: a) Images du dataset small NORB utilisé pour le training b) images du dataset small NORB utilisé pour le testing. Elles représentent les 5 catégories : animal humain avion camion et voiture.

effet du seuil sur 8bits training 200 testing 50								
seuil (pour des images sur 8bits)	Réussite de classification pour un nombre de classes différent (%)							
	2 classes: animal / humain	2 classes: camion / voiture	2 classes: humain / voiture	2 classes : humain / camion	2 classes : avion / camion	3 classes: animal humain voiture	3 classes: voiture camion avion	5 classes
5	84	93	80	87	97	93	87	86
10	79	83	81	88	99	87	82	81
15	79	84	83	83	100	87	80	81
20	86	73	79	81	100	75	81	68
40						69	78	64

HOG classique								
	Réussite de classification pour un nombre de classes différent (%)							
	2 classes	2 classes	2 classes	2 classes	2 classes	3 classes	3 classes	5 classes
	95	89	92	99	95	99	92	89

Table 4-2: Résultats des simulations avec plusieurs catégories et en comparaison avec une méthode de HoG classique sur les mêmes images

Afin de comprendre les effets de la valeur de seuil correspondant à l'inhibition, plusieurs simulations ont été effectuées en changeant ce paramètre. La valeur de seuil n'a donc pas beaucoup d'impact sur les classifications de deux classes uniquement, mis à part sur des objets qui ayant des formes proches comme les voitures et les camions, Figure 4-20-a. Cependant, elle a une forte influence sur les classifications ayant plus que 2 classes, Figure 4-20-b. En effet, pour 5 classes, un taux de classification à 86% avec une inhibition de 5, baisse jusqu'à 68% pour une inhibition de 20. Entre 5 et 15, la différence de réussite de classification n'est pas si importante. Une inhibition de 15 permet d'avoir une carte binaire ayant moins de pixels clignotants, Figure 4-19, c'est cette valeur qui sera retenue. Il faut rappeler qu'une inhibition de 15 d'une valeur sur 8 bits n'est pas une valeur constante dans le domaine temporel, elle dépend du temps auquel l'inhibition est déclenchée. Du fait de la relation en  $1/x$ , le temps séparant 15 valeurs sur 8bits est plus court pour les hautes intensités que pour les faibles intensités. Ainsi, en considérant une image ayant un temps d'intégration maximale de 20ms, il faut  $4.86\mu s$  pour séparer 255 et 240 alors qu'il en faut  $163\mu s$  pour séparer 50 et 35.

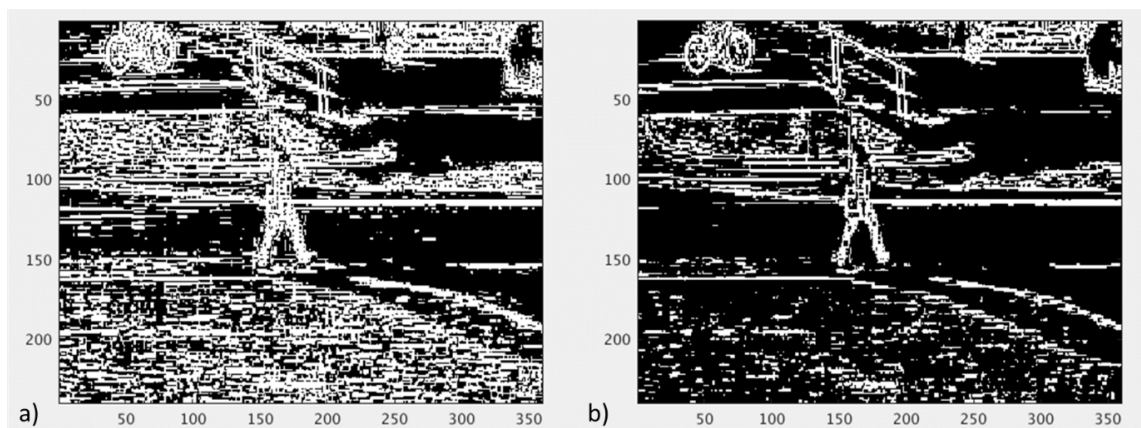


Figure 4-19: a) Carte binaire avec un seuil d'inhibition à 5 b) Carte binaire avec un seuil d'inhibition à 15

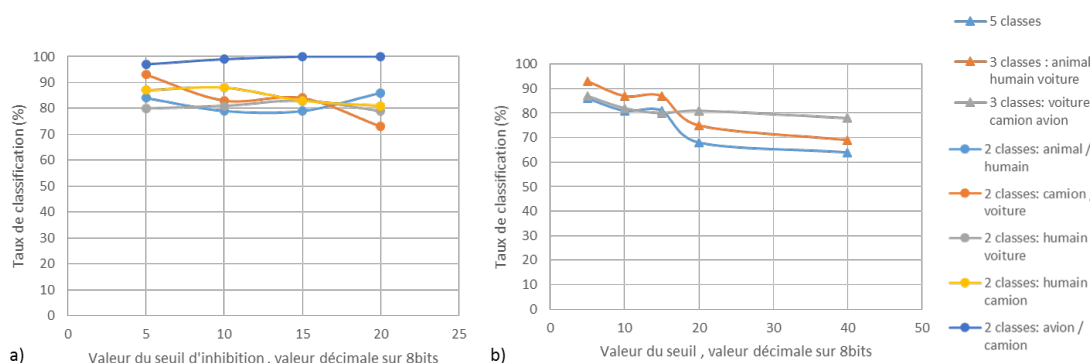


Figure 4-20: Evolution du taux de classifications en fonction de la valeur du seuil d'inhibition pour a) deux classes b) pour 3 et 5 classes

Les matrices de confusions servent à visualiser l'efficacité d'une classification. En ordonnées sont les classes auxquelles appartiennent les images de test et en abscisses celles dans lesquelles elles sont classifiées. La couleur correspond au taux de classification. Par exemple, Figure 4-21, la classe 1 a été classifiée à 85% dans la classe 1, la case 1-1 est donc gris clair. Une matrice de confusion parfaite est une matrice ayant sa diagonale blanche et le reste noir. Dans ce cas testé, la classification est suffisante et ressemble à celle d'une méthode HOG en étant plus facilement implantable.

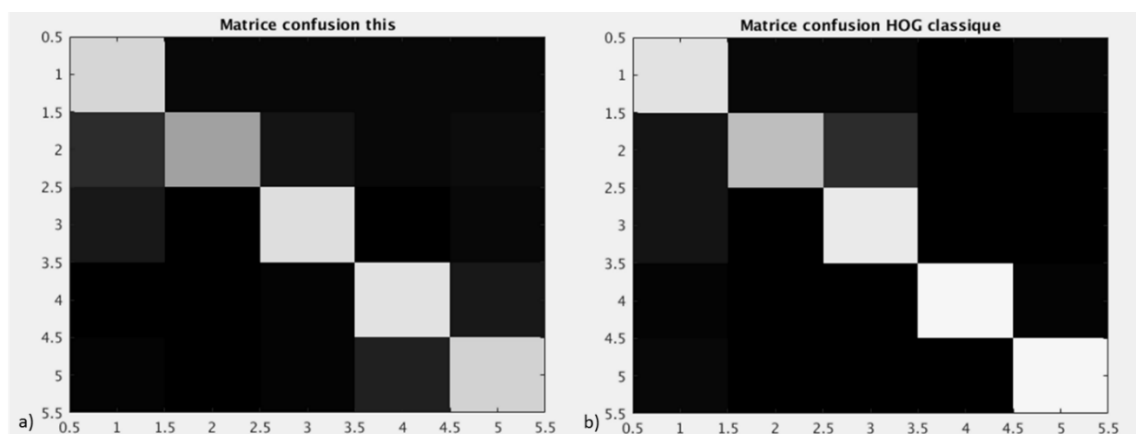


Figure 4-21: Matrices de confusions pour une inhibition de 15 (en valeur 8bits), en ordonnées les classes d'appartenance des images en entrées et en abscisse le résultat de la classification, le blanc correspond à 100% de classification et le noir à 0%. a) Classification à partir de la carte binaire b) Classification d'un HOG classique.

Des simulations en dispersion de bruit ont été effectuées. Le bruit considéré est un bruit gaussien représentant le FPN inter-bloc sur la valeur du seuil d'inhibition. Pour une image uniforme, les différents blocs n'ont donc pas le même seuil d'inhibition ce qui crée des différences dans la carte binaire. Les résultats de classification, Figure 4-22, montrent que plus le seuil d'inhibition est élevé moins il est sensible au bruit FPN sur l'inhibition. En effet, l'écart type du gain a été testé jusqu'à une valeur de 5, en relatif, il est donc moins important pour un seuil de 40 que pour un seuil de 5. Néanmoins, en considérant un seuil d'inhibition à 15, le bruit ne détériore pas trop la classification puisqu'elle reste supérieure à 70% pour un bruit gaussien avec un écart type de 5. Afin de compléter cette analyse, il faudrait réaliser des simulations en ajoutant un bruit sur l'image d'entrée pour reproduire le bruit lié à l'acquisition et

ajouter également des variations de seuil entre les pixels d'un même bloc afin de modéliser le FPN inter-pixel.

effet du bruit de seuil sur 8bits training 200 testing 50					
seuil initial	5 classes				
	pas de bruit	bruit gaussien écart type de 1	bruit gaussien écart type de 2	bruit gaussien écart type de 3	bruit gaussien écart type de 5
5	86	84	80	79	67
10	81	82	80	78	78
15	81	81	78	77	71
20	68	68	70	70	68
40	64	62	63	64	62

Figure 4-22: Effet d'un bruit gaussien avec différents écarts types sur le seuil bloc par bloc. Le bruit n'a été appliqué que sur les images de testing.

## 4.2 Architecture

A la fin de la thèse, une architecture du pixel et d'un bloc de pixels comprenant le système d'inhibition ont été développées, Figure 4-23. Le pixel comprend une photodiode et un comparateur avec un système d'auto-zéro ainsi qu'un point mémoire contrôlé par le système d'inhibition afin de mémoriser l'état du pixel à la fin du temps d'inhibition. Le système d'inhibition comprend un bloc de déclenchement du signal lorsque le premier pixel du bloc flashe et un bloc gérant la largeur du temps d'inhibition. Ce temps doit être plus court pour les pixels recevant une forte intensité lumineuse et plus faible pour les pixels sombres. Cela est dû au passage de l'information lumineuse dans le domaine temporel.

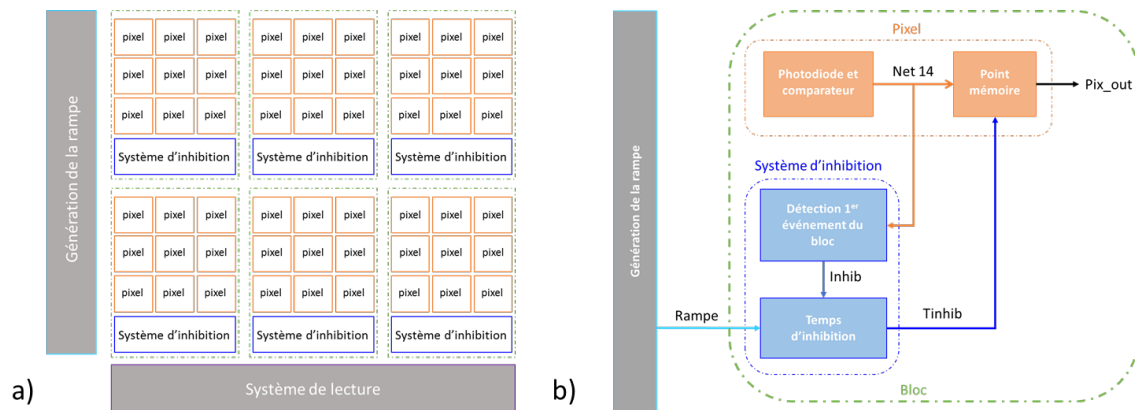


Figure 4-23: a) Schéma de la matrice du capteur b) Schéma des architectures implémentées avec les signaux correspondants

### 4.2.1 Pixel

Le pixel est composé de deux parties, un point mémoire et une photodiode reliée à un comparateur.

