

Titre: Capteur d'images CMOS à pixels numériques et à gamme
Title: dynamique élevée

Auteur: Jean-Luc Trépanier
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Trépanier, J.-L. (2003). Capteur d'images CMOS à pixels numériques et à gamme
Citation: dynamique élevée [Mémoire de maîtrise, École Polytechnique de Montréal].
PolyPublie. <https://publications.polymtl.ca/7306/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7306/>
PolyPublie URL:

**Directeurs de
recherche:** Mohamad Sawan, & Yves Audet
Advisors:

Programme: Non spécifié
Program:

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE MONTRÉAL

**CAPTEUR D'IMAGES CMOS À PIXELS NUMÉRIQUES
ET À GAMME DYNAMIQUE ÉLEVÉE**

JEAN-LUC TRÉPANIÉ

**DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL**

**MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU
DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES**

GÉNIE ÉLECTRIQUE

AOÛT 2003



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitons et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 0-612-90862-3

Our file *Notre référence*

ISBN: 0-612-90862-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

CAPTEUR D'IMAGES CMOS À PIXELS NUMÉRIQUES

ET À GAMME DYNAMIQUE ÉLEVÉE

présenté par: TRÉPANIÉR Jean-Luc

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. KHOUS Abdelhakim, Ph.D., président

M. SAWAN Mohamad, Ph.D., directeur de recherche

M. AUDET Yves, D.Sc.A., codirecteur de recherche

M. MEINGAN Remi, Ph.D., membre

Remerciements

Je souhaiterais remercier mon directeur de recherche, Monsieur Mohamad Sawan et mon codirecteur, Monsieur Yves Audet, professeurs à l'École Polytechnique de Montréal, qui m'ont accueilli dans leur équipe de recherche et m'ont permis d'effectuer mes recherches dans d'excellentes conditions.

J'adresse également mes remerciements à Monsieur Abdelhakim Khouas, professeur à l'École Polytechnique de Montréal, et à Monsieur Remi Meingan, ingénieur chez DALSA Semiconducteurs, pour avoir accepté de participer au jury d'examen de ce mémoire.

Je tiens à saluer mes collègues de l'équipe de recherche PolyStim pour leur bonne compagnie et pour leur aide. En particulier, j'aimerais remercier Christian Fayomi pour m'avoir donné de bons points de départ pour la réalisation de ce projet ainsi que Jonathan Coulombe pour son aide et ses judicieux conseils.

Je ne saurais oublier ma grande copine, Isabelle Rousseau, ainsi que ma famille et amis proches pour leur soutien moral inconditionnel, et pour leurs encouragements tout au long de mes études.

Enfin, mes remerciements vont à la CMC, FCAR et le CRSNG pour leur soutien financier et technique.

Résumé

La technologie à charges couplées (CCD) a été depuis longtemps la technologie de choix pour l'imagerie numérique haute qualité. Cependant, les désavantages liés à la forte consommation de puissance et l'impossibilité d'ajouter d'autres modules de traitement sur la même puce sont des caractéristiques des CCD. Avec l'échelle de technologie réduite du CMOS, il devient possible d'ajouter des modules de contrôle et traitement sur la même puce afin d'obtenir une caméra complètement intégrée sur une seule puce.

L'objectif de ce mémoire est de développer et valider une nouvelle architecture de capteur d'images numérique. Cette architecture doit permettre une flexibilité de façon à utiliser le bloc de base, le pixel numérique, afin de construire des capteurs de différentes résolutions selon l'application visée. De plus, on doit pouvoir contrôler numériquement la matrice photosensible de manière à optimiser la vitesse de capture ou la précision de la représentation numérique. L'architecture doit aussi permettre d'aller chercher le maximum de la gamme dynamique disponible afin de représenter le mieux possible celle de l'œil humain. De cette façon, le capteur doit fonctionner aussi bien dans la lumière intense du soleil que dans l'ombre de la nuit sans aucun ajustement manuel ou intervention externe.

Une nouvelle architecture répondant aux attentes du projet a été conçue et implémentée dans une puce de silicium à l'aide de la technologie CMOS 0.18 μ m. Des pixels numériques sont à la base de cette nouvelle architecture. Ces pixels peuvent être utilisés afin de construire des matrices de tailles arbitraires sans engendrer de modifications majeures des modules de lecture. Une plaquette de test, un circuit discret ainsi qu'un contrôleur numérique ont été conçus afin de valider le fonctionnement des modules présents dans cette nouvelle architecture.

Les observations du fonctionnement de la puce et du système ont mené à des propositions d'améliorations pour les itérations subséquentes de ce capteur d'images numériques.

Abstract

Charge coupled devices (CCD) technology has been for a long time the technology of choice in high quality image sensing. Drawbacks are high power consumption and no possible on-chip processing capabilities. With CMOS reduced feature size technology, it becomes possible to add on-chip control and processing units in order to obtain a fully integrated camera on a single chip.

The main objective of this thesis is to develop and validate a new digital image sensor architecture. This architecture must provide a flexible design in order to use the basic building block, the digital pixel, to implement different resolution sensors depending on the target application. It also must be possible to digitally control the sensor to optimize either the speed of the capture or the precision of the digital representation. The resulting architecture must be able to measure the maximum of the available dynamic range in order to represent the capabilities of the human eye. This implies that the sensor must be able to render good images in bright sunlight as well as in dark shadows without any manual adjustments or external interventions.

A new architecture meeting the project goals has been designed and implemented in a CMOS 0.18 μm silicon chip. Digital pixels are the basic blocs of this new architecture. These pixels can be used to implement any arbitrary size sensor matrix without involving any major modification in the readout module. A test bench, a discrete component circuit and a digital controller have been implemented to validate the functionality of all the modules of this new sensor architecture.

Observations from the experiments conducted on the chip as led to proposals of enhancement for further versions of this digital image sensor.

Table des matières

REMERCIEMENTS	IV
RÉSUMÉ	V
ABSTRACT	VII
TABLE DES MATIÈRES	VIII
LISTE DES FIGURES	XII
LISTE DES TABLEAUX	XV
DÉFINITIONS ET ACRONYMES	XVI
LISTE DES ANNEXES	XVIII
INTRODUCTION	1
CHAPITRE 1 ÉTUDE DES CAPTEURS D'IMAGES	4
1.1 INTRODUCTION	4
1.2 CAPTEURS D'IMAGES À BASE DE SEMI-CONDUCTEURS	5
1.2.1 Caractéristiques des éléments photosensibles	7
1.2.2 Performance des capteurs CMOS et CCD.....	9
1.3 ÉLÉMENTS PHOTOSENSIBLES	10
1.3.1 Photodiode standard	11
1.3.2 Photodiode à couche de semi-conducteur intrinsèque.....	12
1.3.3 Phototransistor	13
1.3.4 Photo-grille	14
1.3.5 Photodiode à silicium amorphe	15
1.3.6 Avenir des capteurs d'images CMOS	15

1.4	CONSTITUTION DU CIRCUIT D'UN PIXEL.....	16
1.4.1	Types de pixels CMOS.....	17
1.4.2	Pixels à circuiterie numérique.....	18
1.5	MATRICE DE PIXELS.....	20
1.5.1	Sources d'erreur dans une matrice de pixels.....	21
1.5.2	Matrice pour la capture d'images couleur.....	22
1.5.3	Perception des images par l'œil humain.....	24
1.6	CONVERSION ANALOGIQUE À NUMÉRIQUE.....	25
1.6.1	Consommation de puissance des convertisseurs A/N.....	27
1.7	GAMME DYNAMIQUE DES CAPTEURS D'IMAGES.....	27
1.7.1	Intrascène versus interscène.....	28
1.7.2	Méthodes possibles pour élargir la gamme dynamique.....	29
1.8	COMPARAISON DE CAPTEURS D'IMAGES.....	30
1.8.1	Capture rapide et pixel numérique de la référence [27].....	32
1.8.2	Conversion analogique à numérique de la référence [46].....	32
1.9	CONCLUSION.....	34
CHAPITRE 2 ARCHITECTURE DU CAPTEUR D'IMAGES.....		35
2.1	INTRODUCTION.....	35
2.2	ARCHITECTURE DU CAPTEUR.....	36
2.2.1	Conversions A/N et lecture des pixels de la matrice.....	36
2.2.2	Expositions multiples.....	38
2.2.3	Lecture rapide.....	46
2.3	ARCHITECTURE DU PIXEL NUMÉRIQUE.....	49
2.3.1	Obturateur et suiveur de la photodiode.....	50
2.3.2	Loquet pour l'obturateur.....	51
2.3.3	Comparateur intérieur au pixel.....	52
2.4	CONCLUSION.....	53
CHAPITRE 3 DESIGN ET CONTRÔLE DU CAPTEUR D'IMAGES.....		55

3.1	INTRODUCTION	55
3.2	DESIGN DU PIXEL NUMÉRIQUE	56
3.2.1	Conversion de l'énergie photonique en énergie électrique	56
3.2.2	Comparateur dynamique.....	57
3.2.3	Design des obturateurs et loquet.....	60
3.2.4	Loquet pour l'obturateur électronique.....	62
3.2.5	Dessin des masques du pixel	64
3.3	CONCEPTION DE LA PUCE.....	66
3.3.1	Matrice de pixels	67
3.3.2	Design des mémoires.....	68
3.3.3	Module de test	70
3.3.4	Interconnexions des blocs et puce complète.....	71
3.4	SYSTÈME DE TEST	73
3.4.1	Circuits de support.....	74
3.4.2	Contrôle numérique	80
3.5	CONCLUSION	82
CHAPITRE 4 MESURES ET RÉSULTATS.....		84
4.1	INTRODUCTION.....	84
4.2	PERFORMANCES DU CAPTEUR.....	84
4.2.1	Paramètres photoélectriques du capteur	86
4.2.2	Performances du suiveur dynamique.....	88
4.2.3	Précision du comparateur	91
4.2.4	Gamme dynamique du capteur	97
4.2.5	Transfert des données	105
4.2.6	Performances de l'architecture proposée.....	108
4.3	DEVELOPPEMENTS FUTURS.....	109
4.3.1	Amélioration potentielle de la qualité des images en CMOS.....	110
4.3.2	Pixel numérique amélioré.....	114
4.3.3	Intégration du capteur dans le cadre du projet d'implant visuel.....	117

4.3.4	Méthodologie pour la suite du projet.....	118
4.4	CONCLUSION	120
	CONCLUSION	122
	BIBLIOGRAPHIE	125

Liste des figures

FIGURE 1.1: CAPTEUR CMOS STANDARD (TIRÉ DE [48]).	5
FIGURE 1.2: QUALITÉ DE SOUS-ÉCHANTILLONNAGE (TIRÉ DE [12]).	8
FIGURE 1.3: PIXEL DE CAPTEUR STANDARD (TIRÉ DE [48]).	11
FIGURE 1.4: PHOTODIODE PPD, STRUCTURE PIN.	12
FIGURE 1.5: DESIGN DE DIFFÉRENTS PHOTOTRANSISTORS	14
FIGURE 1.6: NOMBRE DE TRANSISTORS INTÉGRÉS DANS UN PIXEL DPS (TIRÉE DE [15]).	20
FIGURE 1.7: IMAGE SÉPARÉE PAR UN PRISME (TIRÉ DE [48]).	23
FIGURE 1.8: MOSAÏQUE À PATRON DE BAYER (TIRÉ DE [48]).	23
FIGURE 1.9: TRANSMISSION DES FILTRES DE COULEUR (TIRÉ DE [48]).	25
FIGURE 1.10: IMAGE DU CAPTEUR (TIRÉ DE [27]).	32
FIGURE 1.11: COMPOSANTES DU PIXEL (TIRÉ DE [46]).	33
FIGURE 2.1: SCHÉMA DE L'ARCHITECTURE DU CAPTEUR.	36
FIGURE 2.2: MÉMORISATION DE LA SORTIE NUMÉRIQUE DES PIXELS.	37
FIGURE 2.3: SÉQUENCE LECTURE/ÉCRITURE POUR UNE IMAGE.	38
FIGURE 2.4: ERREUR EN MODE LINÉAIRE.	40
FIGURE 2.5: REPRÉSENTATION EN MODE POINT FLOTTANT.	41
FIGURE 2.6: ERREUR EN MODE POINT FLOTTANT.	42
FIGURE 2.7: SÉQUENCE LECTURE/ÉCRITURE EN MODE EXPOSITIONS MULTIPLES.	44
FIGURE 2.8: EXPOSITIONS MULTIPLES POUR CINQ PIXELS DIFFÉRENTS.	45
FIGURE 2.9: PIPELINAGE DES PHASES DE LECTURE/ÉCRITURE.	46
FIGURE 2.10: ACTIVATION DES COLONNES À L'AIDE D'UN CIRCUIT ENDOCHRONE.	47
FIGURE 2.11: DESIGN DU CIRCUIT ENDOCHRONE.	48
FIGURE 2.12: SCHÉMA BLOC DU PIXEL NUMÉRIQUE.	49
FIGURE 2.13: CIRCUIT DE L'OBTURATEUR ET DU SUIVEUR.	50
FIGURE 2.14: CIRCUIT DU COMPAREUR.	52
FIGURE 3.1: TOPOLOGIES DE PHOTODIODES (PD)	57
FIGURE 3.2: VUE EN COUPE DU PIXEL NUMÉRIQUE.	58

FIGURE 3.3: INJECTION DE CHARGES ET EXCURSION DE L'HORLOGE.....	59
FIGURE 3.4: TRANSISTORS D'UNE PORTE DE TRANSMISSION.....	61
FIGURE 3.5: OPTIMISATION D'UNE PORTE DE TRANSMISSION.....	61
FIGURE 3.6: MÉMOIRE STATIQUE POUR LE LOQUET.....	63
FIGURE 3.7: CONVERSION 3.3V À 1.8V.....	64
FIGURE 3.8: DESSIN DU PIXEL AU NIVEAU TRANSISTORS.....	65
FIGURE 3.9: DESSIN DES MASQUES DU PIXEL.....	65
FIGURE 3.10: SIGNAUX NÉCESSAIRES À CHAQUE PIXEL.....	68
FIGURE 3.11: CELLULE DE MÉMOIRE STATIQUE À DEUX PORTS.....	69
FIGURE 3.12: CELLULE MÉMOIRE AVEC MULTIPLEXEUR.....	70
FIGURE 3.13: DESSIN DES MASQUES DE LA PUCE ICFPMJL1, DEUXIÈME VERSION.....	72
FIGURE 3.14: APPAREIL DE MESURE D'INTENSITÉ LUMINEUSE.....	74
FIGURE 3.15: BANC DE TEST.....	75
FIGURE 3.16: CIRCUIT D'ALIMENTATION.....	76
FIGURE 3.17: SCHÉMA DU CNA MULTICANAUX.....	78
FIGURE 3.18: SIGNAUX DE CONTRÔLE POUR LE CNA MULTICANAUX.....	79
FIGURE 3.19: AJUSTEMENT DE TENSION AVEC DEUX AMPLIFICATEURS OPÉRATIONNELS....	79
FIGURE 3.20: CONTRÔLE NUMÉRIQUE AVEC FPGA.....	81
FIGURE 4.1: PHOTOGRAPHIES DE LA PUCE ICFPMJL1.v2.....	85
FIGURE 4.2: RÉPONSE DU CAPTEUR EN FONCTION DE LA LONGUEUR D'ONDES.....	87
FIGURE 4.3: RÉPONSE DYNAMIQUE DU SUIVEUR.....	89
FIGURE 4.4: ANALYSE TRANSITOIRE DU SUIVEUR.....	90
FIGURE 4.5: SORTIE MAXIMALE DU SUIVEUR EN FONCTION DE LA RÉSISTANCE DE CHARGE.	91
FIGURE 4.6: SIGNAUX DE CONTRÔLE DU COMPAREUR.....	92
FIGURE 4.7: EFFET DU SIGNAL <i>VALIDE</i> SUR LA SORTIE.....	93
FIGURE 4.8: AJOUT DE PRÉCISION AVEC LA PHASE <i>VALIDE</i>	94
FIGURE 4.9: COMPAREUR FONCTIONNANT SUR LA TOTALITÉ DE LA GAMME DYNAMIQUE.	95

FIGURE 4.10: PRÉCISION DU COMPAREUR.....	96
FIGURE 4.11: INTENSITÉ DU PHOTO-COURANT DE LA PHOTODIODE.....	97
FIGURE 4.12: COURANT DE FUITE DE LA PHOTODIODE.....	101
FIGURE 4.13: FUITE DE L'OBTURATEUR.....	103
FIGURE 4.14: QUANTIFICATION DE LA FUITE PRONONCÉE DE L'OBTURATEUR.	103
FIGURE 4.15: CORRECTION DE LA FUITE DE L'OBTURATEUR.....	104
FIGURE 4.16: EXPOSITIONS À LA LUMIÈRE ET TRAITEMENT DES DONNÉES.....	107
FIGURE 4.17: PHOTODIODE AVEC FENTES.....	111
FIGURE 4.18: TEST DU « FLIP-CHIP ».....	113
FIGURE 4.19: APPAREIL DE POLISSAGE POUR L'AMINCISSEMENT DE L'ARRIERE DE LA PUCE.	113
FIGURE 4.20: RAPIDITE DE DECHARGE DES CAPACITES PARASITES.....	115
FIGURE 4.21: VARIATION DE LA RESISTANCE D'UNE PORTE DE TRANSMISSION.	116
FIGURE 4.22: IMPLANT VISUEL.....	118

Liste des tableaux

TABLEAU 1.1: CARACTÉRISTIQUES DES CAPTEURS D'IMAGES	31
TABLEAU 2.1: VALEURS DES PIXELS DE LA FIGURE 2.8	45
TABLEAU 4.1: CARACTÉRISTIQUES DE LA PUCE ICFPMJL1.v2.....	109
TABLEAU 4.2: PROFONDEUR D'ABSORPTION DES PHOTONS PAR LE SILICIUM.....	111

Définitions et acronymes

ADC	Analog to Digital Converter
BJT	Bipolar Junction Transistor
CAN	Convertisseur Analogique à Numérique
CNA	Convertisseur numérique-analogique
CCD	Charge Coupled Device
CDS	Correlated Double Sampling
CMC	Canadian Microelectronics Corporation
CMOS	Complementary Metal Oxide Semiconductor
convertisseur A/N	convertisseur Analogique à Numérique
convertisseur N/A	convertisseur Numérique à Analogique
CRSNG	Conseil de recherche en sciences naturelles et en génie du canada
DAC	Digital to Analog Converter
DEL	Diode électroluminescente
DSP	Digital Signal Processing
FCAR	Fonds par la formation des chercheurs et l'aide à la recherche
FET	Field Effect Transistor
FPGA	Field Programmable Logic Array
FPN	Fixed Pattern Noise
FSM	Finite State Machine
IR	Infrarouge
LED	Light Emitting Diode
LSB	Least Significant Bit
MCBS	Multi Channel Bit Serial
MOS	Metal Oxide Semiconductor

MOSFET	Metal Oxide Field Effect Transistor
MSA	Machine séquentielle algorithmique
MSB	Most Significant Bit
NMOS	N-channel Metal Oxide Semiconductor
PMOS	P-channel Metal Oxide Semiconductor
PPD	Pinned Photodiode
RAM	Random Access Memory
RF	Radio fréquence
SCM	Société canadienne de microélectronique
SNR	Signal to Noise Ratio
SoC	System On Chip
SOI	System On Insulator
TFA	Thin Film on ASIC
UV	Ultraviolet
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration

Liste des Annexes

ANNEXE A: DESSIN DES MASQUES DE LA PUCE ICFPMJL1.V1.....	128
ANNEXE B: DESSIN DES MASQUES DU PIXEL NUMÉRIQUE.....	129
ANNEXE C: CIRCUIT IMPRIMÉ DU MODULE DE TEST.....	130
ANNEXE D: DESCRIPTION VHDL DU CONTRÔLEUR.....	131

Introduction

La capture d'images a beaucoup évolué au cours du dernier siècle. Tout d'abord réalisée à l'aide de films (pellicules photosensibles), il est désormais possible de faire l'acquisition d'images à l'aide de semi-conducteurs. L'avantage de ceux-ci par rapport aux films est qu'ils peuvent être directement reliés à d'autres systèmes électroniques de façon à leur ajouter la possibilité de « voir » ce qui se passe. Les domaines d'applications possibles sont très vastes et vont du contrôle industriel aux systèmes de sécurité en passant par des applications de divertissements. De plus, les progrès réalisés dans le domaine de la microélectronique n'ont cessé de croître au fil des années, offrant de plus en plus de latitude avec une qualité toujours supérieure dans la conception de capteurs à base de semi-conducteurs. Puisque le présent mémoire porte sur les techniques électroniques intégrés de capteurs, seules les caractéristiques liées à ces capteurs seront traitées.

Avec l'augmentation de la densité et de la vitesse d'opération des circuits intégrés, de plus en plus de traitement est effectué de façon numérique, ajoutant ainsi de la flexibilité et éliminant le besoin d'ajustements au niveau analogique. La première génération de capteurs d'images CMOS était basée sur des pixels passifs (PPS) avec lecture analogique de la matrice. Ces capteurs, à base d'éléments passifs, souffraient d'une mauvaise qualité de signal due à la transmission directe de la tension du pixel (charges accumulées) sur le long bus de colonnes. À l'époque, les capteurs CCD étaient encore la technologie privilégiée pour l'imagerie numérique de qualité.

Pour la seconde génération de capteurs d'images CMOS, la qualité fut améliorée avec l'utilisation de pixels actifs (APS), où un transistor d'isolation (suiveur) a été ajouté afin d'empêcher la lecture destructive de la génération PPS. L'augmentation de la vitesse des circuits liée à la diminution d'échelle de la technologie et à la diminution des tensions d'alimentation, impose des contraintes de conception pour la section analogique du pixel

qui deviennent de plus en plus difficile à rencontrer. Grâce aux technologies CMOS sous-microniques, un bon nombre de transistors peuvent être ajouté par pixel, au point où la majeure partie de celui-ci est entièrement numérique. En fait, les tendances liées à la capture d'images se dirigent vers les capteurs à pixels numériques (DPS) qui offrent de nombreux avantages, notamment la résolution spatiale variable, la présence de modules de traitement à même la puce, faible consommation de puissance, large gamme dynamique ainsi qu'un coût de production plus faible. Le traitement numérique à l'intérieur d'un capteur CMOS peut être utilisé afin d'améliorer la qualité d'une image prise avec une large gamme dynamique. L'augmentation de la demande pour des appareils portatifs utilisant des capteurs d'images justifie ainsi la nécessité de concevoir de petits systèmes consommant peu de puissance.

Le présent projet a été réalisé au sein du laboratoire de recherche PolyStim de l'École Polytechnique de Montréal. Une des applications potentielle de ce capteur d'image à base de semi-conducteur est de l'intégrer à un système de stimulation intra-cortical visant à redonner un minimum de vision aux gens atteints de cécité profonde. La caractéristique fondamentale d'un système destiné à remplir cet objectif est de posséder une gamme dynamique élevée afin de représenter le mieux possible celle de l'œil humain. De plus, un tel système doit pouvoir fournir de bonnes performances sans modules de support (flash, lampe, etc.). De cette façon, il doit fonctionner aussi bien dans la lumière intense du soleil que dans l'ombre de la nuit sans aucun ajustement ou intervention externe. Afin de remplir chacun des objectifs mentionnés précédemment tout en ayant la flexibilité d'être intégré à d'autres systèmes développés par l'équipe PolyStim, la puce à base de semi-conducteurs doit être conçue à partir du procédé le plus récent disponible qui est le CMOS à une échelle de $0.18\mu\text{m}$.

À court terme, le projet vise à réaliser un capteur d'images générique dont les blocs de base, les pixels, pourront être utilisés dans un vaste éventail d'applications. Afin de pousser la recherche du côté du développement du pixel et de l'architecture des modules

qui l'entourent, la technologie CMOS offre une quantité incomparable de possibilités de conception par rapport à la technologie CCD qui ne permet pas d'ajouter des modules autres que le pixel analogique. Bien que le CMOS ait donné des images de qualité moindre que celle obtenues avec les capteurs CCD, tel que ceux utilisés dans la majeure partie des appareils photos et vidéo commerciaux, beaucoup de terrain a été rattrapé au cours des dix dernières années.

Dans le cadre de ce projet, nous avons élaboré une nouvelle architecture de capteurs d'images à base de pixel numérique. Cet aspect en fait un fonctionnement global différent des générations précédentes (PPS et APS) puisque chaque pixel intègre une partie des composantes normalement retrouvées à l'extérieur de la matrice photosensible. L'aspect modulaire de cette architecture le rend assez versatile pour être utilisé aussi bien dans un système nécessitant un haut taux de capture d'images que dans un système nécessitant une grande sensibilité à l'intérieur d'une large gamme dynamique.

L'organisation de ce mémoire se divise en quatre chapitres. Tout d'abord, un premier chapitre fait l'objet d'une introduction au fonctionnement des différents modules nécessaires aux capteurs d'images à base de semi-conducteurs tout en comparant ces caractéristiques avec les informations retrouvées dans la littérature. Un second chapitre présente l'architecture nouvelle élaborée dans ce présent projet ainsi que les caractéristiques de fonctionnement qui s'y rattachent. Un troisième chapitre décrit les détails de la conception des différents modules présentés au chapitre qui le précède. En dernier lieu, le chapitre quatre présente les résultats obtenus à partir de la puce de test du capteur conçu et fabriqué pour ce projet, suivit de quelques propositions pour le développement futur visant à améliorer les performances du capteur présenté.

CHAPITRE 1

Étude des capteurs d'images

1.1 Introduction

Les éléments d'intérêt dans la capture d'images électroniques sont très vastes et vont de la conversion de photons à électrons jusqu'au système de contrôle de la matrice photosensible, en passant par la constitution du pixel lui-même. Cette section présente donc les différents éléments de la chaîne de capture d'images à l'aide de semi-conducteurs ainsi que leurs avantages et inconvénients.

Le travail de recherche présenté ici propose une nouvelle architecture de capteur d'images au niveau du pixel et de l'interaction de ces pixels avec le système de contrôle. Le but de l'expérimentation est de maximiser la gamme dynamique de l'image acquise, et ce, sous tous les niveaux d'illumination possibles, soit sous un intense soleil autant que dans l'ombre de la nuit.

Nous commencerons donc par une brève description des différentes technologies qui rendent possible la capture d'images. Dans un deuxième temps, les différents éléments électroniques qui permettent la conversion de l'énergie photonique à de l'énergie électrique seront présentés. Une explication de différentes topologies possibles pour l'élément de base qu'est le pixel est donnée. Puis, on présente les différentes architectures de disposition et d'accès aux pixels ainsi que les méthodes utilisées pour les commander. Ensuite, on explique les méthodes pour les conversions analogiques à numériques des signaux capturés par les éléments photosensibles jusqu'à une représentation sous forme numérique de l'image. Par la suite, on présente les différentes méthodes qui permettent d'aller chercher l'information dans la totalité de la gamme dynamique dont l'image est

composée. Finalement, un tableau comparatif est dressé afin de comparer les principaux capteurs d'images à semi-conducteurs ayant été trouvés dans la littérature.

1.2 Capteurs d'images à base de semi-conducteurs

Deux matériaux différents nous permettent de capturer une image: les films et les semi-conducteurs. Les méthodes de réalisation de capteurs d'images électroniques se résument à deux différents procédés de fabrication de circuits intégrés. La technologie CCD (*Charge Coupled Devices*) communément utilisée dans les capteurs commerciaux que l'on retrouve de nos jours, apparue à la fin des années 1960 et fut optimisée pour l'imagerie dans les années 1980. La technologie MOS (*Metal Oxyde Semiconductor*) quant à elle, introduite au début des années 1960 et fut suivie de la technologie CMOS (*Complementary Metal Oxyde Semiconductor*) qui apparut dans les années 1970. Un capteur CMOS typique comporte une matrice photosensible et d'autres unités fonctionnelles tel qu'illustré à la Figure 1.1, tandis qu'un capteur CCD typique n'a qu'une matrice photosensible et une chaîne de lecture séquentielle (*raster scan*).

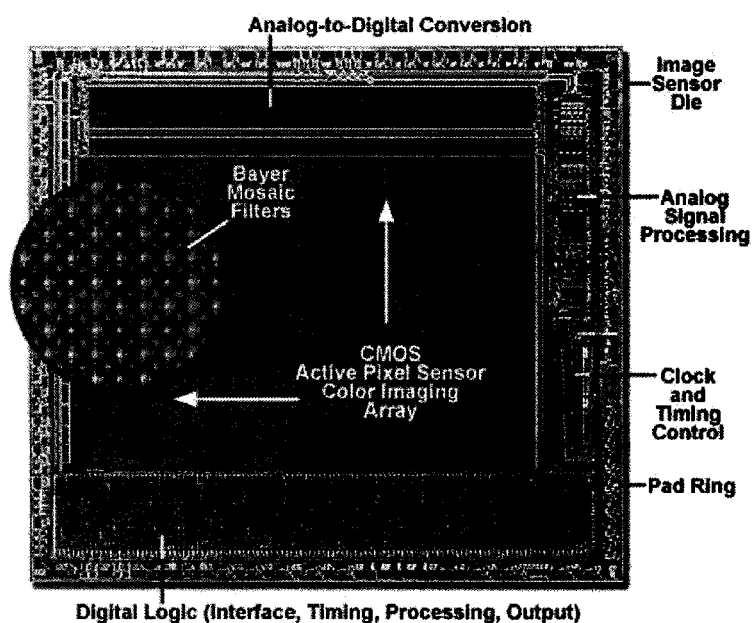


Figure 1.1: Capteur CMOS standard (tiré de [48]).

Du côté de la capture d'images, les premières diodes furent réalisées avec la technologie MOS en 1967 [12] et c'est en 1980 que furent conçus les premiers APS (*Active Pixel Sensor*, section 1.4.1). Il est important de noter que la technologie CMOS, contrairement à la technologie CCD, n'a pas été optimisée pour l'imagerie, mais plutôt pour des circuits numériques rapides. Ces circuits intégrés à applications spécifiques sont appelés ASIC (*Application Specific Integrated Circuits*). Il est cependant possible d'utiliser le procédé CMOS pour fabriquer des capteurs d'images tels que présentés dans ce mémoire.

Le procédé CCD a été inventé vers la fin des années 1960 par les chercheurs du *Bell Laboratories* qui visaient à concevoir un nouveau type de mémoire pour les ordinateurs. Bien que cet objectif ne se soit pas concrétisé, de subséquentes études sur le transfert de charges et les effets photoélectriques des CCD ont conduit ce procédé vers l'imagerie. Celui-ci a commencé à être optimisé en 1980 dans le seul but d'en améliorer les propriétés optiques et d'en augmenter la qualité de l'image captée par le senseur éventuel. Par contre, le procédé CMOS a un meilleur rendement de fabrication que celui du CCD car son plus haut volume lui permet d'être à la fine pointe de la technologie en utilisant des équipements plus modernes [14]. De son côté, le CMOS standard a toujours été optimisé en fonction d'améliorer les performances des ASIC et non afin de rendre ses propriétés optiques meilleures.

Bien que les capteurs CMOS n'aient été utilisés qu'à des fins de recherche jusqu'en 1990, ils gagnent maintenant de plus en plus de popularité par rapport aux CCD et sont devenus intéressants pour plusieurs applications. L'avantage du procédé CMOS sur ce point est le coût de production significativement moindre grâce à l'utilisation de fonderies standard à haut volume. Puisque les capteurs CMOS utilisent le procédé le plus répandu et à plus haut volume, ils profitent de son évolution rapide et bénéficient ainsi de la possibilité grandissante d'intégrer un plus grand nombre de composants sur une même surface de silicium. De plus, ils peuvent capturer beaucoup plus d'images par seconde que les CCD car ils ne sont pas restreints à une lecture sérielle par balayage complet de la

matrice. La production à faible coût des capteurs CMOS a élargie leurs domaines d'application à des appareils tels que fax, scanners, caméras de surveillance, jouets, caméras web, téléphones cellulaires, lecteurs de code barre, souris optique et plusieurs autres. Leur rapidité de capture leur a permis d'entrer dans le domaine de l'inspection industrielle, de la dynamique des fluides, et des diagnostics médicaux pour n'en nommer que quelques-uns.

Bien que le senseur traité dans le présent document soit un capteur générique, une de ses premières applications sera sans doute dans le cadre du projet d'implant visuel de l'équipe de recherche PolyStim [9]. Ceci s'explique par les deux caractéristiques principales, soit la possibilité d'y intégrer des fonctionnalités supplémentaires et la consommation de puissance réduite.

1.2.1 Caractéristiques des éléments photosensibles

Sur une puce de silicium, les éléments de base, c'est-à-dire les pixels, sont disposés sous forme de matrice régulière. Les colonnes et rangées sont donc bien alignées, contrairement aux grains de la pellicule de film. Ces grains sont de différentes tailles (0.5 μm à 10 μm), placés aléatoirement et pouvant se chevaucher. Ceci confère un avantage à la pellicule du film qui n'a pas de problème de sous-échantillonnage, contrairement aux pixels de la puce de silicium. Ce problème survient lorsque l'image capturée comporte des changements de couleur et/ou d'intensité plus faiblement espacés que les pixels eux-mêmes, provoquant ainsi des portions de pixels différemment stimulés. La Figure 1.2 illustre cette propriété.

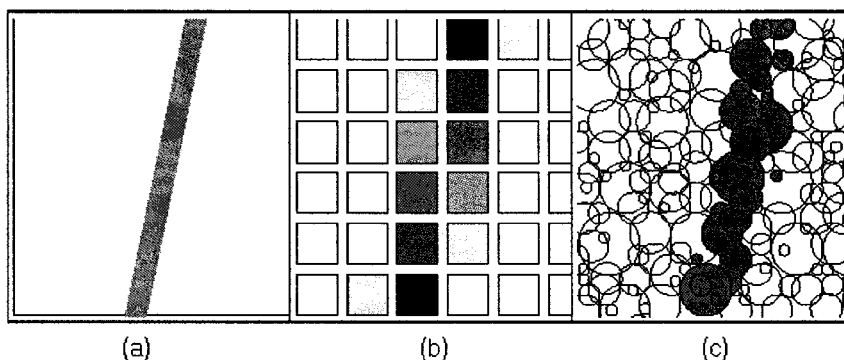


Figure 1.2: Qualité de sous-échantillonnage (tiré de [12])

(a) image réelle; (b) représentation électronique; (c) représentation sur film.

Les grains du film peuvent être noirs, s'ils ont absorbé une quantité suffisante de photons, ou bien rester blanc dans le cas contraire. C'est un processus qui détériore grandement la qualité de l'image due aux forts effets non-linéaires survenant sous de faibles intensités lumineuses. De son côté, le pixel fonctionne par effet photoélectrique où les photons interagissent avec le silicium cristallisé afin de transporter les électrons de la bande de valence vers la bande de conduction [9]. Le courant ainsi généré est proportionnel à l'intensité de la lumière incidente et sa valeur peut être quantifiée sur plusieurs niveaux et ainsi éviter les effets non-linéaires notés ci-dessus.

Une autre différence entre les images capturées sur film et celles capturées à l'aide de semi-conducteurs est leur sensibilité à la portion bleue du spectre de la lumière visible. Les pellicules de films sont hypersensibles à cette lumière, tandis que les semi-conducteurs ont une réponse plus faible dans cette portion que dans le reste du spectre. Ceci est d'autant plus vrai pour les pixels à base de photo-grille, présentés à la section 1.3.4.

1.2.2 Performance des capteurs CMOS et CCD

Au point de vue de l'alimentation électrique du capteur, le CMOS n'en nécessite qu'une seule et de faible niveau, inférieure à 3.3 volts, tandis que le CCD doit en avoir plusieurs et de niveaux beaucoup plus élevés. Ceci est dû aux capacités plus grandes du circuit du pixel en technologie CCD. Ceci implique aussi la nécessité d'avoir des signaux de contrôle et des tensions de polarisation assez élevés. Les grandes variations entre les deux niveaux logiques des signaux, les nombreuses horloges à différentes vitesses, et la nécessité de convertisseurs DC-DC pour les alimentations multiples nécessaires (5 ou plus, [48]), font que les capteurs à base de CCD consomment beaucoup plus de puissance que les CMOS [37]. Par exemple, une caméra vidéo CCD peut facilement consommer un énorme 10W, soit 100 fois plus qu'un capteur CMOS du même type.

Ajoutons qu'il est impossible d'intégrer sur une même puce CCD d'autres fonctions que la matrice de pixels photosensibles et le circuit de lecture. Les systèmes d'imagerie à base de CCD doivent donc utiliser d'autres puces (CMOS) afin de pouvoir traiter l'image. L'information contenant l'image ne peut être transmise à l'extérieur de la puce CCD que sous sa forme analogique avec balayage sériel. Ceci implique que la conversion analogique à numérique doit être faite sur une seconde puce. Le fait d'utiliser plus d'une puce et un certain nombre de composants discrets (pour les alimentations et les horloges multiples) explique que les systèmes à base de CCD sont plus énergivores. Ceci est un aspect limitant important pour les appareils portatifs. Par contre, les capteurs à base de CMOS sont des systèmes plus petits et moins énergivores pouvant être intégrés sur une seule puce.

Malgré tout, les CCD dominent toujours les marchés où la résolution et la sensibilité sont de mise. Cette sensibilité est due à la surface de l'élément photosensible par rapport à celle du pixel qui peut atteindre presque 100% pour certains CCD. D'un autre côté, les capteurs CMOS ont maintenant leur part du marché où la possibilité de miniaturisation est plus avantageuse, au détriment d'une image de qualité légèrement moindre (plus

bruitée). Par contre, avec l'optimisation du circuit du pixel, faisant l'objet de ce mémoire et traité au chapitres 2 et 3, le procédé CMOS permet la capture d'images avec un très faible niveau de bruit et ainsi une excellente qualité. Avec une gamme dynamique de 60dB, les capteurs CMOS commencent à rivaliser sérieusement avec les CCD.

De plus, le procédé CMOS standard tend à avoir une couche épitaxiale (EPI) [26] très mince afin d'éviter les « *latch-up* » dans les circuits numériques, ce qui a pour effet de diminuer la sensibilité dans la portion rouge (650nm - 700nm) du spectre électromagnétique. Aussi, afin de permettre une forte densité de transistors à grille courte nécessaire pour les circuits numériques rapides, les jonctions sont fortement *dopées*. Ceci implique une plus faible réponse dans la partie verte (500nm - 550nm) du spectre.

Selon Wong [43], le procédé CMOS va tendre vers le SOI (*Silicon On Insulator*) et il sera désormais impossible de fabriquer des capteurs photoniques à partir du procédé standard. La solution à ce problème est sans aucun doute la couche mince sur ASIC, nommée TFA (*Thin Film on Asic*), où les variations du procédé n'affecteront à peu près plus la conversion photons à électrons. Ce sujet sera traité à la section 1.3.5.

1.3 Éléments photosensibles

Les capteurs intégrés à base de silicium sont sensibles à la lumière pour les longueurs d'ondes allant des ultraviolets (UV), soit autour de 100nm, jusqu'aux infrarouges (IR), soit dans les environs de 1200nm [12]. Les capteurs qui nous intéressent, destinés aux implants visuels, doivent être sensibles dans le spectre de la lumière visible qui se situe entre 400nm et 700nm. Pour un même procédé de fabrication, il existe plusieurs méthodes et circuits différents pour convertir la lumière en signal électrique. Les éléments de base permettant cette conversion sont présentés dans les sous-sections qui suivent. La Figure 1.3 illustre les principales différences entre les pixels standards des capteurs CCD (a) et ceux des capteurs CMOS (b).

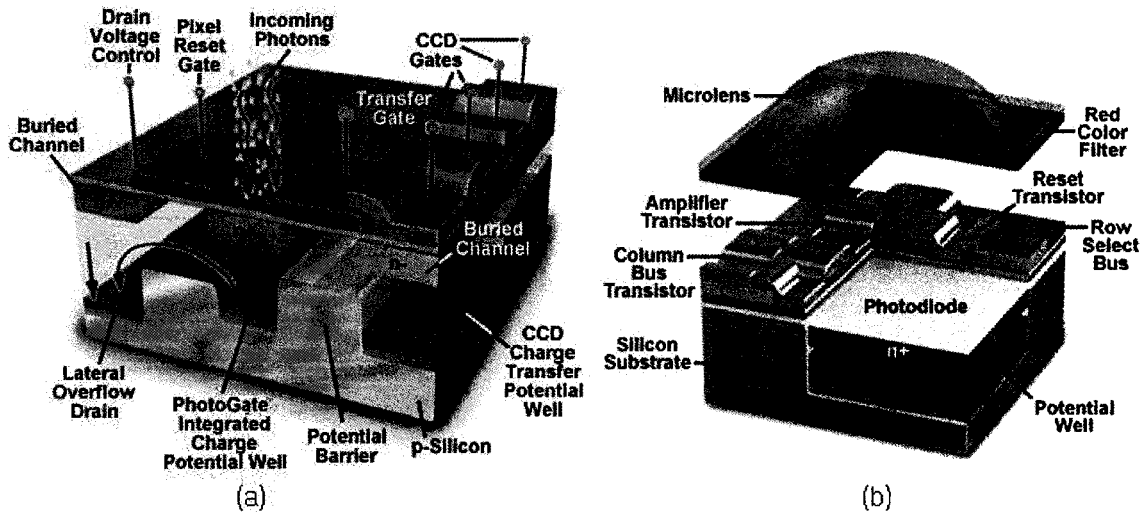


Figure 1.3: Pixel de capteur standard (tiré de [48])

(a) Pixel CCD; (b) Pixel CMOS.

1.3.1 Photodiode standard

La photodiode est probablement l'élément photosensible le plus simple côté topologie, contrôle et fonctionnement. Il s'agit de l'élément qui a d'abord été utilisé dans les premiers capteurs d'images CMOS. Son mode d'opération consiste à polariser la diode en inverse avant la capture d'image. Ensuite, les photons incidents produisent un courant de fuite proportionnel à l'intensité lumineuse pendant le temps d'exposition voulu. Finalement, la lecture de la tension restante aux bornes de la diode donne l'indication de l'intensité lumineuse. La variation idéale de la tension de la diode en fonction de la puissance incidente est donnée par l'équation (1.1) [34].

$$\frac{dV}{dt} = \frac{1.2\lambda P \eta}{c} \quad (1.1)$$

Dans cette formule, λ est la longueur d'onde de la source lumineuse et P représente sa puissance, η représente l'efficacité quantique et c est la vitesse de la lumière. Cette

formule ne tient pas compte de la taille de la diode, qui n'a pas d'effets dans le cas idéal. Par contre, en pratique, la capacité du contour de la photodiode et des transistors connectés à celle-ci s'ajoute à la capacité de la photodiode, laquelle peut affecter de façon non-linéaire la variation de la tension à ses bornes. Cette non-linéarité correspond à un gamma plus grand que un ($V_{out} = V_{idéal} ^ \gamma$). Une grande capacité à l'endroit où s'accumulent les charges suite à l'effet de la lumière peut avoir un impact négatif sur la sensibilité du capteur. Il peut donc être avantageux de réduire la surface de la photodiode sans réduire son périmètre et ainsi ajouter, avec l'espace gagné, des transistors au pixel afin d'améliorer sa sensibilité. Ceci réduit la capacité de la photodiode tout en permettant la collecte de photons par les régions d'appauvrissement latérales. Une méthode plus certaine pour l'amélioration de la sensibilité peut se faire par l'ajout d'une micro-lentille à ménisque positif déposée par photolithographie sur les pixels afin de concentrer les rayons lumineux sur l'élément photosensible, tel qu'illustré à la Figure 1.3b.

