PIXELS FOR FOCAL-PLANE SCALE SPACE GENERATION AND FOR HIGH DYNAMIC RANGE IMAGING

Fernanda Duarte Vilela Reis de Oliveira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: José Gabriel Rodríguez Carneiro Gomes
Ángel Rodríguez-Vázquez

Rio de Janeiro
Março de 2018

PIXELS FOR FOCAL-PLANE SCALE SPACE GENERATION AND FOR HIGH DYNAMIC RANGE IMAGING

Fernanda Duarte Vilela Reis de Oliveira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. José Gabriel Rodríguez Carneiro Gomes, Ph.D.

Prof. Ángel Rodríguez-Vázquez, Ph.D.

Prof. Antonio Petraglia, Ph.D.

Prof. Fernando Antônio Pinto Barúqui, D.Sc.

Prof. Sergio Bampi, Ph.D.

Prof. Davies William de Lima Monteiro, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2018

Oliveira, Fernanda Duarte Vilela Reis de

Pixels for Focal-Plane Scale Space Generation and for High Dynamic Range Imaging/Fernanda Duarte Vilela Reis de Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2018.

XXV, 110 p.: il.; 29, 7cm.

Orientadores: José Gabriel Rodríguez Carneiro Gomes
Ángel Rodríguez-Vázquez

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 95 – 106.

1. Smart cameras. 2. CMOS image sensors. 3. HDR. I. Gomes, José Gabriel Rodríguez Carneiro *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

To my family and friends, who
have supported me
unconditionally during the last
four years.

Acknowledgments

I would like to thank everyone that has supported me not only during this work but also during the graduation and master degree years. This is actually a quite long list of friends and family without whom this work would not have been possible, and which I hope to honor in this acknowledgments section. First, I would like to thank my advisors José Gabriel Rodríguez Carneiro Gomes and Ángel Rodríguez Vázquez, and my unofficial advisors Ricardo Carmona Galán and Jorge Fernández Berni. Thank you for your support and all the knowledge that you have passed to me. Your guidance was of extreme importance for this work to be done, I could not have asked for a better group of advisors.

I would also like to thank professors Sergio Bampi, Davies Monteiro, Antonio Petraglia and Fernando Barúqui for participating in the defense committee. Thank you for your comments, they have helped improving this work and they will also help in future research.

Special thanks to Gustavo Martins da Silva Nunes, my dear friend who has worked with me during the last year of this work. Gustavo has helped me with many tests and ideas. His master degree work has contributed with topics that complement and enrich this work. Our mutual motivational talks were also very important to ensure my progress and I was always inspired by his dedication.

Rocío del Río, thanks for all your contributions in our paper and for agreeing on participating in such short notice. Thank you also for your help with the chip final building blocks.

To all my professors, specially Joarez Bastos, Fernando Barúqui, Antonio Petraglia, Mariane Petraglia, Carlos Teodósio, Antônio Mesquita, Luiz Wagner Biscainho, Jomar Gozzi, Antonio Cláudio, Casé D'Avila, and, my advisor, José Gabriel Rodríguez Carneiro Gomes. You are my role models of dedication and care towards your students, thank you for being such great professors.

Thanks to my friends from the Universidade Federal do Rio de Janeiro, Fabián, Ana, Genildo, Thiago Brito, João Alberto, Allan, Eduardo, Tiago Monnerat, Gustavo, Leonardo, Lucas, Odair, Manoel, Motta, Felipe Petraglia, Felipe Senra, Paulo, Gabriela Chaves, for the help, encouragement and knowledge exchange.

To my high school friends Barbara Bomfim, Stella Melgaço, Bruno Gherman,

Pedro Grojsgold, Daniel Marques, Rodrigo Oliveira, João Augusto Marzagão and Hyrllann Almeida. I know I can always count on you. Thank you for being there for me and for all the reunions and laughter. Thanks also to Gabriel Melgaço, for the encouragement and care.

To my childhood friends Felipe Pacheco, Yasmin, Jonas and Mark, thank you too for all these years of friendship, and Felipe Clark and Felipe Vianna, for joining the group. Felipe Vianna, thanks also for the Skype talks during my first months in Seville, your support was very important.

Rodrigo Arcuri, thank you for showing me the singing classes that you attended and for introducing me to Patricia Evans and to Dani. During the time that I attended the classes, singing seemed to be the solution for every problem. Thanks for the amazing lessons, Patricia.

To all my friends from the Instituto de Microelectrónica de Sevilla, specially, Juan Antonio, Mikel, Evangelos, Jorge, Rocío, James, Angela, Elisa, Marco, Alfredo, Ion, Pablo, Gustavo, Gloria, José Miguel, Lola, José Juan, Luis Carranza, Luis Escudero, Piedad, Adrián, Mazyar, Manuel, Cristina, María del Mar, Ricardo and Ángel, thank you for receiving me extremely well during my one year exchange program.