#### 4.2.1.1 Photodiode et comparateur

La photodiode dans le mode photoconduction est modélisée par un générateur de courant idéal et une capacité. Elle est reliée à un comparateur réalisé par un inverseur avec un auto-zéro [Liu2001]. Les interrupteurs sont contrôlés par les signaux  $Q_{R1}$ ,  $Q_{R2}$  et  $Q_{R3}$ . La Figure 4-24 montre l'architecture ainsi que les chronogrammes des signaux de contrôles et les signaux  $V_{diode}$  et  $V_{out}$  correspondant au signal d'entrée et au signal de sortie du schéma. Les phases 1 et 2 sont l'initialisation. La phase 1 permet de faire l'autozéro du comparateur, c'est-à-dire que la valeur de commutation de l'inverseur est stockée dans la capacité placée entre la photodiode et l'inverseur. Lors de la phase 1, la valeur de  $V_{ref}$  est appliquée à gauche de la capacité alors qu'à droite s'établit le point d'équilibre de l'inverseur. La capacité stocke donc la différence des deux tensions (4-10). Lors de la phase 2, la photodiode est polarisée à  $V_{DD}$  et le nœud entre l'armature droite de la capacité et le comparateur devient un nœud haute impédance. La conservation des charges fait que ce nœud a un potentiel égal à (4-11).

$$\Delta V_1 = V_{ref} - V_{seuil} \quad 4-10)$$

$$Q_{A1} = C_{auto}(V_{ref} - V_{seuil}) \text{ et } Q_{A2} = -C_{auto}(V_{ref} - V_{seuil})$$

$$Q_{B1} = C_{auto}(V_{DD} - V_x) \text{ et } Q_{B2} = -C_{auto}(V_{DD} - V_x)$$

Conservation des charges :

$$Q_{A2} = Q_{B2} \Rightarrow (V_{ref} - V_{seuil}) = (V_{DD} - V_x)$$

$$V_x = V_{DD} - (V_{ref} - V_{seuil}) \quad 4-11)$$

La phase 3 correspond à l'intégration, le temps d'intégration est fixé par l'intensité lumineuse reçue. Plus cette intensité est forte plus le pixel commutera rapidement. Le moment de commutation est la phase 4. La tension de la photodiode passe rapidement à 0 grâce à l'interrupteur de contre réaction de l'inverseur, il n'était pas alimenté pendant la phase 1 et 2 grâce à l'interrupteur contrôlé par  $Q_{R3}$  afin de ne pas perturber l'auto-zéro.

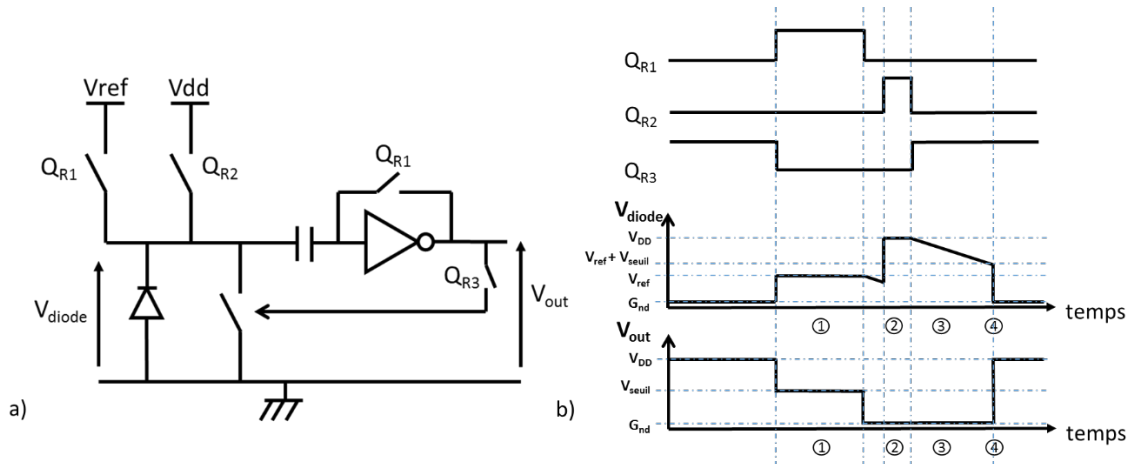


Figure 4-24: a) Architecture du bloc photodiode et comparateur b) chronogrammes des différents signaux.

Lors de l'implémentation, il a été remarqué que les interrupteurs induisaient d'importantes fuites de courant. Cela faisait commuter l'inverseur au bout d'un certain temps même lorsqu'il n'y a pas de courant photo-généré. Il n'est pas question de dark current de la photodiode mais bien de courant de fuite induit par les transistors du pixel. Afin de limiter ces fuites, deux solutions ont été mises en place. Les interrupteurs utilisés sont des PMOS et une tension de back biasing (ou body biasing) leur est appliquée. Cette tension peut aller jusqu'à  $\pm 2V$  en fonction du type de transistors : PMOS ou NMOS. Le courant de fuite détermine la valeur maximale du temps d'intégration, limitant ainsi la dynamique du signal lumineux retranscrit dans les faibles intensités lumineuses.

Lorsqu'il n'y pas de courant photo-généré, le temps de flash devrait être infini. Cependant il est limité par les différentes fuites de courant des différents transistors. Ainsi, lorsqu'il n'y pas de courant photo-généré et sans body biasing appliqué aux transistors, l'inverseur commute au bout de 4,034ms. Avec un body biasing de -0.8V sur les NMOS, et 1.8V sur les PMOS, sans courant photo-généré, l'inverseur commute à 6,407ms. Ces résultats sont obtenus avec une capacité située entre la photodiode et le comparateur, appelée capacité d'autozéro, de 50fF, et une capacité de photodiode de 25fF. Cette capacité a été choisie en concordance avec les résultats exprimés dans le chapitre 2. (Paragraphe 2.7.1.2)

Afin de limiter la consommation de l'inverseur, un transistor de type PMOS a été implémenté entre la borne d'alimentation et la source du PMOS de l'inverseur. Ce transistor est contrôlé en tension et bride l'inverseur en courant, le temps de commutation est plus long mais la consommation est diminué.

#### 4.2.1.2 Point mémoire

Le point mémoire mémorise l'état du pixel à la fin du temps d'inhibition. Le pixel a deux états possibles : le pixel a flashé lors du temps d'inhibition, le pixel n'a pas flashé. La première architecture proposée, Figure 4-25a, était similaire à un point mémoire de SRAM à 6 transistors avec un inverseur en sortie. Cependant, la consommation de ce bloc était trop importante. En effet, l'inverseur placé en amont du point mémoire afin de ne pas détériorer le signal provenant du comparateur quel que soit la charge



de l'étage, n'était pas bridé en courant. C'est pourquoi une deuxième architecture a été proposée, Figure 4-25b. Le nombre de transistors reste le même mais la consommation est divisée par deux, puisque l'inverseur n'est alimenté que lorsque le signal  $T_{inhib}$  est actif. Grâce aux deux inverseurs têtes bêtes la donnée est sauvegardée, elle est écrite lorsque le signal  $T_{inhib}$  est haut. La double porte NMOS et PMOS permet d'écrire un 0 ou un 1 dans le point mémoire.

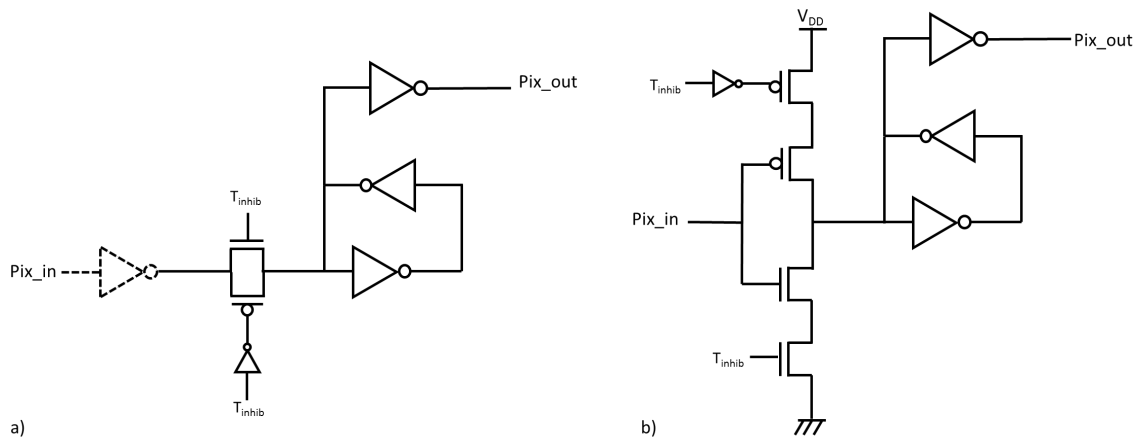


Figure 4-25: Architecture du point mémoire intégré dans le pixel a) première architecture proposée b) amélioration de l'architecture

#### 4.2.1.3 Pixel complet

Une simulation du pixel complet, photodiode comparateur et point mémoire a permis d'attester du fonctionnement du pixel ainsi que de sa consommation d'énergie.

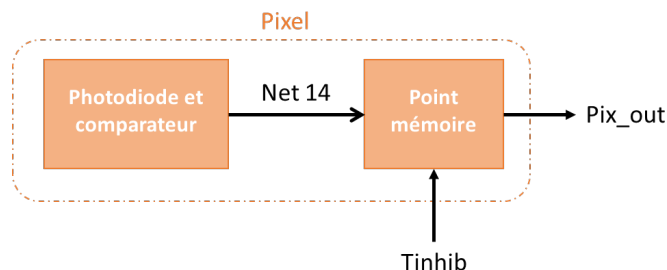


Figure 4-26: Schéma du pixel complet issu de la Figure 4-23.b et signaux associés

Dans la Figure 4-27, les différents signaux du pixel sont affichés dans le temps, issu des simulations sous CADENCE. Le fonctionnement dans les phases 1 et 2 expliqué dans la Figure 4-24 est retrouvé ici. Dans la phase 3, sous l'effet de la lumière incidente, la tension du nœud photosensible  $V(I_{pix})$  décroît. Lorsque  $V(I_{pix})$  passe la valeur  $V_{ref} + V_{seuil}$ , phase 4, une contre réaction accélère sa chute jusqu'à 0 afin d'économiser de l'énergie.  $V_{seuil}$  correspond au point de commutation de l'inverseur. La sortie du comparateur du pixel correspond au signal  $V(Net14)$ . Lorsque la tension, passe sous, le signal  $V(Net14)$  passe à 1V. Lorsque le signal d'inhibition  $V(Tinhib)$  est actif, l'état du pixel  $V(Net14)$  est inscrit dans le point mémoire,  $V(Pix_{out})$ .

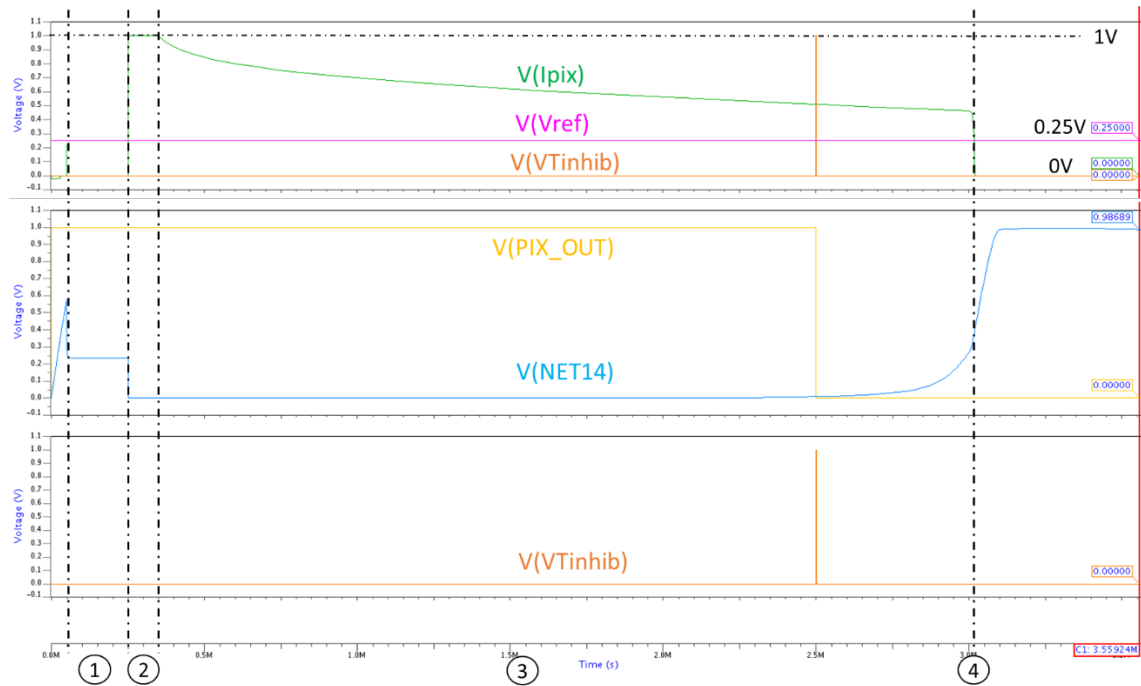


Figure 4-27: De haut en bas : chronogrammes des signaux dans le pixel, décharge de la photodiode ( $V_{Ipix}$ ) sous l'effet de la luminosité incidente et réaction du comparateur du pixel ( $V_{net14}$ ), chronogrammes des signaux pour un pixel, écriture ( $V_{pix\_out}$ ) de l'état du pixel ( $V_{net14}$ ) lors du temps d'inhibition ( $V_{Tinhib}$ ) et chronogrammes de  $V(Tinhib)$ .