1.3.2 Photodiode à couche de semi-conducteur intrinsèque

La photodiode composée d'une structure PIN (silicium dopé P / silicium intrinsèque / silicium dopé N) est connue sous le nom PPD (*Pinned Photodiode*) (Figure 1.4). Lors de son exposition à la lumière, la PPD est initialement sans porteurs [22] et les porteurs majoritaires générés par la lumière sont accumulés dans la région d'appauvrissement, diminuant ainsi la tension à ses bornes [30].

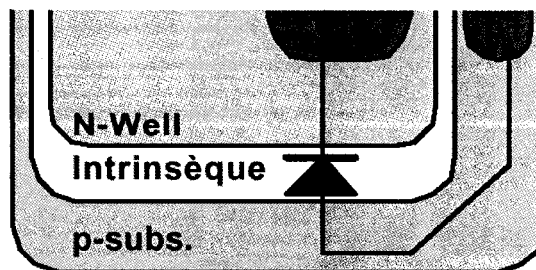


Figure 1.4: Photodiode PPD, structure PIN.

Le courant de fuite généré par la PPD est en général moins élevé que celui de la photodiode. Aussi, l'efficacité quantique de la PPD surpasse celle de la photodiode et de la photo-grille. Ceci est possible à condition de bien contrôler plusieurs paramètres lors de la fabrication, entre autres la profondeur et le niveau de dopage des puits. Deux masques de production supplémentaire sont nécessaires afin d'obtenir des PPD [1]. Ces éléments ajoutent énormément de complexité et augmente le coût du procédé CMOS, déviant ainsi de la méthode standard. Malgré ces considérations, il serait tout de même possible d'utiliser ce procédé de façon rentable pour la fabrication de capteurs d'images, à condition que la demande du marché soit présente.

1.3.3 Phototransistor

Le phototransistor peut être conçu de différentes façons pour la capture de la lumière. Bien que nous parlons ici de phototransistors conçus avec le procédé CMOS, ce ne sont pas les transistors à effet de champ (*Field Effect Transistor, FET*) standard de ce procédé, mais plutôt des transistors bipolaires (*Bipolar Junction Transistor, BJT*). Ils peuvent être verticaux (Figure 1.5a) avec le $p+$ pour l'émetteur, *N-Well* pour la base et le *p-substrat* pour le collecteur [21]. La Figure 1.5 (a), (b) et (c) illustre respectivement différentes méthodes de conception, soit vertical (vue en coupe), latéral (vue en coupe), et circulaire (vue de dessus). Les phototransistors circulaires sont tout simplement des phototransistors latéraux que l'on a fait tourner autour de l'émetteur [2]. En général pour les phototransistors latéraux et circulaires, la grille est laissée flottante, mais on peut lui appliquer une tension afin d'en changer leur gain. La caractéristique qu'ils ont de plus que la photodiode, leur gain (β), amplifie le photo-courant afin de donner possiblement une meilleure lecture. Par contre, cet aspect n'est pas nécessairement avantageux car il donne lieu à de grandes variations d'un transistor à un autre en plus d'être très difficile à contrôler lors de la production. Ceci introduit un bruit à patron fixe (*Fixed Pattern Noise, FPN*) beaucoup trop élevé pour être acceptable. Il est à noter que le FPN peut être réduit en utilisant une photodiode à la place d'un phototransistor [45].

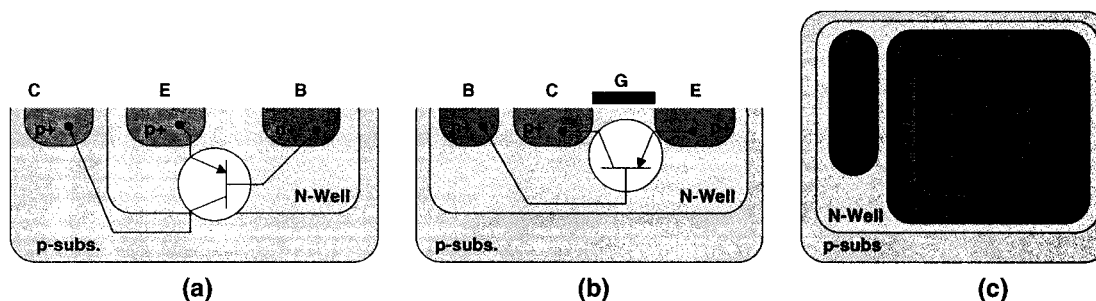


Figure 1.5: Design de différents phototransistors

(a) vertical; (b) latéral; (c) circulaire.

1.3.4 Photo-grille

La photo-grille (PG) est une solution au problème de capacité élevée de la photodiode. Que la PG soit réalisée avec le procédé CCD ou le procédé CMOS, le principe de l'intégration, du transport et de la lecture des charges est sensiblement le même. Une illustration de la PG pour un capteur CCD est donnée à la Figure 1.3a. Lors de l'intégration, les charges s'accumulent sous la grille de polysilicium, dans le puit formé par la tension de polarisation appliquée à cette dernière. Par la suite, les charges sont transférées via une grille de transmission (TG) vers une diffusion flottante où sont transformées en tension pour finalement être lues. Afin d'obtenir un bon transfert, les deux grilles, PG et TG, doivent être très rapprochées pour ainsi mettre en contact leur puit de potentiel. Les procédés CMOS standard ne permettent en général qu'une seule grille. Ainsi pour réaliser PG et TG, il doit y avoir un minimum d'espacement entre les deux et les diffusions ne peuvent donc pas se chevaucher. Le procédé CCD, quant à lui, permet un chevauchement des grilles et donc un meilleur transfert des charges.

Étant donné que les charges s'accumulent sous la grille, le courant de fuite est pratiquement nul. Par contre, la lumière doit d'abord traverser cette grille avant de pénétrer le substrat et permettre à des charges de s'y accumuler, ce que seulement 22% des photons (à 500nm) réussissent à faire dû au matériel peu transparent dont est fait la

grille. À proximité du spectre de la lumière bleue (400nm), la transparence de la grille est encore plus faible [43]. Ceci a pour effet de diminuer l'efficacité quantique du capteur en plus de ne pas représenter adéquatement et justement les couleurs présentes dans la scène observée, en particulier le bleu.

1.3.5 Photodiode à silicium amorphe

Les photodiodes à silicium amorphe sont fabriquées à l'aide du procédé TFA (*Thin Film on ASIC*). Ce procédé est constitué d'une couche mince, formée de plusieurs étages de silicium amorphe, déposée sur un ASIC. L'avantage du silicium amorphe est son coefficient d'absorption de la lumière qui est 20 fois plus élevé que celui du silicium cristallin [7]. Ce dernier étant utilisé pour faire la puce elle-même ainsi que tous les autres éléments photosensibles présentés précédemment. Le résultat est donc une photodiode placée au-dessus de chaque pixel de la matrice. Les micro-lentilles ne sont pas nécessaires ici car le facteur de remplissage est tout près du 100%. Un procédé TFA bien développé permet une excellente isolation entre les pixels [4]. Un autre aspect intéressant est que le TFA est compatible avec tout procédé CMOS en plus d'être réalisable à faible coût. De plus, son utilisation permet de bénéficier de toutes les possibilités du procédé CMOS, telle une plus grande intégration de transistors par pixel et une faible consommation de puissance.

1.3.6 Avenir des capteurs d'images CMOS

Selon Lulé et al. [30], seulement la PPD et le TFA fournissent la sensibilité nécessaire pour permettre une réduction d'échelle sous les 0.18 μ m. L'option d'utiliser la PPD a été écartée car elle nécessitait un procédé CMOS non-standard. Bien que le TFA nécessite lui aussi un traitement spécial (ajout de la couche mince), la fabrication du circuit lui-même se fait avec un procédé CMOS standard disponible pour ce projet. Dans le cas du présent mémoire, une photodiode sera utilisée pour capter la lumière puisque c'est l'élément qui

offre les meilleurs résultats avec un procédé CMOS standard [27]. Avec un circuit de pixel optimisé et une simple photodiode, il serait simple de remplacer cette dernière par le TFA pour obtenir un système à haute performance.

1.4 Constitution du circuit d'un pixel

Le pixel renferme l'élément photosensible et la circuiterie pour l'amplification et le traitement du signal selon des différentes phases. Ces phases d'opération débutent par une phase d'initialisation de l'élément photosensible. Il s'en suit une phase d'intégration, où la lumière incidente modifie la quantité de charges accumulées. Ensuite une phase de lecture retire hors de la matrice la valeur de la charge accumulée de chaque pixel. Certains capteurs utilisent une conversion directe du photo-courant à la tension de sortie, sans intégration [29].

Le rapport de la surface utilisée par l'élément photosensible sur la surface totale du pixel représente le facteur de remplissage. Celui-ci peut se situer autour de 20% pour certains APS, et se rapproche du 100% pour le TFA et pour certains CCD. Afin d'améliorer la performance de certains pixels ayant un faible taux de remplissage, des micro-lentilles, comme celle illustrée à la Figure 1.3b, peuvent être déposées sur la matrice de pixel de façon à focaliser la lumière sur l'élément photosensible.

La taille du pixel peut varier largement selon la technologie utilisée, la dimension minimale des transistors, le nombre de transistors par pixel et les caractéristiques de l'élément photosensible. Par contre, la taille minimale de l'élément photosensible sera limitée par des considérations optiques, et ce même si l'échelle de la technologie utilisée permet l'utilisation de plus petits pixels. Cette taille minimale se situe entre $4\mu\text{m}$ et $5\mu\text{m}$ [15] [12], qui est la limite pour éviter la diffraction lumineuse des micro-lentilles [19] ainsi que pour leur fabrication à un coût raisonnable. Selon Ihara et al. [24], les éléments photosensibles pourraient être plus petits que la limite fixée par les micro-lentilles à

condition de placer quatre éléments photosensibles (2x2 avec patron Bayer, Figure 1.1) sous une même micro-lentille. Ceci pourrait se réaliser en disposant les quatre éléments photosensibles de quatre pixels différents à proximité, par symétrie.

La circuiterie d'un pixel doit être protégée de la lumière incidente afin d'éviter l'influence des photo-courants parasites sur le fonctionnement du pixel. La protection se réalise en ajoutant une couche de métal au-dessous de la circuiterie du pixel. De plus, cette couche réduit la probabilité que les photons incidents causent un «*latch-up*».

1.4.1 Types de pixels CMOS

Les premiers pixels CMOS étaient de type PPS (*Passive Pixel Sensor*) [28]. Ce type de pixels ne contenait qu'un transistor pour la remise à zéro et un autre pour la lecture, mais aucun pour l'isolation des charges intégrées car l'échelle du procédé de fabrication ne permettait pas un niveau d'intégration suffisant pour ajouter d'autres transistors et obtenir une résolution acceptable. Le problème majeur avec ce type de pixel est que la valeur qu'il contient, c'est-à-dire une quantité de charges accumulée, est réduite lors de la lecture suite à l'étalement des charges sur le bus de lecture.

Bien qu'inventés au courant des années 1980, c'est vers les années 1990 que les APS [11] sont devenus la nouvelle façon de faire. Avec l'aide de micro-lentilles, ils sont devenus de sérieux compétiteurs aux CCD. Les APS permettent actuellement quelques variantes quant à leur topologie et à leur sortie qui peut être une tension ou un courant. En mode courant, selon Coulombe et al. [10], la vitesse de lecture est plus élevée car les variations du niveau du signal de sortie sont plus faibles, réduisant les temps de charge et de décharge sur les bus de lecture. Par contre, la transmission d'un signal analogique sur un long bus de lecture est sujette au bruit. Ceci limite le rapport signal sur bruit (SNR, *Signal to Noise Ratio*) qui est directement relié à la gamme dynamique.

1.4.2 Pixels à circuiterie numérique

L'utilisation d'un grand nombre de transistors pour le conditionnement du signal et la conversion analogique à numérique (A/N) à l'intérieur du pixel est maintenant rendue possible et on la qualifie de pixel numérique (DPS, *Digital Pixel Sensor*). Dans le passé, une limitation due à l'échelle de la technologie a longtemps empêché l'utilisation de cette méthode. Aujourd'hui, l'avancement de la technologie a mené à la réduction de la taille du pixel. Ceci permet maintenant de réaliser une caméra sur une seule puce. Bien que cela puisse être l'objectif dans certains cas, cette idée ne vise pas à améliorer la qualité de la capture d'image. De plus, la réalisation de cette caméra sur une seule puce n'est possible qu'avec les nouvelles échelles de technologies sous-microniques car un grand nombre de transistors doivent être intégrés à chaque pixel de relativement faible dimensions. De plus, sous les $4\mu\text{m}^2$ par élément photosensible, il sera impossible de gagner sur la surface du pixel, celui-ci étant limité par des considérations optiques dues aux micro-lentilles, tel que mentionné précédemment. Puisque la taille minimale du pixel reste constante et la taille des transistors diminue avec la progression de l'échelle de la technologie, il devient alors possible d'ajouter des transistors au pixel sans diminuer ses performances optiques. Ceci est encore plus vrai dans le cas du TFA où la totalité de la surface du pixel peut-être remplie de transistors puisque l'élément photosensible est déposé sur celui-ci.

Le DPS apporte de nombreux avantages par rapport aux PPS et APS qui regroupent des modules analogiques à l'extérieur du pixel pour traiter les signaux provenant de ces pixels. Lors de la lecture du DPS, il n'y a pas de détérioration de la qualité du signal analogique par le bruit de lecture sur le bus de colonne puisque cette information est numérique. Avec les PPS et APS, plus le nombre de pixels est élevé, plus le traitement analogique doit se faire rapidement (plus de pixels par colonne) pour traiter le même taux d'image à la seconde, forçant ainsi de nombreuses contraintes sur la circuiterie analogique. En utilisant un DPS, tout le traitement analogique s'effectue à l'intérieur de chaque pixel. La vitesse de la circuiterie analogique est donc indépendante du nombre de

pixels contenus dans la matrice et permet ainsi le traitement de matrices de taille plus élevée.

Les circuits analogiques ne tirent pas beaucoup d'avantages de la réduction d'échelle car les courants de fuite des transistors augmentent et la diminution de la tension d'alimentation réduit la plage de tension disponible pour le traitement analogique. Du côté numérique, le facteur de réduction d'échelle est inversement proportionnel au carré de la surface occupée, les performances en termes de vitesse en sont augmentées et la consommation de puissance en est diminuée. Le fait que le traitement analogique consomme moins de puissance et d'espace comparativement au traitement numérique devient de moins en moins vrai au fur et à mesure que l'échelle de la technologie diminue [15].

En ayant un contrôle numérique sur chacun des pixels, les paramètres d'exposition et de traitement peuvent être ajustés plus adéquatement. Certains types de DPS permettent de tester individuellement chaque pixels sans avoir recours à une source lumineuse, mais plutôt en appliquant des signaux de contrôle, analogiques et numériques [46].

La Figure 1.6 présente une estimation du nombre de transistors en fonction de la taille minimale de la technologie avec un pixel de $5 \times 5 \mu\text{m}^2$ et un facteur de remplissage de 30%. Les deux courbes, analogique (*analog*) et numérique (*digital*), représentent le type de transistors de grande taille et de taille minimale respectivement. On peut voir qu'il est aujourd'hui possible d'intégrer environ douze transistors analogiques et pas loin de 64 petits transistors. Ce nombre ne cesse de grandir avec la poussée technologique, confirmant les avantages du traitement numérique à même le pixel.

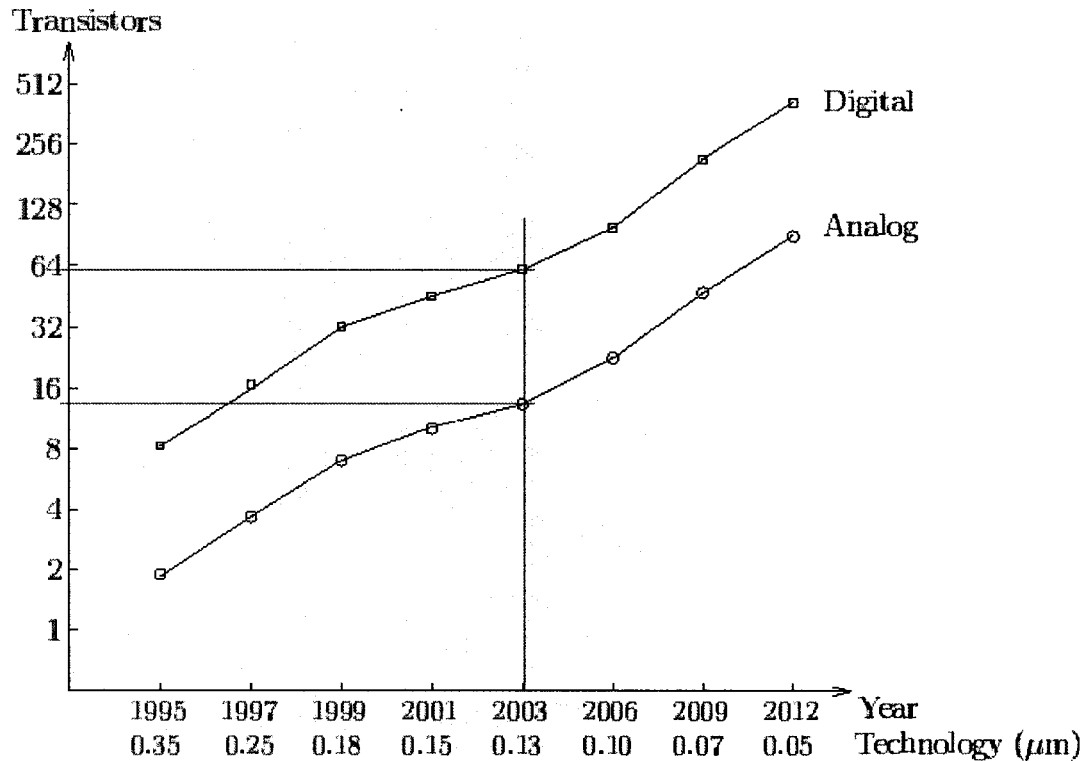


Figure 1.6: Nombre de transistors intégrés dans un pixel DPS (tirée de [15]).

1.5 Matrice de pixels

Contrairement aux capteurs CCD qui sont limités au balayage séquentiel des pixels pour la lecture de la matrice (*raster-scan*), les capteurs CMOS peuvent adresser les pixels de la matrice individuellement, de la même manière que la lecture d'une mémoire RAM. Ceci procure l'avantage de pouvoir réaliser facilement des fonctions, comme le sous-échantillonnage de la matrice ou un zoom numérique, sans avoir à lire la matrice entière. Lorsque le capteur est utilisé pour des applications en temps réel, il peut être intéressant d'obtenir plus rapidement certaines sections d'images afin de les traiter plus vite. Ce cas peut se produire lorsque la luminosité est relativement différente d'une section à l'autre de l'image. Les pixels les plus fortement illuminés compléteront leur intégration plus rapidement que ceux atteints d'une plus faible luminosité où plus de temps sera nécessaire afin d'accumuler une quantité de charges suffisante pour la lecture. Le capteur

réalisé par Brajovic et al. [6] trie les pixels en fonction de leur intensité lumineuse, les rendant disponibles à l'unité de traitement aussitôt que leur quantité de charges accumulées est suffisante.

Une des techniques utilisées permettant de réduire les contraintes de vitesse d'opération sur certaines parties du capteur consiste à répartir les séquences d'opération de lecture de façon séquentielle. Ceci est similaire à la technique de « *pipelining* » utilisée dans les microprocesseurs. Pour un capteur d'images, ceci implique que la lecture et l'intégration ne se font pas en même temps pour toute la matrice. On active et désactive donc l'obturateur continuellement dans différentes sections de la matrice, habituellement une ligne à la fois. Cette technique est appelée « *rolling-shutter* ». Tel que discuté à la section 1.4.2, l'augmentation du nombre de pixels de la matrice ralentit ce type de lecture, laquelle est déjà sensible aux objets en mouvement et à la variation de luminosité. Ce problème est mieux connu sous le nom de « *motion blur* ».

1.5.1 Sources d'erreur dans une matrice de pixels

Lors de la fabrication de la matrice du capteur d'images, de légères variations du procédé peuvent mener à des transistors légèrement différents. Ceci introduit une erreur propre à chaque pixel qui est identique d'une capture d'images à une autre. On qualifie cette erreur de bruit à patron fixe (*fixed pattern noise, FPN*). Afin de corriger cette erreur, certains capteurs permettent de stimuler les pixels par un courant de référence au lieu d'un photocourant afin de calibrer ces derniers. Il est alors possible de quantifier cette erreur fixe. Une autre façon de corriger cette erreur, puisqu'elle est fixe, est de lire la valeur du pixel après sa remise à zéro, puis de lire la valeur qu'il contient après intégration pour finalement faire la différence entre ces deux valeurs. Cette technique porte le nom de double échantillonnage corrélé (*correlated double sampling, CDS*). Peu importe la méthode utilisée pour réaliser le CDS, une mémoire est nécessaire pour retenir la valeur du FPN du pixel. Cette mémoire peut être analogique et incorporée à l'intérieur même du

pixel, mais généralement, cette mémoire n'est pas sur la même puce que le capteur. Cependant, si l'espace le permet, il est possible de l'intégrer sur la même puce (à l'extérieur de la matrice du capteur) et d'obtenir un gain en vitesse.

La fabrication peut aussi causer des variations de propriétés pour les pixels du contour de la matrice, mais ce problème peut être évité en ajoutant des pixels non-utilisés (*dummy*) au pourtour de la matrice [29].

1.5.2 Matrice pour la capture d'images couleur

Tous les types d'éléments photosensibles à base de semi-conducteurs peuvent aussi bien être utilisés pour des capteurs monochromes que pour des capteurs couleurs. Afin d'obtenir des images en couleur, il existe principalement quatre méthodes:

a) La première consiste à illuminer la scène séquentiellement par trois sources de lumières de différentes couleurs et de prendre une photo de chacune de ces scènes. Cette méthode ne peut être utilisée que pour une scène statique et dans un environnement contrôlé. Ceci contraint largement l'utilisation de cette méthode.

b) Une seconde méthode, elle aussi très peu utilisée, consiste à utiliser trois photodiodes réalisant la discrimination spectrale de façon verticale en utilisant différents niveaux de dopage du silicium selon la profondeur d'absorption. Ceci nécessite un procédé CMOS non-standard où il est encore difficile de réaliser la séparation spectrale de façon efficace [12].

c) La troisième méthode implique l'utilisation de trois matrices de senseurs qui obtiennent leur lumière d'un prisme qui a préalablement séparé les couleurs. De cette façon, il n'y a pas de problèmes de recouvrement spectral. La fabrication de ces systèmes est très

délicate et demande beaucoup de précision, mais donne d'excellents résultats et est utilisée dans les caméras de très haute gamme (Figure 1.7).

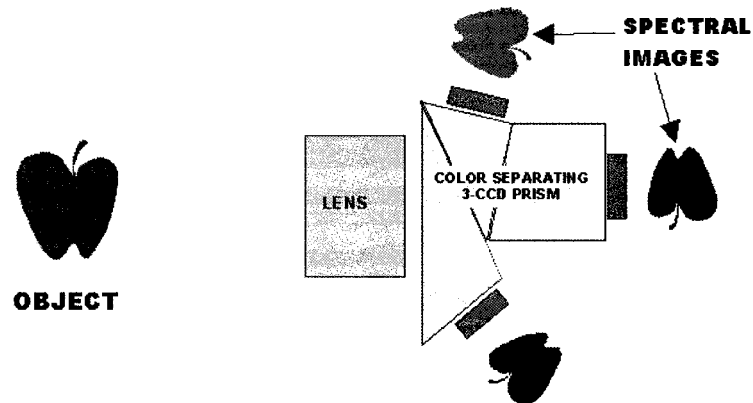


Figure 1.7: Image séparée par un prisme (tiré de [48]).

d) La dernière méthode, et la plus utilisée, consiste à appliquer un filtre de couleur identique pour chaque bloc de 2x2 pixels [48]. Le type de mosaïque utilisée est celle de Bayer, où deux filtres verts, un rouge et un bleu sont utilisés (Figure 1.8). Un inconvénient à cette méthode est le problème de sous-échantillonnage (section 1.2.1) qui peut survenir lorsque la scène comporte un brusque changement de couleurs.

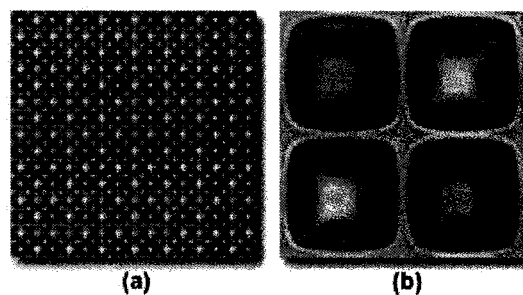


Figure 1.8: Mosaïque à patron de Bayer (tiré de [48])

(a) Matrice de pixels; (b) Un seul pixel comportant quatre éléments photosensibles.

La mosaïque de la Figure 1.8 a été nommée d'après l'ingénieur Bryce E. Bayer de la compagnie Kodak. Elle est formée à partir de polymères des trois couleurs primaires:

rouge, vert et bleu (RVB). Le fait qu'il y ait deux fois plus de filtres verts que bleus ou rouges est dû à la plus grande sensibilité de notre œil dans cette région du spectre de la lumière visible. Notons qu'il est aussi possible d'utiliser un filtre CMY (*Cyan*, bleu pâle; *Magenta*, rose-mauve; *Yellow*, jaune). L'avantage de ce filtre est l'augmentation de la sensibilité du capteur car il laisse passer plus de lumière que le filtre RVB. Son côté négatif est l'augmentation de la complexité des calculs pour la reconstruction de l'image et l'ajout d'une erreur impliquée par cette conversion [48].

1.5.3 Perception des images par l'œil humain

L'application visée du capteur que nous proposons nous motive à comparer les capteurs électroniques et l'œil humain puisqu'elle vise à remplacer un système visuel humain non-fonctionnel.

Bien que quatre éléments photosensibles soient nécessaires afin d'obtenir un seul pixel de couleur, la résolution de la matrice n'en est pas pour autant diminuée. En fait, la résolution spatiale de l'œil humain est principalement déterminé par la luminance (intensité) et non par la chrominance (couleur). Ceci s'explique par le fait que le cerveau humain forme l'image précise à partir de la résolution en intensité, puis il y ajoute la couleur seulement de façon grossière. À l'intérieur d'un capteur électronique, seulement l'intensité de chaque élément photosensible est utilisée. Une interpolation est ensuite faite sur les différentes couleurs du pixel en tenant compte du recouvrement spectral des filtres de Bayer (Figure 1.9). Finalement, la couleur de chaque élément photosensible est déterminée par des algorithmes basés sur les valeurs des pixels voisins.

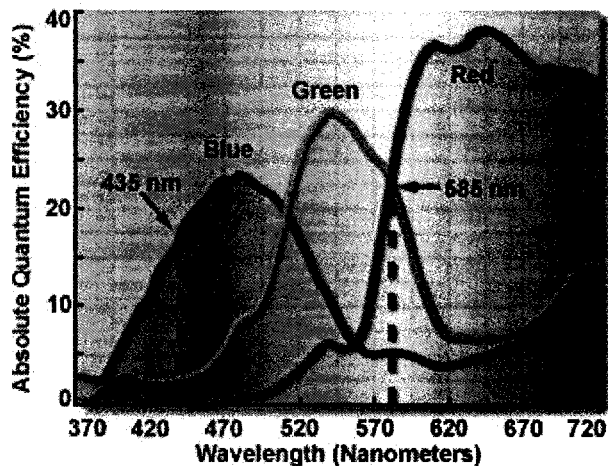


Figure 1.9: Transmission des filtres de couleur (tiré de [48]).

1.6 Conversion analogique à numérique

Lorsque l'on parle de capteurs PPS ou APS, les méthodes de conversion A/N possibles sont nombreuses puisque la conversion se fait à l'extérieur de la matrice où il n'y a pas de contraintes sur la surface utilisée par la circuiterie. On peut alors utiliser des circuits beaucoup plus volumineux renfermant un grand nombre de transistors. Par contre, avec une matrice DPS, l'espace disponible est très limité et on doit faire des compromis entre performance et surface utilisée. Le capteur de McIlrath [32] utilise un convertisseur A/N à un bit sigma-delta ($\Sigma\Delta$) de premier ordre, ce qui est un compromis raisonnable entre surface et fonctionnalité. De plus, il est robuste aux variations du procédé de fabrication. Le problème avec ce type de convertisseur est le haut débit des données de sortie causé par le suréchantillonnage [21].

Les deux méthodes de base pour la conversion analogique à numérique à l'intérieur d'un DPS, le sigma-delta et celle des comparaisons successives, possèdent toutes deux leurs avantages et inconvénients. Par exemple, un convertisseur sigma-delta possède un plus grand FPN, tandis que celui à comparaisons successives a une meilleure robustesse au bruit des transistors [20]. Dans le cas des capteurs infrarouges, le FPN de la matrice n'est

pas critique car il y a déjà un taux élevé de FPN dans le signal incident. Le convertisseur sigma-delta est donc plus approprié dans ce cas. Les capteurs d'images (dans le domaine du visible) nécessitent un faible FPN ainsi qu'un faible débit de données provenant des pixels, étant donné la résolution plus élevée de ceux-ci. Les convertisseurs à comparaisons successives sont donc plus appropriés dans ce cas.

Yang et al. évitent le problème du haut débit des données de sortie du $\Sigma\Delta$ en opérant à vitesse moindre, au taux de Nyquist, et en limitant les données de sorties aux bits nécessaires pour la quantification [46]. Ils proposent le MCBS (*multi-channel bit serial*) qui est composé d'un comparateur et d'une mémoire à un seul bit. La tension de sortie du pixel est comparée à une tension qui est générée par un convertisseur numérique à analogique (CNA) externe à la matrice. Les pixels de la matrice fonctionnent simultanément à l'aide d'un comparateur intérieur à chacun d'eux. Chacun des bits de la matrice est ensuite lu pour laisser place à une autre série de comparaisons successives. Il est à noter que d'autres méthodes de conversion sont possibles pour les DPS utilisant des comparaisons avec des tensions externes à la matrice.

Une méthode un peu plus particulière utilisée par Ripps et al. [38] est inspirée de la vision biologique. Basée sur le fait que les signaux incidents les plus forts provoquent une réponse avant les signaux de plus faible intensité, cette méthode est une sorte de conversion intensité-temps. L'avantage de cette méthode est une faible latence pour les régions de forts signaux incidents. Les sorties ne saturent jamais (au niveau théorique) puisque les pixels sont simplement triés de la plus forte intensité incidente à la plus faible intensité incidente. Cette méthode a été implémentée par Brajovic et al., sur un capteur à N niveaux de sortie où N est le nombre de pixels [6].

1.6.1 Consommation de puissance des convertisseurs A/N

La consommation de puissance est une caractéristique importante à considérer lorsqu'on conçoit un système portable comme un capteur d'images. L'avantage que le CMOS possède sur le CCD est la capacité d'intégrer sur la même puce les convertisseurs A/N, ce qui réduit significativement la consommation de puissance. De la même façon, sur la même puce, il est encore possible de limiter le transport de tensions analogiques en convertissant la valeur du pixel avant de la transporter à l'extérieur de la matrice. Ceci évite d'utiliser des amplificateurs analogiques consommant beaucoup de puissance [37]. Puisque la dissipation de puissance en fonction de la fréquence augmente plus rapidement que linéairement dans les circuits analogiques et linéairement dans les circuits numériques, faire fonctionner N convertisseurs A/N requiert moins de puissance que un seul convertisseur fonctionnant N fois plus vite [36]. Les conversions A/N au niveau du pixel résultent donc en une plus faible consommation de puissance puisqu'ils opèrent en parallèle et à très basse vitesse.

Le type de convertisseur a lui aussi son influence sur la consommation de puissance. Il a été vérifié par Pain et al. [37] que les convertisseurs à approximation successive réalisés par capacités commutées (*switched-capacitors*) consomment le moins de puissance. Tel que mentionné ci-haut, moins loin les tensions analogiques seront transportées et moins la consommation de puissance sera élevée. Il y aura aussi moins de bruit pouvant affecter le signal, résultant ainsi en un plus haut SNR.

1.7 Gamme dynamique des capteurs d'images

La gamme dynamique d'une image est le ratio entre la plus grande et la plus petite valeur de signal pouvant être mesurée pour un même pixel. Un bon capteur d'images doit être en mesure de « voir » aussi bien que l'œil humain, lequel possède une gamme dynamique d'environ 120 dB. Cette valeur équivaut aussi au ratio des plus fortes illuminations sous

un soleil intense (10^3 W/m^2), par rapport aux plus faibles niveaux de luminosité dans l'ombre de la nuit (10^{-3} W/m^2), résultant à 6 décades ou 120dB. Cette valeur représente toute la gamme dynamique sur laquelle l'œil humain est capable de fonctionner, mais est très supérieur à ce qu'il est possible de voir sur une même image. La plus haute sensibilité de l'œil se situe dans la région du vert, entre 500nm et 550nm. La gamme dynamique monochrome de l'œil est limitée autour de seulement 50 niveaux de quantification, mais ceux-ci sont adaptés selon la région de l'image observée [48]. Notre perception d'images de plus haute qualité s'explique par le fait que notre vision a la capacité de s'adapter localement à différents niveaux lumineux en plus de l'aide de notre cerveau qui effectue le travail de reconstituer le tout en une image à plus haute gamme dynamique. Il serait intéressant d'utiliser un capteur dans ce même mode d'opération, soit de diviser l'espace et de « regarder » chaque section avec différents paramètres d'adaptation. Les contraintes de temps à rencontrer afin de balayer l'espace, basées sur les caractéristiques de l'œil humain, seraient de l'ordre 20ms sous de fortes intensités lumineuses et de 100ms dans l'ombre. Il est important de noter que l'œil humain possède une capacité à discriminer le signal du bruit par un ratio de seulement 3:1 dans l'ombre [48]. Ce qu'il faut conclure de tout ceci est que la largeur de la gamme dynamique de la vision humaine est principalement due au travail de notre cerveau et non à celui des yeux. Ce fait s'avère important à considérer pour un projet de stimulateur cortical.

1.7.1 Intrascène versus interscène

On peut définir deux types de gammes dynamiques: la gamme dynamique de l'élément photosensible (interscène) et la gamme dynamique intrascène [13]. La première est mesurée à partir du ratio de la plus grande valeur accumulée dans le pixel avant saturation par rapport à la valeur de l'écart type du bruit sous aucun éclairage. Ce bruit inclut les courants de noirceur, les bruits de lecture ainsi que tous les autres bruits affectant le signal jusqu'à sa conversion en une valeur numérique finale. De son côté, la gamme dynamique intrascène est calculée à partir de ce qui a été possible de mesurer

dans une même capture d'image. Dans ce dernier cas, la plus grande valeur et la plus petite valeur peuvent avoir été toutes deux mesurées sous différents temps d'exposition (multiples captures d'images) et recombinaées ensuite dans la même image. Cette gamme dynamique intrascène est nécessaire lorsque la scène est prise à l'extérieur, avec des régions ensoleillées et des régions d'ombre, afin d'aller chercher les détails dans chacune de ces régions. La gamme dynamique interscène est nécessaire lorsque la scène capturée n'est éclairée que par de faibles sources lumineuses (pas de soleil), où même un long temps d'exposition ne suffira pas pour aller chercher les détails dans les régions où les photo-courants sont trop faibles pour se démarquer des courants de noirceur. Dans ce cas, la gamme dynamique intrascène n'aura pas d'influence et c'est plutôt la gamme dynamique de l'élément photosensible qui déterminera la qualité de l'image. La gamme dynamique intrascène dépend donc de l'architecture du système, de la topologie du circuit ainsi que du travail et calculs effectués pour reconstituer l'image. Sa valeur maximale est celle de la gamme dynamique interscène.

1.7.2 Méthodes possibles pour élargir la gamme dynamique

Afin d'élargir la gamme dynamique, il est possible de retarder la saturation du pixel en lui imposant une réponse logarithmique. Le désavantage de cette méthode se pose lorsqu'on désire effectuer des calculs pour l'ajustement de la couleur ou un traitement quelconque. On doit alors reconvertir ces valeurs sur une échelle linéaire avant de les utiliser. Une autre méthode permettant d'élargir la gamme dynamique consiste à modifier la capacité où sont accumulées les charges (N-Well) une ou plusieurs fois pendant l'intégration afin d'éviter la saturation des pixels. Cette méthode utilise une grille disposée à côté du puit d'accumulation de charges, et celle-ci augmente la capacité du puit lorsqu'une tension est appliquée à ses bornes [11].

Une autre méthode consiste à effectuer de multiples échantillonnages (multiples captures) pour la même image, ce qui augmente significativement le SNR et la gamme dynamique

[44]. Si les temps d'exposition augmentent exponentiellement selon $T, 2T, \dots, 2^k T$ et les valeurs sont quantifiées à l'aide d'un convertisseur A/N à m bits, on obtient alors une valeur de pixel en format point flottant pour un total de $m+k$ bits. La gamme dynamique est ici augmentée par un facteur de 2^k . Ce facteur correspond au ratio du temps d'intégration maximal par rapport au temps minimal, le premier est dicté par le courant de noirceur ou bien par le débit d'images pour une application vidéo, tandis que le second dépend du temps nécessaire pour compléter le plus rapidement possible la lecture d'une image. Cette méthode est intéressante car elle est linéaire. De façon équivalente, il est aussi possible de contrôler individuellement le temps d'intégration de chaque pixel, résultant aussi en l'augmentation de la gamme dynamique [8]. Il est à noter que le SNR et la gamme dynamique sont identiques dans des systèmes linéaires [12].

1.8 Comparaison de capteurs d'images

En faisant une revue des capteurs d'images CMOS répertoriés dans la littérature, il est possible de remarquer que les éléments les caractérisant varient beaucoup de l'un à l'autre. Il n'existe pas une seule méthode pour mesurer les caractéristiques des capteurs. Étant donné leurs différentes topologies (élément photosensible seul, pixel seul, matrice seulement, présence ou non d'un amplificateur et/ou d'un convertisseur A/N, etc.), il est difficile de les classer en ordre de performance. Cependant, à l'aide du Tableau 1.1 on peut remarquer qu'en général, les phototransistors (PT) ont un plus haut FPN que les photodiodes (PD). Ce tableau nous a aussi permis de localiser les capteurs ayant la plus haute gamme dynamique, notamment celui de [21], [45] et [33]. Les travaux de [23], [27] et [47] ont aussi attiré notre attention du point de vue de leur architecture. Les caractéristiques de chacun d'eux ont été étudiées afin de combiner les meilleurs dans le capteur d'images qui fait l'objet de ce mémoire.

Tableau 1.1: Caractéristiques des capteurs d'images

Type de pixel	Échelle du procédé µm	Taille du pixel µm	Taille du pixel λ	Facteur de remplissage %	Dimensions de la matrice	Saturation	Planché de bruit	Gamme dynamique dB	Courant de noircœur	Consommation de puissance mW	Bruit à patron fixe (FPN)	Sensibilité	Sortie	Référence
PG	0.18	9.4	52	15%	352x288	N/A	0.15%	N/A	10 nA/cm ²	50	0.03%	0.107 V/lux*s	Num.	[27]
PG	0.35	8.0	23	25%	512x384	1 V	15 e-	64	11 nA/cm ²	25	1.00%	N/A	N/A	[33]
PD	0.35	5.0	14	47%	1332x1034	1.2 V	0.4 mV	69	N/A	165	0	30.7 fA/lux	Amp.	[23]
PD	0.35	10.5	30	29%	640x512	N/A	N/A	N/A	N/A	80	N/A	N/A	Num.	[47]
PD	0.50	30.0	60	N/A	128x128	50 ke-	20 nA	N/A	N/A	30	1%	N/A	Cour.	[40]
PD	0.70	25.0	36	85%	512x512	100 ke-	400 e-	47	40 ke-/s	50	5.00%	N/A	Num.	[35]
PD	0.70	14.0	20	66%	640x480	100 ke-	200 e-	54	40 ke-/s	N/A	3.00%	280e-6 W/m ²	Num.	[35]
PG	0.80	16.0	20	35%	352x288	1.38 V	525 µV	68	180 pA/cm ²	50	0.29%	4.25 V/lux*s	Amp.	[3]
PD	0.80	20.0	25	30%	128x128	N/A	N/A	83	N/A	1	1%	N/A	Num.	[45]
PD	0.90	20.0	22	27%	256x256	1.5V	29 e-	74	0.36 V/s	80	1.20%	N/A	N/A	[33]
PD	1.20	60.0	50	N/A	64x64	N/A	N/A	93	N/A	1	N/A	N/A	Num.	[21]
PD	1.20	N/A	N/A	N/A	32x32	N/A	N/A	N/A	9 pA	N/A	2.00%	N/A	Cour.	[25]
PT	1.20	N/A	N/A	N/A	32x32	N/A	N/A	N/A	18 pA	N/A	5.00%	N/A	Cour.	[25]
PT	1.20	8.4	7	76%	86x76	N/A	N/A	N/A	N/A	N/A	10%	6.2 nA/lux	Cour.	[16]
PT	2.00	N/A	N/A	N/A	16x16	N/A	N/A	48	5.1 pA	N/A	5.59%	1604 A/W	Cour.	[39]
PT	2.00	N/A	N/A	N/A	16x16	N/A	N/A	66	2.0 pA	N/A	2.96%	69 A/W	Cour.	[39]

LÉGENDE

Type de pixel:

PD Photodiode

PT Phototransistor

PG Photo-grille

Sortie:

Num Numérique

Cour Courrant

Amp Tension amplifiée

1.8.1 Capture rapide et pixel numérique de la référence [27]

Kleinfelder et al. proposent une architecture de pixel numérique à très haut débit. Le capteur, réalisé avec la technologie CMOS à $0.18\mu\text{m}$, peut atteindre une vitesse de 10 000 captures par seconde avec une consommation de puissance de 50mW. C'est l'architecture du pixel numérique qui comporte un convertisseur analogique à numérique à l'intérieur de chaque pixel qui permet d'atteindre une telle vitesse. Toute la circuiterie du pixel est réalisé à l'aide de 37 transistors contenus à l'intérieur d'une surface de $9.4 \times 9.4 \mu\text{m}^2$. Le facteur de remplissage résultant de 15% est suffisant pour obtenir des images de qualité acceptable tel que présenté à la Figure 1.10.