Thanks also to all my family for the encouragement and support. Marcela, Luiz Fernando, Rafael (and Luiza, though she was not born at the time), Carolina, Gustavo, uncle Rico, aunt Rita and mom, thank you for visiting me at Seville, I can not begin to tell how grateful and happy I was for receiving you. Thanks to Priscila, Gabi e José, for receiving me in their house and showing me Madrid. Aunts Patricia, Mônica, Lucynha and Penha, and uncles Ene, Edi, Fernando and Gil, thank you for your love and for always rooting for me.

Alberto Arias, thank you very much for all your love. You have helped me not only emotionally, but also, in many situations, technically. Thank you also for all the words of encouragement and for helping me overcame challenges, many times imposed by my own insecurities. Thanks also to your family, Carmen, Alberto, Andrés, Beatriz, Borja and Ivana, for all the kindness and support.

Finally, a huge thank you to my mother, Cristina Duarte, my greatest supporter. Thank you for your love and care, I could never have done this work without you by my side. I would also like to thank my father, Fernando Oliveira, *in memoriam*.

This work was supported by Brazilian research funding agencies (CAPES, CNPq, and FAPERJ) through projects 309148/2013-8, 479437/2013-0, 309602/2016-5, E-26/201.514/2014 and grants 141288/2014-0, 204382/2014-9 and E-26/200.350/2016, by the Spanish Government through project TEC2015-66878-C3-1-R MINECO (European Region Development Fund, ERDF/FEDER), by Junta de Andalucia through project TIC 2338-2013 CEICE and by the Office of Naval Research (USA) through grant N000141410355.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

PIXELS PARA GERAÇÃO DO ESPAÇO DE ESCALAS NO PLANO FOCAL E PARA CAPTURA EM ALTA FAIXA DINÂMICA

Fernanda Duarte Vilela Reis de Oliveira

Março/2018

Orientadores: José Gabriel Rodríguez Carneiro Gomes
Ángel Rodríguez-Vázquez

Programa: Engenharia Elétrica

O processamento no plano focal de imageadores permite que a imagem capturada seja processada em paralelo por toda a matrix de pixels, característica que pode aumentar a velocidade de sistemas de visão. Ao fabricar circuitos dentro da matrix de pixels, o tamanho do pixel aumenta e a razão entre área fotossensível e a área total do pixel diminui, reduzindo a qualidade da imagem. Para utilizar as vantagens do processamento no plano focal e minimizar a redução da qualidade da imagem, a primeira parte da tese propõe a inclusão de dois transistores no pixel, o que permite que o espaço de escalas da imagem capturada seja gerado. Nós então avaliamos em quais condições o circuito proposto é vantajoso. Nós analisamos o tempo de processamento e o consumo de energia dessa proposta em comparação com uma solução digital. Utilizando um conversor de aproximações sucessivas com frequência de 5.6 MHz, a análise proposta mostra que a abordagem no plano focal é 26 vezes mais rápida que o circuito digital com 10 elementos de processamento, e consome 49 vezes menos energia. Outra maneira de utilizar processamento no plano focal consiste em aplicá-lo para melhorar a qualidade da imagem, como na captura de imagens em alta faixa dinâmica. Esta tese também apresenta o estudo e projeto de um pixel que realiza a captura de imagens em alta faixa dinâmica através do ajuste do tempo de integração de cada pixel utilizando a iluminação média e o valor do próprio pixel. Esse pixel foi projetado considerando pequenas variações estruturais, como diferentes tamanhos do fotodiodo que realiza a captura do valor de iluminação médio. Simulações de esquemático e pós-layout foram realizadas com o pixel projetado utilizando uma imagem com faixa dinâmica de 76 dB, apresentando resultados com detalhes tanto na parte clara como na parte escura da imagem.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PIXELS FOR FOCAL-PLANE SCALE SPACE GENERATION AND FOR HIGH DYNAMIC RANGE IMAGING

Fernanda Duarte Vilela Reis de Oliveira

March/2018

Advisors: José Gabriel Rodríguez Carneiro Gomes
Ángel Rodríguez-Vázquez

Department: Electrical Engineering

Focal-plane processing allows for parallel processing throughout the entire pixel matrix, which can help increasing the speed of vision systems. The fabrication of circuits inside the pixel matrix increases the pixel pitch and reduces the fill factor, which leads to reduced image quality. To take advantage of the focal-plane processing capabilities and minimize image quality reduction, we first consider the inclusion of only two extra transistors in the pixel, allowing for scale space generation at the focal plane. We assess the conditions in which the proposed circuitry is advantageous. We perform a time and energy analysis of this approach in comparison to a digital solution. Considering that a SAR ADC per column is used and the clock frequency is equal to 5.6 MHz, the proposed analysis show that the focal-plane approach is 26 times faster if the digital solution uses 10 processing elements, and 49 times more energy-efficient. Another way of taking advantage of the focal-plane signal processing capability is by using focal-plane processing for increasing image quality itself, such as in the case of high dynamic range imaging pixels. This work also presents the design and study of a pixel that captures high dynamic range images by sensing the matrix average luminance, and then adjusting the integration time of each pixel according to the global average and to the local value of the pixel. This pixel was implemented considering small structural variations, such as different photodiode sizes for global average luminance measurement. Schematic and post-layout simulations were performed with the implemented pixel using an input image of 76 dB, presenting results with details in both dark and bright image areas.