La consommation en énergie du pixel est de 30pJ pour 7ms de fonctionnement avec une capacité de photodiode de 25fF qu'importe l'intensité lumineuse reçue. Le comparateur du pixel est limité en courant à 1nA ce qui réduit fortement la consommation. En effet sans cette limitation, la consommation en énergie du pixel avec un courant photo-généré nul est de 10nJ pour 7ms et 3.11nJ pour 7ms et un courant photo-généré de 50pA.

#### 4.2.2 Système d'inhibition

Le système d'inhibition, Figure 4-28, est composé de deux blocs. Le premier permet de générer le signal inhib qui déclenche le début de l'inhibition. Le deuxième bloc permet de gérer la longueur du signal d'inhibition, il sort un signal nommé  $Tinhib$  qui commande le point mémoire du pixel. Ce bloc prend en entrée un signal Rampe généré en bordure de matrice ou à l'extérieur de la matrice.

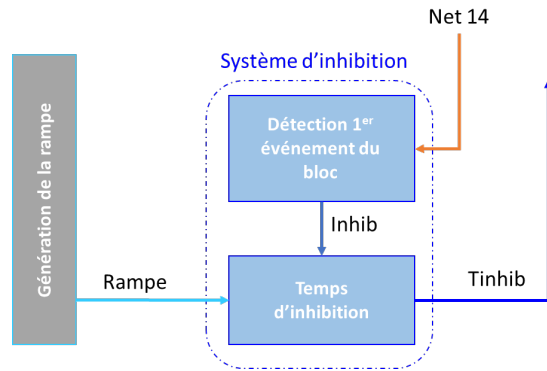


Figure 4-28: Schéma du système d'inhibition, issu de la Figure 4-23.b et signaux associés

#### 4.2.2.1 Génération du signal d'inhibition

La génération du signal d'inhibition, Figure 4-29, se fait par un simple bloc numérique composé de 4 portes OU, un transistor fonctionnant en temps qu'interrupteur et un inverseur à la sortie afin d'avoir un signal numérique stable et pour isoler le bloc inhibition du reste du circuit afin qu'il n'y ait pas de perturbations. Dans une version précédente, une bascule D avait été mise à la place de l'inverseur et de l'interrupteur mais par soucis d'optimisation de surface cette méthode n'a pas été retenue. En effet, une bascule comporte minimum 16 transistors et consomme 18.81pJ au lieu de 8.90pJ dans la version sans la bascule présentée Figure 4-29. Ces chiffres ont été mesurés pour 1ms de fonctionnement et un passage de 1 à 0 du signal inhib\_b.

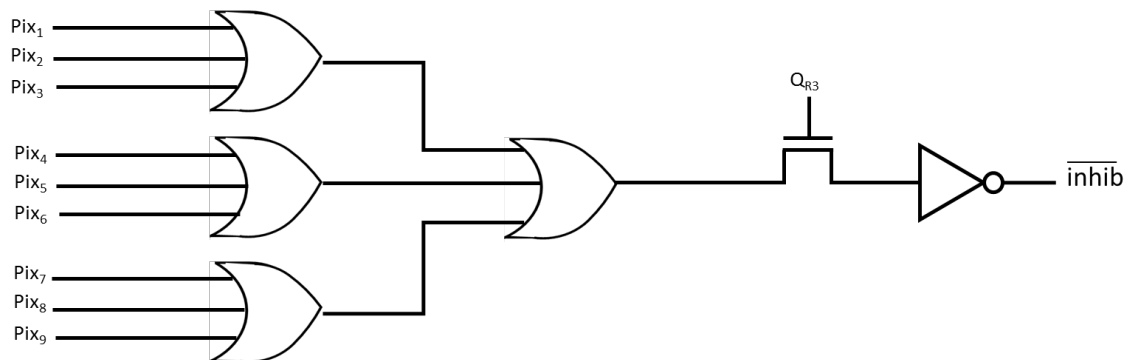


Figure 4-29: Architecture pour la génération du signal d'inhibition suite à un flash d'un des pixels du bloc de 3 par 3 pixels.

#### 4.2.2.2 Gestion du temps d'inhibition

##### 4.2.2.2.1 Architecture

L'architecture ci-dessous est proposée afin de gérer le temps d'inhibition. Il doit être dépendant du moment auquel il est commencé. Pour cela, des capacités commutées sont utilisées, tout d'abord pour mémoriser le point de commutation du comparateur puis afin de stocker une fraction de la valeur de la rampe en entrée, Figure 4-30.

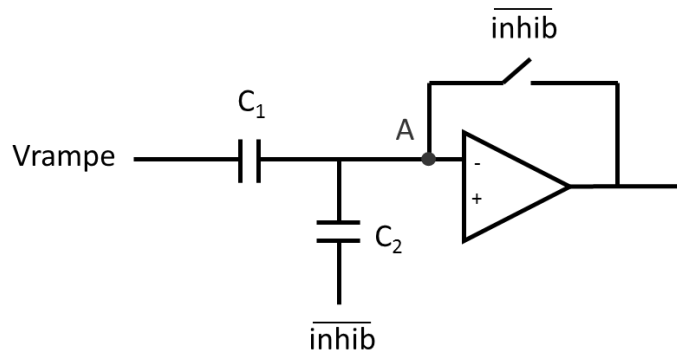


Figure 4-30: Architecture proposée pour la gestion du temps d'inhibition

Dans un premier temps, l'amplificateur est rebouclé avec un gain unitaire et la valeur de seuil de commutation est stockée dans la capacité  $C_1$  puisque la rampe est nulle. La valeur stockée dans  $C_2$  est la différence entre la valeur du signal  $\overline{\text{inhib\_b}}$  et la valeur du seuil. Lorsque l'intégration démarre, la différence entre la valeur de la rampe et la valeur de seuil est stockée dans la capacité  $C_1$ . Le signal d'entrée de l'inverseur, maintenu en contre réaction unitaire par l'interrupteur, est maintenu à la tension de seuil du comparateur, Figure 4-31. Puis lorsque le signal inhibition passe au niveau bas, le nœud A chute à  $V_x$  exprimé en (4-12). Dans les équations ci-dessous  $t_A$  est le temps juste avant la commutation du signal  $\overline{\text{inhib}}$  et  $t_B$  le temps juste après la commutation du signal  $\overline{\text{inhib}}$ .

$$Q_{C_1 2 t_A} = C_1 (V_{\text{seuil}} - V_{\text{rampe}_{t_A}}) \text{ et } Q_{C_2 2 t_A} = C_2 (V_{\text{seuil}} - V_{DD})$$

$$Q_{C_1 2 t_B} = C_1 (V_x - V_{\text{rampe}_{t_B}}) \text{ et } Q_{C_2 2 t_B} = C_2 (V_x)$$

Conservation des charges :

$$Q_{t_A} = Q_{t_B}$$

$$V_{\text{seuil}}(C_1 + C_2) - C_1 V_{\text{rampe}_{t_A}} - C_2 V_{DD} = V_x(C_1 + C_2) - C_1 V_{\text{rampe}_{t_B}}$$

En considérant que :  $V_{\text{rampe}_{t_A}} \approx V_{\text{rampe}_{t_B}}$

$$V_x = V_{\text{seuil}} - \frac{C_2}{C_1 + C_2} V_{DD} \quad (4-12)$$

Le temps d'inhibition est défini par le temps entre le déclenchement du signal  $\overline{\text{inhib}}$  et la commutation du comparateur. Ce temps est illustré sur la Figure 4-31, c'est le temps qu'il faut pour que la tension du nœud A revienne au seuil de commutation de l'inverseur. Le signal  $T_{\overline{\text{inhib}}}$  est donc au niveau haut jusqu'au moment où le signal  $\overline{\text{inhib\_b}}$  commute, alors le signal  $T_{\overline{\text{inhib}}}$  passe au niveau bas et lorsque l'inverseur commute le signal  $T_{\overline{\text{inhib}}}$  passe au niveau haut. Avec une rampe non linéaire

croissante telle que sur la Figure 4-31, le temps d'inhibition est plus court au début de la rampe qu'à la fin, comme souhaité.

Afin de séparer 15 valeurs quantifiées sur 8 bits, par exemple 255 et 240, il faut un temps d'inhibition de  $1.21\mu\text{s}$  pour un temps d'intégration maximal de  $5\text{ms}$  et  $40\mu\text{s}$  pour séparer deux valeurs plus faibles telles que 50 et 35. Ces valeurs sont ajustables grâce aux valeurs des capacités  $C_1$  et  $C_2$ . En effet si la rampe considérée est une charge de filtre RC le temps d'inhibition varie comme indiquée par (4-13).

$$V_1 = 1 - e^{-\frac{t_1}{RC}} \Rightarrow t_1 = -RC * \ln(1 - V_1)$$

$$V_2 = V_1 - \frac{C_2}{C_1 + C_2}$$

$$\Delta T = t_1 - t_2 \Rightarrow \Delta T = RC * \left( \ln\left(1 - V_1 - \frac{C_2}{C_1 + C_2}\right) - \ln(1 - V_1) \right) \quad 4-13$$

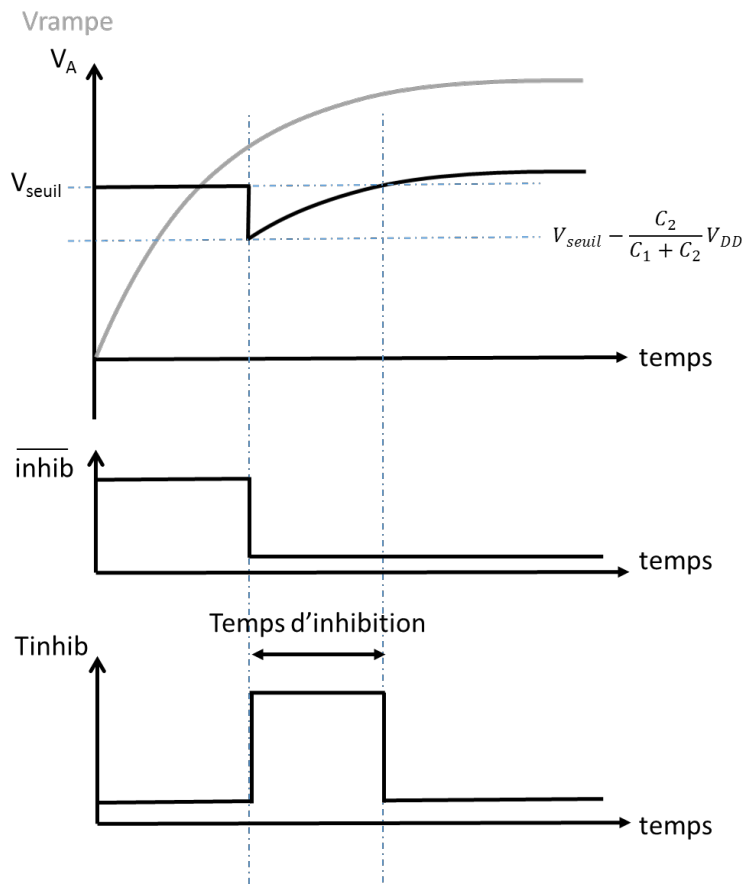


Figure 4-31: Chronogrammes pour la génération du temps d'inhibition variable en fonction du temps de déclenchement du signal inhib.  $V_A$  étant la tension d'entrée du comparateur.