Figure 1.10: Image du capteur (tiré de [27]).

L'élément photosensible de ce capteur est une photo-grille. Dû à des courants de fuite de la grille de transfert trop élevés, les tensions de contrôle ont été fixés à des niveaux permettant d'opérer le circuit en mode photodiode.

1.8.2 Conversion analogique à numérique de la référence [46]

Yang et al. proposent une nouvelle méthode de conversion analogique à numérique à l'intérieur du pixel. Nommé MCBS (*Multi-Channel Bit Serial*), il utilise les comparaisons successives pour effectuer la conversion. Chaque pixel possède donc un comparateur, puis une mémoire (*latch*) pour conserver la valeur du bit converti. La valeur de la tension

aux bornes de l'élément photosensible est comparé à une valeur analogique provenant d'un CNA externe. Les niveaux de tension présentés aux pixels suivent les valeurs d'un code de gray pour les valeurs de chaque bit. Une entrée BITX est présentée au pixel, et c'est cette valeur qui est mémorisée lorsque le comparateur change d'état. Donc, à partir de la Figure 1.11, on peut voir que le niveau de tension du pixel est présenté à l'entrée positive du comparateur, tandis que l'entrée négative est reliée au CNA externe. Le signal BITX est actif seulement lorsque le bit de la valeur de sortie du comparateur est au niveau « 1 » pour la plage entre deux marches de la rampe. Cette rampe, présentée en (a), est générée à partir du CNA externe. Lorsque la valeur de sortie du comparateur change d'état, la valeur courante du bit (BITX) est conservé dans la mémoire. La mémoire de chacun des pixels de la matrice est lue à la fin des comparaisons successives, et ce, pour chaque bit de la valeur finale.

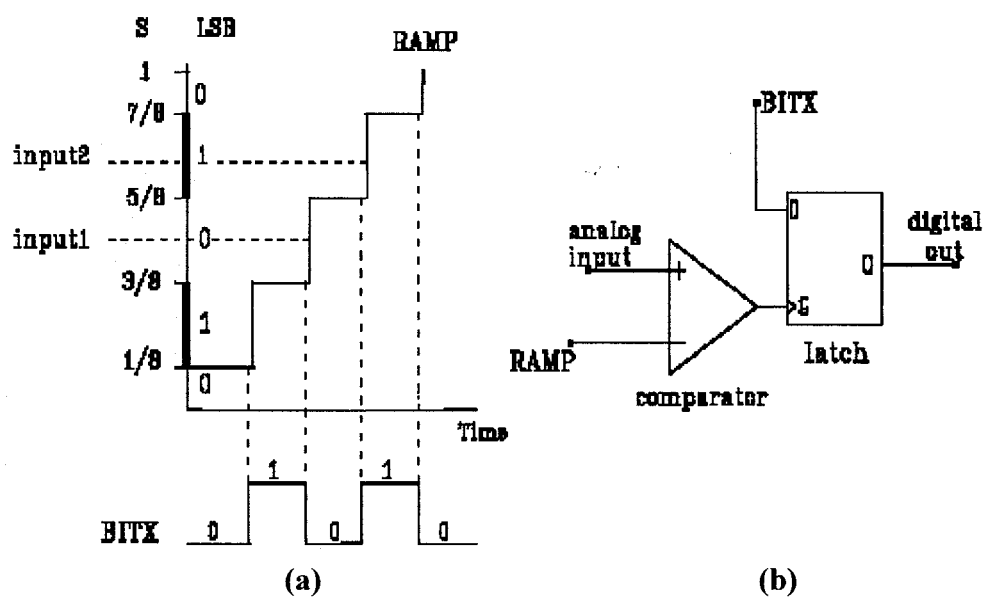


Figure 1.11: Composantes du pixel (tiré de [46])

(a) signaux; (b) circuit.

1.9 Conclusion

Dans ce chapitre, nous avons survolé les différents éléments nécessaires à la réalisation d'un capteur d'images. Pour chacun de ces éléments, les différentes méthodes de conception ont été présentées ainsi que leurs avantages et inconvénients. Nous avons décrit et expliqué les différentes classes de capteurs, les principaux aspects à considérer par rapport aux technologies et aux méthodes analogiques et numériques d'extraction de l'image. De même, un résumé des différents capteurs répertoriés dans la littérature ainsi que leurs caractéristiques a été présenté.

Les sections 1.3 et 1.4 ont permis de dégager les caractéristiques les plus prometteuses pour le design de capteurs d'images CMOS. Bien que le capteur réalisé dans le cadre de ce projet soit monochrome, les détails concernant la couleur ont été présentés. Les contraintes liées à cet aspect seront prises en considération afin de permettre un éventuel ajout de ce côté. Le type d'élément photosensible utilisé aurait certainement été meilleur en technologie TFA mais, compte tenu de nos besoins actuels et de la technologie disponible, nous avons opté pour la réalisation d'un DPS à base de photodiode en mettant l'accent sur l'aspect système. Cependant, le design proposé laisse la porte ouverte à une légère modification du pixel numérique pour une réalisation éventuelle avec la technologie TFA.

CHAPITRE 2

Architecture du capteur d'images

2.1 Introduction

Le présent capteur vise à utiliser les avantages que procure une échelle de technologie réduite. Comme il a été possible de le constater à la Figure 1.6, le procédé CMOS 0.18 μ m permet d'intégrer à l'intérieur d'un même pixel de dimension d'une dizaine de microns, plus de vingt transistors. Ces transistors serviront à réaliser un pixel entièrement numérique vu de l'extérieur, restreignant ainsi à l'intérieur du pixel la conversion photons-électrons et la partie analogique nécessaire pour fournir la sortie numérique désirée.

Avec des pixels numériques, la matrice résultante n'a besoin d'aucune autre circuiterie analogique. Seulement des éléments de circuit numériques pour le contrôle et la mémorisation des données sont nécessaires. Ensuite, seulement l'ajout d'une pile et d'une lentille optique sont nécessaires afin de réaliser un capteur d'images complet dont le cœur est supporté par une seule puce.

Ce type de capteur, un DPS, est ce vers quoi se dirigent les tendances de la capture d'images par dispositifs fabriqués en technologie CMOS. Ceci apporte de nombreux avantages, notamment la simplicité du contrôle, une meilleure gamme dynamique interscène et contrairement au CCD, la possibilité d'ajouter un traitement des données à même la puce. À cela s'ajoute aussi une plus faible dissipation de puissance résultant en une plus longue durée de vie des appareils portatifs. De plus, le coût de production des puces CMOS est moins élevé dû au large volume d'utilisation de cette technologie.

2.2 Architecture du capteur

L'architecture proposée a été conçue afin de maximiser la gamme dynamique par capture d'images multiples. Dépendamment du mode d'opération, la mémorisation de l'image peut se faire de façon linéaire, par représentation à point flottant, ou bien selon toute autre fonction de transfert désirée. Ces différents modes impliquent seulement un changement au niveau du contrôle numérique; l'architecture du pixels, la circuiterie de lecture et la mémoire reste tel quelle. Le schéma de la Figure 2.1 représente l'architecture du système.

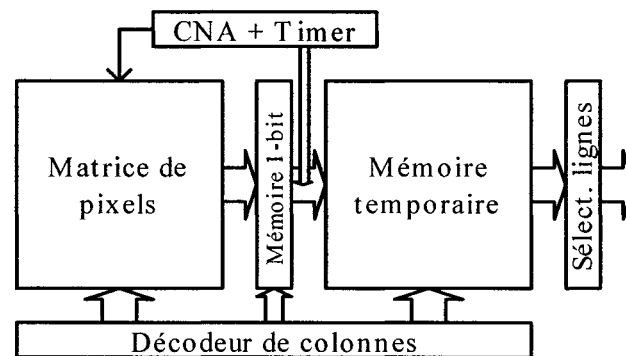


Figure 2.1: Schéma de l'architecture du capteur.

Le système comprend la matrice de pixels à sortie numérique, deux mémoires, les décodeurs pour la sélection des colonnes, les multiplexeurs pour la sélection des lignes vers la sortie, un convertisseur numérique à analogique (CNA) pour effectuer les comparaisons successives de la conversion A/N ainsi qu'un *timer* servant à déterminer à quel moment les pixels franchissent un certain seuil. Le fonctionnement de chacun de ces modules est expliqué dans les sections qui suivent.

2.2.1 Conversions A/N et lecture des pixels de la matrice

Afin d'obtenir la valeur numérique finale du pixel, la méthode de comparaisons successives a été choisie. C'est avec la valeur de la tension de sortie d'un CNA externe à

la matrice que la tension aux bornes de l'élément photosensible de chacun des pixels sera comparée. La comparaison s'effectue en parallèle pour l'ensemble de la matrice à l'aide d'un comparateur interne à chaque pixel.

Lorsque la comparaison doit être effectuée, le CNA est remis à zéro et augmente progressivement jusqu'à sa valeur maximale. Lorsque la sortie du comparateur d'un pixel change, la valeur courante du CNA est inscrite dans la mémoire temporaire, à la case mémoire associée à ce pixel. Une mémoire intermédiaire de un bit par pixel sert à protéger de l'écriture d'une deuxième valeur par-dessus une première valeur déjà inscrite dans la mémoire temporaire. La Figure 2.2 illustre l'interconnexion entre chacun des blocs impliqués dans la mémorisation de la sortie numérique des pixels d'une matrice simplifiée de 2x2 pixels.

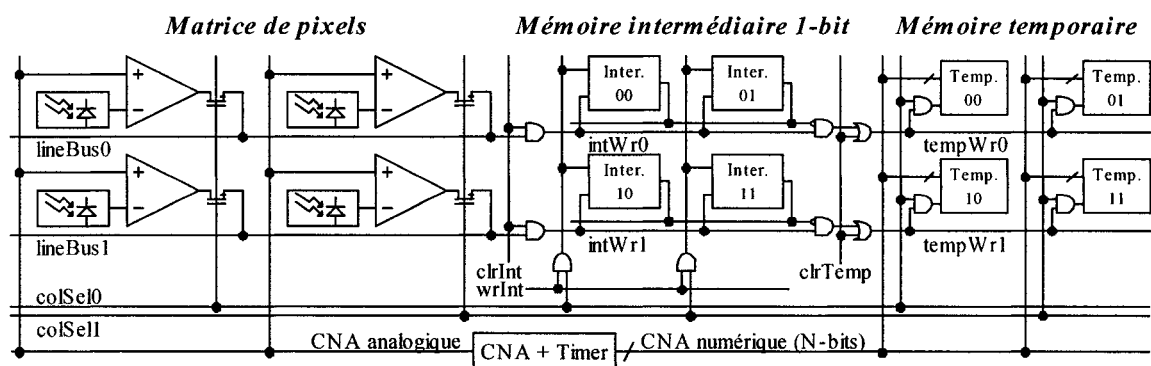


Figure 2.2: Mémorisation de la sortie numérique des pixels.

Avant de débiter une nouvelle capture d'image, la mémoire 1-bit est remise à zéro ainsi que la mémoire temporaire et la sortie du CNA. Ensuite, tel que mentionné ci-dessus, le CNA est incrémenté, le niveau de tension de la photodiode diminuant pendant l'exposition. Pour chaque niveau de sortie du CNA, la matrice est balayée afin d'identifier les pixels dont leur sortie a changé d'état. Cela indique que ces pixels ont mesuré une intensité lumineuse correspondante à la tension analogique générée par le CNA. La valeur numérique associée à cette tension analogique est donc celle appliquée à l'entrée du CNA. Cette valeur est alors inscrite dans la mémoire temporaire. Puisque la

sortie du CNA augmente toujours, un pixel dont la valeur a été convertie ultérieurement aura toujours la sortie de son comparateur activée. Dans ce cas, la séquence de conversion aurait pour effet d'écrire tôt ou tard par-dessus la valeur déjà inscrite dans la mémoire temporaire. Afin de prévenir cette éventualité, une mémoire intermédiaire de 1-bit par pixel a été insérée entre la matrice de pixels et la mémoire temporaire. Lorsqu'un '1' logique est inscrit dans cette mémoire intermédiaire, la case correspondante de la mémoire temporaire est protégée. La Figure 2.3 illustre la séquence des opérations décrite précédemment.

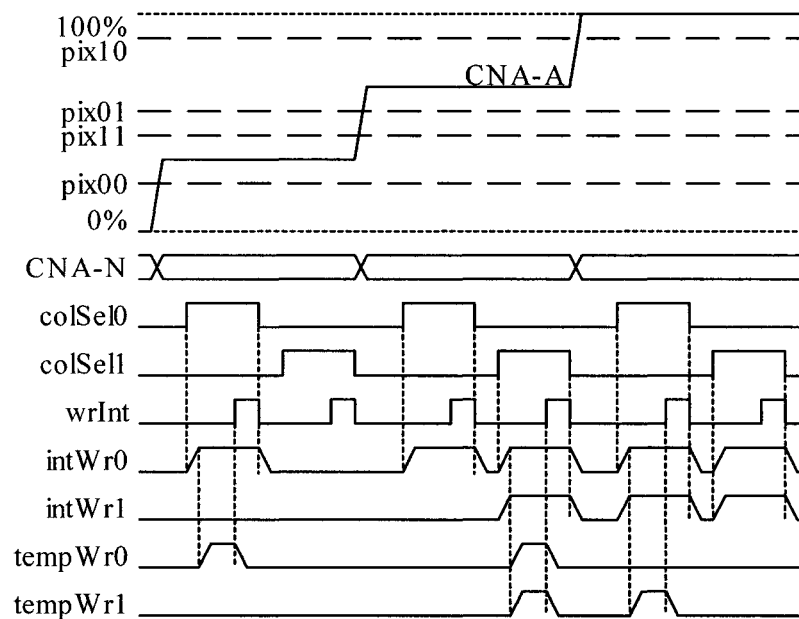


Figure 2.3: Séquence lecture/écriture pour une image.

2.2.2 Expositions multiples

Jusqu'à présent, les conversions A/N présentées opèrent de façon linéaire, avec les comparaisons successives d'une seule exposition. Cependant, l'architecture proposée vise à utiliser les expositions multiples afin d'élargir la gamme dynamique du capteur. Une telle approche permet aussi de maximiser le SNR [46]. L'exposition multiple permet

d'effectuer de multiples captures à des temps d'exposition croissant. Par la suite, les meilleures portions de chaque capture, pixels n'ayant pas saturés ou restés sombres, sont utilisées pour être recombinaées dans une même image.

L'élément nouveau ici est le facteur temps qui est ajouté à la tension mesurée au niveau du pixel. À partir de ce moment, plusieurs méthodes de représentation numérique sont possibles en faisant varier la précision du CNA/CAN et en modifiant les intervalles de temps de mesure. Il est donc important d'évaluer la précision et les avantages de chacune des méthodes.

2.2.2.1 Représentations numériques

Lorsque la valeur d'un pixel est mémorisée de façon numérique, différentes représentations sont possibles. L'objectif d'atteindre une gamme dynamique élevée est un aspect majeur pour ce capteur. Si on se fixe 90dB comme objectif à atteindre, cela signifie que la valeur maximale est de 31 623 fois la valeur minimale d'après l'équation 2.1 où G est la gamme dynamique résultante. L'équation 2.2 donne le nombre de bits B nécessaires pour quantifier cette valeur maximale en mode linéaire, résultant en une valeur de 15 bits pour la valeur maximale précédente. En mode linéaire, le pas de quantification est toujours le même. L'erreur entre les valeurs quantifiées et réelles décroît lorsque cette dernière augmente. Cet aspect est illustré à la Figure 2.4.

$$G=20\log\left(\frac{V_{\max}}{V_{\min}}\right) \quad (2.1)$$

$$B=\log_2\left(\frac{V_{\max}}{V_{\min}}\right) \quad (2.2)$$

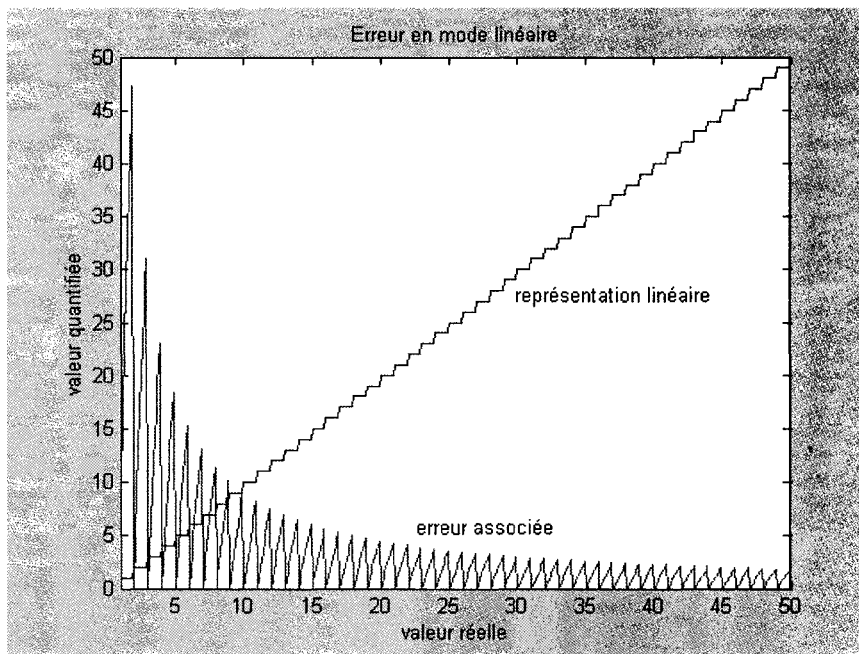


Figure 2.4: Erreur en mode linéaire.

Puisque l'erreur liée à la quantification en mode linéaire diminue pour de grandes valeurs d'entrée, il est possible d'augmenter le pas de quantification tout en gardant l'erreur sous un certain seuil. Si le pas de quantification est augmenté de façon graduelle, la correspondance de la sortie par rapport à l'entrée aura une forme logarithmique. Cependant, un aspect important à considérer est la linéarité de la représentation afin d'aider aux calculs et à toute conversion subséquente. Une méthode bien connue pour rencontrer ces exigences est d'utiliser la représentation à point flottant. Cette représentation est linéaire et nécessite moins de bits que le mode linéaire pour encoder une même valeur. La Figure 2.5 illustre les valeurs possibles pour différents nombres de bits de la mantisse et de l'exposant du nombre représenté en point flottant.

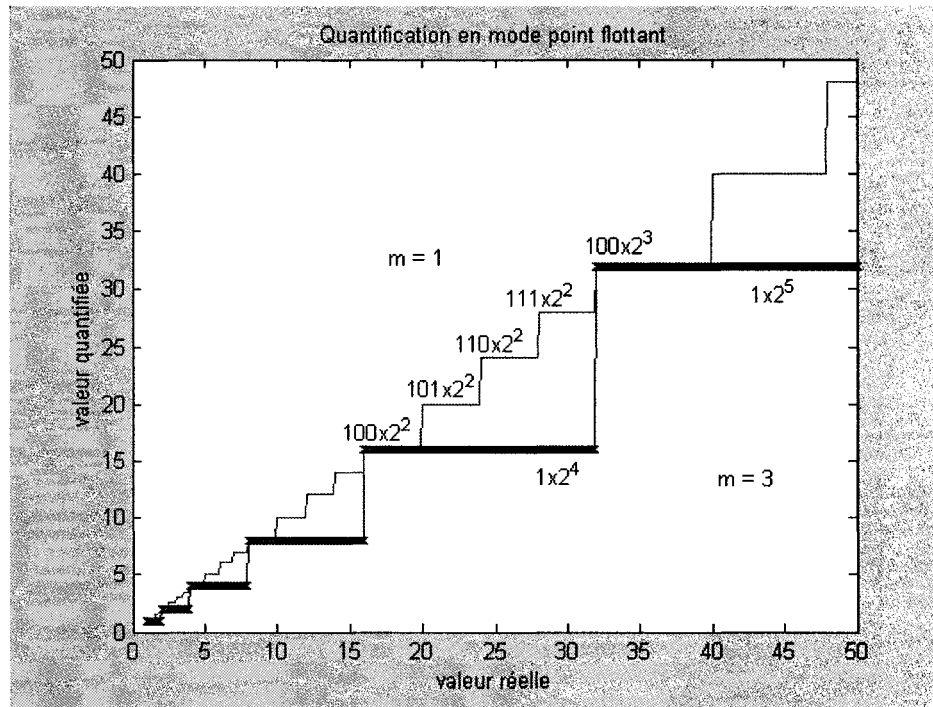


Figure 2.5: Représentation en mode point flottant.

Le trait mince représente une quantification à trois bits pour la mantisse contre un seul pour le trait plus large. Il est à noter que le premier bit de la mantisse est toujours un 1, il n'est donc pas nécessaire de le mémoriser. Le nombre de bits utilisés pour quantifier l'exposant a un effet limitant seulement sur la valeur maximale pouvant être représentée et n'a pas d'influence sur les résultats du graphique de la Figure 2.5. Le nombre de bits utilisé pour représenter la mantisse permet de raffiner le niveau d'erreur pour chaque pas de quantification de l'exposant tel qu'illustré à la Figure 2.6. On peut voir que l'allure de l'erreur est la même pour chaque plage de données reliée à un même exposant.

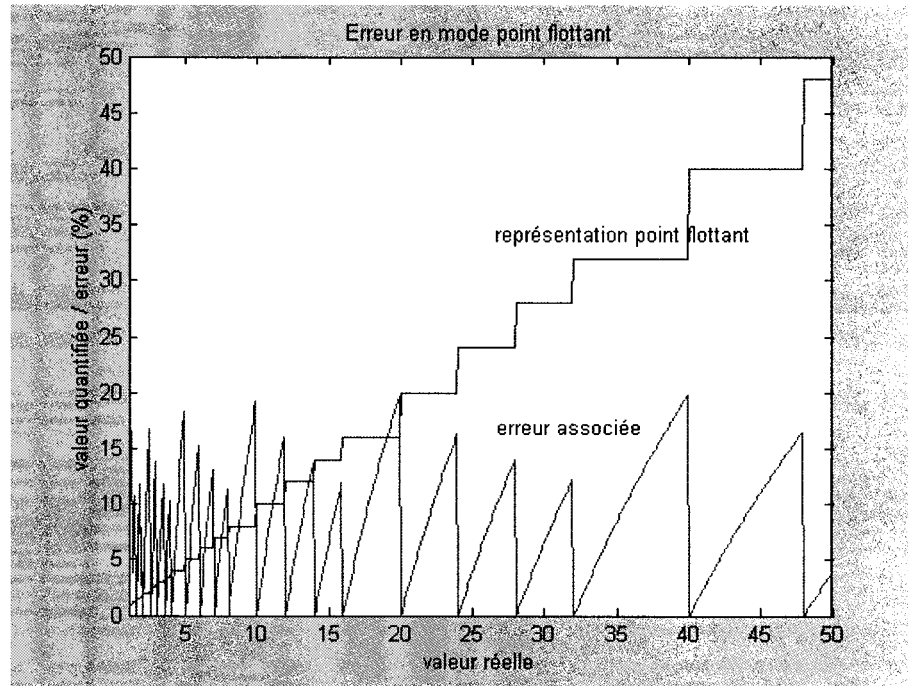


Figure 2.6: Erreur en mode point flottant.

Le lien entre les représentations génériques des Figure 2.4 et Figure 2.6, et la conversion A/N des pixels de ce capteur d'image est le temps et la résolution du CNA. On parle ici de CNA et non de CAN car c'est la précision utile du CNA qui détermine le nombre de comparaisons à effectuer et ainsi la précision de la conversion A/N. En mode linéaire, c'est seulement la précision du CNA (ou seulement le temps) qui détermine la précision de la représentation. En mode point flottant, c'est le temps qui détermine la précision de l'exposant et le CNA pour la résolution de la mantisse à chaque pas de l'exposant.

Toutes ces représentation possibles ne dépendent que de l'instant des comparaisons (temps) et du niveau de quantification (résolution effective du CNA). Ces deux paramètres peuvent facilement être modifiés dans la programmation du contrôleur numérique sans modifications de l'architecture proposée.

2.2.2.2 Conversion A/N en mode expositions multiples

La séquence d'opération de la conversion A/N en mode exposition multiple s'effectue à partir du même capteur que pour les captures à exposition unique, donc la même architecture que celle présentée à la Figure 2.2 est utilisée. Après une réinitialisation de chaque élément, la fuite des charges aux bornes des photodiodes s'effectue en fonction de l'intensité de la lumière incidente et se poursuit pour un temps programmable T . Après ce temps, une vérification est effectuée à savoir quels pixels ont franchi 50% de la largeur totale de la plage dynamique. Ceux qui remplissent cette condition sont marqués dans la mémoire intermédiaire, l'unité de temps est inscrite dans la mémoire temporaire et la valeur du pixel est gelée jusqu'à la prochaine réinitialisation pour une autre capture d'image. Ce gel est réalisé à l'aide d'un obturateur présenté aux sections 2.3.1 et 2.3.2. L'intégration des charges, pendant le temps d'exposition, se poursuit jusqu'au prochain temps programmé dans le système, qui est en général $2T$ pour le mode point flottant. Pour ce même mode, les temps de vérification croissent exponentiellement selon $2^k T$. Lors de la dernière phase de vérification pour une image donnée, la valeur de chacun des pixels est convertie de la même façon que lors d'une exposition simple tel que présenté à la section 2.2.1. Les signaux nécessaires au fonctionnement de ce mode sont présentés à la Figure 2.7.

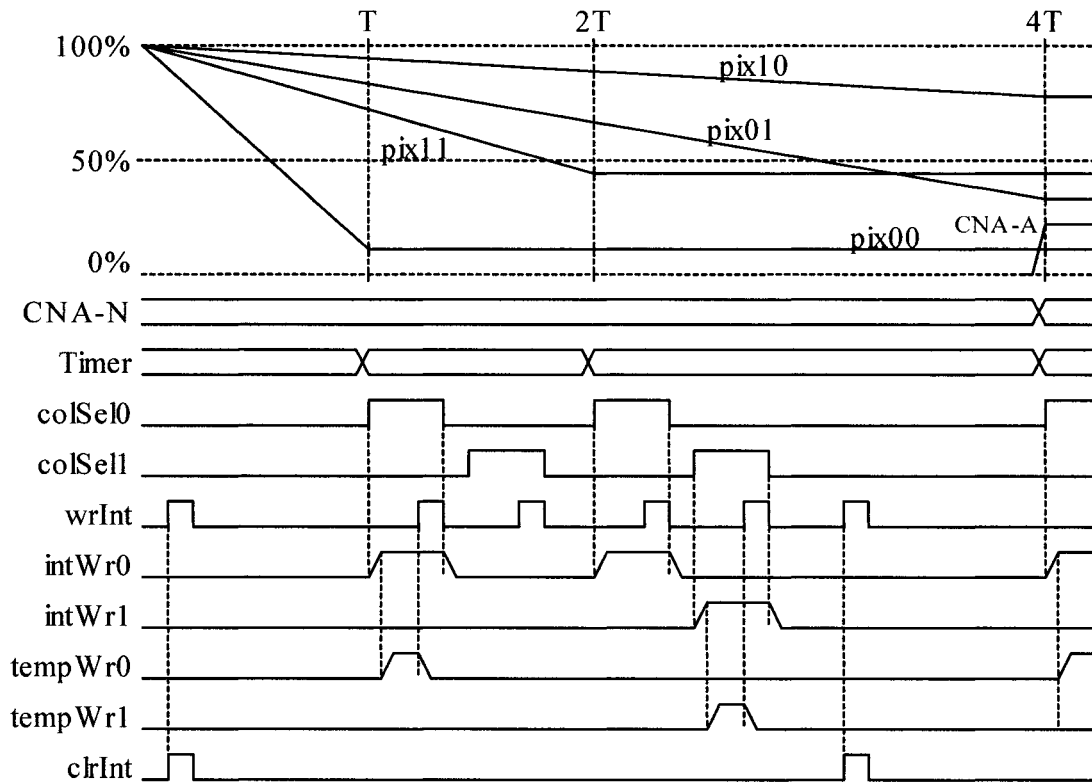


Figure 2.7: Séquence lecture/écriture en mode expositions multiples.

La variation de la valeur de cinq autres pixels, chacun soumis à une différente intensité lumineuse est illustré à la Figure 2.8. Lors de la première phase de vérification au temps T , le pixel le plus illuminé (a) a franchi plus de 50% de la largeur maximale de sa plage. Il est alors marqué dans la mémoire intermédiaire et son temps d'arrêt (T_{arr}) égal à 0 est inscrit dans la mémoire temporaire. La valeur numérique finale (R) en format point flottant est donnée par l'équation 2.3 où $2^B - 1$ est la valeur maximale pouvant être atteinte, V_{pixQ} est la valeur quantifiée de la tension et le deuxième terme de l'équation est lié au temps d'arrêt du pixel concerné.

$$R = (2^B - 1 - V_{pixQ}) (2^{T_{arrMax} - T_{arr}}) \quad (2.3)$$

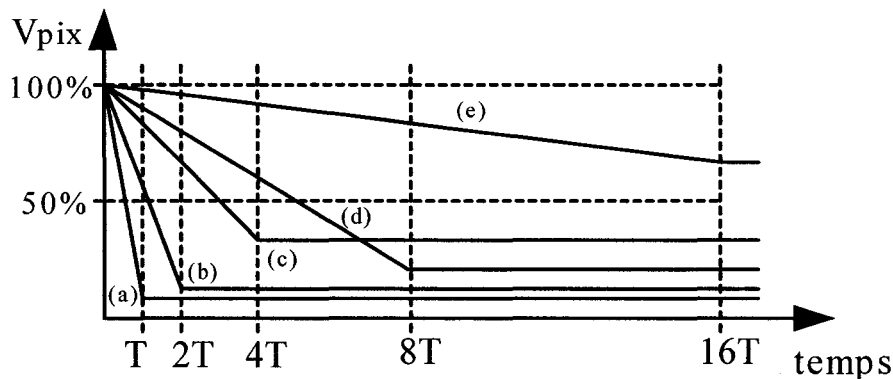


Figure 2.8: Expositions multiples pour cinq pixels différents.

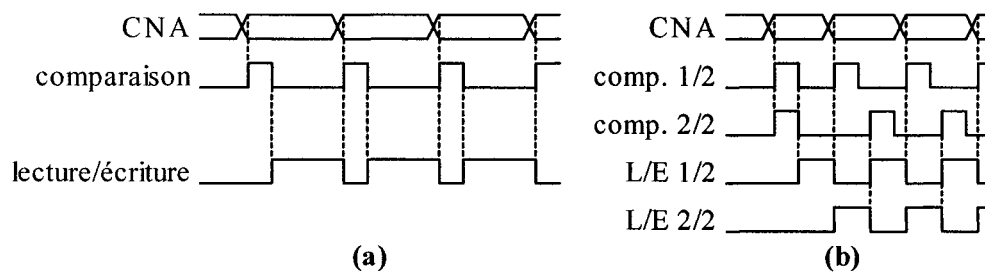
Aux temps de vérification suivants, $2T$, $4T$, $8T$ et $16T$, les pixels (b), (c), (d) et (e) sont marqués respectivement dans la mémoire intermédiaire 1-bit. Ensuite, à $16T$, la conversion A/N est effectuée. La valeur numérique finale pour le pixel (a) est de 3744, résultant de l'équation 2.3, ce qui donne une représentation point flottant. Il est possible de remarquer que la valeur finale calculée d'un certain pixel représente la pente du signal qui est proportionnelle à l'intensité du photo-courant. La même valeur serait obtenue s'il était possible de mesurer des valeurs sur une plus large gamme dynamique que celle de la Figure 2.8, continuant ainsi l'allure de la pente de chaque pixel jusqu'au temps $16T$. Le Tableau 2.1 présente les valeurs des autres pixels.

Tableau 2.1: Valeurs des pixels de la Figure 2.8

Pixel	Marque / T_{arr}					CNA V_{pixQ}	Résultat R
	T	$2T$	$4T$	$8T$	$16T$		
(a)	1 / 000	1 / 000	1 / 000	1 / 000	1 / 000	21	3744
(b)	0 / ---	1 / 001	1 / 001	1 / 001	1 / 001	31	1792
(c)	0 / ---	0 / ---	1 / 010	1 / 010	1 / 010	85	680
(d)	0 / ---	0 / ---	1 / ---	1 / 011	1 / 011	53	404
(e)	0 / ---	0 / ---	0 / ---	1 / ---	1 / 100	170	85

2.2.3 Lecture rapide

Afin d'accroître la vitesse de la conversion A/N de chaque image, il est possible d'utiliser une forme de pipelining des opérations effectuées. Les phases de cette conversion sont: l'ajustement de la sortie du CNA, la comparaison des niveaux de tension analogique et la lecture/écriture dans les mémoires selon les sorties des pixels. L'ajustement du CNA doit être fait avant la conversion A/N (comparaison) et son niveau de tension n'a pas besoin de rester constant pendant la lecture/écriture puisque la valeur numérique est retenue par le comparateur. La deuxième phase, bien que plus courte que la troisième, aurait un effet non négligeable sur le temps total de la conversion A/N si on avait à l'inclure à chaque étape de la conversion (Figure 2.9a). En divisant la matrice en deux sections, gauche et droite, ayant chacune leurs signaux de contrôle vers les pixels correspondant, il est possible d'accélérer le processus. Lorsque la moitié de droite de la matrice du capteur est dans la phase de lecture, il est possible de réajuster la sortie du CNA et d'effectuer la comparaison avec ce nouveau niveau de tension dans la moitié de gauche du capteur (Figure 2.9b). De cette façon, il y a lecture/écriture continue des données et le temps d'exécution des deux premières phases n'influence pas sur le temps de la conversion A/N.



**Figure 2.9: Pipelining des phases de lecture/écriture
(a) sans pipelining; (b) avec pipelining.**

Afin d'accélérer le temps de lecture des colonnes de la matrice du capteur, il est possible de remplacer le démultiplexeur par des blocs auto synchronisés formés de logique

combinatoire. Ce type de circuit, appelé circuit endochrone (*self-timed*), est présenté à la Figure 2.10 montrant la façon de l'utiliser dans notre système.

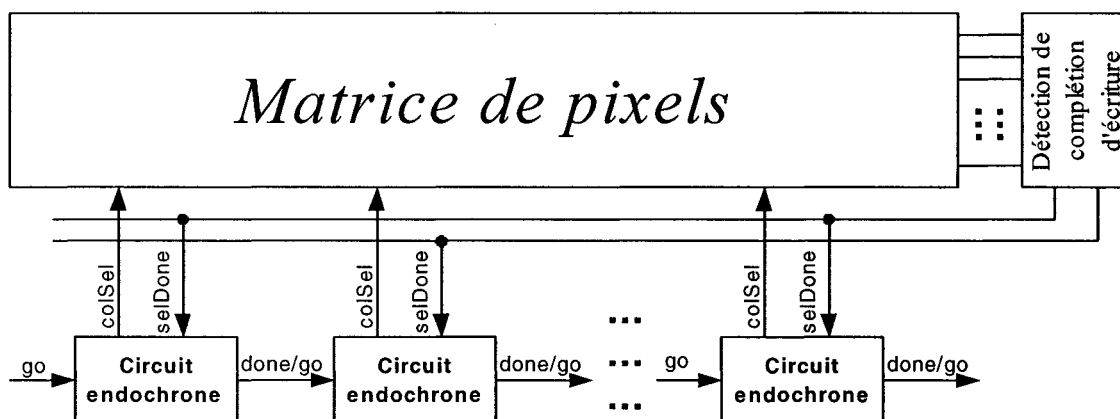


Figure 2.10: Activation des colonnes à l'aide d'un circuit endochrone.

Ce circuit endochrone est constitué de logique asynchrone qui détecte le moment de la fin de ses opérations. Les deux entrées sont le signal *go* provenant du contrôleur qui ordonne le début de la lecture et le signal *selDone* qui détecte la fin de l'écriture dans la mémoire. Après l'activation d'une colonne par le signal *colSel*, le circuit endochrone correspondant ayant reçu le signal *selDone* désactive la colonne ($colSel=0$) et signifie au bloc suivant de procéder aux opérations ($done=1$). La conception d'un tel circuit se fait d'abord par l'écriture de l'équation booléenne correspondante. Cette équation booléenne est définie par les équations 2.4 et 2.5, avec $C=colSel$, $D=done$, $G=go$ et $S=selDone$. Ensuite, une table de Karnaugh (Figure 2.11a) est utilisée afin de déduire le circuit logique correspondant. Dans cette table, les états encerclés sont des états stables, c'est-à-dire ceux dans lesquels la sortie correspond à la sortie du système (valeurs des variables C et D). Les astérisques (*) identifient les chemins possibles pour la transition entre deux états. La séquence des signaux du circuit endochrone est illustré à la Figure 2.11b.

$$C = \overline{D}G\overline{S} \quad (2.4)$$

$$D = \overline{C}D(S+G) \quad (2.5)$$

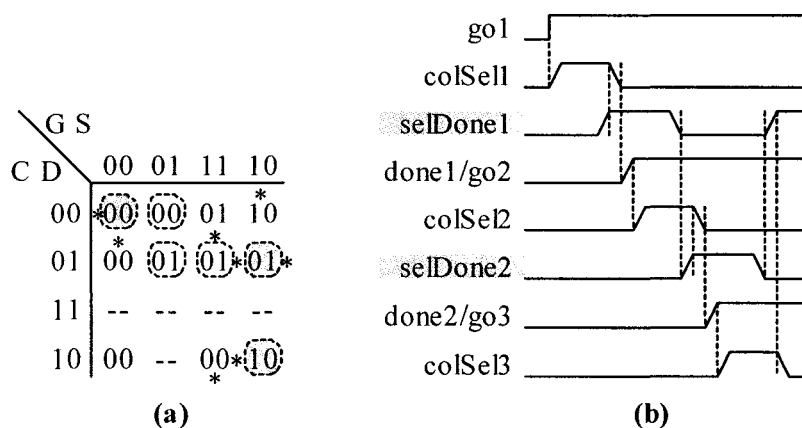


Figure 2.11: Design du circuit endochrone

(a) table de Karnaugh des équations logiques; (b) séquence de signaux.

L'utilisation d'un tel circuit qui gère et optimise toutes les opérations devant être effectuées sans latence et le plus rapidement possible diminue les contraintes sur le contrôleur numérique. En supposant que le temps de lecture/écriture d'une colonne de la matrice est de 20ns et que le nombre de colonnes de la matrice est de 320, nous obtenons un temps de lecture de 6,4µs pour la matrice complète. Le circuit numérique n'ayant qu'à donner le signal de début de la conversion, il est possible de fixer sa période d'horloge à 1µs afin d'être légèrement plus rapide que le temps de lecture de la matrice pour plus de latitude. Ceci est encore loin des 20ns requis comme période d'horloge pour activer chaque colonne à partir du contrôleur et avec la même rapidité que le circuit endochrone. Il est alors possible d'utiliser une horloge fonctionnant à 1 MHz au lieu de 50 MHz et diminuant ainsi largement la consommation de puissance du système. Ceci est d'une importance majeure du côté de la durée de vie autonome du système fonctionnant à l'aide de batteries.

2.3 Architecture du pixel numérique

Ce circuit a aussi été conçu afin de minimiser la consommation de puissance du capteur d'images puisqu'il vise à être utilisé dans un système portable qui doit être autonome le plus longtemps possible. Il est possible de diminuer la consommation de puissance du système en limitant le traitement analogique à l'intérieur de chaque pixel [32]. Tel qu'illustré à la Figure 2.12, chaque pixel contient une photodiode, un obturateur électronique, un comparateur et un mécanisme de loquet afin de maintenir constante la valeur analogique du pixel (V_{pix}) pendant les opérations multiples.

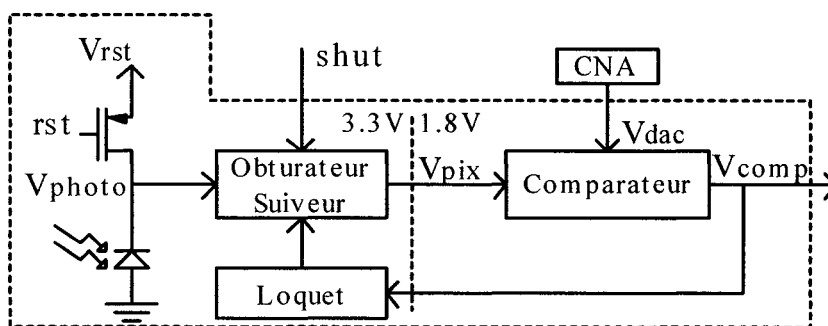


Figure 2.12: Schéma bloc du pixel numérique.

Lors de la réinitialisation, la photodiode est polarisée à une tension de V_{rst} ($V_{th3.3} + 1.8V$) afin de maximiser l'utilisation de la plage de tension à la sortie du suiveur. Pendant l'exposition, les photo-courants générés proportionnellement à l'intensité lumineuse déchargent progressivement la photodiode. Lors des phases de vérification ou conversion A/N, lorsque la matrice est balayée, l'obturateur maintient à un niveau constant les charges à la grille du suiveur. De cette façon, un moment instantané de l'image observée est mémorisé sans être altéré par le processus de lecture.

Chaque pixel contient un comparateur afin de traiter les valeurs analogiques et de fournir une sortie numérique. Ce comparateur détecte la différence entre la tension produite par les charges accumulées à la photodiode et une tension de référence. Cette tension de

référence est commune à toute la matrice de pixels et est générée à partir d'un CNA externe à la matrice

2.3.1 Obturateur et suiveur de la photodiode

Le procédé CMOS 0.18 μ m permet au circuit d'être alimenté avec deux différentes tensions, 1.8V et 3.3V. La taille des transistors produits par cette technologie nécessite une faible tension d'alimentation, soit 1.8V. De tels transistors sont généralement utilisés pour tout le circuit de la puce afin de bénéficier des avantages du gain en surface et d'une plus faible dissipation de puissance. Puisque les tensions d'alimentation réduites peuvent varier d'une puce à l'autre dans un système contenant plusieurs différents circuits intégrés, ils ont tous en commun une tension de 3.3V afin de communiquer entre eux.

Pour le circuit du présent capteur, les transistors de petites taille, fonctionnant à 1.8V, ont toujours été utilisés, à l'exception de la portion analogique intérieure au pixel. Ceci est nécessaire puisque cette section utilise des niveaux de tension supérieurs à 1.8V. Puisque l'espace à l'intérieur du pixel est restreint, la circuiterie analogique doit rester sous sa forme la plus simple. Le suiveur ainsi résultant, Figure 2.13, est donc un simple transistor de type N-MOS qui fournit une tension de sortie (source du transistor) décalée d'un niveau égal à la tension de seuil (V_{th}) par rapport à l'entrée (grille du transistor), causant ainsi une perte de gamme dynamique, au profit d'un gain en simplicité.

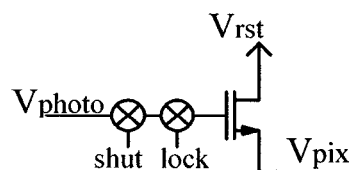


Figure 2.13: Circuit de l'obturateur et du suiveur.