Contents

List of Figures

List of Tables

List of Symbols

C	Floating diffusion node capacitance, p. 49
C_{BL}	Bit line bus capacitance, p. 35
$C_{FD_{Total}}$	Overall capacitance seen by the floating diffusion node, p. 57
C_{FD}	6T pixel floating diffusion node capacitance, p. 34
C_{Rst}	6T pixel reset transistor gate capacitance, p. 34
C_{TX}	6T pixel transfer gate transistor gate capacitance, p. 34
C_{WL}	Word line bus capacitance, p. 35
$C_{W_{bit}}$	W_{bit} input capacitance, p. 36
$C_{\overline{\text{Write}}}$	$\overline{\text{Write}}$ node capacitance, p. 36
C_{ctrl}	Floating diffusion control node capacitance, p. 49
C_n	Node capacitance, p. 33
C_{ph}	Photodiode capacitance, p. 57
C_p	Parasitic capacitance in the floating diffusion node, p. 57
E_{CR}	Energy consumption due to charge redistribution, p. 34
E_{DC}	DC energy consumption, p. 32
E_{Digial}	Digital approach total energy consumption, p. 38
$E_{Dynamic}$	Dynamic energy consumption, p. 32
E_{FP}	Focal plane approach total energy consumption, p. 38
$E_{MACdynamic}$	MAC circuit dynamic energy consumption, p. 35
$E_{MACshortcircuit}$	MAC short-circuit energy consumption, p. 35

$E_{MACstatix}$	MAC circuit static energy consumption, p. 35
$E_{Shortcircuit}$	Short-circuit energy consumption, p. 32
E_{Static}	Static energy consumption, p. 32
$E_{capture}$	Energy for capturing the pixel matrix, p. 34
$E_{matrixStatic}$	Pixel matrix static energy consumption, p. 34
$E_{pixCapture}$	Energy for capturing a single pixel, p. 34
$E_{pixelReadDynamic}$	Matrix readout energy consumption, p. 34
$E_{readDyn}$	Memory read dynamic energy consumption, p. 35
$E_{readDyn}$	Memory read static energy consumption, p. 35
$E_{readStatic}$	Memory read static energy consumption, p. 35
$E_{readTotal}$	Total readout energy consumption, p. 35
$E_{writeDyn}$	Memory DC energy consumption during write operation, p. 36
$E_{writeDyn}$	Memory write dynamic energy consumption, p. 36
$E_{writeShortcircuit}$	Memory write short-circuit energy consumption, p. 36
$E_{writeStatic}$	Memory write static energy consumption, p. 36
I_{leak}	Estimated leakage current for both NMOS and PMOS considering minimum area, p. 32
K_{ADC}	Constant that represents the ratio between an ADC clock period and the circuit overall clock period, p. 25
$K_{Pipeline}$	Ratio between the pipeline ADC clock period and the circuit overall clock period, p. 27
K_{Ramp}	Ratio between the SAR (or cyclic) ADC clock period and the circuit overall clock period, p. 27
K_{Ramp}	Ratio between the ramp ADC clock period and the circuit overall clock period, p. 25
$K_{\Sigma\Delta}$	Ratio between the $\Sigma\Delta$ ADC clock period and the circuit overall clock period, p. 27

K_{ph}	Sensitivity factor used to generate photocurrent values for the HDR pixel simulations based on 12-bit pixel values **P**, p. 51
$N_{ADC_{Max}}$	Maximum number of ADCs, p. 30
N_{ADC}	Number of analog-to-digital converters, p. 23
N_{CR}	Overall number of charge redistribution operations, p. 22
N_{Lev}	Number of image pyramid levels, p. 22
N_{Off}	Number of MAC transistors that contribute with static consumption, p. 35
N_{PE}	Number of processing elements, p. 24
N_{bits}	Number of bits, p. 25
N_{busMem}	Number of possible parallel memory accesses, p. 23
N_{conv}	Number of converted data samples, p. 23
N_{op}	Number of kernel window operations required for Gaussian pyramid generation, p. 24
$Reset$	Reset transistors control signal, p. 46
$V_{dd,Matrix}$	Pixel matrix voltage supply, p. 33
V_{dd}	Voltage supply outside the pixel matrix, p. 33
WB	White balance constant, p. 64
α	Switching activity factor, p. 33
$\mathbf{K_{2\times2}}$	2×2 Binomial kernel., p. 11
$\mathbf{K_{3\times3}}$	3×3 Binomial kernel., p. 12
$\mathbf{K_{5\times5}}$	5×5 Gaussian kernel., p. 8
$\mathbf{P}$	12-bit pixel matrix used for the HDR pixel simulations, p. 51
DR	Dynamic range, p. 39
$\mathrm{I}_{ph_{ctrl}}$	Control photodiode photocurrents, p. 49
I_{ph}	Capture photodiode photocurrent, p. 48
T_G	Maximum integration time control signal, p. 46