#### 4.2.2.2 Génération d'une rampe non linéaire

Le temps d'inhibition doit idéalement être plus court pour les fortes luminosités reçues que pour les faibles luminosités. Pour cela il faut une rampe non linéaire en entrée de l'architecture proposée ci-dessus. Pour les simulations effectuées durant cette thèse, une charge de circuit RC a été utilisée. La rampe est initialisée à 0 en même temps que l'initialisation globale du circuit à chaque trame puis la charge de la capacité à

travers la résistance commence. Le dimensionnement de la capacité et de la résistance dépend du temps maximum d'intégration possible. Celui-ci dépend des fuites de courant dans le pixel comme vu précédemment. Par exemple, si le temps d'intégration maximum est défini à 1ms, la charge de la capacité devra également être de 1ms ce qui correspond à cinq fois le produit  $R \cdot C$ . En prenant une capacité d'un pico Farad, cela donne une résistance d'environ 200M $\Omega$ , Figure 4-32. Il est également possible de déporter hors du circuit la réalisation de la rampe. Cela serait réalisable dans la mesure où la rampe est identique et synchrone pour tous les blocs de la matrice. Cependant cela induirait des délais de propagation dans la matrice qu'il faudrait compenser avec le matériel adéquat.

En prenant des temps d'intégration maximum de 20ms, une résistance de 400M $\Omega$  pour une capacité de 10pF est nécessaire.

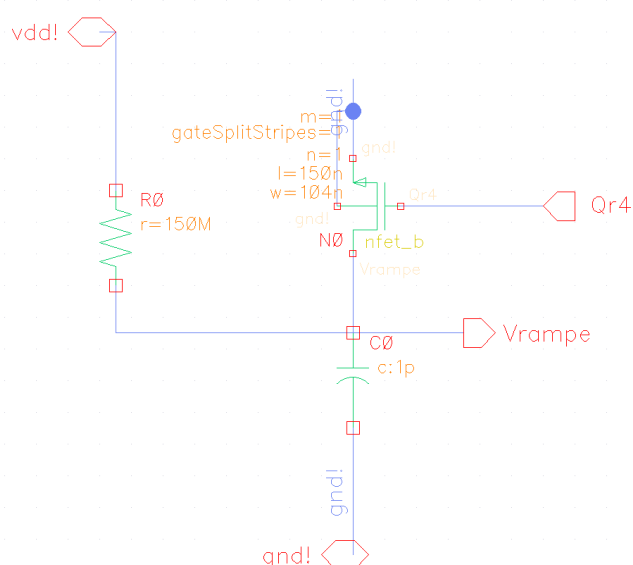


Figure 4-32: Schéma pour le test d'une rampe non linéaire

### 4.2.3 Système de lecture synchrone

L'avantage de cette architecture comparée aux architectures présentées dans le chapitre III est que cette architecture a une lecture synchrone sans ADC. Le mode de lecture est donc le même qu'un capteur classique global shutter. Les lignes sont évacuées une par une en pied de colonne puis lues pixel par pixel. L'absence d'ADC dans cette architecture permet de réduire considérablement la consommation puisque l'ADC est l'élément consommant le plus d'énergie dans un imageur [Choi2015]. Les données de sorties sont des données binaires, le système doit donc lire qu'un seul bit par pixel.

### 4.2.4 Simulation Cadence et résultats

Une simulation a été effectuée pour un bloc de 3 par 3 pixels afin de réaliser une première validation électrique. La rampe utilisée ici est une rampe linéaire avec un temps d'intégration maximum d'une milliseconde. Sur le graphe Figure 4-33, on

remarque que les pixels 4, 5, 6, 7, 8, et 9 ont bien été inhibés. En effet, les valeurs des courants appliqués aux pixels en tant que courants photo-générés sont les suivants : Pix1=100fA, Pix2=200fA, Pix3=500fA, Pix4=1pA, Pix5=2pA, Pix6=5pA, Pix7=6pA, Pix8=7pA, Pix9=8pA. Le temps d'inhibition est de 250µs avec les capacités  $C_1 = 10\text{fF}$   $C_2=25\text{fF}$ . Pour un bloc la consommation est de 77nJ pour un temps de trame de 7ms soit 8.56nJ/fr.pix. Cette consommation devrait pouvoir être améliorée en réduisant les fuites dans le pixel et en réglant la consommation du système d'inhibition.

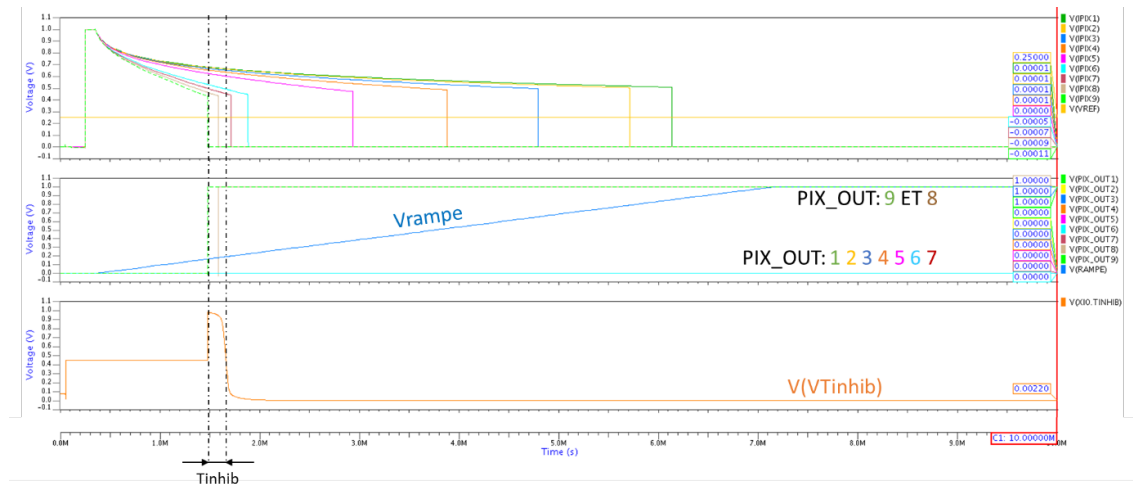


Figure 4-33: Résultats de simulation pour un bloc de 3 par 3 pixels et une rampe linéaire

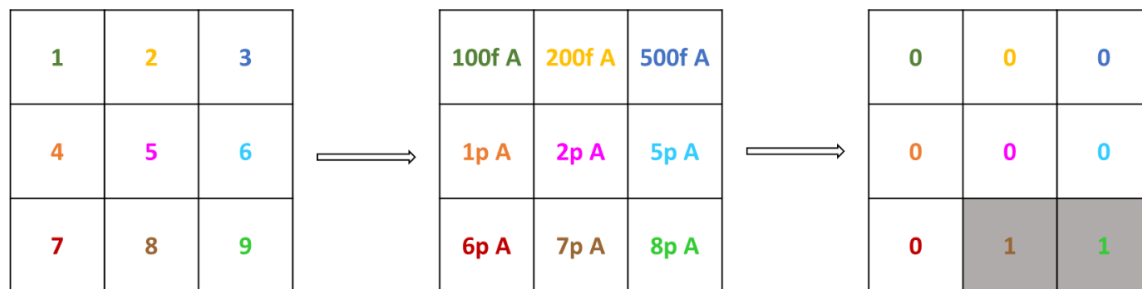


Figure 4-34: Bloc de pixel avec courant photo-généré et carte binaire exporté des simulations décrite dans la Figure 4-33, le code couleur correspond aux signaux de la figure ci-dessus.

#### 4.2.5 Discussions sur la consommation du système.

Le pixel événementiel consomme 29pJ pour un temps trame de 7ms. Cette architecture permet d'inhiber les pixels par bloc ayant la même information lumineuse que le pixel le plus lumineux du bloc considéré. Le but est de sortir une carte binaire représentant les pixels inhibés et les pixels non inhibé. Cette carte est établie par bloc à la fin de l'inhibition, ce qui signifie qu'après ce temps d'inhibition les pixels n'ayant pas encore flashé n'ont pas besoin de continuer leur intégration. Ils peuvent donc être éteints afin de minimiser leur consommation. Ainsi le pire cas à considérer au niveau de la consommation d'énergie par bloc est le cas où tous les pixels flashent pendant le temps d'inhibition, puisque qu'aucun de ces pixels n'aura économisé de l'énergie en étant éteint avant que leur comparateur ne commute, Figure 4-35-a. En effet la principale consommation du pixel vient de la zone de commutation (cross conduction) du comparateur. Pour un bloc de 3 par 3 pixels, la consommation maximale est donc  $9 \times 29\text{p} = 261\text{pJ}$  pour un temps trame de 7ms. A cette consommation il faut ajouter la

consommation liée au système d'inhibition. Le meilleur cas est lorsque pendant le temps d'inhibition seul un pixel flashe (le pixel ayant déclenché l'inhibition), Figure 4-35-b. La consommation des autres pixels est fortement limitée puisque leur comparateur n'entre pas dans le régime de cross conduction. La consommation du bloc est donc 29pJ plus une portion de la consommation maximale des 8 autres pixels et évidemment en ajoutant la consommation du système d'inhibition. De plus, plus il y aura de pixels dans un bloc, plus la consommation du système d'inhibition ramené au nombre de pixel sera faible.

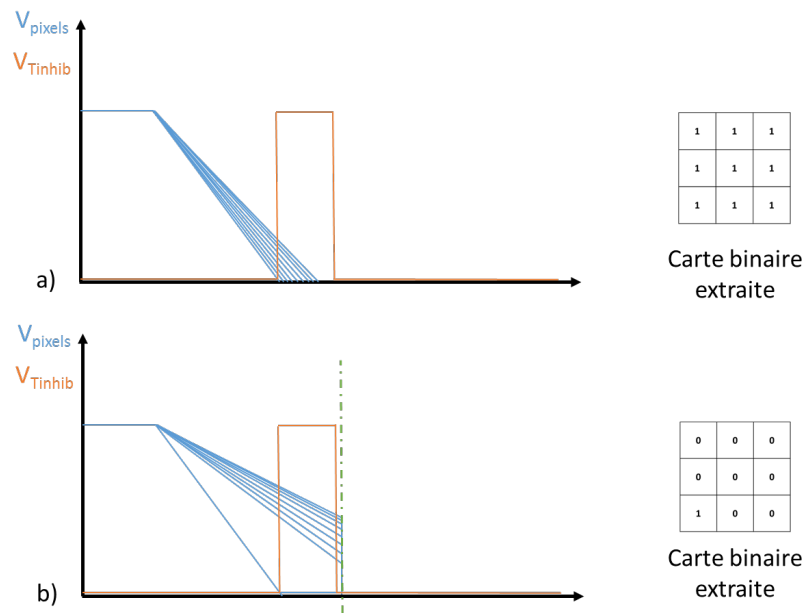


Figure 4-35: Illustration du pire cas (a) et du meilleur cas (b) pour la consommation d'énergie avec la carte binaire extraite associée. Dans le meilleur cas (b) l'intégration des pixels est stoppée après l'inhibition

Dimensionné correctement, le système d'inhibition permet donc de réduire fortement la consommation d'énergie du capteur comparé à un capteur événementiel sans système d'inhibition qui comprendrait un système de lecture AER. Pour exemple de capteur événementiel AER il est possible de prendre celui présenté dans [Guo2007], un capteur TTFS sans inhibition avec un système de lecture AER, consommant 3.1mW pour une matrice de 32 par 32 à 30fps. Le capteur développé dans cette thèse consomme 8.5nJ/fr.pix à 143fps pour une matrice de 3 par 3 pixels. La consommation d'un capteur de ce type de taille 32 par 32 peut donc être estimée à 138 $\mu$ W. Néanmoins le système de lecture synchrone n'est pas estimé en consommation ici. La consommation totale du capteur est cependant nettement inférieure à la consommation du capteur dans [Guo2007].

## 4.3 Les apports de la solution

### 4.3.1 Par rapport à une solution non intégrée

Les avantages d'une solution intégrée monolithiquement par rapport à une solution non intégrée avec le traitement déporté sur une unité de traitement extérieure à la puce sont des avantages de consommation, de délai et d'utilisation de surface.