Afin de contrebalancer la chute de tension du suiveur (de V_{photo} à V_{pix}), toute la circuiterie avant l'entrée de celui-ci fonctionnera à une tension V_{rst} de 1.8V plus $V_{\text{th}3.3}$. De cette façon, la sortie du suiveur (V_{pix}) offrira la gamme dynamique complète (0V – 1.8V) au comparateur afin d'effectuer les conversions A/N avec la plus grande précision possible. Étant donné le fonctionnement à une tension supérieure à 1.8V, le suiveur et l'ensemble de la circuiterie qui le précède utilisent de gros transistors, avec une couche d'oxyde plus épaisse afin de supporter les tensions appliquées.

2.3.2 Loquet pour l'obturateur

Pour les expositions multiples, présentées à la section 2.2.2, il est nécessaire de geler la valeur analogique du pixel jusqu'à la conversion A/N. Pour se faire, il est nécessaire de garder constante la tension à la sortie du suiveur. Ceci peut se faire à l'aide de l'obturateur (*shut*), arrêtant ainsi le mouvement des charges à la grille du transistor suiveur. Bien que l'obturateur puisse effectuer la tâche désirée, il est commandé par un signal global destiné à toute la matrice de pixels. De cette façon, il serait impossible d'arrêter le mouvement des charges des seuls pixels ayant rempli la condition d'avoir franchi 50% de leur gamme dynamique électrique. C'est pourquoi un autre obturateur, désigné ici par loquet et activé par le signal *lock*, a été ajoutée en série avec le premier obturateur (activé par le signal *shut*) de la Figure 2.13. Lorsque ce loquet est activée, il gèle la tension du suiveur que l'obturateur global soit activé ou non. Donc, en mode d'expositions multiples, le loquet est enclenchée pour tous les pixels ayant dépassé 50% de leur gamme dynamique électrique, et elle y reste pendant tout le temps des vérifications subséquentes. C'est seulement lors de la capture d'une nouvelle image, après une réinitialisation complète, que le loquet est désactivé.

2.3.3 Comparateur intérieur au pixel

Une attention spéciale a été apportée afin de construire un comparateur avec une surface minimale dans le but de l'inclure dans chaque pixel. Il doit aussi être capable d'opérer sur la plus grande plage dynamique possible et avoir une tension de décalage minimale afin de fournir la plus grande précision possible. L'architecture choisie est un comparateur à capacité commutée, avec un faible nombre de transistors et un même chemin pour les deux entrées dans le but de réduire la tension de décalage. La Figure 2.14 représente le schémas électrique du comparateur.

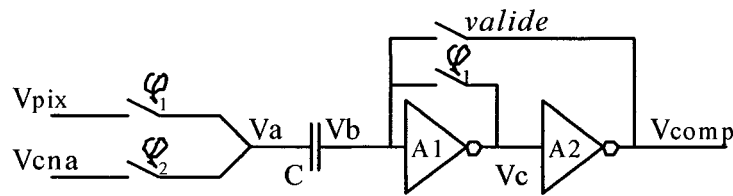


Figure 2.14: Circuit du comparateur.

Trois phases sont nécessaires afin d'effectuer une comparaison, ϕ_1 , ϕ_2 et *valide*. La phase *valide* est utilisée afin de faire basculer la sortie du comparateur vers un signal numérique valide (0 ou V_{dd}) lorsque la différence entre les deux entrées est très petite. D'après la loi de la conservation de la charge donnée à l'équation 2.6, l'expression 2.7 donne la tension aux bornes du condensateur pendant la phase ϕ_1 , où V_a prend la valeur de la tension analogique du pixel (V_{pix}) et V_b est amené au niveau métastable de l'inverseur A_1 , environ $V_{dd/2}$.

$$Q = C(V_a - V_b) \quad (2.6)$$

$$Q = C(V_{pix} - V_{dd/2}) \quad @ \phi_1 \quad (2.7)$$

$$Q = C(V_{cna} - V_b) \quad @ \phi_2 \quad (2.8)$$

Lorsque la phase φ_2 est active, la tension de référence provenant du CNA externe, V_{cna} , est amenée au nœud V_a et le nœud V_b est amené à $V_{dd/2}$ plus la différence entre les entrées du comparateur, menant ainsi à l'équation 2.8 à partir des équations 2.6 et 2.7. Ensuite, la tension V_b (2.9) est amplifiée par l'inverseur A_1 (centré à $V_{dd/2}$) pour donner la tension V_c qui est amplifiée une seconde fois par l'inverseur A_2 pour donner V_{comp} qui est égal à la différence entre les entrées amplifiée par le gain des deux inverseurs (2.10).

$$V_b = (V_{cna} - V_{pix}) + V_{dd/2} \quad (2.9)$$

$$V_{comp} = A_1 A_2 (V_{cna} - V_{pix}) + V_{dd/2} \quad (2.10)$$

Si la différence de tension ($V_{pix} - V_{cna}$) est trop petite pour qu'une fois amplifiée, la sortie ne donne pas une valeur numérique valide (0V ou V_{dd}). La phase *valide* active alors une boucle de rétroaction positive (2.11) afin de faire saturer la sortie rapidement.

$$V_b = V_{comp} \quad @ \text{ valide} \quad (2.11)$$

2.4 Conclusion

Ce chapitre a présenté l'architecture qui est à la base de ce nouveau capteur d'images à pixels numériques. Presque la totalité de la fonctionnalité nécessaire afin de fabriquer une caméra sur une seule puce est intégré à même chaque pixel. Ceci en fait un design très modulaire. Il est donc très facile de partir de l'élément de base qu'est le pixel pour fabriquer une matrice d'un nombre de rangées et de colonnes arbitraire.

Le pixel numérique ne nécessite aucun courant de polarisation et le peu de sections analogiques fonctionnent par la méthode des capacités commutés. Ceci a pour avantage

de ne pas consommer de puissance lorsque aucune image n'est capturée. De plus, aucun délais ou latence n'est observée du système au repos entre la réception de la commande de capturer une image et le début de la capture d'image. Le système au repos ne comporte aucun délais de démarrage. La capture débute aussitôt que l'activation lorsqu'une courte impulsion du signal de réinitialisation est donné.

La vitesse d'opération requise n'est pas compromise par la taille de la matrice puisque le traitement est effectué en parallèle et indépendamment pour chaque pixel. Dépendamment du mode d'opération choisi, il pourra tout de même y avoir une certaine perte de précision dû au temps de lecture si une très grande matrice est utilisée. Cette légère perte est tout de même négligeable si on la compare à celle d'un capteur standard APS. Avec des matrices APS de grande tailles, le système ne peut lire la valeur des pixels dû à une trop grande contrainte de vitesse imposée aux circuits analogiques communs à toute la matrice, où aucun traitement parallèle n'est possible.

Les différents modes d'opérations présentés permettent une flexibilité d'opération, selon l'objectif visé par la capture d'image. Selon que l'on désire prendre des images de très haute résolution ou à un taux d'échantillonnage très élevé, le contrôle numérique peut être reprogrammé et adapté en conséquences.

CHAPITRE 3

Design et contrôle du capteur d'images

3.1 Introduction

Au cours du chapitre précédent, il a été possible de voir les différents blocs qui constituent la base de notre capteur d'images. La plus grande partie des efforts de conception ont été rattachés au pixel car il intègre presque la totalité de la fonctionnalité nécessaire pour fabriquer un capteur d'images complet. Le fait qu'il soit numérique, conversion A/N intégrée, laisse très peu de travail à la circuiterie externe à la matrice de pixels.

Le présent chapitre présentera tout d'abord les détails de la conception du pixel, avec la technologie CMOS 0.18 μ m, qui ont été laissés de côté au chapitre précédent. L'élément photosensible convertissant les photons en électrons sera présentée avec les raisons qui ont motivé le choix du type de celui-ci ainsi que de sa topologie. Toujours à l'intérieur du pixel, les caractéristiques particulières des obturateurs et du suiveur seront détaillées. Ensuite, il sera possible d'avoir une vue d'ensemble des deux versions de puces implémentées avec les différents blocs présenté au chapitre précédent.

Par la suite, le design du système de tests conçu pour évaluer le fonctionnement de ce capteur sera détaillé avec toutes ses composantes. Bien que ce banc de tests ait été conçu pour tester la puce fabriquée dans le cadre de ce projet, il sera démontré à quel point ce système générique pourra être utilisé pour n'importe quelle puce ayant le même boîtier.

3.2 Design du pixel numérique

Le pixel numérique étant le cœur de ce capteur DPS et regroupant pratiquement son entière fonctionnalité, une présentation détaillée de ce dernier s'en suivra. Le design a été fait en tenant compte de l'objectif d'obtenir le meilleur compromis entre la surface utilisée et les performances requises.

3.2.1 Conversion de l'énergie photonique en énergie électrique

Afin de pouvoir convertir les électrons en photons, il est nécessaire d'ajouter un élément photosensible. Parmi ceux présentés à la section 1.3, trois étaient réalisables avec le procédé standard CMOS 0.18 μm , soit le phototransistor, la photo-grille et la photodiode. Le phototransistor a été rejeté à cause de son gain (β) incontrôlable menant à une très grande source de FPN, et à cause de sa grande surface requise à l'intérieur d'un pixel numérique avec un espace très restreint. La photo-grille, bien que nécessitant un contrôle quelque peu plus complexe que les autres éléments, a tout de même mérité un certain intérêt dû à son utilisation performante dans les capteurs CCD. Malheureusement, le procédé CMOS 0.18 μm ne produit pas des photo-grilles aussi performantes que dans le procédé CCD. S. Kleinfelder et al [27], ils ont utilisé une photo-grille dans leur capteur réalisé en CMOS 0.18 μm , et ont pu observer des problèmes liés à une mauvaise isolation des charges par la grille de transmission. Ils ont conséquemment modifié les tensions de polarisation de leur circuit afin de l'opérer comme si c'était une photodiode. Tous ces aspects nous mènent à la conclusion qu'il est nécessaire d'utiliser une photodiode comme élément photosensible pour notre design. À partir de ce moment, les deux options qui s'offrent à nous quant à la topologie de la photodiode sont présentées à la Figure 3.1.

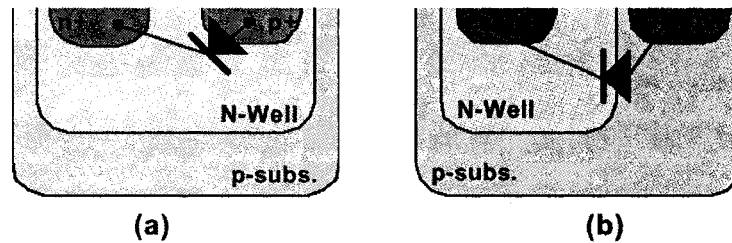


Figure 3.1: Topologies de photodiodes (PD)
(a) PD p+/N-well; (b) PD N-well/p-substrat.

Afin d'utiliser toute la plage de tension disponible, la différence au niveau du circuit existant entre les deux topologies de photodiodes est la nécessité de connecter la cathode au nœud le plus positif du circuit pour la topologie (a) et l'anode à la masse pour la topologie (b). L'implication de ceci est que la photodiode (a) sera confinée dans un puit N-Well. Ceci permet une meilleure isolation des photo-courants adjacents, mais au détriment d'une moins grande surface de jonction PN en plus d'être moins en profondeur que celle de la photodiode (b). Cet aspect joue au niveau de l'efficacité quantique, puisque les photons sont absorbés dans la zone d'appauvrissement (jonction PN) et rares sont ceux qui sont absorbés en surface. Il en résultera donc une meilleure efficacité pour la cellule (b) qui possède une plus grande surface (jonction PN) pour collecter les photons. De plus, cette surface fait que la jonction PN se retrouve plus en profondeur que la zone d'intérêt de la cellule (a), là où les photons sont absorbés.

3.2.2 Comparateur dynamique

La topologie choisie pour le comparateur dynamique est celle présentée à la Figure 2.14. Tous les détails de fonctionnement ayant été expliqués précédemment, l'implémentation matérielle de ce comparateur sera maintenant présentée. Puisque ce circuit se retrouve à l'intérieur de la matrice exposée à la lumière, un écran doit être ajoutée afin de prévenir la génération de photo-courants à des endroits non désirés dans le circuit. En fait, tout le pixel, à l'exception de la photodiode, doit être protégé de la lumière afin de fonctionner

selon le comportement désiré. L'écran est conçu à partir d'une couche de métal, laquelle est opaque et est située au dessus des transistors.

Ce comparateur fonctionne à partir d'un condensateur où deux méthodes pour sa réalisation sont possibles. La première est d'utiliser la capacité de grille d'un transistor, tandis que l'autre consiste à utiliser la capacité existant entre deux couches de métal. Le désavantage de la première méthode est l'augmentation de la surface du pixel dû à l'ajout du transistor de grande taille réalisant cette capacité. De plus, la capacité résultante est non-linéaire en fonction de la tension. Pour le pixel ici concerné, la deuxième méthode possède tous les avantages. La capacité est linéaire en fonction de la tension de sorte que le comparateur fonctionne de la même manière sur la totalité de la gamme dynamique. Aussi, un gain en surface est observé en utilisant les couches de métal supérieures situées à l'intérieur de la même superficie que le circuit du reste du pixel. De cette façon, il est possible de réaliser ce condensateur au-dessus des transistors de la partie numérique du DPS tel qu'illustré à la Figure 3.2. Cette méthode permet de minimiser la taille du pixel.

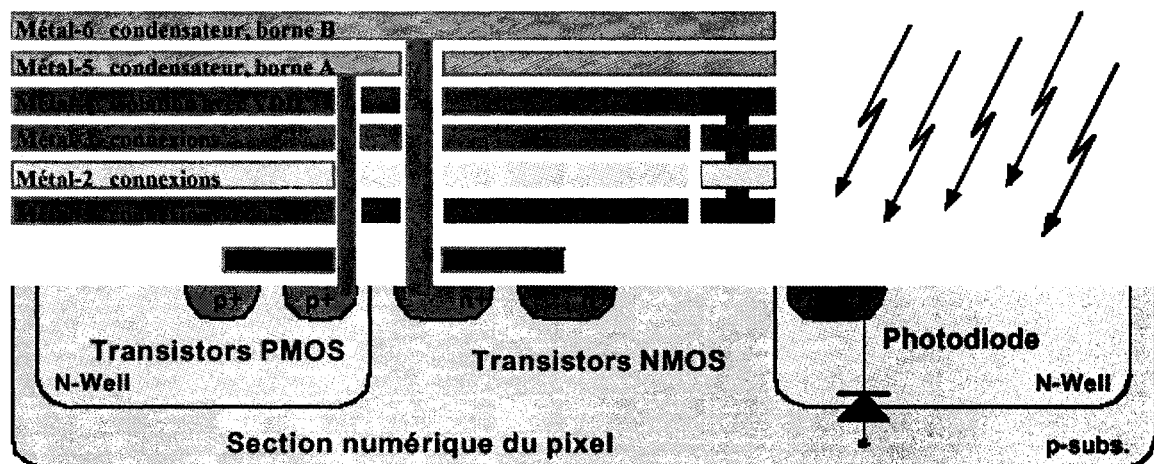


Figure 3.2: Vue en coupe du pixel numérique.

Par la suite, le design des deux inverseurs du comparateur est fait à la manière numérique quasi-standard. La seule restriction est qu'ils soient tous deux identiques. Afin de permettre au comparateur de basculer aussi bien d'un côté que de l'autre, il est préférable

de ramener le niveau métastable des inverseurs le plus près possible de $V_{dd}/2$. Étant donné que la mobilité des charges des transistors pMOS est plus faible que celle des nMOS, quatre transistors pMOS de même dimensions que le nMOS sont requis en parallèle afin de balancer l'inverseur. La relation de la mobilité des charges entre les deux types de transistors est donnée à l'équation suivante:

$$\mu_n \cong 4\mu_p \quad (3.1)$$

Afin de prévenir l'injection de charges lors de l'ouverture (de ON à OFF) des portes de transmissions, des transistors *dummies* ont été ajoutés. Ces éléments limitent les effets d'injection de charges. Cependant, le résultat n'est pas parfait puisque nous opérons sur la totalité de la plage de tension et que l'injection de charges dépend de la tension. Or, la charge n'est pas la même de chaque côté de la porte de transmission. De plus, les *dummies* ont une taille différente (idéalement $W/2$) de celle du transistor duquel ils captent les charges. Une surface de transistor (WL) différente résulte en une capacité parasite entre la grille et la source (C_{gs}) différente de celle entre la grille et le drain (C_{gd}), affectant ainsi l'injection de charges lors du changement de niveau de l'horloge. Ce problème est communément appelé excursion d'horloge (*clock feed-through*). Le circuit est illustré à la Figure 3.3.

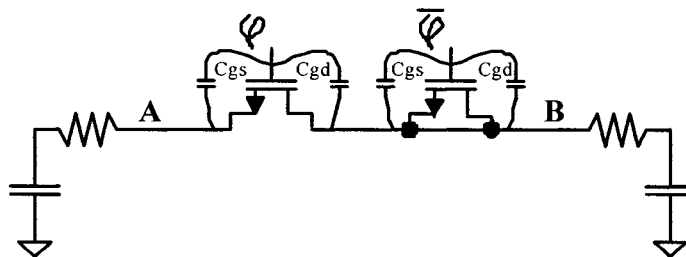


Figure 3.3: Injection de charges et excursion de l'horloge.

La quantité de charges accumulée sur la grille lorsque le transistor est ON est donnée par l'équation 3.2. Lorsque le transistor se ferme, la quantité de charges devant être absorbée

par les *dummies* est donnée par l'équation 3.3 pour les transistors nMOS et l'équation 3.4 pour les transistors pMOS.

$$Q = WLC_{ox}(V_{gs} - V_{th}) \quad (3.2)$$

$$Q_N = \frac{WLC_{ox}(V_{dd} - V_{in} - V_{th})}{2} \quad (3.3)$$

$$Q_P = \frac{WLC_{ox}(V_{in} - V_{th})}{2} \quad (3.4)$$

Suite aux explications de la section 2.3.3, rappelons que la séquence des opérations de comparaison est la suivante:

- ϕ_1 et ϕ_2 ouvert (OFF); *valide* fermé (ON); système au repos;
- ϕ_1 se ferme; *valide* s'ouvre; départ de la comparaison;
- ϕ_1 s'ouvre; possibilité d'injection de charges;
- ϕ_2 se ferme; la sortie tend vers son état final;
- *valide* se ferme; ajout de précision par feedback positif;

Le seul moment où il risque d'y avoir injection de charges est lorsque ϕ_1 se ferme. Des *dummies* doivent donc être ajoutés à la porte de transmission activée par ϕ_1 ainsi qu'à la boucle de feedback, elle aussi activée par ϕ_1 , reliant l'entrée et sortie de l'inverseur A_1 .

3.2.3 Design des obturateurs et loquet

Les obturateurs sont les éléments électroniques qui permettent de séparer la photodiode du reste du circuit du pixel numérique. Au niveau circuit, ils se doivent d'isoler la tension variant à la photodiode de celle se retrouvant à la grille du suiveur afin de garder cette dernière constante pour la conversion A/N. L'implémentation au niveau transistor (Figure 3.4) consiste à concevoir des portes de transmission à l'aide de transistors nMOS et

pMOS. Afin de limiter les injections de charges de ces portes, des dummies ont été ajoutés selon la méthode présentée dans la sous-section précédente. Le substrat des transistors de type pMOS et nMOS est connecté à V_{dd} et à V_{ss} , respectivement, sauf lors d'indications contraires.

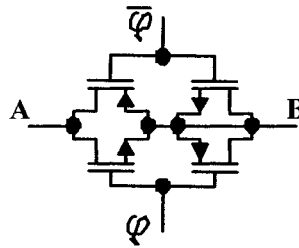


Figure 3.4: Transistors d'une porte de transmission.

Afin d'obtenir une fonction de transfert linéaire pour les portes de transmission, on doit en vérifier le fonctionnement à l'aide d'un circuit comme celui de la Figure 3.5. Ceci est important afin d'éviter de causer la distorsions du signal de sortie.

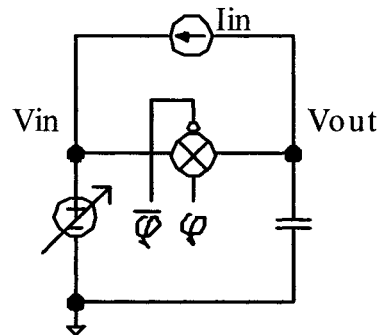


Figure 3.5: Optimisation d'une porte de transmission.

Avec ce circuit, on fait varier la tension V_{in} en appliquant un courant de retour I_{in} . Ensuite, on calcule la variation de la résistance en fonction de la tension d'entrée (V_{in}). Cette variation est un indice de non-linéarité qui peut causer des distorsions dans le signal transmis. Dans notre circuit, la non-linéarité n'est pas une priorité puisque les tensions transférées ne varient pas rapidement et c'est le transfert des charges qui

importe. Par contre, ce circuit peut aussi être utile afin de mesurer la résistance de la porte de transmission au travers de laquelle le signal devra passer.

3.2.4 Loquet pour l'obturateur électronique

La caractéristique du circuit qui permet d'obtenir une très large gamme dynamique est l'adaptation locale de l'exposition à la lumière. Ceci résulte en de multiples temps d'exposition différent d'un pixel à l'autre. La méthode des expositions multiples se traduit au niveau du circuit par le loquet de l'obturateur. Présent dans chaque pixel, il permet d'arrêter la variation de tension à la sortie de la section analogique. Ce loquet contrôle un second obturateur placé en série avec le premier qui est contrôlé par un signal global à la matrice. Les deux ont la même fonction qui est de garder la valeur analogique du pixel à un niveau constant. La différence est que l'obturateur contrôlé par le loquet est propre à chaque pixel, tandis que le premier obturateur est toujours activé en même temps pour toute la matrice. Cette adaptation locale à l'aide du loquet donne à ce capteur une ressemblance avec le fonctionnement de l'œil humain.

Du côté du fonctionnement du loquet, il est activé lorsque la tension du pixel dépasse 50% de sa valeur maximale (signal *compar* activé) et sa colonne est sélectionnée (signal *colSel* activé). Il est en fait une mémoire statique qui ouvre (état OFF) une des portes de transmission (obturateur). Le loquet est remis dans son état initial par le signal *clr*, global à la matrice, avant la capture d'une nouvelle image. Cette cellule mémoire est illustrée à la Figure 3.6.

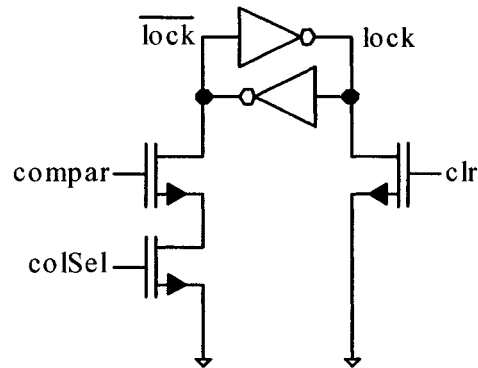


Figure 3.6: Mémoire statique pour le loquet.

La cellule de mémoire statique fournit directement les signaux nécessaires à la porte de transmission puisque cette dernière a besoin du signal d'activation et de son inverse pour être contrôlée. Un autre aspect très intéressant de cette cellule est qu'il commande un circuit 3.3V à partir d'un circuit 1.8V. À première vue, cette fonctionnalité peut sembler banale, mais contrairement à la conversion inverse, la solution est non triviale. Aucun circuit contrôlé par de signaux de 1.8V ne peut générer directement des sorties numériques de 3.3V de façon efficace (sans pertes). La solution à la conversion 1.8V – 3.3V est donc d'utiliser une cellule mémoire comme celle de la Figure 3.6 implémentée avec seulement des transistors de 3.3V. Étant donné que la conductivité des transistors nMOS (signaux *clr*, *colSel* et *compar*) est environ quatre fois plus grande que celle des pMOS, l'activation d'un transistor nMOS par une tension inférieure à 3.3V est suffisante pour faire basculer la boucle de rétroaction de 3.3V. Avec la rétroaction positive de cette boucle, le système est toujours dans un état de valeur numérique valide, avec une sortie soit à 0V ou bien à 3.3V. Pour la conversion inverse, il suffit d'utiliser un circuit tel celui de la Figure 3.7 où les astérisques (*) dénotent l'utilisation de transistors 3.3V.

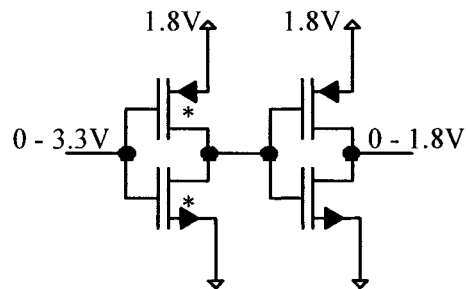


Figure 3.7: Conversion 3.3V à 1.8V.

3.2.5 Dessin des masques du pixel

Chaque pixel contient un total de 33 transistors, soit 20 petits (1.8V) et 13 gros (3.3V). Les transistors 3.3V sont dans la portion de gauche du schéma de la Figure 3.8. Les transistors pMOS des inverseurs du comparateur sont quatre fois plus larges que les nMOS afin de ramener le seuil de ces inverseurs près de $V_{dd}/2$. En fait, il y a quatre transistors pMOS en parallèle pour chacun des deux inverseurs, mais la Figure 3.8 en illustre un seul pour des questions de lisibilité. Le schéma du loquet de l'obturateur est celui de la Figure 3.6 qui contient sept transistors. Le total des 33 transistors par pixels est donc composé des 20 transistors de la Figure 3.8, des six transistors pMOS non-illustrés des inverseurs ($8 - 2 = 6$) et des sept transistors du loquet (non-illustré). La photodiode représente 29% de la surface totale de $13.8 \times 13.8 \mu\text{m}^2$ du pixel représenté à la Figure 3.9.

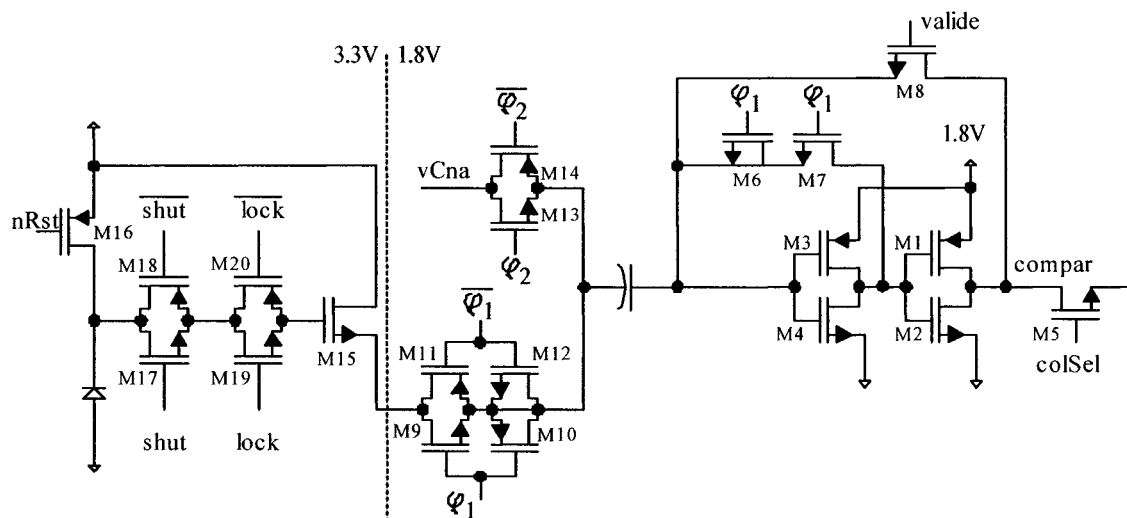


Figure 3.8: Dessin du pixel au niveau transistors.

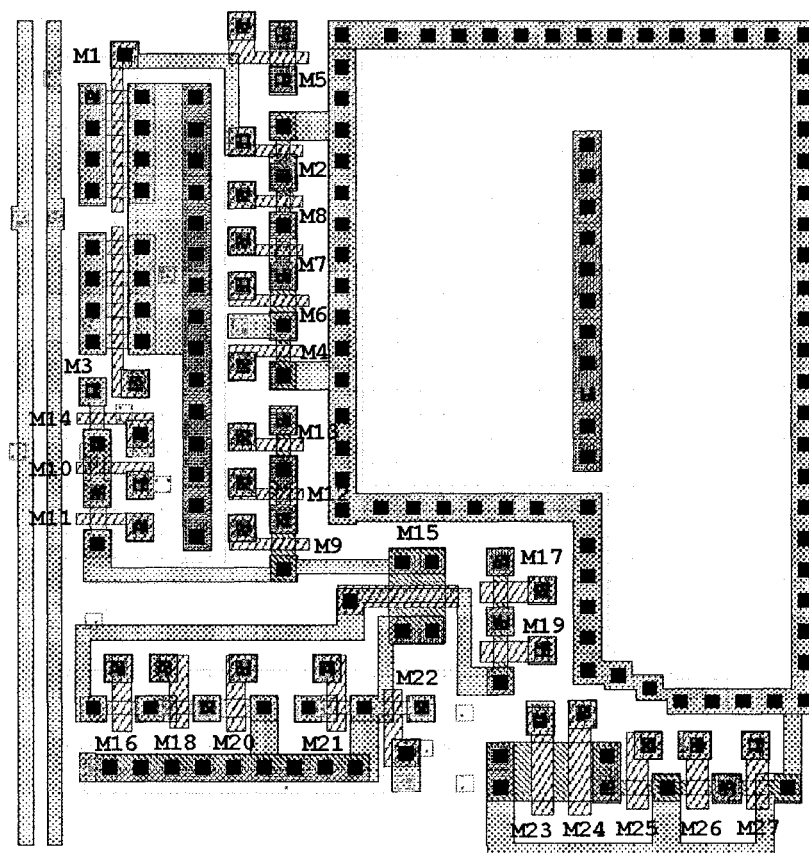


Figure 3.9: Dessin des masques du pixel.

La Figure 3.9 ne contient pas le dessin des masques des interconnexions des métaux de la couche 2 à la couche 6 pour fins de lisibilité. Le DPS avec ses couches de connexions est présenté à l'Annexe B. Les couches de métaux 2 et 3 sont utilisés pour les connexions internes au pixel, le métal 4 transporte l'alimentation de 1.8V, tandis que les couches 5 et 6 constituent le condensateur nécessaire au comparateur (voir Figure 3.2). Le métal 4, transportant une tension toujours constante, est nécessaire au fonctionnement de la capacité du comparateur en l'isolant du reste des signaux du circuit.

3.3 Conception de la puce

Maintenant que les détails du pixel numérique ont été explorés, il ne reste plus qu'à ajouter les mémoires et les connexions. La complexité ayant été entièrement intégrée au pixel, ce qui reste à concevoir est donc minime avant d'assembler le tout dans le but d'obtenir un capteur complet.

Le capteur résultant peut paraître simpliste sans plusieurs circuits de support l'entourant. On pourrait bien présenter une multitude de fonctionnalités, mais la matrice DPS a la flexibilité d'être manipulée à la guise de l'utilisateur lors de la conception puisqu'elle est composée de blocs modulaires, tous indépendants les uns des autres. Par exemple, une variation de la résolution spatiale peut être souhaitable d'une application à l'autre, mais pas d'un jour à l'autre pour une même application. Il est donc inutile de dédier une fonctionnalité à cet effet. À partir des blocs de base, soit le pixel et la cellule mémoire, il serait facile d'implanter un processus automatique qui construirait le circuit complet du capteur de la même manière que fonctionne un générateur de mémoire. Par la suite, l'utilisateur pourrait choisir le mode d'opération ainsi que les détails de conversion et de manipulation des données dans un circuit numérique programmable. Il serait aussi possible d'automatiser la génération d'un tel circuit. Au niveau du contrôle, le DPS contient tous les éléments nécessaires pour permettre au contrôleur numérique d'effectuer

le type de conversion A/N désirée. De cette façon, l'optimisation en vitesse et précision, ainsi que l'ajout de fonctionnalités, ne dépend que du contrôleur.

3.3.1 Matrice de pixels

Afin de minimiser l'espace entre les pixels, l'avantage des six métaux disponibles, avec le procédé CMOS 0.18 μ m a été utilisé à profit. La tâche d'amener quinze signaux à chaque pixel n'est pas triviale lorsque l'espace est restreint comme dans le cas d'un capteur d'images. La Figure 3.10 rappelle les signaux allant vers chaque pixel. Les noms de ceux situés du côté gauche du schéma contrôlent le pixel, ceux du haut pour la section analogique et ceux du bas pour la section numérique. Le signal *colSel* active les colonnes une à une afin de lire la sortie résultante sur la ligne *lineBus*. Tous ces signaux (à l'exception du *lineBus*) atteignent le pixel selon une distribution verticale, tandis que les autres se distribuent de façon horizontale. Les signaux marqués d'une astérisque (*) identifient ceux ayant 3.3V pour leur niveau logique haut, et ceux marqués d'un « *a » identifient les signaux analogiques et les alimentations. Les signaux *colSel* et *lineBus* sont propres à chaque colonne et ligne respectivement, tandis que les autres sont globaux, c'est-à-dire communs à toute la demie-matrice. Les signaux de contrôle globaux sont doublés, un pour chaque portion de la matrice coupée en deux pour des questions de pipelining discutées au chapitre précédent.

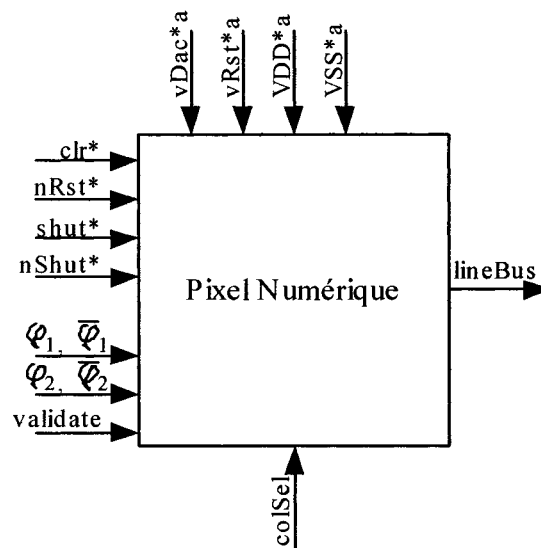


Figure 3.10: Signaux nécessaires à chaque pixel.

3.3.2 Design des mémoires

L'architecture présentée peut fonctionner avec ou sans mémoire embarquée sur la même puce que le capteur, tel qu'expliqué aux sections 2.2.1 et 2.2.2. Dans le cas où deux mémoires sont utilisées, une première constituée d'un seul bit par pixel contrôle l'écriture dans la deuxième, tandis que cette deuxième est constituée d'autant de bits par pixel que la précision de l'application le requiert pour mémoriser la valeur finale de chaque pixel.

Dans la première version de la puce ICFPMJL1, les deux mémoires ont été implantées sur la puce. Le dessin des masques de cette puce est présenté à l'Annexe A. La mémoire temporaire comportait 10-bits par pixel pour une matrice de 32x32 pixels, résultant en une mémoire de 10.24 kbits ou 1.28 koctets. Étant donné cette quantité de mémoire non négligeable, un design optimisé d'une cellule de mémoire statique à deux ports a été fait pour répondre aux exigences de fonctionnalité et dimensions restreintes (Figure 3.11).

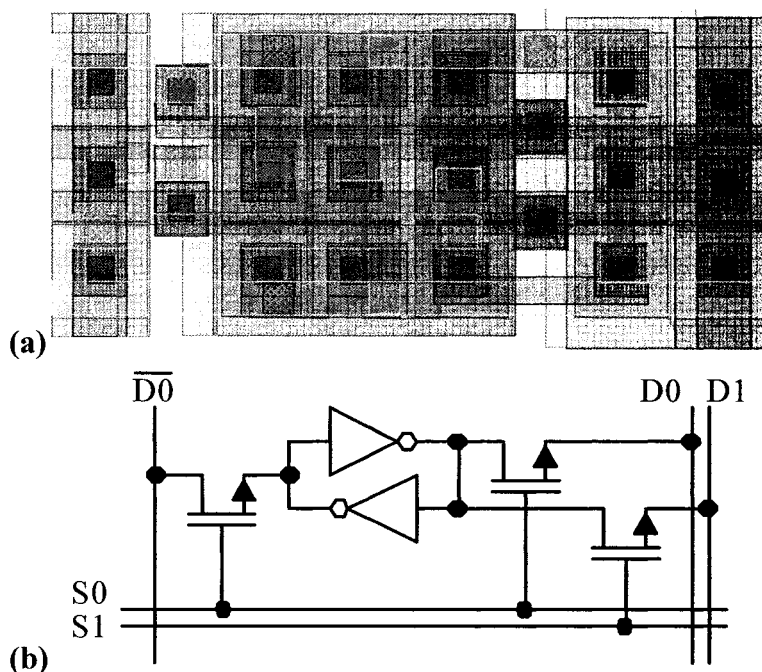


Figure 3.11: Cellule de mémoire statique à deux ports
(a) dessin des masques; (b) schéma des transistors.

La cellule de la Figure 3.11a (avant rotation de 90°) est très étroite par rapport à sa hauteur pour permettre aux dix bits de s'insérer les uns à côté des autres. Ceci sans causer des dimensions rectangulaires 10:1 qui résulterait en une mémoire trop large, empêchant ainsi son intégration adéquate sur une puce.

Le double port de la cellule mémoire a été conçu pour permettre sa lecture à partir du contrôle extérieur pendant l'écriture interne des valeurs de pixels dans celle-ci. Les ports S_0 et D_0 de la Figure 3.11b sont utilisés pour écrire de nouvelles valeurs dans la mémoire, tandis que la lecture se fait par D_1 après sélection de la ligne S_1 .

La deuxième version de la puce ICFMJL1, ne contient pas de mémoire temporaire par manque d'espace d'une part, et pour obtenir d'autre part une plus grande flexibilité côté performances et capacité lorsque réalisée à l'extérieur de la puce. Seule la mémoire intermédiaire à 1-bit est intégrée sur la puce. Afin de faciliter son contrôle, elle a été

réalisée à l'aide d'un multiplexeur et d'un transistor de sélection, tel qu'illustré à la Figure 3.12.

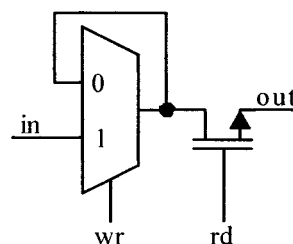


Figure 3.12: Cellule mémoire avec multiplexeur.

Pour relier la matrice de pixels à la mémoire intermédiaire, un circuit d'entrée achemine les valeurs des signaux de sortie de la matrice (signaux *lineBus* de chacune des lignes) via un *buffer* pour accélérer la transmission du signal sur ces longues lignes capacitives. Le circuit d'entrée permet aussi de forcer le niveau logique désiré à l'entrée de cette matrice afin de valider le fonctionnement de cette dernière sans l'utilisation de la sortie de la matrice de pixels de la Figure 2.2.

3.3.3 Module de test

L'élément principal visant à être validé par la fabrication de la présente puce est le pixel numérique. Ce pixel étant composé de trois blocs de base, des cellules de test ont été créées afin de tester chacun de ces blocs. Le dessin des masques du pixel numérique a donc été subdivisé de manière à accéder indépendamment à la photodiode, au suiveur et au comparateur. Les cellules suivantes ont donc été ajoutées à la puce pour fins de tests.

- 1) 1 photodiode + suiveur + comparateur
- 2) 2 photodiodes + suiveur + comparateur
- 3) 1 photodiode + suiveur
- 4) 2 photodiodes + suiveur

- 5) 1 suiveur
- 6) 1 suiveur + 1 comparateur
- 7) 1 comparateur

Les cellules 1 et 3 comportent une photodiode afin de caractériser les paramètres photoélectriques en comparant ces résultats à ceux des cellules 2 et 4 comportant deux photodiodes. Les cellules 5 et 6 permettent d'appliquer la tension désirée en entrée afin de caractériser le circuit avec des variations de tension en entrée différentes de celles provenant de la photodiode. La cellule 7 permet de tester le comparateur indépendamment du reste du circuit. Ces trois dernières cellules permettront de vérifier le fonctionnement du pixel advenant un problème photoélectrique au niveau de la photodiode qui n'a pu être validé par simulations.

3.3.4 Interconnexions des blocs et puce complète

La matrice du capteur numérique réalisée sur la puce ICFPMJL1 comporte 128 colonnes et 54 lignes de pixels. Les informations de contrôle entrent par le bas des colonnes et les données résultantes sortent par le côté droit des lignes. Étant donné qu'il est impossible d'avoir un aussi grand nombre de broches sur la puce qu'il y a de lignes et de colonnes, des décodeurs et multiplexeurs sont nécessaires. À l'entrée, un décodeur achemine le signal d'activation des colonnes à la bonne colonne en fonction de sept lignes d'adresses. Les autres signaux de contrôle de la partie numérique des pixels (Figure 3.10) sont disponibles en double, les premiers pour la moitié gauche de la matrice et les seconds pour sa moitié droite. La raison pour laquelle ces signaux sont doublés est qu'ils permettent une forme de pipelining des opérations de conversion A/N et celles de lecture.

En sortie, des multiplexeurs sont nécessaires afin d'acheminer séquentiellement les signaux vers les broches de sortie. Puisque quatre broches de sorties sont disponibles, il y a donc quatre multiplexeurs 16 : 1 placés en parallèle et contrôlés par quatre lignes

d'adresses. L'implémentation des circuits a été réalisée à la main avec les cellules numériques disponibles dans la librairie. Le schéma de la puce finale fabriquée, ICFPMJL1.v2, est présenté à la Figure 3.13.