En effet, la consommation d'un système imageur standard comportant un ADC est élevée [Choi2015]. A celle-ci, il faut ajouter le coût de transmission des données (I/O ring, canal de transmission externe) et le coût en consommation des traitements à effectuer sur l'image transmise, à savoir la réalisation de la carte binaire à partir de l'image acquise, ainsi que la classification avec un SVM. Cependant cette classification a le même coût, qu'il soit réalisé au sein du CPU contenu dans la puce que dans un CPU externe.

### 4.3.2 Comparaison à l'état de l'art

Dans cette partie, l'architecture étudiée durant cette thèse est mise en regard de l'état de l'art. Cependant, les chiffres de consommation sont à modérer du fait que l'architecture développée n'est qu'une preuve de concept et n'a pas été optimisée pour de la faible consommation. Des améliorations sont imaginables telles que l'extinction des pixels du bloc à la fin du temps d'inhibition, ainsi la zone de commutation des inverseurs est réduite voire supprimée pour certains pixels.

La comparaison avec l'état de l'art permet de mettre en évidence qu'il n'y a pas encore de système de vision en technologie FDSOI 28nm considéré comme une technologie avancée pour l'imagerie. De plus, seul très peu de systèmes de vision n'ont que peu de transistors dans le pixel ce qui impacte fortement le facteur de remplissage.

Seul un bloc complet de 3 par 3 pixels avec le système d'inhibition a été simulé, les chiffres de consommations ont été estimés pour une matrice de 32 par 32 pixels et pour une matrice de 128 par 128 pixels. Cependant la consommation du système de lecture n'est pas comprise dans ces consommations.

#### 4.3.2.1 *Extraction de contours*

Avec ce capteur, l'extraction de contour se fait sans ajouter de traitement par la suite. Ces informations sont utilisées pour des applications telles que la robotique où une forte cadence d'image est requise afin de prendre les décisions de manière précise et rapide. Plusieurs publications ont été répertoriées dans la Table 4-3. Bien que dans certaines publications, l'acquisition est événementielle, la lecture est majoritairement synchrone comme dans l'architecture étudiée dans cette thèse. Cela permet de s'affranchir du matériel asynchrone.

Sans optimisation design, les valeurs obtenues sont raisonnables et laissent à penser qu'avec une architecture plus optimisée des performances basses consommations pourront être atteintes. De plus, seule cette architecture est en FDSOI 28nm dans l'état de l'art. Cette technologie a été choisie afin de permettre un raisonnement de basse consommation au niveau système.

	Travaux de thèse	Kim 2017	Schmitz 2017	Berkovich 2015	Gottardi 2009	Costas-Santos 2007
Fonctionnalités	Détection de mouvement, extraction de contours, HoG	Extraction de contours	Image statique, extraction de contours, image filtering	Contraste spatial	Contraste temporel et spatial	Rétine, contraste spatial
Taille de la matrice	128 x 128	128 x 160	80 x 64	110 x 110	128 x 64	32 x 32
Process	FDSOI 28nm	0.18µm	0.13µm	0.35µm	0.35µm	0.35µm
N. de transistor dans le pixel	7T	4T	16T	34T	45T	104T
Fill Factor (%)	N/A	N/A	12	23	20	3
Lecture	Synchrone	Synchrone	Synchrone	Synchrone	Asynchrone	Asynchrone
Consommation d'énergie	7mW	4.3mW	31mW	30µW	100µW	10mW
Frame rate	143fps	3200fps	801fps	30fps	50fps	N/A
FOM J/frame.pixel	3nJ/fr.pix	70pJ/fr.pix	7.03nJ/fr.pix	83pJ/fr.pix	244pJ/fr.pix	6.5nJ/fr.pix

Table 4-3: Etat de l'art pour de l'extraction de contour grâce à des systèmes de vision

### 4.3.2.2 Détection de mouvement

Comme vu précédemment, la détection de mouvement peut s'effectuer par différence de deux trames, ce qui consiste en un traitement bas niveau et bas coût (en surface et en consommation d'énergie). Pour ces applications également les lectures sont majoritairement synchrones, Table 4-4. Comparé à des applications de détection de contours, l'écart de consommation entre cette architecture et l'état de l'art est plus important. Cependant, la consommation est comparable aux architectures d'acquisition temporelle telles que [Lichtsteiner2006] et [Mallik2005]. De plus, comme pour la détection de contour, il n'y a pas d'architecture en technologie FDSOI 28nm.

	Cette thèse	Ohmaru 2016	Choi 2013	Cottini 2013	Kim 2013	Gottardi 2009	Lichtsteiner 2006	Mallik 2005
Fonctionnalités	Détection de mouvement, extraction de contours, HoG	Détection de mouvement	Contraste temporel: détection de mouvement et HoG	Soustraction d'arrière plan: détection de mouvement	Détection de mouvement et moyennage temporel	Contraste temporel et spatial	Changement temporel	Détection de mouvement
Taille de la matrice	128 x 128	240 x 160	128 x 128	64 x 64	48 x 16	128 x 64	128 x 128	90 x 90
Process	FDSOI 28nm	0.5µm/0.18µm	0.18µm	0.35µm	0.13µm	0.35µm	0.35µm	0.5µm
N. de transistor dans le pixel	7T	5T	11T	45T	10T	45T	26T	6T
Fill Factor (%)	N/A	27.5	30	12	38	20	8.1	17
Lecture	Synchrone	Synchrone	Synchrone	Synchrone	synchrone	Asynchrone	Asynchrone	Asynchrone
Consommation d'énergie	7mW	25.3µW	3.31µW	33µW	467nW	100µW	30mW	4.3mW
Frame rate	143fps	60 fps	15fps	13fps	5fps	50fps	282fps	30fps
FOM J/frame.pixel	3nJ/fr.pix	10.9pJ/fr.pix	13.46pJ/fr.pix	620pJ/fr.pix	122nJ/fr.pix	244pJ/fr.pix	15nJ/fr.pix	17.7nJ/fr.pix

Table 4-4: Etat de l'art pour de la détection de mouvement sur puce

A noter qu'un autre mode dans l'architecture présentée dans la publication [Choi2013] permet d'extraire les caractéristiques de l'image à la manière d'une approche d'histogramme des gradients orientés.

## 4.4 Conclusion

Ce chapitre décrit une des contributions de la thèse, le développement des descripteurs d'image et leur implémentation dans la matrice. Le capteur développé crée une carte binaire de l'image qui représente les contrastes locaux de l'image. Cette carte binaire est réalisée en numérisant l'information de contraste, le seuil de numérisation est défini par bloc de 3 par 3 pixels. La carte binaire peut être réalisée soit grâce à un capteur de type TTFS soit en traitement externe avec un capteur qui sort basiquement une image. Elle peut être utilisée pour faire des traitements directement comme de la détection de mouvement ou de la détection de contour. Ces traitements peuvent servir comme premiers traitements bas coût effectués dans le cadre de réveil progressif d'un nœud IoT mais leur faible fiabilité ne permet pas de les utiliser en tant que traitement principal.

C'est pourquoi des descripteurs ont été créés afin de rendre l'information contenue dans la carte binaire plus robuste et plus exploitable. Pour cela, l'approche choisie s'appuie sur les histogrammes des gradients orientés. Chaque bloc de pixels permet de déterminer le gradient majoritaire en orientation et en intensité du bloc. Cette étape est réalisée en dénombrant par masque angulaire les valeurs binaires et leurs positions dans le bloc puis en faisant une convolution avec un signal triangulaire afin d'éviter les conflits. Un descripteur de l'image est donc créé sous forme de vecteur représentant les histogrammes des gradients pour chaque macro-bloc glissant de l'image. Enfin ce descripteur est utilisé pour classifier les images. Des simulations ont été effectuées avec des images de la base de données « small orb » et les résultats ont été comparés avec les résultats de classification pour un algorithme d'histogramme des gradients orientés (HoG) classique. En effet, pour 5 classes, la classification est efficace à 81% avec la carte binaire et un seuil fixé à 15, la classification avec du HOG classique est efficace à 89%. La différence de classification est acceptable en considérant que les ressources nécessaires pour effectuer les traitements sont moindres avec la carte binaire.

Dans une seconde partie de ce chapitre, l'architecture permettant de créer la carte binaire a été explicitée. Elle comprend des pixels et un système d'inhibition. Les pixels sont composés d'une photodiode en mode intégration, d'un comparateur et d'un point mémoire. Les limites liées aux fuites de courant ont été explicités et les valeurs des capacités utilisées ainsi que des courants photo-générés utilisés pour les simulations CADENCE, ont été pris en compte grâce au circuit de test des photodiodes en FD-SOI 28nm réalisé durant la thèse. L'architecture du point mémoire a été réalisée afin de limiter au maximum la consommation. Le rôle du point mémoire du pixel est de mémoriser l'état du pixel à la fin du temps d'inhibition. La consommation totale du pixel a été simulée à 30pJ pour 7ms. L'architecture du système d'inhibition reste à améliorer. En effet la consommation d'un bloc de 3 par 3 pixels revient à 8.5nJ/fr.pix. Mis au regard de la consommation du pixel seul, on peut en déduire que le système d'inhibition a une forte consommation d'énergie. Le système d'inhibition est composé d'un bloc inhibition totalement digital qui permet de repérer lorsqu'un pixel du bloc flashe ce qui déclenche l'inhibition pendant un temps donné. La gestion de ce temps

est laissée à un bloc dédié, composé de la génération d'une rampe non linéaire croissante et de la génération du signal  $T_{inhib}$ . Il y a un compromis entre le temps d'intégration maximal, l'intégration dans la puce de la génération de la rampe et le temps d'inhibition.

Enfin, le système de lecture est un système de lecture synchrone qui ne comporte pas d'ADC puisque l'information est déjà numérisée. Cela représente un avantage au niveau de la consommation d'énergie comparé à une approche non intégrée. Ainsi qu'un avantage par rapport à la surface de silicium utilisé pour réaliser le capteur dans son intégralité.

Mis au regard de l'état de l'art sur des capteurs d'image capables d'effectuer des traitements tels que de l'extraction de contours ou de la détection de mouvement, la consommation en énergie est d'un ordre de grandeur raisonnable. L'architecture actuellement fonctionnelle, mais basique, reste à optimiser, notamment la consommation du système d'inhibition. Cependant, dans l'état de l'art, aucune architecture de capteur d'image n'a été développée en FDSOI 28nm. Cette technologie permet d'avoir accès à des ressources de traitements digitaux très performantes et basse consommation, ce qui peut rendre le système global, capteur et traitement, basse consommation, même avec un capteur qui ne présente pas des performances impressionnante en basse consommation.

# Conclusion et perspectives

Cette thèse s'inscrit dans le contexte d'applications de l'internet des objets ou IoT dans lequel la contrainte en consommation d'énergie est forte. L'approche d'intégration système via un système sur puce ou SoC hétérogène permet d'avoir un nœud IoT polyvalent qui peut répondre à différents besoins applicatifs en optimisant le système afin d'obtenir de faibles consommations d'énergie au niveau des différents blocs ou du système global. Ce nœud comporte une partie « always-on » toujours active et fortement contrainte en consommation et une partie « on-demand » qui sera réveillée pour effectuer des tâches plus complexes à la demande.

Cette thèse a permis de commencer à étudier les possibilités d'utilisation de capteurs d'images dédiés à des applications très basse consommation. Ces consommations sont comparables à celles demandées par une application de type IoT. Dans ce cadre, l'architecture générale de l'imageur classique est à revoir. Deux objectifs sont à atteindre. D'une part la contrainte d'énergie en proposant une architecture basse consommation permettant le prétraitement des images afin de relaxer les traitements périphériques en effectuant notamment des prises de décision au plus tôt. Et d'autre part la contrainte de robustesse par rapport à la technologie ciblée non dédiée à l'imagerie, mais la plus à même de répondre aux attentes d'un nœud IoT. Des études ont également été menées afin d'évaluer les potentialités d'imagerie dans des technologies ou des composants prototypes du CEA leti.

Dans ce contexte IoT, la technologie FDSOI 28nm a été choisie. En effet, elle permet d'implémenter des fonctions numériques avec un très bon rapport performance de calcul / consommation. Cette technologie permet également d'implémenter des fonctions analogiques / RF très basse consommation avec de très bonnes performances. Cette thèse se focalise particulièrement sur la partie always-on du nœud IoT. L'imageur est donc soumis à de fortes contraintes de consommation d'énergie tout en étant implémenté dans une technologie très avancée. Il aura pour charge de détecter un événement de la scène et de déclencher un réveil dans la partie on-demand.

Un premier circuit a été réalisé afin d'évaluer la réponse électro-optique de pixels classique 3T en technologie FDSOI 28nm. Les 14 pixels du circuit ont différents types de photodiode. La valeur du courant photo-généré obtenu pour un rendement de 100% de la photodiode donnée sous un éclairage de 7000lux a été évaluée en fonction de la taille et du modèle de la photodiode. Cela a permis de conclure sur le type de photodiode à utiliser dans une future implémentation. Le meilleur rendement global obtenu est de 45% pour une photodiode de type N<sup>+</sup>P de taille 5 par 5  $\mu\text{m}^2$ . Un second circuit a été réalisé afin d'évaluer les potentialités en termes de performances et de compacité qu'offre l'implémentation d'un pixel dans une technologie prototype dite 3D séquentielle basée sur la FD SOI 28nm. De plus, une réflexion a été menée sur des architectures nouvelles intégrant une photodiode sous le BOX du transistor FDSOI.

Une étude a été menée sur l'impact des non-idéalités de la technologie pour une acquisition d'image en tension comparé à une acquisition d'image dans le domaine temporel. Dans l'optique d'optimiser la consommation d'énergie, une acquisition d'image dans le domaine temporel paraît plus pertinente qu'une acquisition en

tension. Ainsi deux architectures de type time-to-first spike (TTFS) ont été étudiées et simulées sous MATLAB : l'IM-TTFS ainsi que l'architecture « multi-bus ». L'architecture multi-bus utilise la densité de métaux disponibles en FDSOI 28nm et permet de réduire le flot de données, mais ne semble cependant pas l'architecture la plus pertinente pour intégrer du traitement dans le pixel. Au contraire, l'architecture IM-TTFS permet de réduire le flot de données plus efficacement, tout en permettant facilement l'intégration de traitement grâce au système d'inhibition par blocs.

Ainsi, une dernière architecture a été réalisée reprenant le principe de l'IM-TTFS. L'architecture a un fonctionnement synchrone, permettant de s'affranchir du mécanisme de lecture asynchrone (AER) coûteux en surface et en consommation. L'architecture intègre des traitements d'image permettant d'extraire une carte binaire représentant les contrastes locaux d'une image. Cela permet d'être insensible aux changements de lumière globaux (passage de nuage ou scintillement des néons lumineux par exemple) ainsi que d'être robustes aux fuites de courant inhérentes à la technologie avancée utilisée. Les performances de l'architecture finale sont raisonnables comparées à l'état de l'art mais restent à optimiser.

Ainsi, en perspective, l'architecture finale pourrait être optimisée au niveau de la consommation d'énergie et dans l'implémentation de la rampe non linéaire servant de signal d'entrée au système d'inhibition. De plus, un prototype pourrait être conçu ce qui permettra d'avoir des mesures de consommation globale comprenant le système de lecture synchrone. Puis cette architecture pourrait être intégrée dans un nœud IOT afin de mettre en place les mécanismes de réveil tels que la détection de mouvement et d'exploiter les ressources de calcul pour effectuer de la classification par exemple. Enfin, cette architecture a été pensée afin d'être compatible pour une implémentation 3D séquentiel, notamment en n'utilisant que des transistors GO1. Un circuit 3D séquentiel aux performances et compacités intéressantes pourrait donc être réalisé.

# Références

- Andoh2000 F. Andoh, H. Shimamoto, and Y. Fujita, “A digital pixel image sensor for real-time readout,” *IEEE Transactions on Electron Devices*, vol. 47, no. 11, pp. 2123–2127, 2000.
- Beigne2015a E. Beigné, J.-F. Christmann, A. Valentian, O. Billoint, E. Amat, and D. Morche, “Uttb fdsoi technology flexibility for ultra low power internet-of-things applications,” in *Solid State Device Research Conference (ESSDERC), 2015 45th European*, 2015, pp. 164–167.
- Beigne2015b E. Beigne *et al.*, “A 460 MHz at 397 mV, 2.6 GHz at 1.3 V, 32 bits VLIW DSP Embedding F MAX Tracking,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 125–136, Jan. 2015.
- Berkovich2015 A. Berkovich, M. Lecca, L. Gasparini, P. A. Abshire, and M. Gottardi, “A 30  $\mu$ W 30 fps 110 x 110 Pixels Vision Sensor Embedding Local Binary Patterns,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 9, pp. 2138–2148, Sep. 2015.
- Bigas2006 M. Bigas, E. Cabruja, J. Forest, and J. Salvi, “Review of CMOS image sensors,” *Microelectronics Journal*, vol. 37, no. 5, pp. 433–451, May 2006.
- Bol2014 D. Bol, G. de Streel, F. Botman, A. K. Lusala, and N. Couniot, “A 65-nm 0.5-V 17-pJ/frame. pixel DPS CMOS image sensor for ultra-low-power SoCs achieving 40-dB dynamic range,” in *VLSI Circuits Digest of Technical Papers, 2014 Symposium on*, 2014, pp. 1–2.
- Brouk2007 I. Brouk, K. Alameh, and Y. Nemirovsky, “Design and Characterization of CMOS/SOI Image Sensors,” *IEEE Transactions on Electron Devices*, vol. 54, no. 3, pp. 468–475, Mar. 2007.
- Brunetti2016 A. M. Brunetti and B. Choubey, “A low dark current wide dynamic range CMOS pixel,” in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2016, pp. 2523–2526.
- Canny1986 J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- Chae2011 Y. Chae *et al.*, “A 2.1 M Pixels, 120 Frame/s CMOS Image Sensor With Column-Parallel  $\Sigma\Delta$  ADC Architecture,” *IEEE*

- 
- Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 236–247, Jan. 2011.
- Chen2011 D. G. Chen, D. Matolin, A. Bermak, and C. Posch, “Pulse-Modulation Imaging: Review and Performance Analysis,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 1, pp. 64–82, Feb. 2011.
- Choi2012 J. Choi, S. Park, J. Cho, and E. Yoon, “A 1.36  $\mu$ W adaptive CMOS image sensor with reconfigurable modes of operation from available energy/illumination for distributed wireless sensor network,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, pp. 112–114.
- Choi2013 J. Choi, S. Park, J. Cho, and E. Yoon, “A 3.4  $\mu$ W CMOS image sensor with embedded feature-extraction algorithm for motion-triggered object-of-interest imaging,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, 2013, pp. 478–479.
- Choi2015 J. Choi, S. Park, J. Cho, and E. Yoon, “An Energy/Illumination-Adaptive CMOS Image Sensor With Reconfigurable Modes of Operations,” *IEEE Journal of Solid-State Circuits*, vol. 50, no. 6, pp. 1438–1450, Jun. 2015.
- Choi2016 J. Choi, J. Shin, D. Kang, and D.-S. Park, “Always-On CMOS Image Sensor for Mobile and Wearable Devices,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 130–140, Jan. 2016.
- Chou2012 W.-F. Chou, S.-F. Yeh, and C.-C. Hsieh, “A 143dB 1.96% FPN linear-logarithmic CMOS image sensor with threshold-voltage cancellation and tunable linear range,” in *Sensors, 2012 IEEE*, 2012, pp. 1–4.
- Chung2012 M.-T. Chung and C.-C. Hsieh, “A 0.5 V 4.95  $\mu$ W 11.8 fps PWM CMOS imager with 82dB dynamic range and 0.055% fixed-pattern noise,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, pp. 114–116.
- Costas-Santos2007 J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco, “A Spatial Contrast Retina With On-Chip Calibration for Neuromorphic Spike-Based AER Vision Systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 7, pp. 1444–1458, Jul. 2007.



- Cottini2013 N. Cottini, M. Gottardi, N. Massari, R. Passerone, and Z. Smilansky, “A  $33\ \mu\text{m} \times 64$  Pixel Vision Sensor Embedding Robust Dynamic Background Subtraction for Event Detection and Scene Interpretation,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 850–863, Mar. 2013.
- Crooks2009 J. P. Crooks *et al.*, “A CMOS Image Sensor With In-Pixel ADC, Timestamp, and Sparse Readout,” *IEEE Sensors Journal*, vol. 9, no. 1, pp. 20–28, Jan. 2009.
- Culurciello2003 E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, “A biomorphic digital image sensor,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.
- Dalal2005 N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.
- Darwish2015 A. Darwish, L. Fesquet, and G. Sicard, “RTL simulation of an asynchronous reading architecture for an event-driven image sensor,” in *Event-based Control, Communication, and Signal Processing (ECCSP), 2015 International Conference on*, 2015, pp. 1–4.
- Darwish2016 A. Darwish, “Capteur d’images événementiel, asynchrone à échantillonnage non-uniforme,” Université Grenoble Alpes, 2016.
- Dehaene2016 D. Dehaene, “Time code image sensor”, U.S. Patent 0 323 530 (A1) , Nov. 03, 2016
- ElGamal2005 A. El Gamal and H. Eltoukhy, “CMOS image sensors,” *IEEE Circuits and Devices Magazine*, vol. 21, no. 3, pp. 6–20, 2005.
- Elouardi2007 A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.-O. Klein, and R. Reynaud, “Image Processing Vision Systems: Standard Image Sensors Versus Retinas,” *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1675–1687, Oct. 2007.
- Gottardi2009 M. Gottardi, N. Massari, and S. A. Jawed, “A  $100\ \mu\text{m} \times 128 \times 64$  Pixels Contrast-Based Asynchronous Binary Vision Sensor for Sensor Networks Applications,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 5, pp. 1582–1592, May 2009.

- Gruev2010 V. Gruev, Z. Yang, J. Van der Spiegel, and R. Etienne-Cummings, "Current mode image sensor with two transistors per pixel," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1154–1165, 2010.
- Guidash1997 R. M. Guidash *et al.*, "A 0.6/spl mu/m CMOS pinned photodiode color imager technology," in *Electron Devices Meeting, 1997. IEDM'97. Technical Digest., International*, 1997, pp. 927–929.
- Guo2007 X. Guo, X. Qi, and J. G. Harris, "A Time-to-First-Spike CMOS Image Sensor," *IEEE Sensors Journal*, vol. 7, no. 8, pp. 1165–1175, Aug. 2007.
- Hanson2010 S. Hanson, Z. Foo, D. Blaauw, and D. Sylvester, "A 0.5 V Sub-Microwatt CMOS Image Sensor With Pulse-Width Modulation Read-Out," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 759–767, Apr. 2010.
- Hartmann2012 J. Hartmann "Planar FD-SOI Technology at 28nm and below for extremely power-efficient SoCs." in the presentation at Symposium Fully Depleted Transistors Technology - 10 december 2012, San Francisco, CA
- Ieng2014 S.-H. Ieng, C. Posch, and R. Benosman, "Asynchronous Neuromorphic Event-Driven Image Filtering," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1485–1499, Oct. 2014.
- Jendernalik2013 W. Jendernalik, G. Blakiewicz, A. Handkiewicz, and M. Melosik, "Analogue CMOS ASICs in Image Processing Systems," *Metrology and Measurement Systems*, vol. 20, no. 4, Jan. 2013.
- Jeon2017 B.-K. Jeon, S.-K. Hong, and O.-K. Kwon, "A Low-Power 12-bit Extended Counting ADC Without Calibration for CMOS Image Sensors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2017.
- Kadura2016 L. Kadura *et al.*, "Extending the functionality of FDSOI N- and P-FETs to light sensing," in *Electron Devices Meeting (IEDM), 2016 IEEE International*, 2016, pp. 32–6.
- Kim2010 D. Kim and E. Culurciello, "A compact-pixel tri-mode vision sensor," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, 2010, pp. 2434–2437.
- Kim2013 G. Kim *et al.*, "A 467nW CMOS visual motion sensor with temporal averaging and pixel aggregation," in *Solid-State*

- Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International, 2013, pp. 480–481.*
- Kim2017 H.-J. Kim, S.-I. Hwang, J.-H. Chung, J.-H. Park, and S.-T. Ryu, “A Dual-Imaging Speed-Enhanced CMOS Image Sensor for Real-Time Edge Image Extraction,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 9, pp. 2488–2497, Sep. 2017.
- Kitchen2005 A. Kitchen, A. Bermak, and A. Bouzerdoum, “A Digital Pixel Sensor Array With Programmable Dynamic Range,” *IEEE Transactions on Electron Devices*, vol. 52, no. 12, pp. 2591–2601, Dec. 2005.
- Lazaveric2006 N. Lazarevic-McManus, J. Renno, D. Makris, and G. A. Jones, “Designing evaluation methodologies: the case of motion detection,” in *Proceedings of 9th IEEE International Workshop on PETS*, 2006, pp. 23–30.
- Lee2003 Jonghwan Lee, G. Bosman, K. R. Green, and D. Ladwig, “Noise model of gate-leakage current in ultrathin oxide MOSFETs,” *IEEE Transactions on Electron Devices*, vol. 50, no. 12, pp. 2499–2506, Dec. 2003.
- LeneroBardallo2014 J. A. Leñero-Bardallo, J. Fernández-Berni, and Á. Rodríguez-Vázquez, “Review of ADCs for imaging,” in *IS&T/SPIE Electronic Imaging*, 2014, p. 18.
- LeneroBardallo2016 J. A. Lenero-Bardallo, P. Hafliger, R. Carmona-Galan, and A. Rodriguez-Vazquez, “A Bio-Inspired Vision Sensor With Dual Operation and Readout Modes,” *IEEE Sensors Journal*, vol. 16, no. 2, pp. 317–330, Jan. 2016.
- Lichtsteiner2006 P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change,” in *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, 2006, pp. 2060–2069.
- Lichtsteiner2008 P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128 $\times$ 128 120 dB 15  $\mu$ s Latency Asynchronous Temporal Contrast Vision Sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- Lim2010 Y. Lim *et al.*, “A 1.1 e-temporal noise 1/3.2-inch 8Mpixel CMOS image sensor using pseudo-multiple sampling,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 396–397.