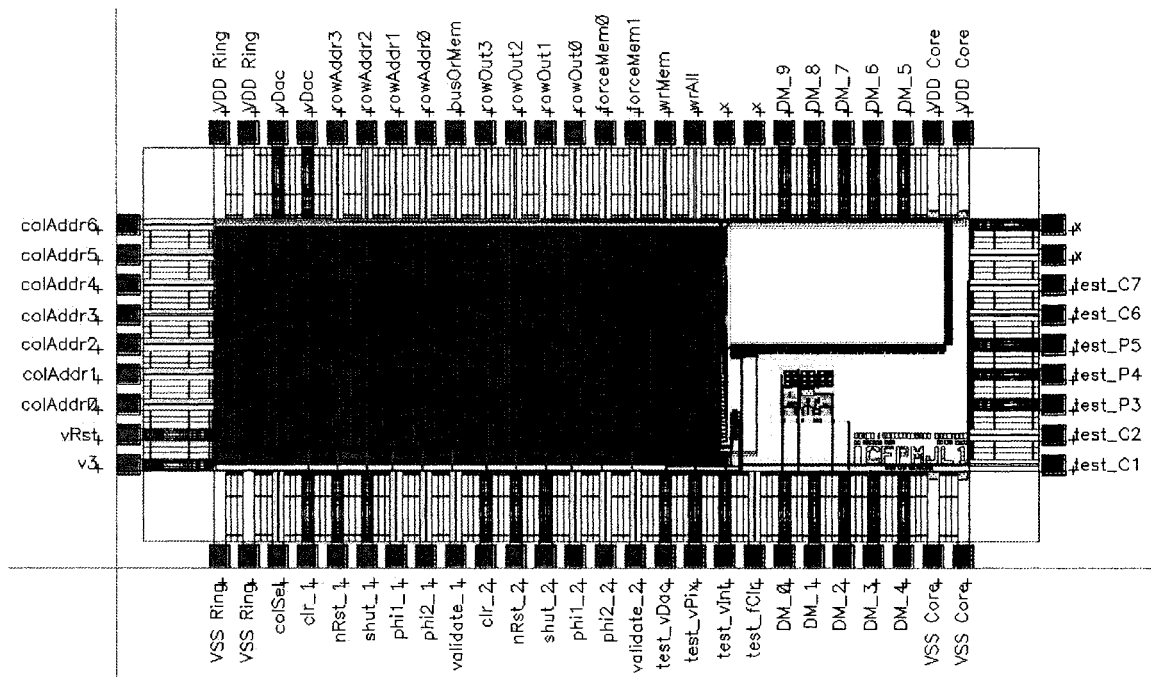


Figure 3.13: Dessin des masques de la puce ICFPMJL1, deuxième version.

Un autre aspect à gérer dans le contrôle des pixels est que certains signaux doivent être limités à 1.8V tandis que d'autres doivent atteindre 3.3V. Une attention particulière doit donc être apportée dans le choix des transistors d'amplification des différents signaux de chacune des colonnes. Le contrôle numérique externe communiquant avec la puce se fait à 3.3V. Les signaux devant avoir une tension de 3.3V sont alors connectés vers les broches de la puce par un plot analogique. De cette façon, la tension de 3.3V n'est pas modifiée avant d'être envoyée au circuit interne de la puce. Contrairement à cette méthode, les signaux du pixel devant recevoir un niveau de tension de 1.8V sont connectés aux broches via des plots numériques. Ceux-ci font la conversion 3.3V – 1.8V à l'interne. Tous les signaux de contrôle doivent être amplifiés et/ou inversés avant d'être envoyés à la matrice de pixels. Ces circuits doivent être réalisés à la main pour les

signaux fonctionnant à 3.3V car les cellules de portes logiques disponibles dans la librairie numérique ne supportent qu'une tension de 1.8V.

3.4 Système de test

La puce à tester étant complexe, il est nécessaire de concevoir un système de test autour de celle-ci afin de pouvoir en évaluer tous les paramètres. Ce système doit avoir la capacité d'envoyer des séquences complexes de signaux numériques sur plusieurs canaux simultanément. De plus, il doit être possible de faire varier les signaux des différents canaux analogiques afin de stimuler les entrées des modules analogiques de test de la puce ainsi que de fournir les tensions de référence pour les conversions A/N.

Il a été décidé de concevoir et de fabriquer un tel système puisque les outils de développement, le réceptacle pour la puce dans un boîtier 84-PGA et les alimentations propres à la technologie CMOS 0.18 μ m n'existaient pas. Les sorties numériques seront évaluées à partir du contrôleur numérique pour ensuite être acheminées vers un ordinateur qui collectera les données. D'autre part, les sorties analogiques seront analysées à partir d'un oscilloscope analogique puisque le contrôle d'une puce de conversion A/N pour lire les données vers un ordinateur nécessite un travail qui n'est pas nécessaire et trop complexe pour ce que l'on désire mesurer.

Afin de mesurer les paramètres optiques du capteur, des sources lumineuses émettant une longueur d'ondes précise seront utilisées. L'émission de quatre longueurs d'ondes différentes sera réalisée à partir de quatre diodes électroluminescentes (DEL) très intenses: une bleue (465 nm), une verte (565 nm), une jaune (590 nm) et une rouge (640 nm). Leur intensité sera mesurée à l'aide du wattmètre *FieldMaster* de la compagnie *Coherent*, présenté à la Figure 3.14. À partir des résultats de ces mesures, il nous sera possible de vérifier si le capteur possède une réponse acceptable et capable de représenter une scène telle que perçue par l'œil humain.

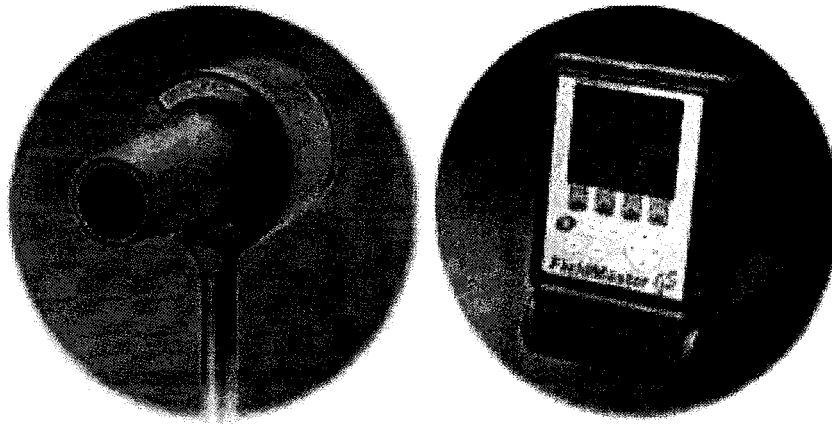


Figure 3.14: Appareil de mesure d'intensité lumineuse.

3.4.1 Circuits de support

Le circuit conçu pour tester la puce comporte trois principales sections: l'alimentation du système, une interface analogique et un contrôleur numérique. Le tout est assemblé sur un même circuit imprimé qui contient les interfaces physiques nécessaires aux connections des composantes externes. Ce système est illustré à la Figure 3.15 vue de face, et vue de derrière à la Figure 3.20. Le dessin de masques de ce circuit imprimé de test à une seule couche est donné à l'Annexe C.

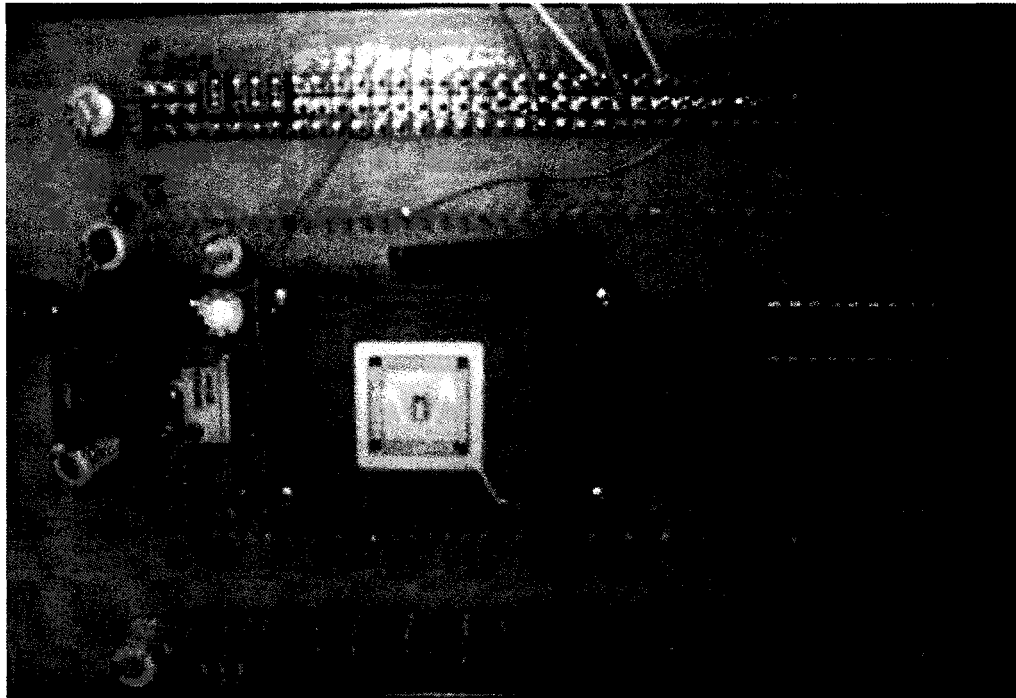


Figure 3.15: Banc de test.

Le détail du circuit et du fonctionnement des différentes puces de support sera présenté dans les sous-sections suivantes. Toutes les broches qui ressortent vers le haut du circuit imprimé ont été disposées de cette façon afin de permettre un prototypage rapide à l'aide de « *wire-wrap* ». Les deux rangées horizontales de broches les plus rapprochées du milieu de la plaquette font partie du réceptacle arrière pour le circuit numérique. Ensuite, au haut et au bas de la plaquette, il y a trois rangées de broches avec celles du centre directement reliées à chacune des broches du réceptacle pour la puce. Les deux autres rangées portent les signaux d'alimentation (V_{dd}) et de masse (Gnd). Selon que l'on veuille tester une partie ou une autre de la puce, il y a toujours plusieurs entrées (parmi les 84) qui doivent être connectées soit à la masse, soit à un niveau logique haut. La disposition des trois rangées permet ici de connecter ces signaux sans perdre de temps à l'aide de cavaliers (*jumpers*).

3.4.1.1 Alimentation

Il y a quelques années, le standard pour l'alimentation des circuits numériques était de 5V. Suite à la réduction d'échelle des procédés de fabrication et afin de diminuer la consommation de puissance des circuits, ce standard est passé à 3.3V. Aujourd'hui, avec la taille minimale des transistors qui diminue encore plus, ce niveau de tension d'alimentation des circuits numériques est maintenant sous les 3.3V. Afin de préserver une certaine compatibilité entre les circuits de nouvelle génération, le cœur du circuit a la possibilité d'être alimenté par une tension inférieure à 3.3V, par contre les entrées/sorties communiquent encore à 3.3V.

L'alimentation du banc de test doit donc être capable de fournir des niveaux de tension de 5V, de 3.3V ainsi qu'une autre variable sous les 3.3V. À partir d'une entrée standard à centre positif, des régulateurs fournissent les niveaux de tension désirés. Des cavaliers permettent de choisir lequel des trois niveaux doit aller sur chacune des deux rangées du milieu des broches d'alimentation. Le circuit réalisant cette fonctionnalité est présenté à la Figure 3.16.

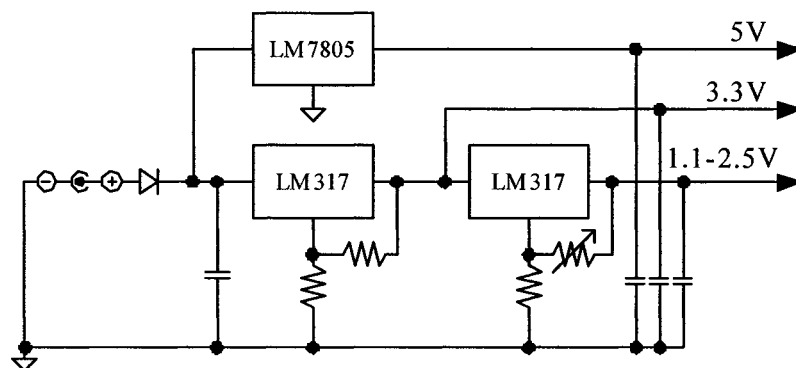


Figure 3.16: Circuit d'alimentation.

3.4.1.2 Ajusteur de niveau logique

La section précédente a expliqué la nécessité des trois alimentations présentes sur le circuit du banc de test. Bien que les différentes puces récentes communiquent à 3.3V, les plus anciennes sont toujours à 5V. Afin de permettre à ces deux générations de communiquer dans un même circuit, des ajusteurs de niveau logique sont nécessaires. De la même façon que l'ajusteur de niveau 1.8V – 3.3V intégré à même la puce ICFPMJL1, le convertisseur MAX3379, de la compagnie *Maxim-Semiconductors*, ajuste les niveaux entre 3.3V et 5V. Ceci est nécessaire dans notre circuit car une puce du circuit fonctionne à 5V, contrairement au reste du circuit qui fonctionne à 3.3V.

3.4.1.3 CNA rapide

Un des éléments essentiel à l'architecture proposée est un CNA. Ce module fournit la référence de tension qui est comparée au niveau de tension atteint par chaque pixel pour effectuer la conversion analogique à numérique. Cette conversion doit être effectuée le plus rapidement possible pour obtenir de courts temps d'exposition (limités par le temps de conversion). Le CNA doit donc être très rapide afin de balayer chacun de ses niveaux de tension à l'intérieur d'un court laps de temps. La puce choisie pour répondre aux exigences est le MAX5180 de la compagnie *Maxim-Semiconductors*. Il s'agit d'une des seules puces sur le marché, à ce jour, capable de fournir un taux de conversion de 40 Méga échantillons par seconde, avec une sortie en mode tension et une précision de 10-bits. Le fonctionnement se résume à présenter les 10-bits du niveau de tension à convertir, puis à chaque front d'horloge la puce ajuste la tension de sortie en fonction de l'entrée numérique.

3.4.1.4 CNA multicanaux

Dans la sous-section précédente, nous avons vu la nécessité d'avoir une référence de tension qui varie rapidement afin d'effectuer les comparaisons successives de la conversion A/N. Cependant, la puce ICFPMJL1 nécessite d'autres références de tension, notamment pour l'alimentation de la portion analogique du pixel (V_{rst}) et pour des modules de test. Ces niveaux de tension ne doivent pas varier rapidement, mais plusieurs canaux sont nécessaires. Au lieu d'ajuster manuellement des potentiomètres, l'utilisation d'un CNA à plusieurs canaux peut s'avérer plus utile. L'interface numérique de ce CNA se commande par un contrôleur numérique, ce qui peut éviter les maux de tête liés à des ajustements manuels. Le schéma de cette puce est présenté à la Figure 3.17.

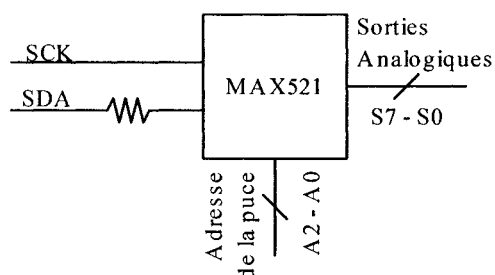


Figure 3.17: Schéma du CNA multicanaux.

Le contrôle de ce CNA se fait de façon sérielle. Après un signal de début de commande, la commande est envoyée, suivie de l'adresse de la puce, puis de celle du canal de sortie, ensuite c'est la valeur numérique de la tension de sortie désirée et finalement le signal de fin de commande est donné. Une portion de cette séquence est illustrée à la Figure 3.18 et la description VHDL du contrôleur est donné à l'Annexe D. Toute les commandes sont des mots de 8-bits, envoyés du MSB au LSB, suivis d'une réponse (*acknowledgment*) du MAX521 qui force la ligne de donnée (SDA) à zéro pour la neuvième période haute de l'horloge (SCK). L'adresse de la puce dépend de ses trois lignes d'adresses et est encodé selon la forme $0101A_2A_1A_00$. Le mot de commande qui encode la sélection du canal de

sortie est de la forme $00000S_2S_1S_0$ et sa valeur de tension désirée s'en suit sur un autre mot de 8 bits.

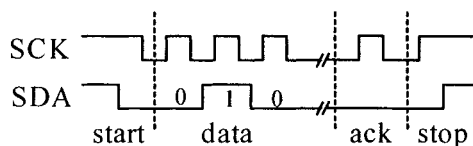


Figure 3.18: Signaux de contrôle pour le CNA multicanaux.

3.4.1.5 Amplificateur opérationnel

Dans ce circuit, les deux amplificateurs opérationnels contenus sur la puce MAX4249, sont reliés à des broches qu'il est possible de connecter à n'importe quel endroit désiré par le concepteur. Cependant, le but de placer ces amplificateurs sur le circuit imprimé est d'ajuster les niveaux de tension (équation 3.5) provenant du CNA rapide. Celui-ci a une sortie en différentiel que l'on pourrait vouloir ramener entre deux autres valeurs arbitraires. Le schéma de ce circuit est illustré à la Figure 3.19.

$$V_{out} = \frac{R_b}{R_a} (V_{in+} - V_{in-}) + V_{off} \quad (3.5)$$

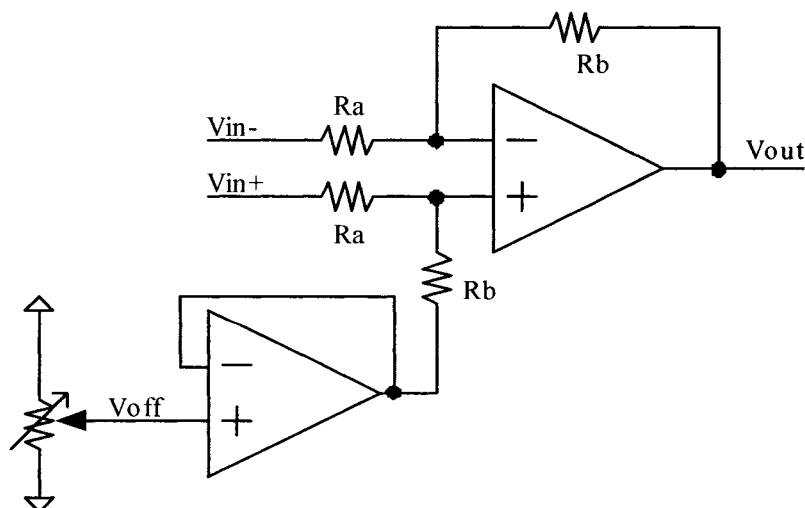


Figure 3.19: Ajustement de tension avec deux amplificateurs opérationnels.

Avec le circuit de la Figure 3.19, il est possible de ramener la tension différentielle d'entrée entre n'importe quels niveaux de tension à la sortie. Il suffit de vérifier quelles est la plage d'opération de l'entrée (V_{in}) pour calculer le gain que l'on doit y apporter pour en arriver à la plage de tension de sortie désirée (équation 3.5). Ensuite, une tension de décalage (V_{off}) est calculée afin de mener la valeur de la tension minimale en entrée vers celle désirée en sortie. Pour ajuster la sortie de notre CNA rapide, avec chacune de ses sorties différentielles allant de 0 à 400mV, nous devons obtenir un ratio de 4.125 pour R_b/R_a et un V_{off} de 1.65V afin d'obtenir une sortie allant de 0 à 3.3 volts.

3.4.2 Contrôle numérique

Maintenant que le circuit de support est complet, avec le réceptacle pour la puce à boîtier 84-PGA, les alimentations et le circuit analogique, il ne reste plus que le contrôle numérique pour gérer tout ce système. Comme il a été possible de le voir à la Figure 3.15, deux rangées de broches reliées à des prises femelles ont été installées spécialement pour recevoir la partie numérique du système. Cette partie est constituée d'une plaquette de développement de la compagnie Xess, qui contient comme éléments principaux un FPGA (*Field Programmable Gate Array*) Xilinx Spartan IIe avec une capacité de 100 000 portes ainsi qu'un CPLD (*Complex Programmable Logic Device*) Xilinx XC9572XL. Ce système est illustré à la Figure 3.20.

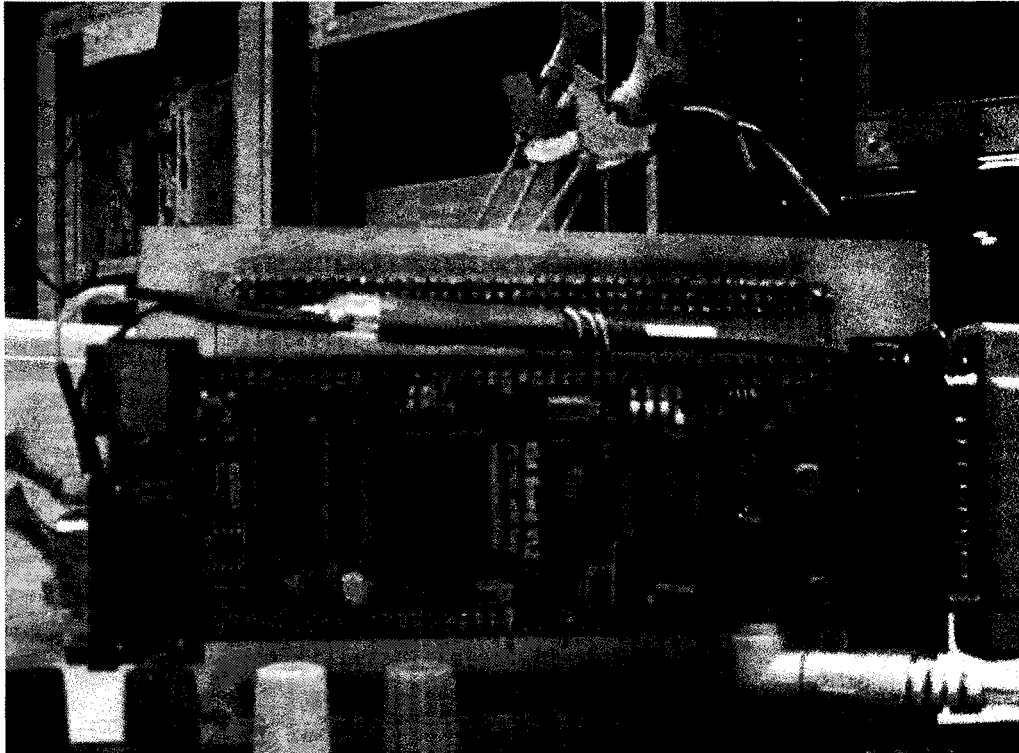


Figure 3.20: Contrôle numérique avec FPGA.

Un port parallèle est inclus sur la plaquette de développement afin de communiquer et programmer le FPGA et le CPLD. La description VHDL du code téléchargé dans le contrôleur FPGA est présentée à l'Annexe D. La première version du code développé pour ce projet comportait un processus de communication bidirectionnel entre l'ordinateur et la carte de développement, mais suite à des erreurs fréquentes de communication, ce code a été remplacé par une sélection manuelle du programme et du mode de test désiré. Après discussion avec l'ingénieur en chef de *Xess*, Dave Van der Bout, il a été confirmé que l'erreur était dans le CPLD (programmé par *Xess*) qui assure la communication entre le FPGA et le port parallèle sur la plaquette de développement. L'alternative de changer le code s'est donc avéré la bonne solution.

Ce circuit numérique programmable contrôle le circuit analogique de support de la manière présentée ci-dessus. Il contrôle aussi la puce principale en lui envoyant les

signaux décrits à la Figure 2.3, les 18 (2*9) signaux des pixels numériques décrits à la Figure 3.10, ainsi que l'adressage des décodeurs et multiplexeurs.

3.5 Conclusion

Dans ce chapitre, nous avons vu les détails de l'implémentation du pixel numérique qui a concrétisé les concepts présentés dans le chapitre précédent. Les subtilités au niveau du circuit ont été décrites ainsi que l'optimisation du dessin des masques qui a rendu le pixel de 33 transistors assez compact pour occuper une superficie de $13.8\mu\text{m}$ de côté, incluant une photodiode dont le facteur de remplissage est de 29%.

En comparant le capteur résultant du nouveau pixel numérique présenté dans ce mémoire et un capteur standard APS [9], il est possible de quantifier le gain qui est apporté par la nouvelle architecture. Pour une matrice de 128×128 , le capteur à base de DPS requiert une surface de $3.12\mu\text{m}^2$, tandis que le capteur à base de APS requiert une surface de $3.30\mu\text{m}^2$ où une taille équivalente à 40% de la taille de la matrice est nécessaires aux convertisseurs analogiques à numériques. Avec le capteur DPS, ces convertisseurs sont inclus dans chaque pixel, réduisant ainsi la surface utilisée en plus de permettre une conversion A/N en parallèle pour tous les pixels. Ceci permet aux matrices DPS d'être de plus grandes tailles que celles APS puisque les contraintes sur les convertisseurs A/N ne sont pas un facteur limitant.

Le résultat final de la conception de la puce ICFPMJL1, incluant tous ses sous blocs ainsi que leurs interconnexions, a été décrit avec les méthodes de design de la technologie $0.18\mu\text{m}$. Ensuite, le banc de test permettant de recevoir le boîtier 84-PGA de la puce fabriquée a été présenté avec ses fonctionnalités. Bien que cette plaquette ait été conçue afin de tester la puce ICFPMJL1, la topologie avec le contrôleur numérique, les circuits intégrés analogiques et les broches reliant les composantes entre elles de façon générique,

à l'aide de « *wire-wrap* », laisse ouverte la possibilité de l'utiliser pour le test d'autres puces dans un boîtier PGA à 84 broches ou moins.

CHAPITRE 4

Mesures et résultats

4.1 Introduction

Cette section présente les résultats des tests effectués avec les appareils et le système de test présentés à la section 3.4. Ces résultats expérimentaux sont comparés aux simulations effectuées à l'aide du logiciel *Spectre* de *Cadence*.

Les nouveautés apportées par ce projet de capteur d'images sont en majorité au niveau de l'architecture du système. Certain des concepts ont été validés au niveau théorique dans le chapitre 2 et 3 et leur vérification expérimentale est rapportée dans ce quatrième chapitre. D'autres concepts ayant été développés après la fabrication de la puce ont été validés par simulation.

Suite aux résultats des tests effectués et aux idées survenues en cours de route, nous proposons des améliorations et de nouveaux concepts pour les futures versions de ce capteur d'images. Ces idées et voies à explorer dans le but d'améliorer le système sont effectuées à partir de l'architecture de base présentée dans cet ouvrage.

4.2 Performances du capteur

La puce contenant le capteur d'images du présent projet, nommée ICFPMJL1, a été conçue en deux versions. La première version n'a pas été fabriquée suite à des erreurs dans le module du contrôleur intégré où des cellules numériques distribuées sous forme de boîte noire ont été utilisées. La seconde version, ICFPMJL1.v2 présentée à la Figure 4.1, est une version hautement améliorée du ICFPMJL1.v1 avec un pixel complètement redessiné et optimisé. Dans cette seconde version, tout le contrôle numérique a été enlevé

afin de bénéficier de la flexibilité d'un contrôleur numérique programmable externe et ainsi éviter des erreurs non-réversibles rendant la puce inutilisable.

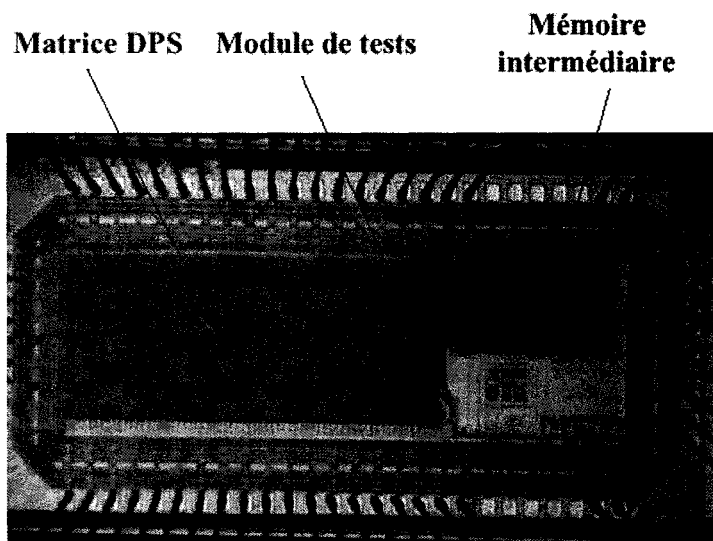


Figure 4.1: Photographies de la puce ICFPMJL1.v2.

Bien que les mesures nécessaires permettant d'éviter les erreurs aient été prises, des contraintes de fabrication ont imposé des modifications de dernière minute, causant quelques mauvais contacts et trois erreurs. La première erreur est la déconnexion de la ligne d'activation des colonnes utilisée pour la lecture de la matrice de pixels. Bien que ce ne soit qu'un contact manquant, cette erreur a pour effet d'empêcher toute lecture de la matrice de pixels. Malgré cette conséquence néfaste au fonctionnement du capteur, il y a toujours les modules de test qui pourront être utilisés afin de valider le fonctionnement des blocs intérieurs au pixel. Malheureusement, une deuxième erreur s'est glissée en court-circuitant les sorties des comparateurs des différents modules de test. Ces deux erreurs font qu'il est impossible de tester le fonctionnement du comparateur. Toutefois, une validation du fonctionnement de ce dernier sera présentée à la section 4.2.3. La troisième erreur est un court-circuit entre une entrée numérique et la tension de réinitialisation (V_{rst}) de la portion analogique du pixel. Le nœud V_{rst} ne peut donc pas être porté à une tension optimale (située entre 1.8V et 3.3V) qui maximise la gamme dynamique de la photodiode. Cependant, le fonctionnement du circuit touché par cette

erreur pourra tout de même être validé en fixant ce nœud à 1.8V. Ceci se fait en appliquant un niveau logique haut à l'entrée numérique avec laquelle ce nœud est court-circuité.

Les erreurs citées ci-dessus nous empêchent de lire l'image capturée par la matrice de pixels. Par contre, les modules analogiques intérieurs au pixel peuvent être testés, incluant la conversion photonique à électrique de la photodiode. Afin d'être validés, ces modules fonctionnels devaient absolument être intégrés sur une puce de silicium fabriquée à l'aide de la technologie CMOS 0.18 μm puisque le fonctionnement d'un circuit analogique (et optique) varie énormément d'une technologie à l'autre. D'un autre côté, les modules numériques du pixel, le transfert des données numériques ainsi que l'écriture dans les mémoires ne nécessitent pas une validation dans une technologie précise pour en prouver le fonctionnement puisqu'un circuit numérique peut être adapté d'une technologie à l'autre pour obtenir un fonctionnement à toutes fins pratiques identique.

4.2.1 Paramètres photoélectriques du capteur

Dans le but de concevoir un capteur d'images, la conversion de l'énergie photonique à électrique doit se faire à l'intérieur du spectre de la lumière visible. Par contre, lorsqu'une technologie standard est choisie, peu de latitude est possible du côté de l'optimisation de l'efficacité quantique. La littérature nous a indiqué qu'il était possible de capturer le rayonnement électromagnétique de longueur d'onde visible pour l'œil humain à l'aide de la technologie CMOS 0.18 μm . L'objectif était donc de caractériser cette sensibilité pour une puce fabriquée par la *Taiwan Semiconductor Microelectronics Corporation (TSMC)*, puisque les paramètres peuvent fluctuer légèrement d'un fabricant à l'autre.

La lumière visible ayant un spectre allant de 400nm à 700nm, quatre diodes électroluminescentes (DEL) de longueur d'onde différentes ont été utilisées afin de caractériser la réponse spectrale du capteur fabriqué (Figure 4.2).

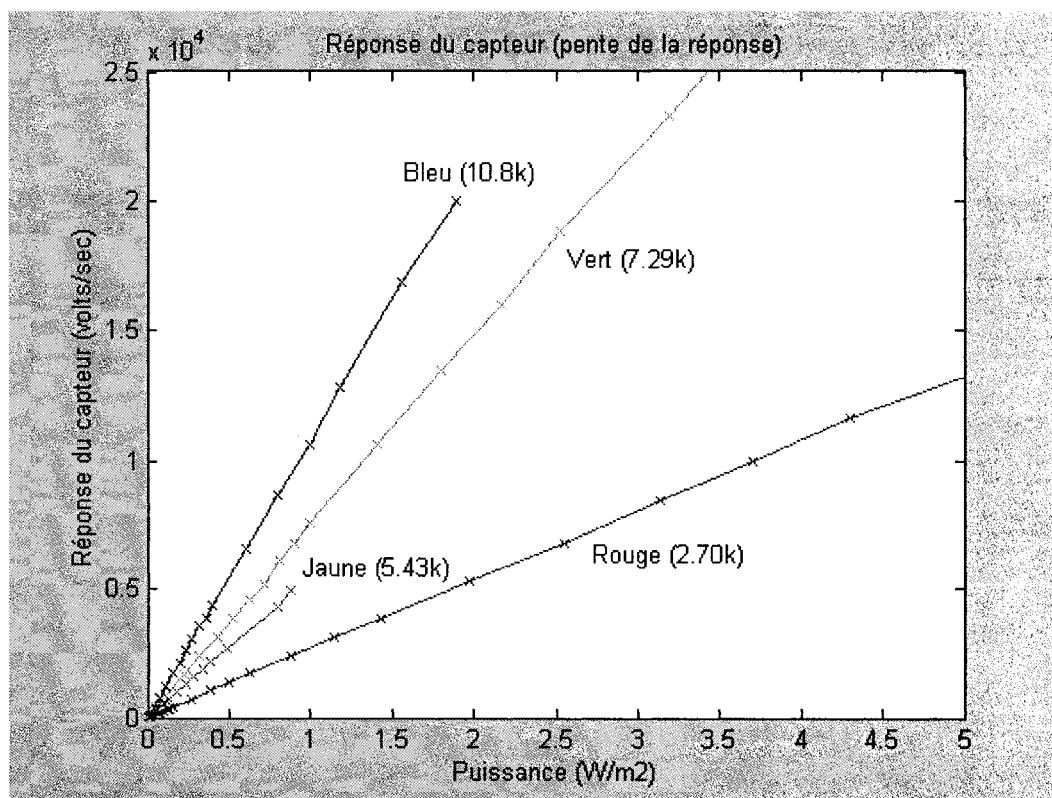


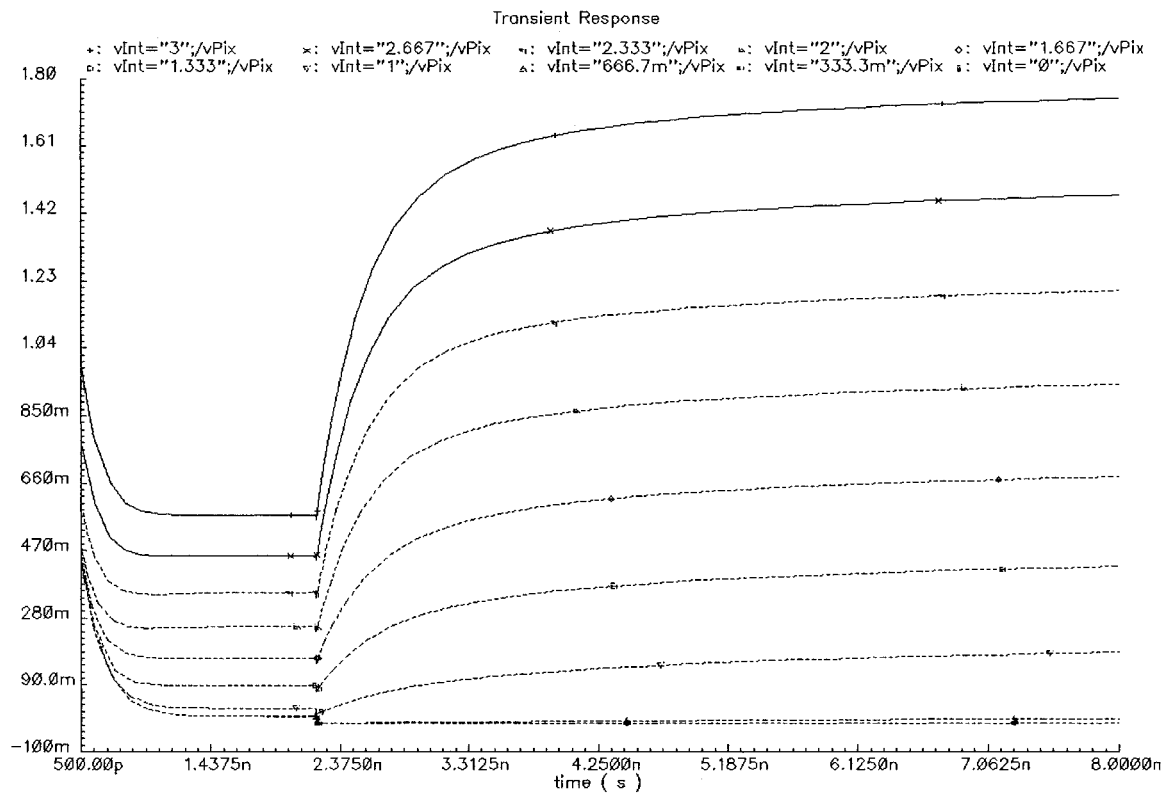
Figure 4.2: Réponse du capteur en fonction de la longueur d'ondes
 (a) bleu (465 nm); (b) vert (565 nm); (c) jaune (590 nm); (d) rouge (640 nm).

Il est intéressant de comparer la réponse du capteur à celle de l'œil humain qui est le plus sensible aux longueurs d'ondes de 500nm pour la vision photopique (vision de jour) et de 550nm pour la vision scotopique (vision de nuit). On peut alors voir que le capteur a la meilleure sensibilité au environ de 465nm (région du bleu), puis celle-ci décroît avec la longueur d'ondes qui augmente. Toutefois, le capteur a une bonne sensibilité dans la région utile à l'œil humain (500nm à 550nm). Le nombre entre parenthèses pour chaque trait de la Figure 4.2 correspond à la pente du signal, en volts* m^2 * sec^{-1} * watts $^{-1}$.

Des vérifications subséquentes à l'aide d'infrarouges (à partir d'une DEL) et d'ultraviolets (à partir d'une lampe à UV) ont permis d'affirmer que le capteur possède un spectre de sensibilité beaucoup plus large que celle de l'œil humain. Cependant, ceci ne signifie pas qu'il résulte en une image de qualité supérieure, mais seulement que ce capteur peut voir de la lumière que l'œil humain ne peut pas voir, notamment les infrarouges et les ultraviolets.

4.2.2 Performances du suiveur dynamique

En général, un transistor est utilisé à la source du suiveur (Figure 2.13) afin de polariser ce dernier par un courant avant d'y lire la tension. Cependant, lorsque seulement un échantillon à un instant donné est utilisé, il n'est pas nécessaire d'utiliser ce transistor de polarisation qui n'aurait pour effet que de consommer trop de surface et de courant pour chaque pixel. Le fonctionnement de ce suiveur sans transistor de polarisation débute donc par la remise à zéro de ses charges accumulées en sortie, via les portes de transmission des entrées du comparateur, en liant V_{cna} à la masse. Après cette réinitialisation, la tension augmente rapidement pour tendre vers un certain niveau de tension. Si le temps écoulé entre la remise à zéro et la lecture est toujours le même, la tension à la sortie sera toujours proportionnelle à celle de l'entrée. La Figure 4.3 illustre cette réponse dynamique avec la montée rapide de la tension à la source du suiveur après la remise à zéro pour différentes tensions de grilles et une tension de 3V au drain du transistor suiveur (nœud V_{rst}).



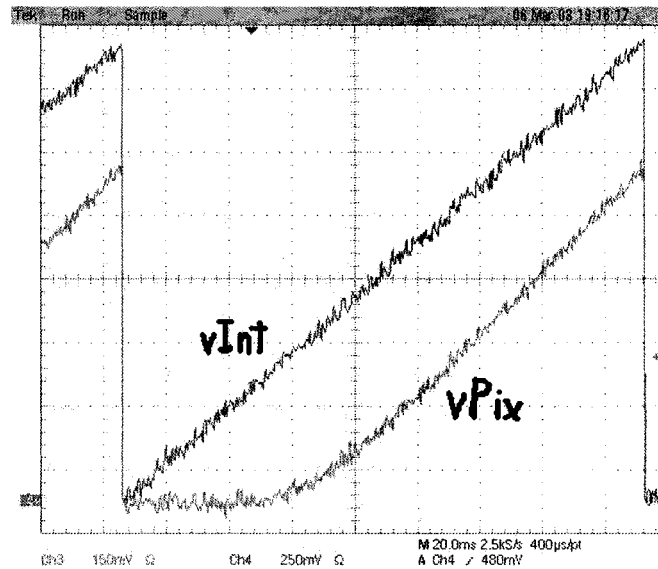


Figure 4.4: Analyse transitoire du suiveur.

En état de fonctionnement normal, la tension au drain du suiveur est beaucoup plus grande (3V) que celle de la figure précédente (1.8V) puisqu'une erreur dans le dessin des masques court-circuite en permanence le drain du suiveur à cette tension de 1.8V. On peut alors voir à la Figure 4.5 la sortie maximale du suiveur en fonction de la charge pour des tensions de 1.8V et 3.3V. Avec une charge capacitive et une tension de drain plus élevée, la pente du signal de sortie de la Figure 4.4 serait plus accentuée, résultant donc en une fonction de transfert (ou gain) plus efficace.

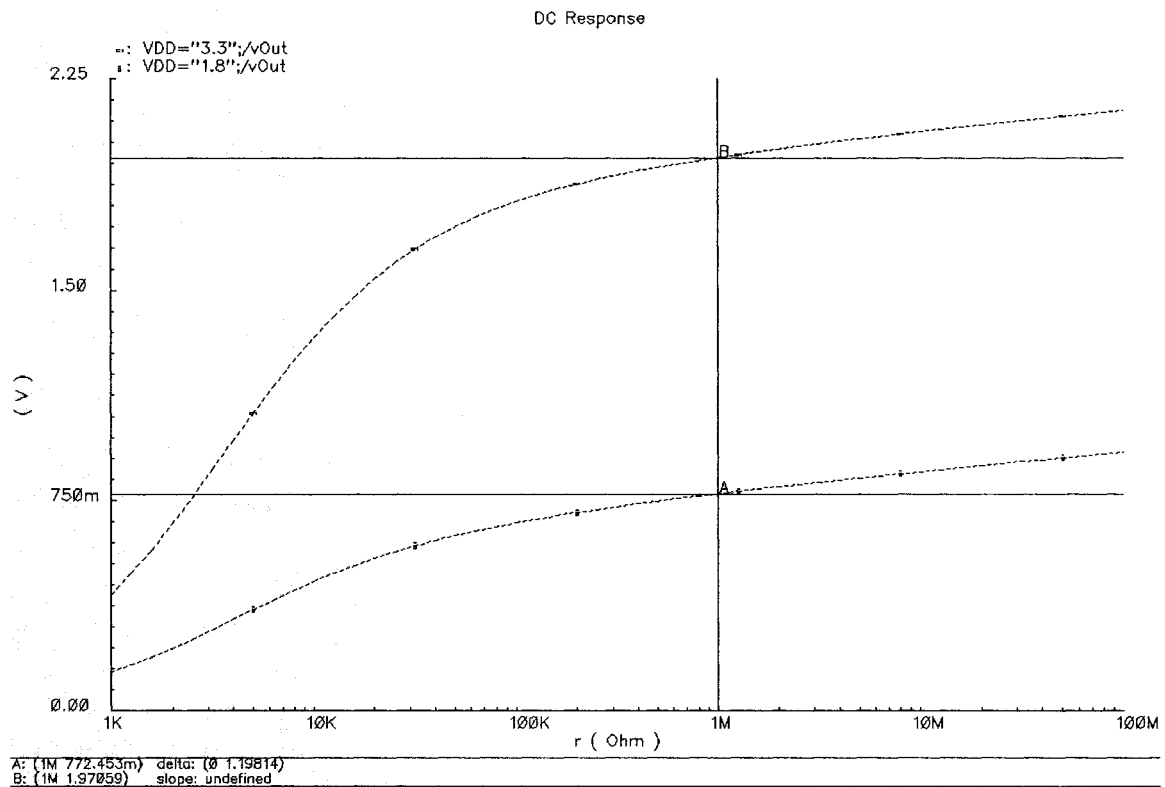


Figure 4.5: Sortie maximale du suiveur en fonction de la résistance de charge.

4.2.3 Précision du comparateur

Suite aux quelques erreurs de connexion de la puce, lesquelles ont été expliquées précédemment, le fonctionnement du comparateur intégré n'a pu être vérifié et testé. Cependant, les simulations ont démontré que le fonctionnement était possible et un circuit a été monté à l'aide de composants discrets afin d'en prouver le fonctionnement électronique réel. Les mesures prises sur ce circuit discret sont présentées à la Figure 4.6. En (a), l'entrée positive du comparateur (V_{cna}) est inférieure à l'entrée négative (V_{pix}), résultant ainsi en une sortie (A_2) qui tombe à zéro, tandis que l'inverse se produit en (b).

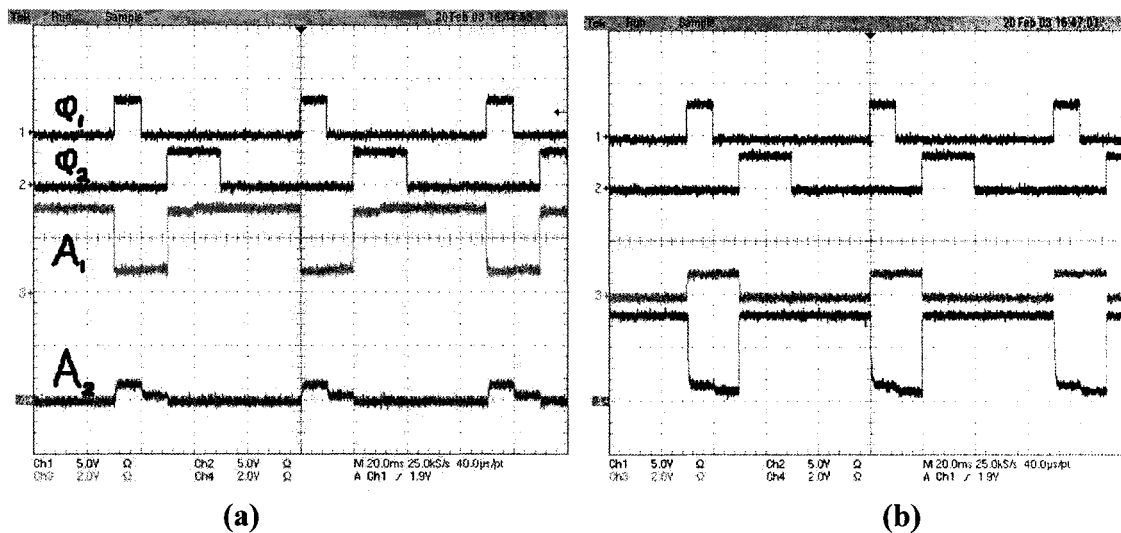


Figure 4.6: Signaux de contrôle du comparateur

(a) $V_{pix} > V_{cna}$; (b) $V_{pix} < V_{cna}$.

Lorsque la différence entre les deux entrées est relativement grande comme à la Figure 4.6, la sortie (A_2) bascule immédiatement après l'activation du signal de contrôle ϕ_2 . Par contre, lorsque cette différence entre les entrées est très petite (quelques millivolts, Figure 4.7a), la sortie s'approche quelque peu de 0V, mais est loin de l'atteindre après l'activation de ϕ_2 . C'est le signal *valide*, représenté sur le schéma de la Figure 2.14, mais non visible sur la Figure 4.7, qui est activé au milieu de la période active de ϕ_2 et qui fait immédiatement basculer la sortie (A_2) à 0V. Tandis qu'à la Figure 4.7b, le signal *valide* aide le comparateur à basculer vers V_{dd} .

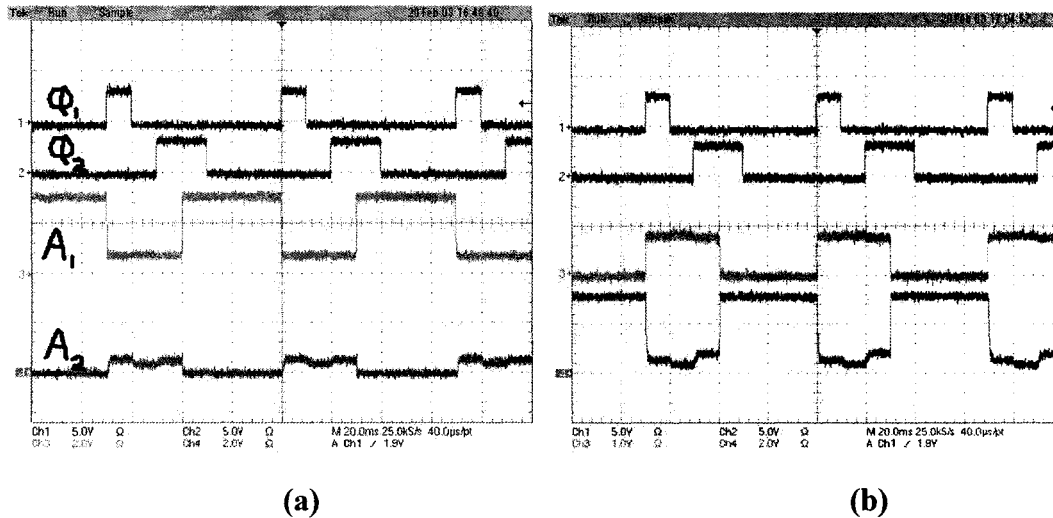
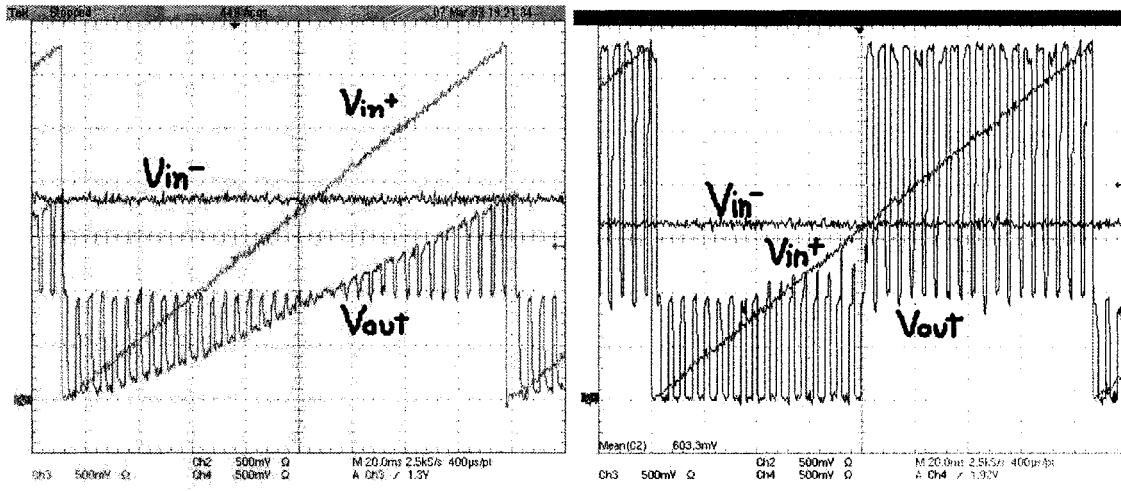


Figure 4.7: Effet du signal *valide* sur la sortie

(a) $V_{pix} > V_{cna}$; (b) $V_{pix} < V_{cna}$.

Afin de mieux comprendre l'effet du signal *valide* sur la précision du comparateur, plusieurs comparaisons ont été effectuées sur toute la plage de tension. Sans l'activation du signal *valide*, la sortie reste dans un état logique indéterminé de plus en plus prononcé à mesure que la différence entre les entrées est petite (Figure 4.8a). Par contre, avec l'activation du signal *valide* (Figure 4.8b), la sortie est toujours valide d'un point de vue numérique grâce à la rétroaction positive activée par ce signal. Rappelons que cette figure représente les mesures effectuées sur le comparateur conçu à l'aide de pièces discrètes. Cependant, les simulations du comparateur intégré de la puce ICFPMJL1 nous donnent des résultats identiques.



(a)

(b)

Figure 4.8: Ajout de précision avec la phase *valide*
(a) sans la phase *valide*; (b) avec la phase *valide*.

Le fonctionnement du comparateur ayant maintenant été prouvé, il reste à en vérifier la précision. Puisque le montage avec pièces discrètes s'avère très bruyant et que les pièces utilisées sont très différentes des transistors de la version intégrée, la précision réelle du comparateur ne peut être évaluée que par simulations.

Certaines topologies de comparateurs, ne fonctionnent pas de la même façon dépendamment du niveau de tension des entrées. Il arrive qu'avec des entrées sous un certain seuil, le comparateur ne fonctionne plus du tout. Nous avons donc vérifié le fonctionnement sur la totalité de la gamme dynamique des entrées pour une différence de 30mV entre elles, et pour une tension d'alimentation de 3.3V. Le résultat présenté à la Figure 4.9 nous confirme que notre design de comparateur fonctionne aussi bien peu importe le niveau de tension des entrées.

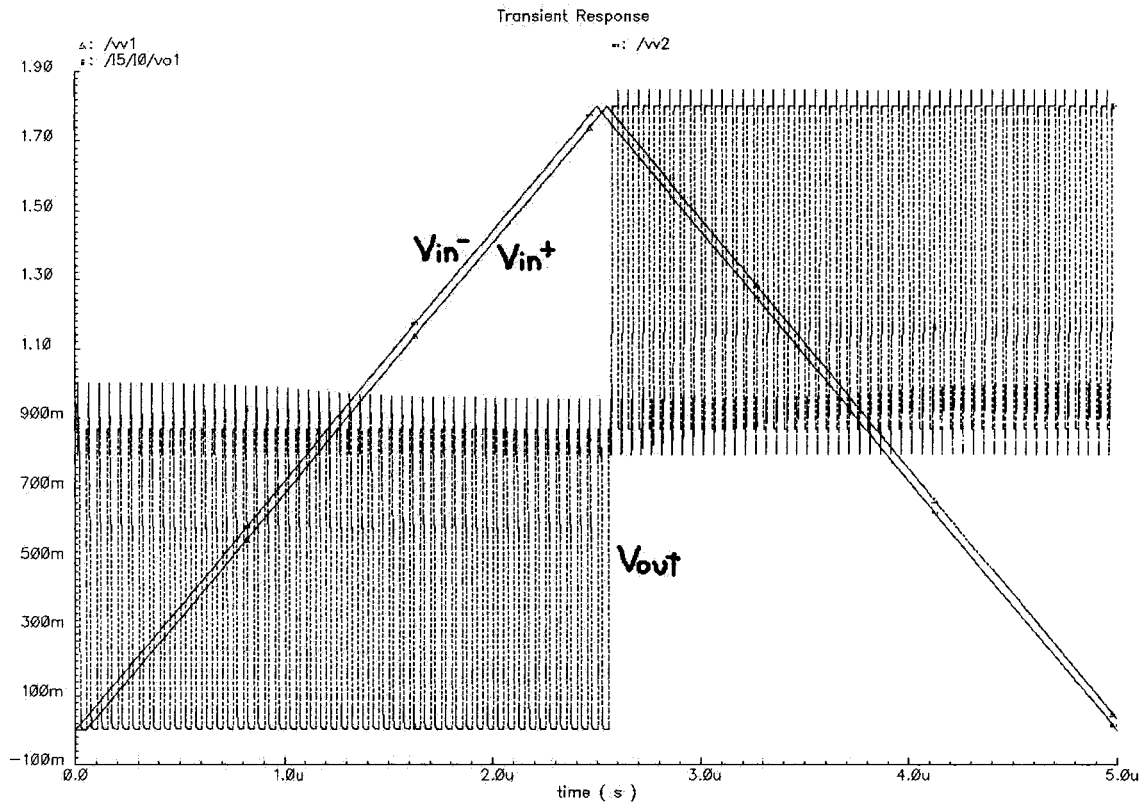


Figure 4.9: Comparateur fonctionnant sur la totalité de la gamme dynamique.

Les simulations de base du circuit du comparateur lui donnent une précision de plus de 11-bits, ou un décalage de 0.65mV. Cette erreur n'est pas causée par les amplificateurs (inverseurs) du comparateur, mais de l'injection des charges des portes de transmission. Bien que minimisée, tel qu'expliqué aux sections 3.2.2 et 3.2.3, une contrainte d'espace importante nous empêche de porter plus loin cette optimisation. Par contre, si ce comparateur était utilisé dans un circuit non limité à la surface du pixel du présent design, une augmentation significative de la précision pourrait être atteinte par l'augmentation de la capacité du comparateur. De cette façon, une plus grande quantité de charges « retiendrait » la valeur des tensions d'entrée, étant ainsi beaucoup moins affectées par l'injection de quelques charges. Rappelons que le condensateur, qui couvre la totalité de la surface du pixel (métal 5 et 6) à l'exception de la photodiode, est le plus gros qu'il était possible de construire.

Maintenant que les erreurs de circuit ont été minimisées, il est important de se prémunir des erreurs de fabrication. Ces erreurs peuvent avoir pour effet que les composants ne soient pas appareillés et que les seuils des inverseurs résultants soient légèrement différents. Afin d'évaluer l'erreur causée par des inverseurs non appareillés, la largeur de l'un des deux transistors nMOS a été modifiée de $\pm 5\%$ et de $\pm 10\%$, tel que présenté à la Figure 4.10 (a) et (b) respectivement. On y voit représenté la tension de sortie qui est à un niveau logique haut pour la première moitié de la simulation pour ensuite retomber à un niveau logique bas. Une des entrées du comparateur est fixe et l'autre augmente linéairement de 890mV à 910mV.

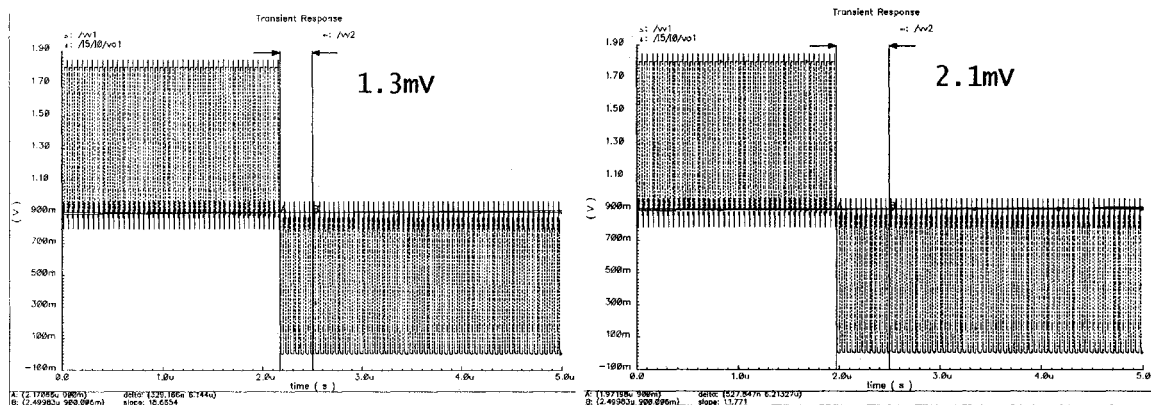


Figure 4.10: Précision du comparateur

(a) erreur d'appareillement de 5%; (b) erreur d'appareillement de 10%.

Lors de la fabrication d'une puce, on peut s'attendre à un maximum de 10% ($\pm 5\%$) d'erreur de la largeur des traces constituant les composants, affectant ainsi directement leurs paramètres. Les erreurs représentées à la Figure 4.10 (a) et (b) sont de 1.3mV ($20\text{mV} \cdot 0.325\mu\text{s} / 5\mu\text{s}$) et de 2.1mV ($20\text{mV} \cdot 0.525\mu\text{s} / 5\mu\text{s}$) respectivement.

En résumé, le comparateur fonctionne bien sur toute la gamme dynamique d'entrée. Avec les injections de charge, la tension de décalage est de 0.65mV, ce qui donne une précision de plus de 11-bits. Cette tension a été mesurée de la même manière que les erreurs de la Figure 4.10, mais en n'introduisant aucune erreur d'appareillement. Si on joute à cela les

erreurs de fabrication, on obtient une tension de décalage d'un maximum de 2.1mV, ce qui résulte en une précision de plus de 9-bits. Avec une implémentation de ce comparateur dans un espace non-restreint à la surface du pixel, une précision beaucoup plus grande serait facile à atteindre en augmentant la taille du condensateur afin de minimiser encore plus les erreurs liées aux injections de charges, et en utilisant des transistors plus gros afin de minimiser les erreurs liées au procédé de fabrication.

4.2.4 Gamme dynamique du capteur

La gamme dynamique maximale que le capteur peut atteindre est celle de la photodiode. Elle se mesure par le rapport entre le photo-courant maximal et minimal pouvant être généré par la photodiode sous illumination intense et sous obscurité totale (courant de fuite) respectivement. Les taux de décharge maximal et minimal (Figure 4.11) sont inversement proportionnel au photo-courant et sont de l'ordre de $0.4286 \text{ volts} \cdot \text{sec}^{-1}$ et de $37500 \text{ volts} \cdot \text{sec}^{-1}$ respectivement. La gamme dynamique résultante pour la photodiode est de 98.8dB, ce qui équivaut à une précision de plus de 16-bits ou de 2^{16} niveaux de quantification pour la représentation numérique.

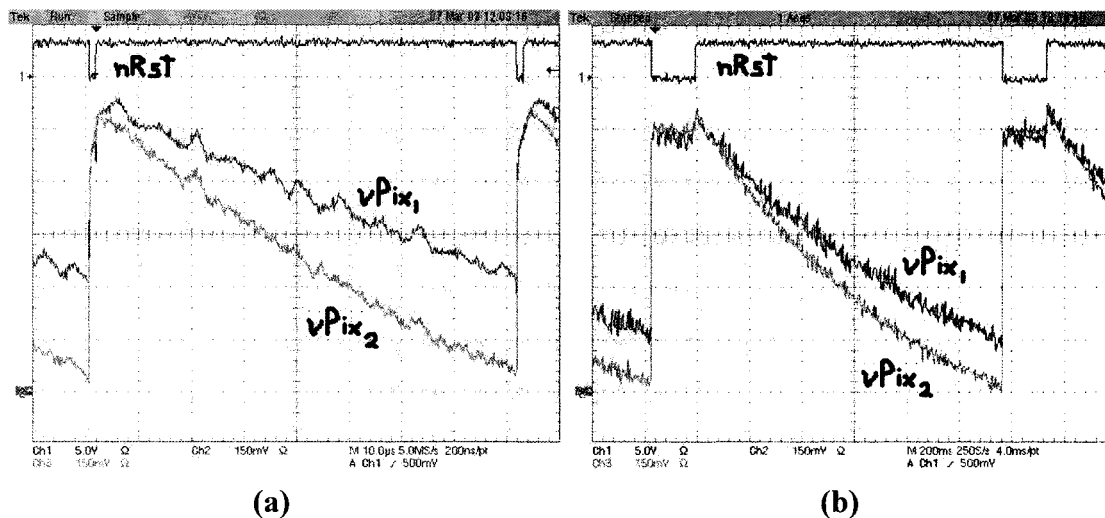


Figure 4.11: Intensité du photo-courant de la photodiode
(a) sous forte intensité lumineuse; (b) sous obscurité.

La Figure 4.11 (a) et (b) contient le signal $nRst$ qui réinitialise la réponse des photodiodes. PD1 représente un pixel à une photodiode, tandis que PD2 représente un autre pixel avec deux photodiodes en parallèle. En utilisant la différence de réponse entre les deux pixels (1-PD et 2-PD) de la Figure 4.11a à forte intensité lumineuse (courant de fuite négligeable), il est possible de quantifier la valeur du photo-courant généré par la lumière (i_{photo}). La capacité totale rattachée à la photodiode est la capacité de cette dernière plus celle de la capacité parasite des transistors l'entourant. Il est possible de mesurer cette capacité par simulation en imposant des valeurs arbitraires au courant et à la capacité de la photodiode (C_{photo}) pour deux simulations différentes et résoudre l'équation 4.1 à deux inconnues. Bien que les valeurs choisies soient arbitraires, il est nécessaire de choisir des valeurs près de la réalité afin d'obtenir de bons résultats de l'équation 4.1. La valeur obtenue par simulation pour la capacité parasite (C_{par}) est de 6.2fF. Maintenant, avec cette valeur et celles de la Figure 4.11a, il est possible de calculer la valeur de la capacité de la photodiode ainsi que l'intensité du photo-courant. Le circuit à deux photodiodes est donné par l'équation 4.2 et celui à une seule photodiode est donné par l'équation 4.3. La résolution de ces deux équations donne une valeur de 5.26fF pour C_{photo} et 77.36pA pour i_{photo} .

$$(C_{par} + C_{photo}) \Delta v = i_{photo} \Delta t \quad (4.1)$$

$$(6.2fF + 2C_{photo}) (790mV - 50mV) = 2i_{photo} (80\mu s) \quad (4.2)$$

$$(6.2fF + C_{photo}) (590mV - 50mV) = i_{photo} (80\mu s) \quad (4.3)$$

En simulation, il est très important de vérifier le paramètre G_{min} lorsque l'on travaille avec de très petits courants. Ce paramètre est la conductance entre chaque nœud du circuit et la masse nécessaire afin de faire converger la simulation. Si ce paramètre est trop grand, les courant de fuite de chaque nœud seront de valeur comparable à celle des courants réel du circuit, et fausseront ainsi les résultats. Pour la simulation des photo-courants, la valeur de G_{min} a été portée à 10^{-19} .

Afin de parvenir à mesurer la gamme dynamique complète, il est nécessaire de procéder par la méthode des expositions multiples. Ceci s'explique du fait qu'à un temps donné, les tensions des intensités minimales et maximales ne peuvent être quantifiées sur la même échelle à moins d'être capable d'aller chercher une précision de 16-bits pour une seule conversion A/N, ce qui est très difficile à atteindre. Par exemple, à 48 μ s, l'intensité maximale est de 1.8V (48 μ s * 37500 volts*sec⁻¹) et l'intensité minimale est de 20.6 μ V (48 μ s * 0.4286 volts*sec⁻¹). Même avec une conversion A/N de 10-bits, un niveau de tension de 20.6 μ V se retrouve largement sous 1 LSB qui est de 1.76mV (1.8V / 2¹⁰). Avec un niveau de quantification plus conservateur de 8-bits, le temps nécessaire à l'intensité minimale (courant de fuite) pour atteindre 1 LSB est de 16.4ms (1.8V / 2⁸ / 0.4286 volts*sec⁻¹). Afin d'aller chercher de la précision dans la totalité de la gamme dynamique, les temps d'expositions doivent débiter à 48 μ s et dépasser les 16.4ms. Alors, si on désire obtenir une représentation numérique à point flottant, il serait nécessaire de procéder à 9 expositions (équation 4.4), selon la séquence 48 μ s, 96 μ s, 192 μ s, ..., 24.57ms. À partir de la gamme dynamique réelle de 98.8dB de la photodiode et étant donné la méthode de quantification, un maximum de 96dB pourra être mesuré comme gamme dynamique (dépendant des temps d'exposition).

$$N_{exp}=1+\log_2\left(\frac{I_{photoMax}}{2^B I_{photoMin}}\right) \quad (4.4)$$

La période maximale disponible pour faire un système vidéo opérant à une séquence standard de 30 images par secondes est de 33.3ms. Notre exposition maximale de 24.57ms est donc la dernière exposition possible pour une quantification sur 8-bits. Afin de capturer des images à plus grande vitesse, il est possible d'augmenter le niveau de quantification à 9-bits. Dans ce cas, le plus grand temps d'exposition serait de 12.29ms ($B=9$ et $N_{exp}=8$ dans l'équation 4.1). Ces exemples sont toujours en ayant 96dB de gamme dynamique. Par contre, il est possible de diminuer les contraintes sur le nombre

de niveaux de quantification pour une même vitesse de capture au coût d'une gamme dynamique moindre. Tel qu'expliqué à la section 2.2.2, il est possible de faire la concession d'un bit de précision pour la quantification contre un temps d'exposition deux fois plus long.

Lorsque le nombre de niveaux de quantification change, le temps requis pour effectuer la conversion A/N est directement affecté. Il faut ajouter ce temps au temps d'exposition maximal afin d'obtenir le temps total de capture d'une image sous expositions multiples. Avec un temps de lecture/écriture de 80ns pour une matrice de 320 colonnes et une quantification sur 8-bits, le temps de conversion résultant est de 6.55ms ($80\text{ns} * 320 * 2^8$) puisqu'il y a une lecture/écriture à tous les niveaux. Ajouté au temps d'exposition maximal de 24.57ms, le temps total est de 31.02ms qui est sous les 33.3ms de période maximale pour la vidéo. Avec 9-bits, le temps total est de 25.4ms, encore bien en dessous des 33.3ms. Rappelons que cette limite de 33.3ms ne doit être respectée que lorsque l'on doit fournir une séquence vidéo standard. Il est possible de faire de la vidéo à un taux de 20 images par seconde comme le font plusieurs appareils photo/vidéo commerciaux actuellement. Lorsque l'image capturée n'est qu'une seule photo, la seule limite dans le temps est liée au courant de fuite.

4.2.4.1 Effet non linéaire

Sous aucune exposition à la lumière (Figure 4.11b), il est possible d'observer un effet non linéaire lors de la décharge de la photodiode, contrairement au cas où l'intensité lumineuse est plus forte (Figure 4.11a). La décharge de la photodiode est due à deux courants, soient le courant de fuite et le courant de photo exposition (équation 4.5). Lorsque le courant de photo exposition est nul, on observe directement le courant de fuite. Lorsque l'intensité lumineuse est très forte, le courant de fuite est négligeable par rapport au courant de photo exposition. Il est donc permis de supposer que c'est uniquement ce dernier que l'on observe dans ce cas. À partir de ces observations, il est

possible de conclure que le courant de photo exposition est linéaire en fonction de la tension à la photodiode, tandis que le courant de fuite est non linéaire. La Figure 4.12 permet de bien voir cette non-linéarité.

$$I_{\text{decharge}} = I_{\text{photo}} + I_{\text{fuite}} \quad (4.5)$$

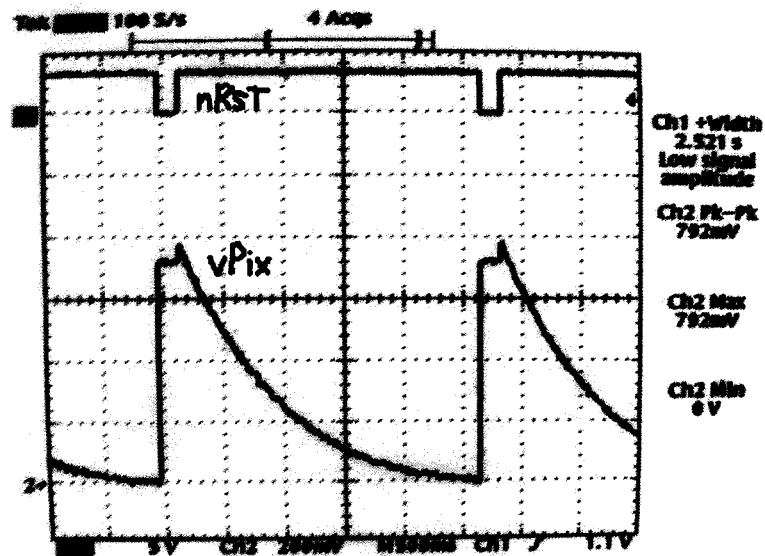


Figure 4.12: Courant de fuite de la photodiode.

4.2.4.2 Amélioration de la gamme dynamique

Nos objectifs de conception d'un capteur à gamme dynamique élevée sont largement atteints avec une gamme dynamique de 96dB. Cependant, afin d'obtenir une gamme dynamique encore plus grande, il serait possible de réduire la résistance dans le chemin de la décharge en enlevant les obturateurs et d'ajouter des fentes dans la photodiode afin d'en améliorer l'efficacité quantique. Ces aspects seront discutés plus en détail à la section 4.3.

4.2.4.3 Fuite de l'obturateur

Un effet non prévu du pixel, qui a été découvert lors de mesures, est la fuite accélérée de l'obturateur qui survient un certain temps après son ouverture (porte de transmission à OFF). Après plusieurs mesures, il a été vérifié que cette fuite accélérée dépendait seulement du niveau de tension atteint à la photodiode. Ce problème affecte donc de la même manière (en fonction de la tension) les pixels sous forte intensité lumineuse que ceux qui sont dans l'obscurité. Des simulations à l'aide de *Spectre* sont parvenues à reproduire les effets observés sur le circuit, éliminant ainsi la possibilité que la cause soit d'origine photoélectrique. Puisque la simulation est représentative de la réalité, il a été possible d'observer les niveaux de tension des différents nœuds et de vérifier l'intensité des courants dans le circuit. Ces observations nous ont mené à conclure qu'un courant de fuite augmentait largement lorsque la tension de la photodiode descendait légèrement sous les 0V. Ceci s'explique par la polarisation avant de la photodiode, subissant le même effet qu'une pile photovoltaïque, avec la cathode fixée à la masse, la tension à l'anode baisse sous zéro volt. Il en résulte une remise en conduction des transistors NMOS des portes de transmission qui laissent ainsi fuir les charges accumulées à l'entrée du suiveur. La Figure 4.13 présente les mesures et les simulations de ces observations en (a) et (b) respectivement. La Figure 4.14 met en évidence la fuite de l'obturateur par un agrandissement de la Figure 4.13b, et permet de mesurer la vitesse de cette chute de tension qui est de l'ordre de 624 kV/s.

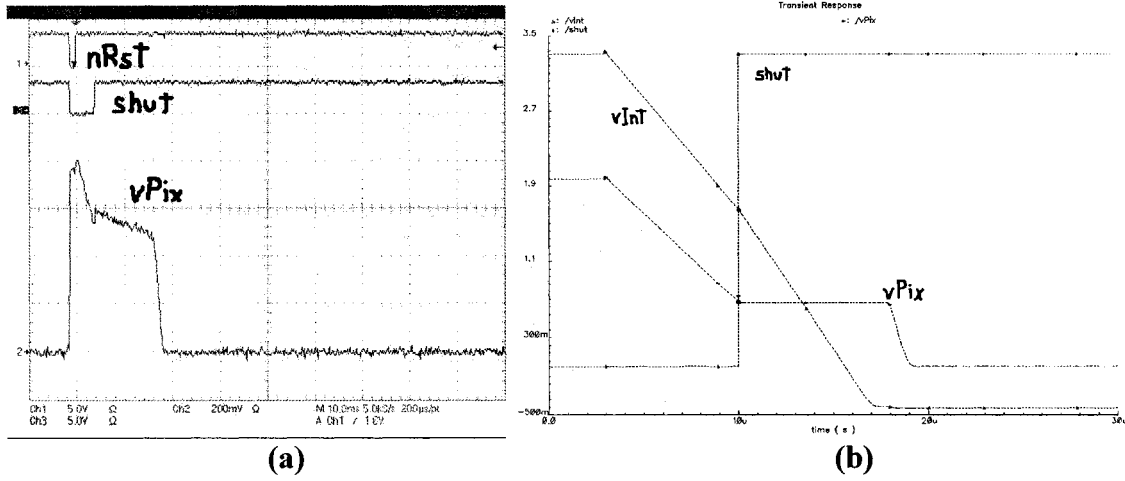


Figure 4.13: Fuite de l'obturateur
(a) mesure; (b) simulation.

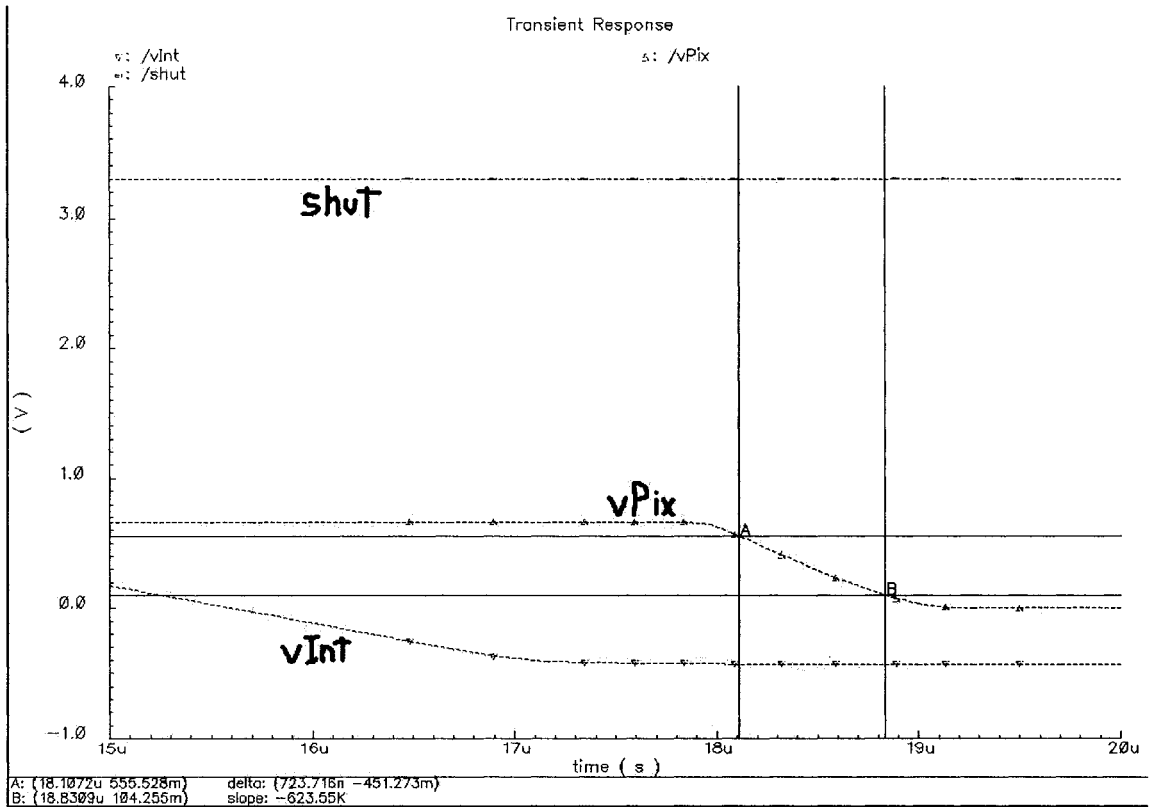


Figure 4.14: Quantification de la fuite prononcée de l'obturateur.

La fuite de l'obturateur nuit grandement au bon fonctionnement du système, surtout en mode expositions multiples où l'obturateur doit retenir les charges pendant un temps assez long. Heureusement, il a été possible d'éliminer cet effet néfaste avec un simple changement dans la séquence des signaux de contrôle. Cette correction a donc pu être effectuée dans le contrôleur et vérifiée par des mesures (Figure 4.15a). Les simulations subséquentes ont confirmé le résultat des mesures (Figure 4.15b). Bien que cette correction empêche la fuite prononcée de l'obturateur, une fuite minime est observée lorsque la simulation se poursuit sur une seconde. Le taux de cette chute de tension est de $0.037 \text{ volts} \cdot \text{sec}^{-1}$, ce qui équivaut à 8.6% du courant de fuite de la photodiode. On peut donc conclure que la correction de cette fuite fonctionne selon les résultats des simulations. De plus, avec d'aussi petits courants, il se pourrait qu'ils soient liés aux paramètres de simulation malgré l'ajustement du G_{\min} discuté précédemment. La conséquence d'avoir à utiliser le signal $nRst$ pour corriger le problème impose un moment de correction identique à chaque pixel étant donné la globalité de ce signal. Cependant, une méthode de correction propre à la présente architecture devrait être plus élaborée dans une version subséquente de ce capteur, ce problème n'ayant pas été prévu lors de la conception. La solution évidente à ce point est d'ajouter un autre transistor de remise à zéro (*reset*), mais celui-ci contrôlé par le loquet.

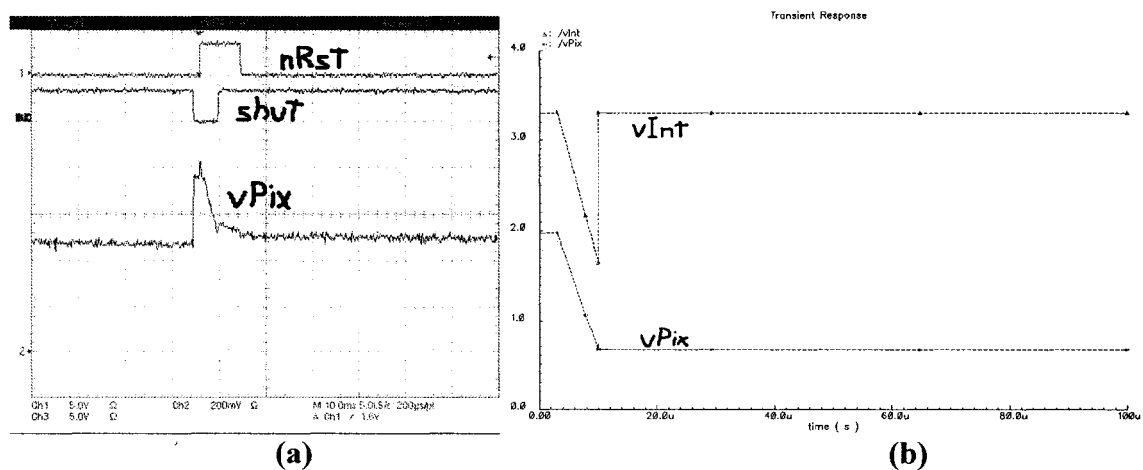


Figure 4.15: Correction de la fuite de l'obturateur
(a) mesure; (b) simulation ($V_{rst}=3.3$; $C_{photo}=15f$; $C_{par}=6.2f$).

4.2.5 Transfert des données

Les expositions multiples sont nécessaires pour atteindre une gamme dynamique élevée. Parmi les capteurs qui exploitent cet aspect et recensés dans la littérature, plusieurs images sont capturées et transférées une à une vers l'extérieur de la puce. Ce n'est que plus tard et après plusieurs calculs que ces images sont remises ensemble afin d'en faire une seule image à large gamme dynamique. Cette façon de procéder est donc possible seulement en ayant une ou plusieurs autres puces de support assez performantes pour effectuer beaucoup de calculs et suffisamment de mémoire pour stocker toute l'information provenant des différentes captures d'images.

Dans l'architecture proposée, le loquet de l'obturateur permet d'adapter localement les temps d'expositions à la lumière incidente. À partir de ce moment, l'image complète n'a pas besoin d'être transférée hors de la puce après chaque temps d'exposition intermédiaire. Ceci diminue considérablement la consommation de puissance du capteur du fait que la circuiterie est active moins souvent.

Lors de l'utilisation d'une architecture standard (multiples transferts d'images [27]), la quantité de données transférée est de NBE bits (**N**ombre de pixels, **B**its par pixel, différentes **E**xpositions). Avec l'architecture proposée, les bits sont seulement transférés dans la mémoire temporaire et aucun bit n'est transféré hors de la puce du début de l'exposition à la fin de la conversion A/N. Après ce moment, il y a B -bits par pixels pour la mantisse et E expositions codées sur $\log_2 E$ bits qui seront transférés hors de la puce. La quantité de données transférée est donc réduite par un ratio de 6, donné par l'équation 4.6 pour des valeurs de 8 et 9 pour B et E respectivement.

$$\eta = \frac{BE}{B + \log_2 E} \quad (4.6)$$

Cette architecture nécessite une mémoire temporaire (Figure 2.1) afin d'atteindre une telle réduction de la quantité de données transférée. Bien que cette mémoire ne soit pas nécessairement présente dans les capteurs utilisant les multiples transferts d'images, ces derniers ne pourraient pas réduire leur quantité de données transférée de la sorte car une mémoire temporaire de très grande taille devrait être ajoutée pour contenir les multiples captures. Ceci résulterait ainsi en une architecture compliquée et inefficace. De plus, les capteurs standards nécessitent un post-traitement coûteux en temps et puissance de calcul, ainsi qu'en énergie consommée afin de reconstituer l'image. Ceci n'est pas le cas avec notre capteur puisque les données sont déjà en format point flottant, prêt pour un rendu d'affichage.

Bien que l'architecture avec la mémoire temporaire demande une moins grande quantité de données transférées, ces données doivent éventuellement être lues hors de la puce. Cette opération requiert un temps non négligeable. Il ne serait donc pas praticable de l'utiliser pour faire de la vidéo à 30 images par secondes. Le but d'une telle architecture est de capturer des images à très haute précision avec une gamme dynamique très élevée, et ce, à un taux d'images par seconde plus bas que celui du vidéo standard. Un objectif éventuel très approprié à cette architecture pourrait être d'y ajouter un module de compression à même la mémoire de la puce du capteur.

Avec l'utilisation d'une matrice de pixels numériques seule, c'est-à-dire sans les mémoires, une réduction de la quantité de données transférée hors de la puce ainsi qu'une diminution de l'énergie consommée sont toujours possible. Le capteur standard [27] transfert B bits par pixel par exposition, mais seulement suite à 2^B comparaisons. Ses opérations lors de l'exposition finale est identique à celle des expositions précédentes. Avec l'architecture proposée, seulement une comparaison est effectuée et un seul bit est transféré lors de chaque exposition intermédiaire. Après l'exposition finale, 2^B bits sont transférés suite à 2^B comparaisons (un après chaque comparaison). Ce ratio d'amélioration est donné par l'équation 4.7 et équivaut à 0.27 pour des valeur de 8 et 9

pour B et E respectivement. Bien que ce ne soit pas une amélioration dans ce cas, il est possible de travailler avec une quantification sur 6 bits. Dans ce cas, l'architecture proposée devient avantageuse lors qu'il y a plus de 13 expositions. Dépendamment de la résolution et des autres caractéristiques de l'application visée, il est possible de fonctionner dans ce mode d'opération avantageux.

$$\eta = \frac{BE}{2^{B+E}} \quad (4.7)$$

Il est important de noter que la quantité de données transférée n'est pas la seule responsable de la consommation de puissance du circuit. Le taux d'activité sur la puce joue pour beaucoup plus de ce côté. Celui-ci est nettement plus élevé dans l'architecture standard puisqu'il y a 2^B comparaisons à chaque exposition intermédiaire, contre une seule dans l'architecture proposée.

Du côté de la qualité de l'image, la gamme dynamique est un paramètre à ne pas négliger. Celui-ci est directement relié au temps d'exposition minimal (exp_0) qui est contraint à être supérieur (ou égal) au temps de traitement de l'exposition initiale (conv_0), tel qu'illustré à la Figure 4.16.