- Linn2011 A. M. T. Linn, C. Shoushun, Y. K. Seng, and others, “Adaptive priority toggle asynchronous tree arbiter for AER-based image sensor,” in *VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on*, 2011, pp. 66–71.
- LIOT2015 <http://electroiq.com/blog/2015/07/internet-of-things-platform-silicon-impulse-energy-efficiency-and-consumer-applications-focus-of-letidays-discussion/>
- Liu2001 M.-H. Liu and S.-I. Liu, “An 8-bit 10 MS/s folding and interpolating ADC using the continuous-time auto-zero technique,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 1, pp. 122–128, 2001.
- Liu2014 X. Liu, M. Zhang, and J. Van der Spiegel, “A Low-Power Multifunctional CMOS Sensor Node for an Electronic Facade,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 9, pp. 2550–2559, Sep. 2014.
- Liu2016 H. Liu, D. P. Moeys, G. Das, D. Neil, S.-C. Liu, and T. Delbrück, “Combined frame-and event-based detection and tracking,” in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2016, pp. 2511–2514.
- Mallik2005 U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, “Temporal change threshold detection imager,” in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, 2005, pp. 362–603.
- Manghisoni2007 M. Manghisoni, L. Gaioni, L. Ratti, V. Re, V. Speziali, and G. Traversi, “Impact of gate-leakage current noise in sub-100 nm CMOS front-end electronics,” in *Nuclear Science Symposium Conference Record, 2007. NSS’07. IEEE*, 2007, vol. 4, pp. 2503–2508.
- Martineau2016 B. Martineau, C. Jany, F. Todeschini, D. Morche, and E. Mercier, “Towards fully integrated 28nm UTBB FD-SOI IoT node: The sub-50 $\mu$ W RF receiver,” in *SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2016 IEEE*, 2016, pp. 1–2.
- Massari2005 N. Massari, M. Gottardi, L. Gonzo, D. Stoppa, and A. Simoni, “A CMOS Image Sensor With Programmable Pixel-Level Analog Processing,” *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1673–1684, Nov. 2005.
- Massari2007 N. Massari, S. A. Jawed, and M. Gottardi, “A collision-free time-to-first spike camera architecture based on a winner-take-all network,” in *Circuit Theory and Design, 2007*.

- ECCTD 2007. 18th European Conference on*, 2007, pp. 950–953.
- Matsuo2008 S. Matsuo *et al.*, “A very low column FPN and row temporal noise 8.9 M-pixel, 60 fps CMOS image sensor with 14bit column parallel SA-ADC,” in *VLSI Circuits, 2008 IEEE Symposium on*, 2008, pp. 138–139.
- McIlrath2001 L. G. McIlrath, “A low-power low-noise ultrawide-dynamic-range CMOS imager with pixel-parallel A/D conversion,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 5, pp. 846–853, 2001.
- Nakamura1997 J. Nakamura, B. Pain, T. Nomoto, T. Nakamura, and E. R. Fossum, “On-focal-plane signal processing for current-mode active pixel sensors,” *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1747–1758, 1997.
- Njuguna2012 R. Njuguna and V. Gruev, “Low Power Programmable Current Mode Computational Imaging Sensor,” *IEEE Sensors Journal*, vol. 12, no. 4, pp. 727–736, Apr. 2012.
- Njuguna2014 R. Njuguna and V. Gruev, “Current-Mode CMOS Imaging Sensor With Velocity Saturation Mode of Operation and Feedback Mechanism,” *IEEE Sensors Journal*, vol. 14, no. 3, pp. 710–721, Mar. 2014.
- Ohmaru2016 T. Ohmaru *et al.*, “A 25.3  $\mu$ W at 60 fps  $240 \times 160$  Pixel Vision Sensor for Motion Capturing With In-Pixel Nonvolatile Analog Memory Using CAAC-IGZO FET,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2168–2179, Sep. 2016.
- Ohta2008 Jun Ohta, *Smart CMOS Image Sensors and Applications*, 2008, CRC Press 2007 ISBN: 978-0-8493-3681-2
- Panovic2006 M. Panovic and A. Demosthenous, “A Low-Power Analog Motion Estimation Processor for Digital Video Coding,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 673–683, Mar. 2006.
- Park2009 J.-H. Park, S. Aoyama, T. Watanabe, K. Isobe, and S. Kawahito, “A High-Speed Low-Noise CMOS Image Sensor With 13-b Column-Parallel Single-Ended Cyclic ADCs,” *IEEE Transactions on Electron Devices*, vol. 56, no. 11, pp. 2414–2422, Nov. 2009.
- Pelloux-Prayer2014 B. Pelloux-Prayer, “Optimisation de l’efficacité énergétique des applications numériques en technologie FD-SOI 28-14nm,” Grenoble, 2014.

- Philip2007 R. M. Philipp, D. Orr, V. Gruev, J. Van der Spiegel, and R. Etienne-Cummings, "Linear current-mode active pixel sensor," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 11, pp. 2482–2491, 2007.
- Posch2011 C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.
- Rebecq2016 H. Rebecq, G. Gallego, and D. Scaramuzza, "EMVS: Event-based multi-view stereo," in *British Machine Vision Conference (BMVC)*, 2016.
- Sakurai2006 T. Sakurai, A. Matsuzawa, and T. Douseki, *Fully-depleted SOI CMOS circuits and technology for ultra-low power applications*. Dordrecht, The Netherlands: Springer, 2006.
- Schmitz2017 J. A. Schmitz, M. K. Gharzai, S. Balkir, M. W. Hoffman, D. J. White, and N. Schemm, "A 1000 frames/s Vision Chip Using Scalable Pixel-Neighborhood-Level Parallel Processing," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 556–568, Feb. 2017.
- Scholkopf2002 B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.
- Shoushun2007 C. Shoushun and A. Bermak, "Arbitrated Time-to-First Spike CMOS Image Sensor With On-Chip Histogram Equalization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 346–357, Mar. 2007.
- Shoushun2008 C. Shoushun, F. Boussaid, and A. Bermak, "Robust Intermediate Read-Out for Deep Submicron Technology CMOS Image Sensors," *IEEE Sensors Journal*, vol. 8, no. 3, pp. 286–294, 2008.
- Sparso2001 J. Sparsø and S. Furber, Eds., *Principles of Asynchronous Circuit Design*. Boston, MA: Springer US, 2001.
- Sukegawa2013 S. Sukegawa *et al.*, "A 1/4-inch 8Mpixel back-illuminated stacked CMOS image sensor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, 2013, pp. 484–485.

- Suleiman2016 A. Suleiman and V. Sze, “An Energy-Efficient Hardware Implementation of HOG-Based Object Detection at 1080HD 60 fps with Multi-Scale Support,” *Journal of Signal Processing Systems*, vol. 84, no. 3, pp. 325–337, Sep. 2016.
- Takahashi2017 T. Takahashi *et al.*, “A 4.1 Mpix 280fps stacked CMOS image sensor with array-parallel ADC architecture for region control,” in *VLSI Circuits, 2017 Symposium on*, 2017, pp. C244–C245.
- Tang2016 F. Tang, B. Wang, A. Bermak, X. Zhou, S. Hu, and X. He, “A Column-Parallel Inverter-Based Cyclic ADC for CMOS Image Sensor With Capacitance and Clock Scaling,” *IEEE Transactions on Electron Devices*, vol. 63, no. 1, pp. 162–167, Jan. 2016.
- Tsai2012 S.-J. Tsai, Y.-C. Chen, C.-C. Hsieh, W.-H. Chang, H.-H. Tsai, and C.-F. Chiu, “A column-parallel SA ADC with linearity calibration for CMOS imagers,” in *Sensors, 2012 IEEE*, 2012, pp. 1–4.
- Xu2014 J. Xu, D. Li, and S. Yao, “A Time Error Correction Method Applied to High-Precision AER Asynchronous CMOS Image Sensor,” *Journal of Signal Processing Systems*, vol. 75, no. 1, pp. 1–13, Apr. 2014.
- Yadid-Pecht2004 O. Yadid-Pecht, Tutorial on CMOS Sensors, and International Symposium on Circuits and Systems, Eds., *CMOS imagers: from phototransduction to image processing*. Boston: Kluwer Academic, 2004.
- Yamazaki2017 T. Yamazaki *et al.*, “4.9 A 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing,” in *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*, 2017, pp. 82–83.
- Yang1994 W. Yang, “A wide-dynamic-range, low-power photosensor array,” in *Solid-State Circuits Conference, 1994. Digest of Technical Papers. 41st ISSCC., 1994 IEEE International*, 1994, pp. 230–231.
- Yin2016 C. Yin, C.-F. Chiu, and C.-C. Hsieh, “A 0.5 V, 14.28-kframes/s, 96.7-dB Smart Image Sensor With Array-Level Image Signal Processing for IoT Applications,” *IEEE Transactions on Electron Devices*, vol. 63, no. 3, pp. 1134–1140, Mar. 2016.
- Zhu2014 Hongbo Zhu and T. Shibata, “A Real-Time Motion-Feature-Extraction VLSI Employing Digital-Pixel-Sensor-Based Parallel Architecture,” *IEEE Transactions on Circuits and*

*Systems for Video Technology*, vol. 24, no. 10, pp. 1787–1799,  
Oct. 2014.



## A. Annexe 1 : Architecture IM-TTFS

Logigramme du pixel :

Un logigramme du pixel IM-TTFS a été réalisé afin d'illustrer l'acquisition avec le mécanisme d'inhibition.

### Architecture du Handshake

Lorsqu'un pixel flash et que l'inhibition n'est pas active dans son bloc, le signal RowRequest\_b commun à toute la ligne est mis à 0. Dans la matrice, plusieurs RowRequest\_b peuvent être mis à 0 simultanément, un arbitrage est donc nécessaire afin de choisir quelle requête traiter en premier. Cela se fait à l'aide d'un arbre binaire d'arbitrage. A chacune de ses feuilles, Figure A-1, l'arbre va gérer un conflit entre deux requêtes, et acquitter l'une des deux. Par exemple, à sa racine, l'arbre choisira d'acquitter une des deux moitiés de l'arbre. L'acquiescement descend ainsi le long de l'arbre, l'unique signal de ligne à acquiescer sera au bout d'une feuille. Chaque nœud de cet arbre est composé d'une bascule RS, d'une porte AND et de deux portes NOR. La porte AND sert à transmettre à l'étage supérieur de l'arbre si un des deux signaux à traiter au niveau du nœud est à 0. Une porte logique OR est utilisée pour des signaux à logique basse. La bascule RS sert à implémenter l'arbitrage au niveau local. Elle est déséquilibrée en augmentant la taille de l'une des entrées de la porte NOR. De cette manière, si les deux signaux de requête arrivent en même temps l'un des deux aura la priorité. Cette bascule retient la priorité entre les deux signaux. Enfin, les portes NOR génèrent le signal d'acquiescement pour le niveau de dessous. Ces portes prennent en entrée le signal d'acquiescement du niveau supérieur, la priorité contenue dans la bascule RS, et le signal de requête correspondant. Prendre directement le signal de requête correspondant permet de gagner du temps lors de la désactivation de ce signal. Dans le cas contraire, ce signal devrait remonter puis redescendre l'arbre. Ainsi, lorsqu'au moins une des lignes génère une requête, le signal de requête va se propager par la porte AND jusqu'à la racine de l'arbre. La racine va générer le signal d'acquiescement qui va redescendre jusqu'aux feuilles, en choisissant à chaque embranchement une des deux directions en fonction de la priorité contenue dans la bascule RS.

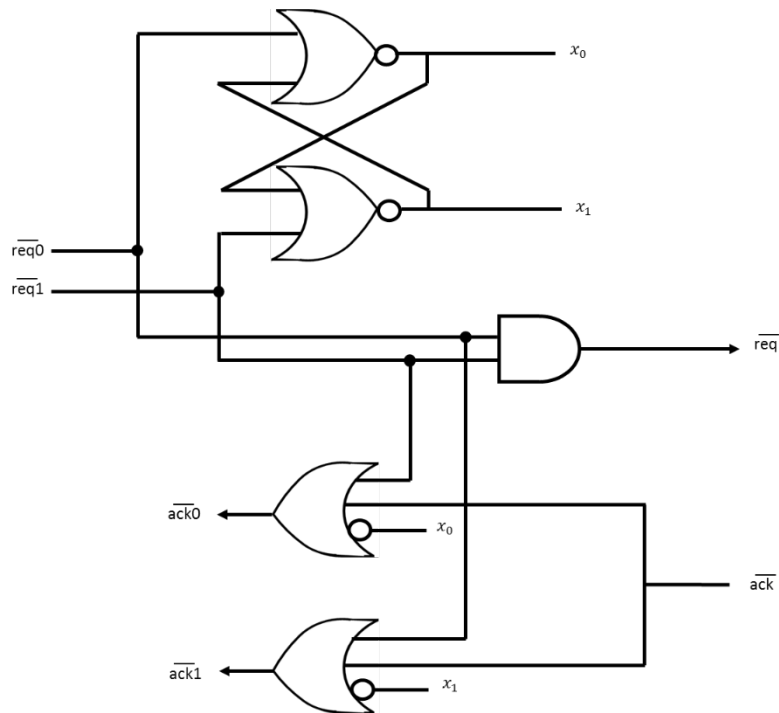


Figure A-1: Feuille de l'arbre d'arbitrage

Lorsqu'une ligne reçoit le signal RowAck, les pixels ayant flashé et non acquités de cette ligne descendent le signal ColRequest\_b à 0, ce signal est unique à chaque colonne. Afin de pouvoir s'affranchir d'une contrainte de temps sur la remontée à 1 de ColRequest\_b, ColRequest\_b met directement ColAck à 1, ce qui éteint le pixel et on peut alors remonter immédiatement ColRequest\_b à 1. Les ColAck à 1 sont enregistrés pour être traités en parallèle, et ne seront mis à 0 que lorsqu'ils seront écrits sur le bus de sortie.

## Architecture de l'inhibition

L'inhibition pour l'architecture IM-TTFS, doit durer un temps fixe, et l'inhibition ne se déclenche qu'une seule fois par trame. Il faut donc avoir une mémoire 1 bit pour retenir si l'inhibition a eu lieu pendant la trame actuelle. Afin d'implémenter un temps d'inhibition court et constant sans utiliser un grand nombre de transistors, les propriétés de fuite de grille des transistors en FDSOI 28nm ont été utilisées. L'architecture des signaux d'inhibition au niveau des blocs est décrite par la Figure AA-2. Block\_en est le signal qui garde en mémoire si une inhibition a déjà eu lieu au cours de la trame actuelle. Il empêche la mise à zéro du signal Inhib\_b si une inhibition a déjà eu lieu. Lorsqu'un pixel du bloc est acquitté, son signal Shutdown (SD) est mis à 1 afin d'éteindre le pixel, ce signal est utilisé au niveau du bloc pour mettre Inhib\_b à 0, et Inhib à 1. La valeur de Block\_en s'inverse, ce qui a pour conséquence de laisser Inhib en haute impédance. Inhib\_b n'est plus utilisé, et l'inverseur n'est plus alimenté donc il ne consomme plus. Inhib est connecté à la grille d'un transistor où source et drain sont à 0. Les fuites de grille vont le descendre à 0 en quelques centaines de nanosecondes, ces fuites permettent de définir le temps d'inhibition. Le temps

d'inhibition peut néanmoins être réglé dans une certaine mesure par la taille du transistor de fuite d'Inhib, ce qui permet aussi d'éviter du mismatch dans le temps d'inhibition.

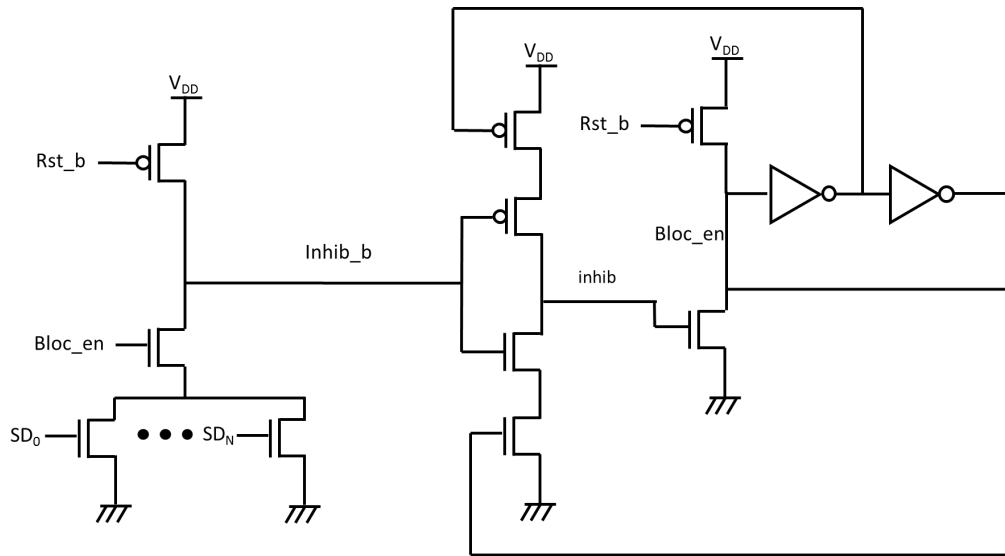


Figure AA-2: Architecture de l'inhibition au niveau du bloc



## Résumé

---

Le sujet de thèse a pour objectif de revisiter le paradigme d'acquisition d'images dans les circuits intégrés pour le rendre robuste et scalable en technologies nanométriques (telles que le 28nm FDSOI). Ceci, afin de répondre aux contraintes d'imagerie imposée par des applications de type internet des objets. Dans ce cas, un système sur puce (SoC) hétérogène conçu en technologie avancée permettrait de répondre aux contraintes de consommation d'énergie. L'utilisation des imageurs standard actuels n'est alors pas compatible avec cette exigence à cause de leur consommation excessive et leur non compatibilité avec les technologies FDSOI 28nm. De plus, les ressources importantes de calcul numérique disponibles dans ces types de SoC couplées avec de nouveaux modes de captures d'images permettraient d'atteindre des niveaux de consommation d'énergie extrêmement bas tout en offrant la possibilité d'implémenter des algorithmes de traitement d'image complexes. Après une étude bibliographique sur les différentes méthodes d'acquisition d'image ainsi qu'une étude bibliographique sur les imageurs en technologies dites avancées pour l'imagerie et pour des applications basse consommation, il a été montré qu'il était nécessaire de numériser au plus tôt l'information lumineuse reçue par le capteur. C'est pourquoi le sujet a été orienté vers une architecture de type événementielle. L'architecture d'un capteur d'image événementiel avec traitement intelligent associé a été développée, en prenant en considération les contraintes liées à la technologie. Afin de définir ces contraintes, un circuit de test de pixel en FDSOI 28nm a été réalisé permettant d'évaluer la réponse électro-optique. Les pixels ont chacun des types et des tailles de photodiodes différentes afin de valider le type et la taille les plus efficaces. Deux architectures événementielles ont été étudiées durant cette thèse afin de répondre aux contraintes d'une implémentation en technologies FDSOI 28nm : une architecture de type « Time-to-first-Spike » (TTFS) avec un système d'inhibition et une architecture dite « multi-bus » utilisant les possibilités d'interconnexions denses offertes par la technologie. Ces deux architectures visent à réduire le flot de données sortant ainsi que la consommation d'énergie. Les traitements associés à l'acquisition ont été validés par des simulations MATLAB émulant l'acquisition événementielle et les prétraitements. Ce système de vision extrait donc une carte binaire correspondant aux contrastes locaux en utilisant un principe d'inhibition par bloc. Cette architecture de traitement est basée sur le pixel TTFS (et son principe d'inhibition) en adaptant son implémentation. La carte binaire est extraite de manière synchrone ce qui permet d'éviter l'ajout de matériel lié à une implémentation purement événementielle. Cette carte binaire peut servir dans des applications telles que de la détection de mouvement, ou de la classification telles que la méthode des histogrammes des gradients (HoG) le permet. La carte binaire extraite se rapproche des motifs binaires locaux (LBP) qui sont des outils fréquemment utilisés dans la détection et la reconnaissance de visage. Une partie de la thèse a également été consacrée à l'exploitation des possibilités qu'offre la technologie FDSOI 28nm. Notamment des architectures pixels utilisant une photodiode sous le transistor ont été étudiées. Il a également été développé dix matrices de 3 par 3 pixels en intégration 3D séquentielle utilisant la technologie CoolCube™ du LETI.

## Abstract

---

The goal of this thesis is to study new image acquisition paradigm in integrated vision circuits to enhance their robustness and scalability using nanometric technologies (such as the 28nm FDSOI) in order to satisfy the imaging constraints imposed by applications such as Internet of Things. In this case, a heterogeneous system-on-chip (SoC) designed in advanced technology would meet the energy consumption constraints. Using standard imagers is not compatible with this requirement because of their excessive power consumption and their architectures non-compatible with 28nm FDSOI technologies. In addition, in these SoC, significant available digital computational resources coupled with new image acquisition modes would allow ultra-low power consumption while providing the ability to implement complex image processing. After a bibliographic study on the state of the art on image acquisition methods and a study on imagers designed with advanced technologies and on low-power applications, it has been shown that it is necessary to quickly digitize light information received by the sensor (i.e. in the pixel). This is why the subject has been oriented towards an event-based vision sensor architecture. The architecture of an event-based image sensor with its associated smart processing has been developed, taking into account technology constraints. In order to define these constraints, a 28nm FDSOI pixel test circuit has been carried out to evaluate the electro-optical response. Each pixel has a different type and size of photodiodes in order to validate the most effective type and size. Two event-based architectures were studied during this thesis in order to fit with the constraints of an implementation in 28nm FDSOI technologies: a "Time-to-first-Spike" (TTFS) architecture with an inhibition system and an architecture called "multi-bus" using the dense interconnections possibilities offered by the technology. These two architectures aim to reduce the data throughput as well as energy consumption. The processing associated to the acquisition have been validated by MATLAB simulations emulating the event acquisition and pre-processing. This vision system therefore extracts a binary map corresponding to the local contrasts using block inhibition mechanism. This processing architecture is based on TTFS pixel (and its inhibition mechanism) with a dedicated pixel schematic. The binary map is extracted in a synchronous manner, thus avoiding hardware addition inherent to an AER (Adress Event Representation) implementation. This binary map can be used for applications such as motion detection, or classification such as histogram of gradient method (HoG). This extracted binary map approaches local binary patterns (LBP), which are frequently used tools in face detection and recognition. A part of this thesis has been dedicated also to the exploration of FDSOI 28nm capabilities in terms of pixel implementation. Notably, by studying pixels using a photodiode under the FDSOI transistor. It has also been developed ten 3 by 3 pixels matrices using 3D integration with LETI technology CoolCube™.