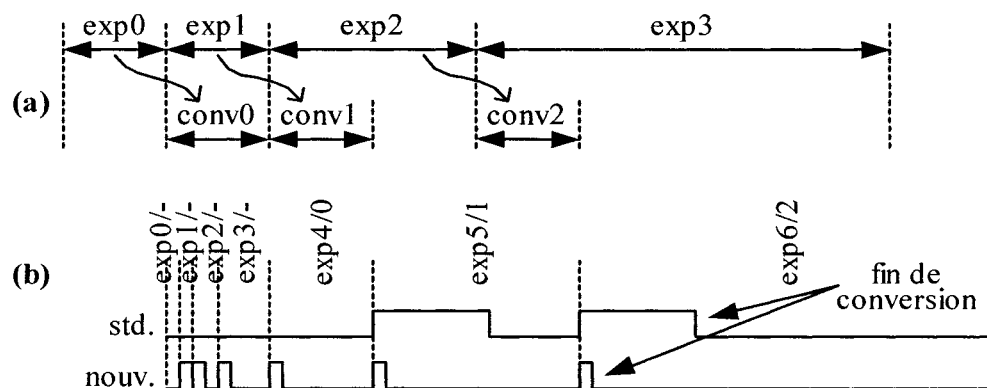


Figure 4.16: Expositions à la lumière et traitement des données

(a) temps d'exposition et de conversion; (b) temps d'exposition minimal.

Pour l'architecture proposée, le temps de traitement se résume à une comparaison et au transfert d'un bit par pixel, tandis que 2^B comparaisons sont nécessaires pour sortir B -bits avec l'architecture standard. En assumant que le temps d'une comparaison est le même que celui de la lecture d'une colonne, l'architecture proposée nécessite 8.77 fois moins de temps (équation 4.8) pour effectuer le traitement sur une exposition intermédiaire pour une matrice de 320 colonnes (C) et une quantification sur 8-bits. Avec le temps d'exposition maximal (16.4ms) et minimal (48 μ s), le nombre d'expositions et la gamme dynamique sont de 12 contre 9 et de 96dB contre 78dB, respectivement. Rappelons que ces calculs sont liés au temps d'exposition minimal.

$$\eta = \frac{BC + 2^B}{C + 1} \quad (4.8)$$

4.2.6 Performances de l'architecture proposée

L'architecture proposée est composée d'une matrice de pixels numériques supportée par une mémoire intermédiaire d'un bit par pixel et d'une mémoire temporaire contenant les données numériques de l'image capturée. Cette architecture a pour but de réaliser une caméra sur une seule puce. Dans ce cas, la résolution doit être limitée à une matrice de taille suffisamment petite pour avoir assez d'espace sur la puce pour intégrer les mémoires et le contrôle numérique. Avec tous les éléments de support à la matrice photosensible, la puce ne doit pas être de trop grande taille pour ne pas diminuer les rendements de production de cette caméra sur une seule puce.

Pour une utilisation dans un système nécessitant une plus haute résolution spatiale, seulement la matrice de pixels numériques est utilisé, sans les mémoires. La grande taille de la puce requise pour réaliser un tel capteur nécessite qu'il y ait seulement la matrice photosensible sur la puce. De cette façon, les rendements de production peuvent être

gardés élevés puisqu'une défectuosité d'un pixel peut être toléré et ne compromet pas le fonctionnement du système, contrairement à une défectuosité au niveau des circuits de contrôle.

Le point central de l'architecture proposée est le pixel numérique. Ce pixel utilisé pour fabriquer un capteur d'images a la flexibilité de satisfaire à différents objectifs de conception, que ce soit côté vitesse de capture, gamme dynamique, consommation de puissance ou précision de la conversion A/N dans différentes portions de la gamme dynamique. Les caractéristiques de ce pixel et de la puce conçue afin de valider son fonctionnement sont résumées au Tableau 4.1.

Tableau 4.1: Caractéristiques de la puce ICFPMJL1.v2

Technologie	0.18 μ m CMOS Nwell
Tension d'alimentation	3.3V et 1.8V
Boîtier	84-PGA
Type d'élément photosensible	Photodiode
Taille du pixel	13.8 μ m x 13.8 μ m
Facteur de remplissage	29%
Transistors / pixel	33
Courant de noirceur	0.4286 volts*sec ⁻¹ 8.89 nA/cm ²
Réponse maximale	37500 volts*sec ⁻¹
Gamme dynamique	96 dB

4.3 Développements futurs

La recherche et le développement de ce capteur s'étant surtout déroulé au début de la période allouée au projet, plusieurs idées innovatrices sont survenues en cours de développement. Certaines d'entre elles ont pu être intégrées à la puce, tandis que d'autres ont seulement été développées au niveau théorique dû à l'impossibilité de fabriquer une autre puce par contrainte du délais lié au cycle de fabrication.

Entre autres, le circuit de lecture rapide présenté à la section 2.2.3 a été développé au niveau théorique, pendant le temps où la puce était en production, l'idée étant survenue bien après l'envoi à la production des dessins de masques.

Les autres idées, n'ayant pu être intégrées à la puce et n'ayant pas pu bénéficier d'un temps suffisant pour leur élaboration, sont présentés dans cette section. Les aspects présentés ici découlent surtout des résultats des tests et mesures sur la puce fabriquée.

Pour une prochaine version du capteur d'images, la réalisation de deux matrices photosensibles, l'une dont l'activation des colonnes est contrôlée à la manière standard par un décodeur et l'autre par un circuit endochrone, permettrait de mesurer le gain en vitesse ainsi qu'en économie de puissance dissipée. À partir de ces résultats, il serait possible d'en extrapoler le gain acquis sur une matrice de très grande taille d'un produit fini.

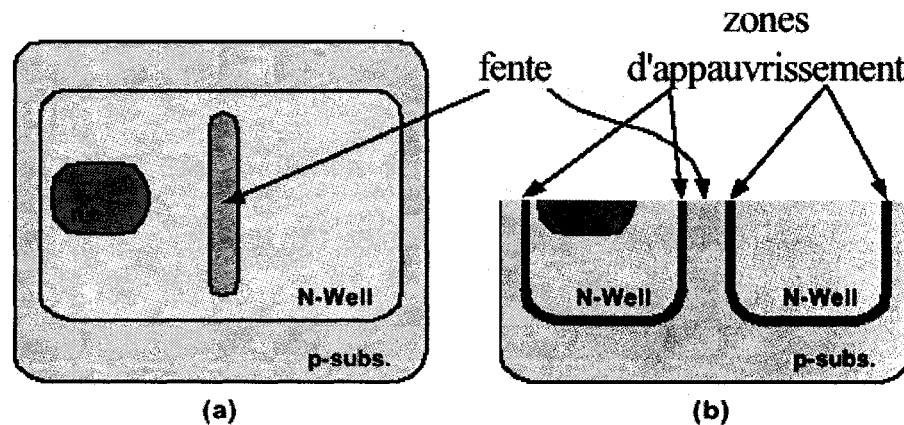
4.3.1 Amélioration potentielle de la qualité des images en CMOS

Dans le but d'améliorer l'efficacité quantique, peu d'options sont disponibles une fois la technologie choisie. Avec le procédé CMOS 0.18 μ m standard utilisé dans le cadre de ce projet, l'épaisseur et le niveau de l'implantation ionique des couches constituant la puce ne peuvent être modifiés. Sachant que l'absorption des photons par le silicium est tel que présenté au Tableau 4.2, il est possible de vérifier si la profondeur des zones d'appauvrissement de la technologie choisie permettront son utilisation en tant que capteur d'images.

Tableau 4.2: Profondeur d'absorption des photons par le silicium

λ	Absorption
400nm	0.19 μm
500nm	2.3 μm
600nm	5.0 μm
700nm	8.5 μm
800nm	23 μm

Puisqu'il est impossible de modifier la profondeur d'implantation des puits, il reste deux options possibles afin d'améliorer la conversion photons-électrons. La première est le choix du type de photodiode (Figure 3.1). Le type P+/N-well est situé très près de la surface et aura du mal à capter les photons pénétrant plus creux dans le substrat que la photodiode de type N-well/p-sub. L'autre option qui vise à améliorer l'efficacité quantique est l'ajout de fentes à l'intérieur de la photodiode. Bien que l'ajout de ces fentes diminue la surface vue du dessus (Figure 4.17a), elle augmente la surface de la zone d'appauvrissement, ce qui accroîtra la quantité de photons capturés. Une vue en coupe de ces configurations de zones d'appauvrissement est illustré à la Figure 4.17b.

**Figure 4.17: Photodiode avec fentes**

(a) vue de haut du pixel; (b) vue en coupe d'une fente.

4.3.1.1 Alternative au TFA

Un autre moyen d'augmenter l'efficacité quantique est d'agrandir le facteur de remplissage de la photodiode par rapport au pixel. La meilleure méthode pour y parvenir est d'utiliser le TFA (section 1.3.5) qui dépose la photodiode par dessus le circuit électronique à l'aide d'une étape subséquente dans un deuxième procédé.

Une alternative simple au procédé menant au TFA est de concevoir une première puce contenant seulement le circuit électronique de la matrice de pixels, comme celui nécessaire au TFA, et ensuite de fabriquer une deuxième puce contenant seulement les photodiodes. L'avantage de cette méthode proposée est de fabriquer la matrice de photodiodes à l'aide d'un autre procédé que celui utilisé pour le circuit à l'aide d'une plus grande échelle, qui est aussi moins coûteux. Ceci sans avoir recours aux appareils coûteux et complexes d'opérations nécessaires au dépôt de la couche mince de photodiodes sur la première puce comme pour la méthode de fabrication de TFA.

Avec le DPS présenté dans ce mémoire, il suffirait, pour un premier test, d'enlever le 29% de la surface du pixel occupé par la photodiode pour y ajouter un simple contact vers le métal de niveau supérieur. Ce métal servirait à une éventuelle connexion avec la deuxième puce contenant les photodiodes. Puisque les métaux supérieurs de chacune des puces sont disposés de manière à se connecter ensemble, les deux puces devront être connectées face contre face (Figure 4.18).

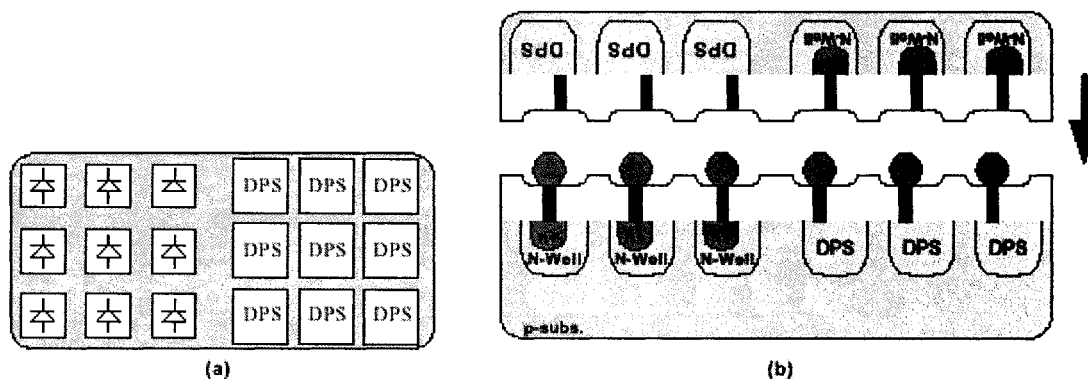


Figure 4.18: Test du « Flip-chip »

(a) vue de dessus; (b) vue en coupe.

À partir de ce moment, la matrice de photodiodes de la deuxième puce sera illuminée de l'arrière. Pour se faire, cette puce devra être amincie par polissage afin de respecter l'épaisseur nécessaire (Tableau 4.2) pour obtenir une bonne absorption des photons par les photodiodes situées à l'autre extrémité. L'amincissement commencera à être transparent (et suffisant pour un bon fonctionnement) lorsque la puce aura atteint $10\mu\text{m}$ à $15\mu\text{m}$ d'épaisseur [48]. Il serait possible d'atteindre ce niveau d'amincissement à l'aide du *623 Disc Grinder* de la compagnie *Gatan* [49] présenté à la Figure 4.19.

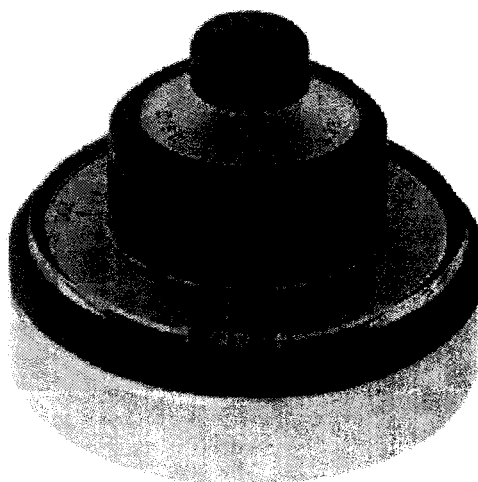


Figure 4.19: Appareil de polissage pour l'amincissement de l'arrière de la puce.

L'illumination par l'arrière d'une puce de capture d'images n'est pas nouveau. Bien que non commercialisé, un amincissement de puces CCD a déjà été effectué pour permettre une illumination par l'arrière de la puce [48]. Cette expérience avait pour but d'améliorer l'efficacité de la puce dans le spectre de la lumière bleue qui n'avait ainsi plus à passer au travers de l'oxyde de la grille. La différence avec ce projet est que nous proposons d'utiliser deux puces au lieu d'une seule afin de combiner les avantages d'un pixel entièrement numérique CMOS avec le haut facteur de remplissage d'une matrice de photodiodes réalisée sur une autre puce.

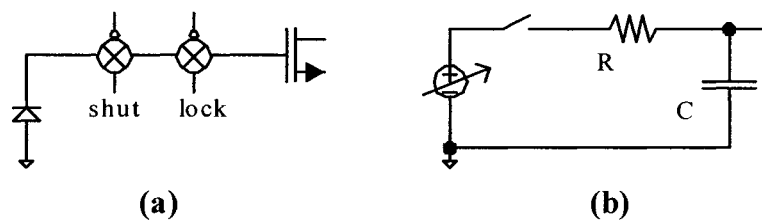
Il serait aussi possible d'utiliser un procédé autre que le CMOS, ou une variante de celui-ci, pour construire la puce contenant la matrice de photodiodes afin d'obtenir de meilleurs performances optoélectroniques. Certains procédés comme le SOI pourraient donner de meilleurs résultats en ayant un courant de fuite moins grand. Puisque les pixels numériques conçus dans la première puce contiennent plusieurs transistors et interconnexions et que la deuxième puce ne contient qu'une photodiode et une connexion, le niveau de complexité de cette dernière puce est beaucoup plus faible et ne nécessite pas d'être réalisé à l'aide d'une technologie à très petite échelle. Il est donc possible d'utiliser une technologie plus vieille à plus faible coût. Un autre avantage de ce capteur à deux puces est que le facteur de remplissage avoisine les 100%. L'ajout d'une micro-lentille est donc inutile, évite une étape de production et sauve donc les coûts qui y sont rattachés.

4.3.2 Pixel numérique amélioré

Le premier endroit où il pourrait y avoir optimisation du pixel est dans la partie analogique. Dans cette section, l'obturateur permet d'isoler l'effet de la lumière et la tension analogique à convertir en format numérique. Cependant, comme nous avons pu l'observer à la section 4.2.4.3, cet obturateur n'est pas parfait et une fuite prononcée survient lorsque la tension à la photodiode baisse sous zéro volt. Cet effet négatif aurait

remis en question la fonctionnalité de l'architecture présentée s'il n'y existait aucun moyen de palier à ce problème. La solution à ce problème a été trouvée et s'avère très simple d'implémentation. L'idée principale consiste à empêcher la tension de la photodiode de descendre sous les zéro volt lorsque l'obturateur s'ouvre (OFF) de façon définitive pour la capture courante (loquet enclenché). À ce moment, le transistor de réinitialisation force une tension de V_{rst} volts à la photodiode, empêchant ainsi que cette dernière se décharge complètement. Afin d'effectuer ce changement, une simple modification au niveau du contrôleur numérique est nécessaire.

Un autre élément pouvant possiblement être amélioré est la réponse du capteur sous une forte intensité lumineuse. Lorsque la lumière est maximale, le photo-courant l'est aussi. Ce photo-courant décharge la capacité de la photodiode ainsi que la capacité parasite de la grille du transistor suiveur. Les charges de cette dernière capacité se déplacent de la grille du suiveur vers la masse en passant au travers des deux obturateurs. Ces obturateurs sont en fait des portes de transmission dont la résistance n'est pas nulle (Figure 4.20) même lorsqu'ils sont fermés (ON). Ceci impose donc une constante de temps RC qui pourrait être un facteur limitatif lorsque l'illumination est très forte (courant plus élevé). Une solution simple à ce problème serait de concevoir les portes de transmission à l'aide de transistors plus larges afin d'en minimiser la résistance.



**Figure 4.20: Rapidité de décharge des capacités parasites
(a) obturateur; (b) circuit RC équivalent.**

La valeur de la résistance et du condensateur peut être estimée à $128\text{k}\Omega$ et à 6.2fF respectivement. La valeur du condensateur est déterminée d'après les calculs présentés à

la section 4.2.3. La valeur de la résistance a été calculée d'après la résistance de deux portes de transmission, avec le circuit de la Figure 3.5. Avec ce circuit l'équation 4.9, la variation de la résistance en fonction de la tension est donné à la Figure 4.21, avec un maximum de 63kΩ. Ces valeurs résultent en une constante de temps (RC) égale à 390ps. Ce temps est négligeable par rapport au temps de décharge le plus court de 80μs, sous une forte intensité lumineuse. Il n'est donc pas nécessaire d'optimiser cet aspect du circuit.

$$\Delta R = \frac{V_{in} - V_{out}}{I_m} \quad (4.9)$$

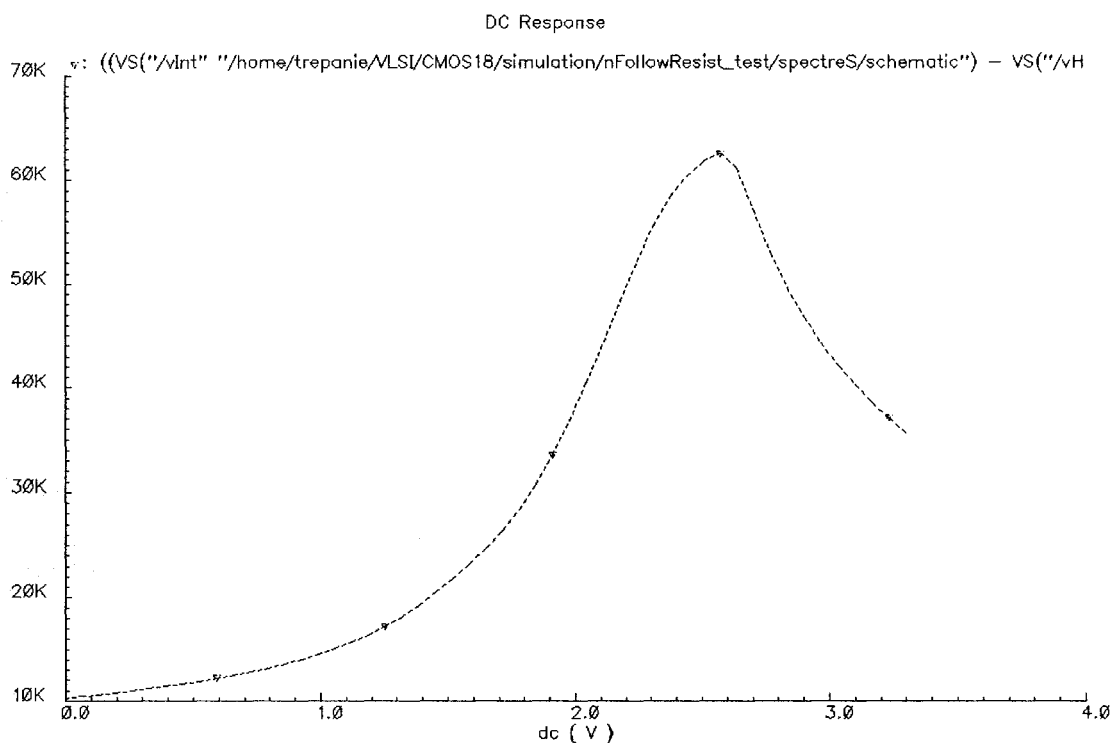


Figure 4.21: Variation de la résistance d'une porte de transmission.

Dans le cas, où les mémoires de support seraient utilisées, une partie de la puce doit être réservée pour l'implémentation des mémoires. Cependant, il serait intéressant de minimiser cet espace qui est occupé par une mémoire statique à six transistors dans la

première version de la puce et une mémoire à multiplexeur dans la deuxième version. L'utilisation d'une mémoire dynamique, à un ou deux transistors, occuperait moins de surface et serait sans doute profitable au design du capteur. Si cette nouvelle mémoire donne de bons résultats, il serait possible de pousser l'optimisation plus loin en incorporant la mémoire dynamique à l'intérieur de chaque pixel. Le coût de cette optimisation serait d'une modification du design du pixel numérique déjà optimisé et d'un plus grand nombre de transistors à l'intérieur de celui-ci.

4.3.3 Intégration du capteur dans le cadre du projet d'implant visuel

Bien que ce capteur soit générique et qu'il puisse être utilisé dans de multiples applications, son utilisation première sera sans doute dans le cadre du projet d'implant visuel du laboratoire de recherche *PolyStim*. Ce grand projet d'implant visuel comporte différents modules devant tous être intégrés dans le même système autonome, autour de la région du cortex visuel du cerveau du non-voyant. Ces modules externes sont notamment le capteur d'images (le présent projet), le module de traitement de l'image, le module de transmission sans fil des paramètres de stimulation, sans oublier l'alimentation de tout ce système. Une méthode intéressante pour supporter cette partie externe à proximité de l'implant est d'utiliser le modèle de lunette *Overthetop* de la compagnie *Oakley*, tel que présenté à la Figure 4.22. De cette façon, tous les modules pourraient être intégrés de façon compacte sans engendrer d'encombrement inutiles de la part de l'utilisateur. Le module de stimulation n'est pas illustré dans cette figure car il est sous cutané et ne fait donc pas parti du système externe.

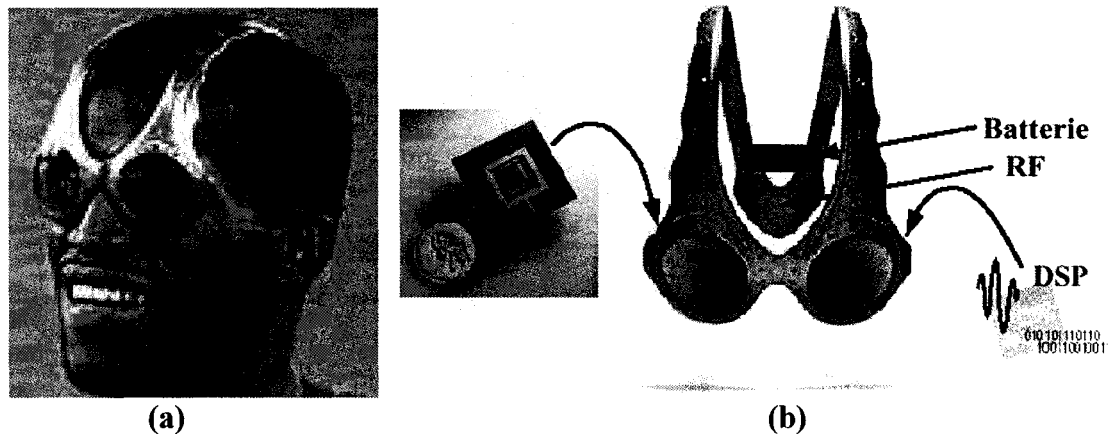


Figure 4.22: Implant visuel
(a) lunette de support pour l'implant; (b) composantes.

4.3.4 Méthodologie pour la suite du projet

Afin de faire de ce projet un produit fini, plusieurs autres étapes s'imposent. Suite aux problèmes rencontrés, solutions trouvées et expérience acquise dans le cadre de ce projet, une méthodologie pour la suite du projet est proposée.

Dans un premier temps, étant donné le délais de fabrication d'une puce, nous recommandons de faire un premier design (nouvelle puce) immédiatement en commençant le projet. Ce design ne devrait contenir qu'une très petite matrice de pixels numériques (10x10) afin d'en caractériser le fonctionnement global et la consommation de puissance, avec le comparateur intégré qui n'a pu être testé ici. Ce module de test pourrait en fait aussi être intégré à même une autre puce ayant déjà entamé le cycle de design afin d'en minimiser le délais de fabrication. Deux de ces matrices pourraient être conçues, une première avec une activation standard des colonnes et une deuxième avec une activation des colonnes par le circuit endochrone pour la lecture rapide présenté à la section 2.2.3

Pendant le temps de fabrication de cette première puce, une deuxième puce devrait être immédiatement conçue afin de tester l'alternative au TFA qu'est le *flip-chip* tel que présenté à la section 4.3.1.1. Cette puce pourrait être réalisée à l'aide d'une échelle de technologie moins récente ($0.35\mu\text{m}$ au lieu de $0.18\mu\text{m}$) afin de diminuer les coûts et délais de production. Les zones de contacts entre les deux puces devant être conçues avec une très grande surface pour faciliter l'assemblage, une échelle de technologie réduite ne serait pas utilisée à profit. L'objectif de ce design serait d'assembler deux puces après avoir poli la première (Figure 4.19). Le design devrait contenir les deux sections nécessaires à l'assemblage, soit la matrice de photodiodes et le circuit du pixel tel que présenté à la Figure 4.18.

Les contraintes de la réalisation d'un pixel numérique à 33 transistors dans un petit espace carré de $13.8\mu\text{m}$ de côté ayant été atteint, l'utilisation d'une autre échelle de technologie pour valider les autres caractéristiques de la puce peut être utilisée. Ceci peut se justifier par les délais de fabrication moins long, l'achalandage moins grand et la disponibilité de plus de surface de silicium. Par la suite, lorsque viendra le temps d'assembler le tout, il sera encore plus facile d'intégrer les composantes d'un tel pixel numérique dans une petite taille avec des échelles de technologies sous les $0.18\mu\text{m}$.

Pendant la fabrication des deux puces, le design d'un contrôleur numérique flexible devrait être fait de manière à permettre de valider tous les modes de conversion A/N présentés aux sections 2.2.1 et 2.2.2. D'autres fonctionnalités pourront y être ajoutées, soit le zoom numérique, la détection de mouvements, la détection de contours et la stabilisation de l'image pour n'en nommer que quelques-unes.

L'expérience acquise dans le cadre de ce projet nous porte à la conclusion qu'il est inutile de concevoir un circuit numérique intégré sans que celui-ci ne soit la version finale. La complexité, l'impossibilité d'y apporter des modifications subséquentes ainsi que les possibilités d'erreurs de connexions lors de l'intégration sont trop grandes pour que

l'investissement en temps en vaille la peine. Il est fortement suggéré d'utiliser un circuit reprogrammable comme un microprocesseur ou un FPGA pour contrôler la puce du capteur d'images fabriqué.

D'autres aspects devant faire le cas d'une recherche plus approfondie seraient du côté des mémoires et du circuit endochrone. Tel que mentionné précédemment, les mémoires pourraient être réalisées à l'aide de cellules dynamiques. Du côté du circuit endochrone, son utilisation a pour but d'accélérer le processus de lecture tout en minimisant la complexité du contrôle de la lecture. Une optimisation de ce circuit ainsi que de celui des amplificateurs de signaux voyageant sur les longues lignes de la matrice de pixels serait à étudier plus en détails. Il serait aussi intéressant de voir comment ce circuit endochrone permet de repousser les limites de la taille maximale de la matrice pouvant être contrôlée et s'il était possible de se rendre à un capteur de la taille d'une gaufre (*waferscale sensor*).

4.4 Conclusion

Dans ce chapitre, nous avons présenté les résultats des mesures qui ont validé l'effet photoélectrique du capteur ainsi que la portion analogique de la puce. Bien que quelques problèmes de connexions à l'intérieur de la puce nous empêchent de prouver hors de tout doute le fonctionnement du comparateur lorsque celui-ci est intégré avec la technologie CMOS 0.18 μm , la validation de son fonctionnement à l'aide de pièces discrètes ainsi que des simulations effectuées concordent avec le comportement attendu. Aucune raison ne porte donc à croire que le circuit ne fonctionnerait pas. Toutes les données du design sont disponibles pour une itération subséquente qui confirmera le tout.

Le mode d'opération avec expositions multiples a été démontré comme nécessaire afin d'obtenir la plus grande bande dynamique possible. Nous avons ensuite présenté les

performances de l'architecture proposée qui a spécialement été conçue pour ce mode d'opération et qui fait l'objet principal de ce mémoire.

Plusieurs développements pour de futures itérations du capteur ont ensuite été proposés. Certains tendent à améliorer la qualité du capteur, tandis que d'autres tendent à explorer de nouvelles voies. L'objectif principal de ces développements proposés est de pousser la recherche vers de meilleurs capteurs d'images numériques.

Selon les aspects à valider, un minimum de deux autres itérations de cette puce seront nécessaires avant d'obtenir le produit fini d'une caméra sur une seule puce. Cette présente itération avait pour but de valider l'architecture proposée ainsi que de quantifier l'effet photoélectrique. La prochaine itération aura pour but de faire fonctionner le design complet et d'en évaluer les performances. La troisième et dernière itération consistera à mettre ensemble tous les modules et à prendre en considération toutes les contraintes d'optimisation afin de réaliser une caméra sur une seule puce.

Conclusion

Le projet présenté dans ce mémoire avait pour objectif de capter des images dans la plus grande gamme dynamique possible afin de « voir » comme l'œil humain. Tout le système réalisant cet objectif devait être le plus petit possible, léger et autonome, d'où la conception d'une puce de silicium pour contenir ce système.

L'aspect majeur du présent mémoire était de concevoir et d'optimiser la partie de l'architecture électronique numérique du système. Cependant, une grande partie du travail a été consacrée au choix des composantes pour la conversion photonique à électrique ainsi qu'à l'optimisation de la portion analogique intérieure au pixel. Bien que ce projet réalise la capture de photons afin d'en former une image, le côté optique n'a pas été exploré, seulement le côté électronique a été étudié. Il est considéré qu'une faible quantité de travail est nécessaire afin d'ajouter un module optique (lentilles) au présent projet afin d'en faire un capteur d'images pour des scènes réelles.

La première étape du projet fut d'explorer les caractéristiques des capteurs d'images existants. À partir de nos objectifs, nous avons choisi une technologie de puces en silicium pour la fabrication de notre capteur d'images. La nécessité d'intégrer tout le circuit de support au capteur d'images sur une même puce ainsi que les contraintes de faible consommation de puissance devant être rencontrées ont porté notre choix vers une technologie CMOS au détriment des CCD.

En second lieu, toujours à partir d'une exploration de la littérature, les caractéristiques des éléments photosensibles ont été prélevées. La première partie du premier chapitre fait l'étude de chacun de ces éléments. La conclusion de cette étude nous mène à utiliser la photodiode qui est l'élément photosensible le plus simple à contrôler et qui donne aussi les meilleurs résultats pour la technologie choisie du CMOS 0.18 μ m. La revue de

l'ensemble des topologies de pixels réalisées jusqu'à maintenant ont porté à la conclusion que celui qui offre le plus de flexibilité avec la meilleure qualité de signal est le DPS. Ce type de pixel est à la base de l'architecture présentée dans ce projet. De plus, l'amélioration de la réduction de l'échelle de la technologie va permettre de plus en plus d'innovations de ce côté, tandis que les PPS et APS ne pourront pas tirer profit de cet avancement technologique. Une autre caractéristique essentielle du DPS proposé est qu'il pourrait être utilisé avec une technologie TFA sans modifications majeures afin d'améliorer de plus de 300% l'efficacité quantique du pixel tout en permettant l'intégration d'un plus grand nombre de transistors à l'intérieur de celui-ci.

Ensuite, une nouvelle architecture permettant d'aller chercher le maximum de gamme dynamique dans l'image observée et impliquant le design innovateur d'un DPS a été présentée. Un nouveau comparateur dynamique de très faible taille a été conçu afin de permettre la conversion A/N à même le pixel en étant intégré à chacun de ceux-ci. Ce comparateur permet d'atteindre des niveaux de précision de plus de 9-bits en simulation, avec les erreurs d'appareillement et d'injection de charge. Bien que son fonctionnement au niveau intégré n'ait pas pu être mesuré et vérifié, une caractérisation au niveau de pièces discrètes a confirmé son potentiel d'utilisation. Par ailleurs, plusieurs étudiants du laboratoire PolyStim ont décidé de l'utiliser pour leur projet, notamment pour les conversions A/N d'un stimulateur cortical ainsi que pour un détecteur de proximité par infrarouges. De plus, il est présenté dans la thèse de C. Fayomi [17].

Du côté du DPS, son architecture interne permet une grande flexibilité quant au mode d'opération du capteur d'images. Dépendamment de l'objectif de performance visé, il est possible d'augmenter la rapidité de la capture ou bien la largeur de la gamme dynamique ainsi que la précision de la représentation numérique. La méthode ayant le plus d'effet sur la largeur de la gamme dynamique atteinte est le fonctionnement par expositions multiples. L'architecture de ce capteur, avec son pixel numérique, sa méthode de conversion A/N ainsi que la lecture de ses données a été conçue en fonction d'une

utilisation en mode d'expositions multiples. Une autre caractéristique majeure de ce capteur DPS est que son fonctionnement n'est pratiquement pas influencé par la taille de la matrice, contrairement aux PPS et APS.

Dans un autre temps, les détails de l'implémentation de la puce au niveau des transistors et du dessin des masques ont été présentés. Deux versions de puces, ICFPMJL1 v1 et v2, ont été conçues, mais seulement la deuxième version a été fabriquée. Au niveau du circuit de tests, une plaquette a été conçue afin de permettre la caractérisation de la puce à partir de son boîtier 84-PGA. Il est possible d'y insérer un circuit numérique programmable ainsi que d'utiliser les circuits intégrés analogiques présent sur cette plaquette.

En dernier lieu, les résultats des tests de la puce ICFPMJL1.v2 ont été présentés. Ces résultats constituent la validation de la majorité des idées avancées dans les chapitres précédents. Une autre section présente plusieurs possibilités pour les développements futurs.

Finalement, nous pouvons conclure que l'objectif de concevoir un capteur à large gamme dynamique a été atteint avec un résultat de 96dB. Rappelons que ceci a été possible seulement suite à l'élaboration de la nouvelle architecture présentée précédemment.

Tout est maintenant en place pour obtenir une caméra numérique sur une seule puce. Une subséquente itération sera effectuée afin de valider d'autres innovations en matière de capteur d'images ainsi que pour tester le système dans son ensemble. Ensuite, il sera possible de fabriquer une caméra complète avec une résolution standard.

Bibliographie

- [1] AGWANI, S., et al., *"A 1/3" VGA CMOS imaging system on a chip"*, Proc. 1999 IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors, p. 21, 1999.
- [2] AUDET, Y., DEEN, J., *"Performance of a Gate-Controlled Lateral Bipolar Phototransistor Fabricated in a Standard CMOS Process"*, McMaster University, Hamilton, Ontario.
- [3] BLANKSBY, A.J., and LOINAZ, M.J., *"Performance Analysis of a Color CMOS Photogate Image Sensor"*, IEEE Trans. Electron Devices, vol. 47, no. 1, pp. 55-64, Jan 2000.
- [4] BOHM, M., et al., *"High Dynamic Range Image Sensors in Thin Film on ASIC technology for automotive applications"*, Advanced Microsystems for Automotive Applications, Berlin, Germany: Springer-Verlag, p. 157, 1998.
- [5] BOUSSAID, F., BERMAK, A., and BOUZERDOUM, A., *"A Wide Dynamic Range CMOS Imager with Extended Shunting Inhibition Image Processing Capabilities"*, School of Engineering and Mathematics, Perth, 6027 WA, Australia, 2001.
- [6] BRAJOVIC, V., and KANADE, T., *"A VLSI Sorting Image Sensor: Global Massively Parallel Intensity-to-Time Processing for Low-Latency Adaptive Vision"*, IEEE Trans. On Robotics and Automation, vol. 15, no. 1, Feb. 1999.
- [7] CARLSON, D.E., and WRONSKI, R., *"Amorphous Silicon Solar Cells"*, Topics in Applied Physics, M.H. Brodsky, Ed. Berlin, Germany: Springer-Verlag, Amorphous Semiconductors, p. 287, 1985.
- [8] CHEN, S., GINOSAR, R., *"Adaptive Sensitivity CCD Image Sensor"*, Proc. SPIE, San Jose, CA, vol. 2415, pp. 303-309, Feb. 1995.
- [9] COULOMBE, J., *"Capteur d'images intégré en mode courant et à résolution variable"*, Mémoire de maîtrise, École Polytechnique de Montréal, Nov. 2000.

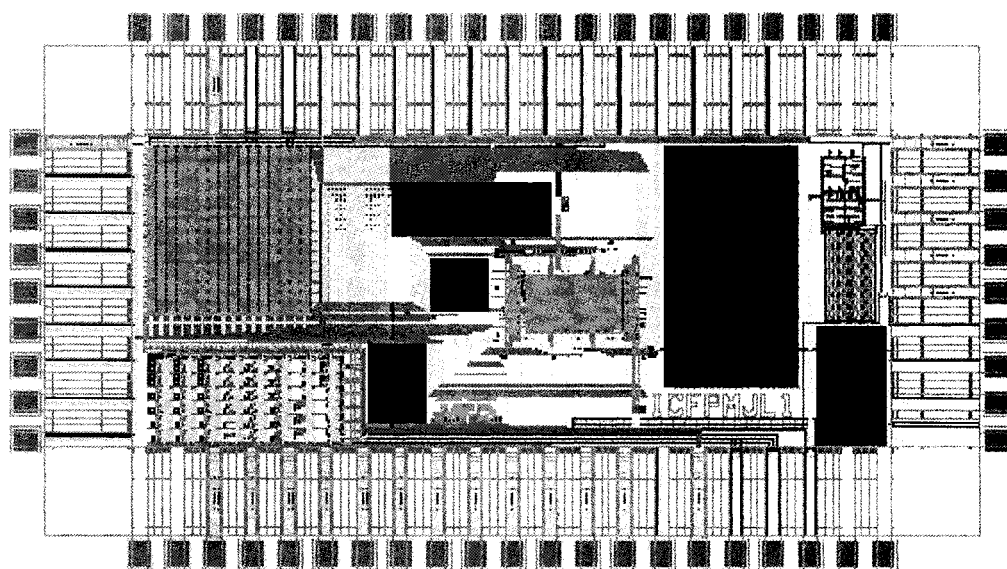
- [10] COULOMBE, J., SAWAN, M., WANG, C., "*Variable Resolution CMOS Current Mode Active Pixel Sensor*," IEEE International Symposium on Circuit and Systems, Geneva, Switzerland, May 2000.
- [11] DECKER, S., McGRATH, R. D., BREHMER, K., and SODINI C.G., "*A 256x256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output*," J. Solid State Circuits, vol. 33, pp. 2081-2091, Dec. 1998.
- [12] DIERICKX, B., "*CMOS Image Sensors concepts*", Photonics West 2000 Short Course, 2000.
- [13] EASTMAN KODAK COMPANY, Rochester NY, "*Kodak CMOS Image Sensors*", Nov. 2000.
- [14] EG&G RETICON, "*Image Sensing Products 1992/1993*", 345 Portero Ave., Sunnyvale, CA 94086-4197 USA, 1992.
- [15] EL GAMAL, A., YANG, D.X.D., and FOWLER, B., "*Pixel-Level Processing -- Why, What, and How?*", Information Systems Laboratory, Stanford University, Stanford, CA 94305, USA.
- [16] ETIENNE-CUMMINGS, R., "*Single-Capacitor-Single-Contact Active Pixel Sensor*", IEEE Int. Symposium on Circuits and Systems, Geneva, Switzerland, May 2000.
- [17] FAYOMI, C., "*Reliable Circuit Techniques for Low-Voltage Deep Sub-Micron CMOS Analog to Digital Converters*", Thèse de Doctorat, École Polytechnique de Montréal, Août 2003.
- [18] FOSSUM, E., "*Active Pixel Sensors: are CCD's dinosaurs*", Proc. SPIE, vol. 1900, pp. 2-14, San Jose, CA, Feb. 1993.
- [19] FOSSUM, E.R., "*CMOS image sensor: Electronic camera-on-a chip*", IEEE Trans. Electron Devices, vol. 44, p. 1689, Oct. 1997.
- [20] FOWLER, B., EL GAMAL, A., and YANG, D.X.D., "*Techniques for Pixel Level Analog to Digital Conversion*", Information Systems Laboratory, Stanford University.

- [21] FOWLER, B., EL GAMAL, A., and YANG, D.X.D., "*A CMOS Area Image Sensor With Pixel Level A/D Conversion*", Information Systems Laboratory, Stanford University, Stanford, CA 94305-4055, June 2001.
- [22] FOX, E.C., et al, "*A High Speed Linear CCD Sensor with Pinned Photodiode Photosite for Low Lag and Low noise Imaging*", Proc. SPIE, vol. 3301, p. 17, 1998.
- [23] HO, J.-S., "*A New design for a 1280x1024 Digital CMOS Image Sensor with Enhanced Sensitivity, Dynamic Range and FPN*", Electronic Research & Service Organisation (ERSO), Industrial Technology Research Institute (ITRI), Taiwan, R.O.C.
- [24] IHARA, H., et al., "*A 3.7 x 3.7 μm^2 square pixel CMOS Image Sensor for Digital Still Camera Application*", IEEE ISSCC Dig. Tech. Papers, p. 182, 1998.
- [25] KALAYJIAN, Z.K., and ANDREOU, A.G., "*Mismatch in Photodiode and Phototransistor Arrays*", John Hopkins University, Baltimore.
- [26] KAVADIAS, S., et al., "*A Logarithmic Response CMOS Image Sensor with On-Chip Calibration*", IEEE J. of Solid State Circuits, vol. 35, no. 8, pp. 1146-1152, August 2000.
- [27] KLEINFELDER, S., LIM, S., LIU, X., and EL GAMAL, A., "*A 10000 Frames/s CMOS Digital Pixel Sensor*," J. Solid State Circuits, vol. 36, pp. 2049-2059, Dec. 2001.
- [28] KOZLOWSKI, L.J., LUO, J., KLEINHANS, W.E., and LIU, T., "*Comparison of passive and active pixel schemes for CMOS visible imager*", Proc. SPIE 3360, Infrared Readout Electronics IV, 1998.
- [29] LOOSE, M., MEIER, K., and SCHEMMEL, J., "*A Self-Calibrating Single-Chip CMOS Camera with Logarithmic Response*", J. Solid State Circuits, vol. 36, no. 4, pp. 586-596, April 2001.
- [30] LULÉ, T., et al. "*Sensitivity of CMOS Based Imagers and Scaling Perspectives*", IEEE Trans. On Electron Devices, vol. 47, no. 11, Nov. 2000.

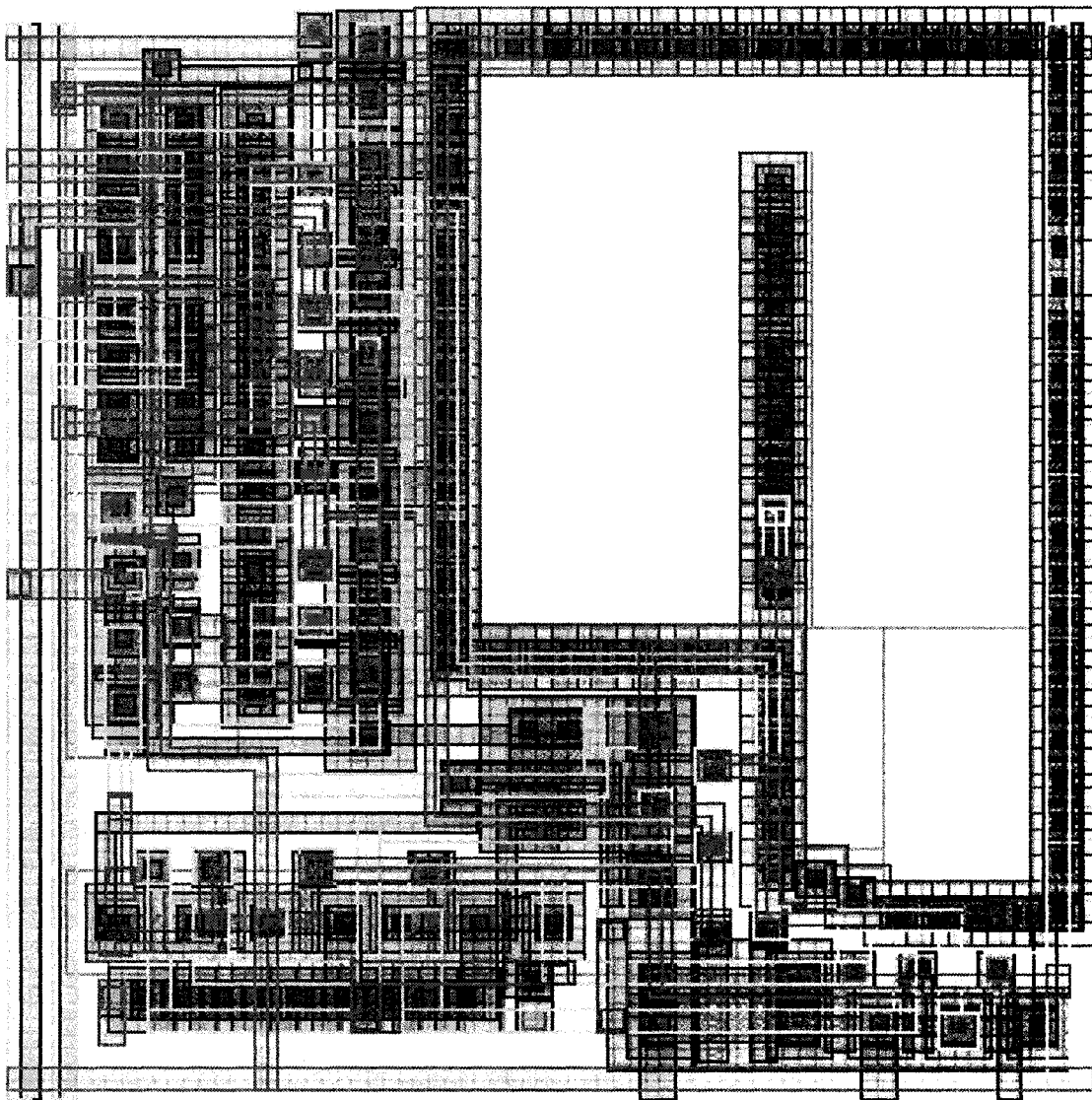
- [31] MARSHALL, G.F., and COLLINS S., "*A high-dynamic-range front end for automatic image processing applications*", Proc. SPIE 3410, Advanced Focal Plane Arrays and Electronic Cameras, Zurich, Switzerland, May 1998.
- [32] McILRATH, L.G., "*A Low-Power Low-Noise Ultrawide-Dynamic-Range CMOS Imager with Pixel-Parallel A/D Conversion*", IEEE J. of Solid State Circuits, vol. 36, pp. 846-853, May 2001.
- [33] NAKAMURA, J., "*Recent Progress of CMOS Image Sensors*", Corporate Research Laboratories, Olympus Optical Co. Ltd.
- [34] NOBLE, P., "*Self-scanned solid image detector arrays*", IEEE Tran. Electron Devices, vol. ED-15, p. 202, 1968.
- [35] OGIERS, W., "*Compact CMOS Vision Systems for Space Use*", IMEC, Belgium.
- [36] PAIN, B., and FOSSUM, E.R., "*Approaches and analysis for on-focal-plane analog-to-digital conversion*", Infrared Readout Electronics II, Proc. SPIE, vol. 2226, pp. 208-218, 1994.
- [37] PAIN, B., et al., "*A Single-chip Programmable Digital CMOS Imager with Enhanced Low-light Detection Capability*", California Institute of Technology, Jet Propulsion Laboratory.
- [38] RIPPS, H., and WEALE, R.A., "*Temporal Analysis and Resolution*", The Eye, H. Davson, Ed. New York: Academic, vol. 2A, pp. 185-217, 1976.
- [39] SANDAGE, R.W., and CONNELLY, A., "*Producing Phototransistors in a Standard Digital CMOS Technology*", Georgia Institute of Technology, Atlanta, Georgia.
- [40] SHIH, Y.-C., "*The Design of High-Performance 128x128 CMOS Image Sensors using New Current-Readout Techniques*", Institute of Electronics National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
- [41] TREPANIER, J.-L., SAWAN, M., AUDET, Y., and COULOMBE, J., "*A Wide Dynamic Range CMOS Digital Pixel Sensor*", IEEE Mid-West International Symposium on Circuits and Systems, Aug. 2002.

- [42] TU, N., HORNSEY, R., and INGRAM, S.G., "*CMOS Active Pixel Image Sensor with Combined Linear and Logarithmic mode Operation*", University of Waterloo, 1998.
- [43] WONG, H., "*Technology and Device Scaling Considerations for CMOS Imagers*", IEEE Trans. Electron Devices, vol. 43(12), pp. 2131-2141, 1996.
- [44] YANG, D.X.D., and EL GAMAL, A., "*Comparative Analysis of SNR for Image Sensors with Widened Dynamic Range*", Proc. SPIE, San Jose, CA, vol. 3649, pp. 197-211, Feb. 1999.
- [45] YANG, D.X.D., FOWLER, B. and EL GAMAL, A., "*A 128x128 pixel CMOS Area Image Sensor with Multiplexed Pixel Level A/D Conversion*", Information Systems Laboratory, Stanford University, Stanford, CA 94305-4055.
- [46] YANG, D.X.D., FOWLER, B., and EL GAMAL, A., "*A Nyquist-Rate Pixel-Level ADC for CMOS Image Sensors*," J. Solid State Circuits, vol. 34, pp. 348-356, Mar. 1999.
- [47] YANG, D.X.D., EL GAMAL, A., FOWLER, B., and TIAN, H., "*A 640x512 CMOS Image Sensor with Ultrawide Dynamic Range Floating-Point Pixel-Level ADC*," J. Solid State Circuits, vol. 34, pp. 1821-1834, Dec. 1999.
- [48] <http://micro.magnet.fsu.edu>
- [49] <http://www.gatan.com/specimenprep623discgrinder.html>

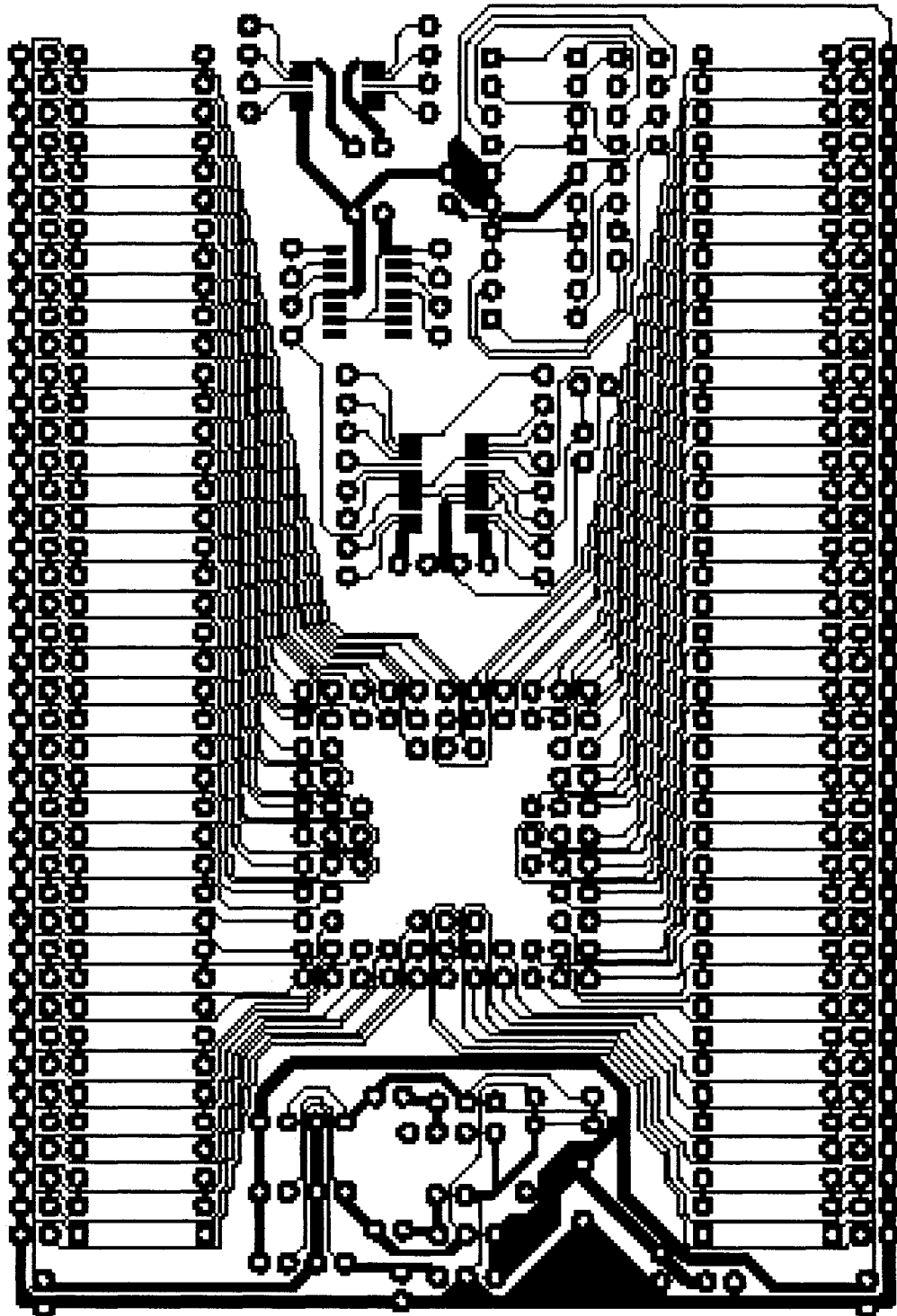
Annexe A
Dessin des masques de la puce ICFPMJL1.v1



Annexe B
Dessin des masques du pixel numérique



Annexe C
Circuit imprimé du module de test



Annexe D

Description VHDL du contrôleur

```

-----
-----
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--use std.textio.all;
--use ieee.std_logic_textio.all;
--use txt_util.ALL;
-----
-- Entity
-----
entity testPhoto is
  Port (
    ck      : in std_logic;
    nBtIn  : in std_logic;
    dipIn  : in std_logic_vector(3 downto 0);
    seg    : out std_logic_vector(7 downto 0);
    max521_sck      : out std_logic;
    max521_sda      : out std_logic;
    compar_phi1     : out std_logic;
    compar_phi2     : out std_logic;
    compar_validate : out std_logic;
    photo_fClr      : out std_logic;
    photo_nRst      : out std_logic;
    photo_shut      : out std_logic
  );
end testPhoto;
-----
-- Architecture begin
-----
architecture arch1 of testPhoto is
-----
-- Functions
-----
-- converts an integer into a character
-- for 0 to 9 the obvious mapping is used, higher
-- values are mapped to the characters A-Z
-- (this is usefull for systems with base > 10)
-- (adapted from Steve Vogwell's posting in comp.lang.vhdl)

function chr(int: integer) return character is
  variable c: character;
begin
  case int is
    when 0 => c := '0';
    when 1 => c := '1';
    when 2 => c := '2';
    when 3 => c := '3';
    when 4 => c := '4';
    when 5 => c := '5';
    when 6 => c := '6';
    when 7 => c := '7';
    when 8 => c := '8';
    when 9 => c := '9';
    when 10 => c := 'A';
    when 11 => c := 'B';
    when 12 => c := 'C';
    when 13 => c := 'D';
    when 14 => c := 'E';
  end case;
end function;

```

```

        when 15 => c := 'F';
        when 16 => c := 'G';
        when 17 => c := 'H';
        when 18 => c := 'I';
        when 19 => c := 'J';
        when 20 => c := 'K';
        when 21 => c := 'L';
        when 22 => c := 'M';
        when 23 => c := 'N';
        when 24 => c := 'O';
        when 25 => c := 'P';
        when 26 => c := 'Q';
        when 27 => c := 'R';
        when 28 => c := 'S';
        when 29 => c := 'T';
        when 30 => c := 'U';
        when 31 => c := 'V';
        when 32 => c := 'W';
        when 33 => c := 'X';
        when 34 => c := 'Y';
        when 35 => c := 'Z';
        when others => c := '?';
    end case;
    return c;
end chr;
-----
-- convert integer to string using specified base
-- (adapted from Steve Vogwell's posting in comp.lang.vhdl)

function str(int: integer; base: integer) return string is

    variable temp:    string(1 to 10);
    variable num:     integer;
    variable abs_int: integer;
    variable len:     integer := 1;
    variable power:   integer := 1;

begin

    -- bug fix for negative numbers
    abs_int := abs(int);

    num := abs_int;

    while num >= base loop
        len := len + 1;
        num := num / base;
    end loop ;

    for i in len downto 1 loop
        temp(i) := chr(abs_int/power mod base);
        power := power * base;
    end loop ;

    -- return result and add sign if required
    if int < 0 then
        return '-' & temp(1 to len);
    else
        return temp(1 to len);
    end if;

end str;
-----
procedure printMsgVal( msg :string; val :integer ) is
    variable tmp : integer := 0;
begin

```

```

        assert( tmp=-1 ) report msg & str(val,10) severity note;
end printMsgVal;
-----
-- Returns how many bits is needed to code 'maxVal'
function nbBits( maxVal : integer ) return integer is
    variable nb : integer;
    variable max : integer;
begin
    nb := 1;
    max := 2;

    while( max < maxVal ) loop
        nb := nb + 1;
        max := max * 2;
    end loop;
    printMsgVal( "Nb bits is = ", nb );

    return nb;
end nbBits;
-----
function count_periodUs( ckFreqIn, periodUsIn : integer ) return integer is
    variable us_s : integer := 1000000;
    variable ckFreq : integer := ckFreqIn;
    variable periodUs : integer := periodUsIn;
    variable result : integer;
begin

    printMsgVal( "Period in is = ", periodUs );

    -- to avoid overflow
    while( (ckFreq MOD 10)=0 and us_s>1 ) loop
        ckFreq := ckFreq / 10;
        us_s := us_s / 10;
    end loop;

    while( (periodUs MOD 10)=0 and us_s>1 ) loop
        periodUs := periodUs / 10;
        us_s := us_s / 10;
    end loop;

    result := ckFreq / us_s;
    result := result * periodUs;
    printMsgVal( "Result value is = ", result );

    return result;
end count_periodUs;
-----
-- Signals
-----
constant CK_FREQ : integer := 50000000;

constant UP_DOWN_COUNT_MAX : integer := CK_FREQ;--count_periodUs( CK_FREQ, 100000 );
constant UP_DOWN_WIDTH : integer := nbBits( UP_DOWN_COUNT_MAX );
signal upDownCount : std_logic_vector(UP_DOWN_WIDTH-1 downto 0);
signal upDown : std_logic;

constant CK_LOW_P_MAX : integer := count_periodUs( CK_FREQ, 10000 ) / 2;
constant CK_LOW_P_WIDTH : integer := nbBits( CK_LOW_P_MAX );
signal ckLow_period : std_logic_vector(CK_LOW_P_WIDTH-1 downto 0);
signal ckLowCount : std_logic_vector(CK_LOW_P_WIDTH-1 downto 0);
signal ckLow : std_logic;

constant RST_DELAY2 : integer := count_periodUs( CK_FREQ, 10000 );

```



```

constant NRST_P_MAX      : integer := count_periodUs( CK_FREQ, 700000 );
constant NRST_P_WIDTH   : integer := nbBits( NRST_P_MAX );
signal   nRst_period    : std_logic_vector(NRST_P_WIDTH-1 downto 0);
signal   periodCount    : std_logic_vector(NRST_P_WIDTH-1 downto 0); -- main counter

constant NRST_W_MAX     : integer := count_periodUs( CK_FREQ, 1000 );
constant NRST_W_WIDTH   : integer := nbBits( NRST_W_MAX );
signal   nRst_width     : std_logic_vector(NRST_W_WIDTH-1 downto 0);

constant SHUT_D_MAX     : integer := count_periodUs( CK_FREQ, 4000 );
constant SHUT_D_WIDTH   : integer := nbBits( SHUT_D_MAX );
signal   shut_delay     : std_logic_vector(SHUT_D_WIDTH-1 downto 0);

constant SHUT_W_MAX     : integer := count_periodUs( CK_FREQ, 200000 );
constant SHUT_W_WIDTH   : integer := nbBits( SHUT_W_MAX );
signal   shut_width     : std_logic_vector(SHUT_W_WIDTH-1 downto 0);

constant COMPAR_D_MAX   : integer := count_periodUs( CK_FREQ, 10000 );
constant COMPAR_D_WIDTH : integer := nbBits( COMPAR_D_MAX );
--signal   compar_delay : std_logic_vector(COMPAR_D_WIDTH-1 downto 0);

constant FCLR_D_MAX     : integer := count_periodUs( CK_FREQ, 20000 );
constant FCLR_D_WIDTH   : integer := nbBits( FCLR_D_MAX );
signal   fClr_delay     : std_logic_vector(FCLR_D_WIDTH-1 downto 0);

constant FCLR_W_MAX     : integer := count_periodUs( CK_FREQ, 100 );
constant FCLR_W_WIDTH   : integer := nbBits( FCLR_W_MAX );
signal   fClr_width     : std_logic_vector(FCLR_W_WIDTH-1 downto 0);

signal   rst            : std_logic;
--signal   incStep      : std_logic;
signal   bPressed      : std_logic;
signal   bActive       : std_logic;
signal   bt1           : std_logic;
signal   dip            : std_logic_vector(3 downto 0);
signal   ledNum        : std_logic_vector(3 downto 0); -- number (hex) to display on leds

constant max521_outId   : std_logic_vector(2 downto 0) := "000";
signal   max521_en      : std_logic;
signal   max521_dacVal  : std_logic_vector(7 downto 0);
signal   max521_busy    : std_logic;
signal   sMax521_sck    : std_logic;
signal   sMax521_sda    : std_logic;

signal   compar_en      : std_logic;
signal   compar_busy    : std_logic;
signal   compar_compar  : std_logic;
signal   compar_comparVal : std_logic;
-----
-- Components
-----
COMPONENT led7seg
  PORT(
    num : IN std_logic_vector(3 downto 0);
    seg : OUT std_logic_vector(6 downto 0)
  );
END COMPONENT;
-----
COMPONENT max521
  PORT(
    ck : IN std_logic;
    rst : IN std_logic;
    enable : IN std_logic;
    outId : IN std_logic_vector(2 downto 0);
    dacVal : IN std_logic_vector(7 downto 0);
    busy : OUT std_logic;
    sck : OUT std_logic;
  );

```

```

        sda : OUT std_logic
    );
END COMPONENT;
-----
COMPONENT compar
    PORT(
        ck : IN std_logic;
        rst : IN std_logic;
        enable : IN std_logic;
        compar : IN std_logic;
        busy : OUT std_logic;
        comparVal : OUT std_logic;
        phil1 : OUT std_logic;
        phi2 : OUT std_logic;
        validate : OUT std_logic
    );
END COMPONENT;
-----
begin
-----
-- Port Map
-----
U_led7seg: led7seg PORT MAP(
    num => ledNum,
    seg => seg(6 downto 0)
);
-----
U_max521: max521 port map(
    ck => ckLow,
    rst => rst,
    enable => max521_en,
    outId => max521_outId,
    dacVal => max521_dacVal,
    busy => max521_busy,
    sck => sMax521_sck,
    sda => sMax521_sda
);
-----
U2_compar: compar PORT MAP(
    ck => ckLow,
    rst => rst,
    enable => compar_en,
    busy => compar_busy,
    compar => compar_compar,
    phil1 => compar_phil1,
    phi2 => compar_phi2,
    validate => compar_validate,
    comparVal => compar_comparVal
);
-----
-- Connections
-----
rst <= '1' when dip = "0000" else '0';

--max521_outId <= "000";
max521_sck <= sMax521_sck;
max521_sda <= sMax521_sda;
-----
-- Processes
-----
p_upDown: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            upDownCount <= (others=>'0');
            upDown <= '0';

```

```

elseif( upDownCount = UP_DOWN_COUNT_MAX ) then
    upDownCount <= (others=>'0');
    upDown <= not upDown; -- toggle up-down indicator
else
    if( bt1 = '1' ) then
        -- button is pressed, reset count
        upDownCount <= (others=>'0');
    else
        -- increment counter
        upDownCount <= upDownCount + 1;
    end if;
end if;
end if; -- ck event

end process;
-----
p_buttonOneCycle: process( ck )
begin
    if( ck'event and ck='1' ) then

        if( rst = '1' ) then
            bPressed <= '0';
            bActive <= '0';
        end if;

        if( bActive = '1' ) then
            bActive <= '0'; -- one cycle only
        end if;

        if( bt1 = '1' and bPressed = '0' ) then
            bPressed <= '1';
            bActive <= '1';
        end if;

        if( bt1 = '0' and bPressed = '1' ) then
            bPressed <= '0';
        end if;
    end if;
end process;
-----
p_ckLow: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            ckLowCount <= (others=>'0');
            ckLow <= '0';
        elsif( ckLowCount = ckLow_period ) then
            ckLowCount <= (others=>'0');
            ckLow <= not ckLow;
        else
            ckLowCount <= ckLowCount + 1;
        end if;
    end if;
end process;
-----
p_syncAsyncInput: process( ck )
begin
    if( ck'event and ck='1' ) then
        -- synchronizer on asyn inputs
        bt1 <= not nBtIn;
        dip <= not dipIn;

        -- buffer LED output
        if( dip = "1111" ) then
            -- quiet mode (no blinking LEDs, reduce noise)
            seg(7) <= '1';
        else

```

```

                seg(7) <= upDown;
            end if;
        end if;
    end process;
-----
p_setValues: process( ck )
begin
    if( ck'event and ck='1' ) then

        -- default values
        ledNum <= dip;

        case dip is
            when "0000" => -- 0 -- ResetAll
                nRst_period <= (others=>'1');
                nRst_width <= (others=>'1');
                shut_delay <= (others=>'1');
                shut_width <= (others=>'1');
                fClr_delay <= (others=>'1');
                fClr_width <= (others=>'0');      fClr_width(0) <= '1';
                ckLow_period <= (others=>'1');
                max521_en <= '0';
                max521_dacVal <= (others=>'1');
                compar_delay <= (others=>'0');
            --

            when "0001" => -- 1 -- Change 'nRst' period
                if( bActive = '1' ) then
                    if( upDown = '0' ) then
                        nRst_period(NRST_P_WIDTH-2 downto 0) <=
nRst_period(NRST_P_WIDTH-1 downto 1);
                        nRst_period(NRST_P_WIDTH-1) <= '0';
                    else
                        nRst_period(NRST_P_WIDTH-1 downto 1) <=
nRst_period(NRST_P_WIDTH-2 downto 0);
                        nRst_period(0)
<= '0';
                    end if;
                end if;
                if( nRst_period = 0 ) then
                    nRst_period(0) <= '1';
                end if;

            when "0010" => -- 2 -- Change 'nRst' width
                if( bActive = '1' ) then
                    if( upDown = '0' ) then
                        nRst_width(NRST_W_WIDTH-2 downto 0) <=
nRst_width(NRST_W_WIDTH-1 downto 1);
                        nRst_width(NRST_W_WIDTH-1)
<= '0';
                    else
                        nRst_width(NRST_W_WIDTH-1 downto 1) <=
nRst_width(NRST_W_WIDTH-2 downto 0);
                        nRst_width(0)
<= '0';
                    end if;
                end if;
                if( nRst_width = 0 ) then
                    nRst_width(0) <= '1';
                end if;

            when "0011" => -- 3 -- Change 'shut' delay
                if( bActive = '1' ) then
                    if( upDown = '0' ) then
                        shut_delay(SHUT_D_WIDTH-2 downto 0) <=
shut_delay(SHUT_D_WIDTH-1 downto 1);
                        shut_delay(SHUT_D_WIDTH-1)
<= '0';
                    end if;
                end if;
            end if;
        end case;
    end if;
end process;

```

```

else
    shut_delay(SHUT_D_WIDTH-1 downto 1) <=
shut_delay(SHUT_D_WIDTH-2 downto 0);
    shut_delay(0)
    <= '0';
    end if;
end if;
if( shut_delay = 0 ) then
    shut_delay(0) <= '1';
end if;

when "0100" => -- 4 -- Change 'shut' width
    if( bActive = '1' ) then
        if( upDown = '0' ) then
            shut_width(SHUT_W_WIDTH-2 downto 0) <=
shut_width(SHUT_W_WIDTH-1 downto 1);
            shut_width(SHUT_W_WIDTH-1)
            <= '0';
        else
            shut_width(SHUT_W_WIDTH-1 downto 1) <=
shut_width(SHUT_W_WIDTH-2 downto 0);
            shut_width(0)
            <= '0';
        end if;
    end if;
    if( shut_width = 0 ) then
        shut_width(0) <= '1';
    end if;

    when "0101" => -- 5 -- Change 'fClr' delay
        if( bActive = '1' ) then
            if( upDown = '0' ) then
                fClr_delay(FCLR_D_WIDTH-2 downto 0) <=
fClr_delay(FCLR_D_WIDTH-1 downto 1);
                fClr_delay(FCLR_D_WIDTH-1)
                <= '0';
            else
                fClr_delay(FCLR_D_WIDTH-1 downto 1) <=
fClr_delay(FCLR_D_WIDTH-2 downto 0);
                fClr_delay(0)
                <= '0';
            end if;
        end if;
        if( fClr_delay = 0 ) then
            fClr_delay(0) <= '1';
        end if;

        when "0110" => -- 6 -- Change 'fClr' width
            if( bActive = '1' ) then
                if( upDown = '0' ) then
                    fClr_width(FCLR_W_WIDTH-2 downto 0) <=
fClr_width(FCLR_W_WIDTH-1 downto 1);
                    fClr_width(FCLR_W_WIDTH-1)
                    <= '0';
                else
                    fClr_width(FCLR_W_WIDTH-1 downto 1) <=
fClr_width(FCLR_W_WIDTH-2 downto 0);
                    fClr_width(0)
                    <= '0';
                end if;
            end if;
            if( fClr_width = 0 ) then
                fClr_width(0) <= '1';
            end if;

            when "1000" => -- 8 -- ckLow period
                if( bActive = '1' ) then

```

```

                                if( upDown = '0' ) then
ckLow_period(CK_LOW_P_WIDTH-1 downto 1);      ckLow_period(CK_LOW_P_WIDTH-2  downto  0)  <=
                                ckLow_period(CK_LOW_P_WIDTH-1)
                                <= '0';
                                else
ckLow_period(CK_LOW_P_WIDTH-2 downto 1);      ckLow_period(CK_LOW_P_WIDTH-1  downto  1)  <=
                                ckLow_period(0)
                                <= '0';
                                end if;
                                end if;
                                if( ckLow_period = 0 ) then
                                    ckLow_period(0) <= '1';
                                end if;
                                if( compar_busy = '0' ) then
                                    compar_en <= '1';
                                else
                                    compar_en <= '0';
                                end if;

when "1010" => -- A -- MAX521 change DAC value and enable output
    if( bt1 = '1' and max521_busy = '0' ) then
        if( upDown = '1' ) then
            max521_dacVal <= max521_dacVal + 10;
        else
max521_dacVal + 10;
            max521_dacVal      <=      (others=>'0');--
            end if;
            max521_en <= '1';
        else
            max521_en <= '0';
        end if;

when "1011" => -- B -- Comparator direct control (with button)
    if( bt1 = '1' and compar_busy = '0' ) then
        compar_en <= '1';
    else
        compar_en <= '0';
    end if;

when "1100" => -- C -- Comparator direct control (not button)
    if( compar_busy = '0' ) then
        compar_en <= '1';
    else
        compar_en <= '0';
    end if;

    when others =>
        NULL;
    end case;

    end if;
end process;
-----
p_runSequence: process( ck )
begin
    if( ck'event and ck='1' ) then

        if( rst = '1' ) then
            periodCount <= (others=>'0');
            photo_nRst <= '1';
            photo_shut <= '0';
            photo_fClr <= '0';
        else
            -- default action, unless other assignation
            periodCount <= periodCount + 1;
        end if;
    end if;
end process;

```

```

end if;

-- Control for 'nRst' signal
if( periodCount = 0 ) then
    photo_nRst <= '0';
elsif( periodCount = nRst_width ) then
    photo_nRst <= '1';
elsif( periodCount = nRst_period ) then
    periodCount <= (others=>'0');
end if;
if( dip = "0011" ) then
if( periodCount = RST_DELAY2 and dip="0011") then
    -- to test photon blooming block
    photo_nRst <= '0';
end if;
end if;

-- Control for 'shut' signal
if( periodCount = 0 ) then
    photo_shut <= '0';
elsif( periodCount = shut_delay ) then
    photo_shut <= '1';
elsif( periodCount = shut_delay + shut_width ) then
    photo_shut <= '0';
end if;
if( dip="0001" or dip="0010" ) then
    photo_shut <= '0';
end if;

-- Control for 'compar_en' signal
--if( periodCount = 0 ) then
--    compar_en <= '0';
--elsif( periodCount = compar_delay ) then
--    if( bt1 = '1' and compar_busy = '0' ) then
--        compar_en <= '1';
--    else
--        compar_en <= '0';
--    end if;
--end if;

-- Control for 'fClr' signal
if( periodCount = 0 ) then
    photo_fClr <= '0';
elsif( periodCount = fClr_delay ) then
    photo_fClr <= '1';
elsif( periodCount = fClr_delay + fClr_width ) then
    photo_fClr <= '0';
end if;
photo_fClr <= '0';--tmp

    end if;
end process;
-----
p_template: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            else
                end if;
        end if;
    end process;
-----
end arch1;
-----
-----
-----
-----

```

```

-----
-----
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-----
-- Entity
entity genericTest is
  Port (
        ck      : in std_logic;
        rstIn   : in std_logic;
        nBt1    : in std_logic;
        seg     : out std_logic_vector(7 downto 0); -- 7-segments LEDs
        inPar   : in  std_logic_vector(6 downto 0); -- Parallel port input
        outPar  : out std_logic_vector(3 downto 0); -- Parallel port
output
        stimIn  : in  std_logic_vector(3 downto 0);
        stimOut: out std_logic_vector(3 downto 0)
        );
end genericTest;
-----
architecture arch1 of genericTest is
-----
-- Signals
--constant waitCountMax : std_logic_vector(7 downto 0) := "11111111";
--signal waitCount      : std_logic_vector(7 downto 0);

-- signaux synchronises
signal sInPar : std_logic_vector(6 downto 0);
signal rst    : std_logic;

constant WIDTH : integer := 24;
signal ckLow   : std_logic;
signal ckDiv   : std_logic_vector(WIDTH-1 downto 0);
signal ckCount : std_logic_vector(WIDTH-1 downto 0);

signal ledNum : std_logic_vector(3 downto 0); -- number (hex) to display on leds
signal lastParId : std_logic; -- Parallel port toggle bit, last value read

--constant NB_VECT : integer := 2;
--type tVectAddr is array (NB_VECT-1 downto 0) of std_logic_vector(WIDTH-1 downto 0);
--type tVectTime is array (NB_VECT-1 downto 0) of std_logic_vector(WIDTH-1 downto 0);
--signal stimAddr : tVectAddr;
--signal stimTime : tVectTime;
--signal stimVal  : std_logic_vector(NB_VECT-1 downto 0);

type tState is ( eIdle,
                 eResetAll,
                 eSetPeriod,
                 eSetPeriod_1,
                 eSetPeriod_2,
                 eSetPeriod_3,
                 eSetPeriod_4,
                 eSetStimul,
                 eXXX
               );

signal curState : tState;
signal nextState : tState;
-----
-- Components
-----
COMPONENT led7seg
  PORT(
    num : IN std_logic_vector(3 downto 0);
    seg : OUT std_logic_vector(6 downto 0)
  )

```



```

    );
END COMPONENT;
-----
-- Architecture begin
begin
-----
-- Connections
seg(7) <= ckLow;
ledNum <= inPar(3 downto 0);
-----
-- Port Map
-----
U_led7seg: led7seg PORT MAP(
    num => ledNum,
    seg => seg(6 downto 0)
);
-----
-- Process
-----
syncAsync: process( ck )
begin

    if( ck'event and ck='1' ) then
        -- Ajout d'un synchroniseur pour les entrees asynchrones
        sInPar    <= inPar;
        rst       <= rstIn;
        --lastParId <= sInPar(6);
    end if;

end process;
-----
dataIO: process( ck )
begin

    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            --
            ledNum    <= "1110"; -- E
            ckDiv     <= (others=>'1');
            --outPar   <= (others=>'1');
            outPar    <= inPar(3 downto 0);

            --
            stimAddr(0) <= (others=>'0');
            --
            stimTime(0) <= (others=>'0');
            --
            stimVal    <= (others=>'0');
            stimOut    <= (others=>'0');
        else
            -- default values
            outPar <= (others=>'1');
            --
            ledNum <= "1101"; -- D

            case curState is

                when eIdle =>
                    --
                    ledNum <= "0000"; -- 0
                    outPar <= (others=>'0');

                when eResetAll =>
                    ckDiv <= (others=>'1');
                    stimAddr(0) <= (others=>'0');
                    --
                    stimTime(0) <= (others=>'0');
                    --
                    stimVal    <= (others=>'0');
                    --
                    stimOut    <= (others=>'0');
                    --
                    ledNum <= "0001"; -- 1

                when eSetPeriod =>
                    ckDiv <= (others=>'0');
            end case;
        end if;
    end if;
end process;

```

```

--
--
--          ledNum <= "0010";
--          when eSetPeriod_1 =>
--              ckDiv(5 downto 0) <= sInPar(5 downto 0);
--          when eSetPeriod_2 =>
--              ckDiv(11 downto 6) <= sInPar(5 downto 0);
--          when eSetPeriod_3 =>
--              ckDiv(17 downto 12) <= sInPar(5 downto 0);
--          when eSetPeriod_4 =>
--              ckDiv(23 downto 18) <= sInPar(5 downto 0);
--
--          when eSetStimul =>
--              ledNum <= "0011"; -- 3
--              stimVal(0) <= '1';
--
--          when others =>
--              ledNum <= "1111"; -- F
--
--          end case;
--
--          lastParId <= sInPar(6);
--
--          if( nBt1 = '0' ) then
--              -- Button is pressed
--              ckDiv <= "000000001000000000000000";
--              ledNum <= "1010"; -- A
--          end if;
--
--          -- Output stimul
--          stimOut(1) <= ckLow;
--signal stimAddr : tVectAddr;
--signal stimTime : tVectTime;
--          stimOut(0) <= stimVal(0);
--
--          end if;--rst
--          end if;--ck'event
--
--end process;
-----
periodCountSync: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            ckLow <= '1';
            ckCount <= (others=>'0');
        else
            -- Clock divider
            if( ckCount = ckDiv ) then
                ckCount <= (others=>'0');
                ckLow <= not ckLow;--'1';
            else
                --ckLow <= '0';
                ckCount <= ckCount + 1;
            end if;
        end if;
    end if;
end if; -- ck event
end process;
-----
fsmSync: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            curState <= eIdle;
        else -- rst
            curState <= nextState;
        end if;
    end if;
end process;

```

```

        end if;
    end if;

end process;
-----
fsmAsync: process( curState, sInPar, lastParId )
begin
    case curState is
        when eIdle =>
            nextState <= eIdle; -- default value
            -- Incoming parallel port data
            if( lastParId /= sInPar(6) ) then
                if( sInPar(5 downto 4) = "00" ) then
                    nextState <= eResetAll;
                elsif( sInPar(5 downto 4) = "01" ) then
                    nextState <= eSetPeriod;
                --
                --
                elsif( sInPar(5 downto 4) = "10" ) then
                    nextState <= eSetStimul;
                end if;
            end if;

        when eResetAll =>
            nextState <= eResetAll;
            if( lastParId /= sInPar(6) ) then
                nextState <= eIdle;
            end if;

        when eSetPeriod =>
            nextState <= eSetPeriod; -- default value
            if( lastParId /= sInPar(6) ) then
                nextState <= eSetPeriod_1;
            end if;

        when eSetPeriod_1 =>
            nextState <= eSetPeriod_1; -- default value
            if( lastParId /= sInPar(6) ) then
                nextState <= eSetPeriod_2;
            end if;

        when eSetPeriod_2 =>
            nextState <= eSetPeriod_2; -- default value
            if( lastParId /= sInPar(6) ) then
                nextState <= eSetPeriod_3;
            end if;

        when eSetPeriod_3 =>
            nextState <= eSetPeriod_3; -- default value
            if( lastParId /= sInPar(6) ) then
                nextState <= eSetPeriod_4;
            end if;

        when eSetPeriod_4 =>
            if( lastParId /= sInPar(6) ) then
                nextState <= eIdle;
            end if;

        --
        --
        --
        when eSetStimul =>
            if( lastParId /= sInPar(6) ) then
                nextState <= eIdle;
            end if;

        when others =>
            nextState <= eXXX;

    end case;
end process;

```

```
end process;
```

```
-----
```

```
end arch1;
```

```
-----
```

```
-----
```

```
-----
```

```

-----
-----
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-----
entity max521 is
    Port ( ck : in std_logic;
          rst : in std_logic;
          enable : in std_logic;
          outId : in std_logic_vector(2 downto 0);
          dacVal : in std_logic_vector(7 downto 0);
          busy : out std_logic;
          sck : buffer std_logic;
          sda : buffer std_logic);
end max521;
-----
architecture arch1 of max521 is
-----
-- Signals
type tState is ( stateIdle,
                 stateAddr,
                 stateCmd,
                 stateData );

signal curState : tState;
signal reg : std_logic_vector( 9 downto 0 );

signal i2cIn      : std_logic;
signal i2cEnable  : std_logic;
signal i2cBusy    : std_logic;
signal i2cLoad    : std_logic;
-----
begin
-----
-- Connections
i2cIn <= reg(9);
busy <= i2cBusy;
-----
-- Processes
-----
-- MSB to LSB
main: process( ck )
begin
    if( ck'event and ck='1' ) then
        if( rst = '1' ) then
            curState <= stateIdle;
            reg <= (others=>'-');
            i2cEnable <= '0';
        elsif( curState=stateIdle and (enable='1' or i2cEnable='1') ) then
            i2cEnable <= '1';
            if( i2cBusy='1' ) then
                -- i2c has started, so send data
                curState <= stateAddr;
                reg(9 downto 2) <= "01010000"; -- Physical address
                reg(1 downto 0) <= "01"; -- ACK=0, END=1
            end if;
        elsif( curState/=stateIdle and sck='1' ) then
            -- change data
            if( reg(8 downto 0) = "100000000" ) then
                -- change state
                if( curState = stateAddr ) then
                    curState <= stateCmd;
                    reg(9 downto 7) <= "000"; -- reserved
                    reg(6) <= '0'; -- reset DACs to 0
                end if;
            end if;
        end if;
    end if;
end process;

```

```

        reg(5)                <= '0'; -- power down
        reg(4 downto 2) <= outId; -- DAC select
        reg(1 downto 0) <= "01"; -- ACK=0, END=1
    elsif( curState = stateCmd ) then
        curState <= stateData;
        reg(9 downto 2) <= dacVal;
        --reg(9) <= dacVal(0);
        --reg(8) <= dacVal(1);
        --reg(7) <= dacVal(2);
        --reg(6) <= dacVal(3);
        --reg(5) <= dacVal(4);
        --reg(4) <= dacVal(5);
        --reg(3) <= dacVal(6);
        --reg(2) <= dacVal(7);
        reg(1 downto 0) <= "01"; -- ACK=0, END=1
    elsif( curState = stateData ) then
        curState <= stateIdle;
        reg <= (others=>'-');
        i2cEnable <= '0';
    end if;
else
    -- shift bits, LSB to MSB
    reg(9 downto 1) <= reg(8 downto 0);
    reg(0) <= '0';
end if;
end if; -- i2cBusy=0
end if;

end process;
-----
-- Start communication with i2cEnable=1, anytime
-- Stop communication with i2cEnable=0, when sck=0
-- Change data when sck=1
i2c: process( ck )
begin

    if( ck'event and ck='1' ) then
        if( rst='1' ) then
            sck <= '1';
            sda <= '1';
            i2cBusy <= '0';
            i2cLoad <= '0'; -- no data
        elsif( i2cEnable='1' ) then --or i2cBusy='1' ) then
            -- data to communicate
            if( sck='1' and sda='1' and i2cBusy='0' ) then
                -- start condition 1
                sda <= '0';
                i2cBusy <= '1';
            elsif( sck='1' ) then
                -- sck returning to 0
                sck <= '0';
                i2cLoad <= '0';
            elsif( sck='0' and i2cLoad='0' ) then
                -- load data
                sda <= i2cIn;
                i2cLoad <= '1';
            elsif( sck='0' and i2cLoad='1' ) then
                -- send (loaded) data
                sck <= '1';
            else
                sda <= '-';
                sck <= '-';
                i2cBusy <= '-';
            end if;
        else -- i2cEnable='0'
            -- stop condition
            if( sck='0' and sda='0' ) then

```

```
        sck <= '1';
        i2cBusy <= '1';
    elsif( sck='1' and sda='0' ) then
        sda <= '1';
        i2cBusy <= '0';
    elsif( sck='1' and sda='1' ) then
        -- do nothing
        sck <= '1';
        sda <= '1';
        i2cBusy <= '0';
    else
        -- goto stop condition 1
        sck <= '0';
        sda <= '0';
        i2cBusy <= '1';
    end if;
end if; -- i2cEnable
end if;

end process;
-----
end arch1;
-----
-----
-----
```