T_s	Signal that controls the switches which connects control floating diffusion nodes in the HDR circuit, p. 46
T_{Max}	Maximum integration time, p. 48
T_{int}	Pixel integration time, p. 54
T_{mid}	Time it takes for a pixel floating diffusion voltage to reach the middle of the range, p. 49
T_{sMax}	Maximum value T_s can assume, p. 50
$V_{FD_{capture}}$	Floating diffusion capture node voltage, p. 48
$V_{FD_{control}}$	Floating diffusion control node voltage, p. 49
V_{HDR}	HDR pixel output voltage, p. 52
V_{min}	Minimum detectable floating diffusion voltage, p. 52
V_{rst}	Floating diffusion voltage level after reset, p. 48
m_C	Ratio between the capture and the control floating diffusion nodes capacitances, p. 49
m_{ph}	Ratio between the capture and the control photodiodes photocurrents, p. 49
σ	Gaussian filter standard deviation, p. 17
$\tau_{ADC_{Total}}$	Total analog-to-digital conversion time, p. 23
τ_{ADC}	Time required for one ADC to perform one sample conversion, p. 23
τ_{CR}	Time required for performing one charge redistribution, p. 22
$\tau_{Clk_{Pipeline}}$	Pipeline ADC clock period, p. 27
$\tau_{Clk_{Ramp}}$	Ramp ADC clock period, p. 25
$\tau_{Clk_{SAR,Cyclic}}$	SAR and cyclic ADC clock period, p. 27
$\tau_{Clk_{\Sigma\Delta}}$	$\Sigma\Delta$ ADC clock period, p. 27
τ_{Clk}	Clock period, p. 22
$\tau_{FP_{Proc}}$	Overall focal plane processing time, p. 22

τ_{Mem}	Time required for accessing a single memory position, p. 23
τ_{comp}	Time for a comparison operation, p. 67
$\tau_{convolutionWindow}$	Time required for one MAC unit to compute the kernel window operations, p. 24
$\tau_{convolution}$	Overall time required for performing the convolution operations, p. 24
τ_{div}	Time for a division operation, p. 67
$\tau_{memRead}$	Time to read the necessary pixels for the kernel window operation during convolution, p. 24
$\tau_{memWrite}$	Time required to write the result of the kernel window operation in the memory, p. 24
τ_{mult}	Time for a multiplication operation, p. 67
τ_{op}	Time for a single multiply and accumulate operation, p. 24
τ_{sum}	Time for a summing operation, p. 67
f_s	ADC reported conversion rate, p. 27
f'_s	ADC normalized conversion rate considering 8 bits, p. 27
$f_l(i,j)$	Pixel value in position (i,j) for image pyramid level l., p. 6
i_{max}	Maximum current that the imager is capable of reading before saturation, p. 39
i_{min}	Minimum current perceived by the imager, p. 39
$k\sigma$	Standard deviation for the Gaussian filter at the nearby scale in the derivative approximation function, where k represents the distance between the scales, p. 17
$n_k \times n_k$	Final kernel size used for the image pyramid generation considering 2×2 binomial kernel associations, p. 22
p	12-bit pixel value used for the HDR pixel simulations, p. 53
$p''_{i,j}$	sub-sampled image pixel positions after the third charge redistribution in the 8×8 6T operation example, p. 12

$p'_{i,j}$	sub-sampled image pixel positions after the second charge redistribution in the 8×8 6T operation example, p. 11
p_{HDR}	HDR pixel output value converted to a range of $[0, 1]$, p. 52
$p_{i,j}$	sub-sampled image pixel positions after the first charge redistribution in the 8×8 6T operation example, p. 11
ph	Capture photodiode, p. 45
ph_{ctrl}	Control photodiode, p. 45
EN_C	Signal that controls the switch connecting the floating diffusion nodes of two neighboring pixels from the same row, that is, connects two columns., p. 15
EN_R	Signal that controls the switch connecting the floating diffusion nodes of two neighboring pixels from the same column, that is, connects two rows., p. 15
FD_{ctrl}	Control floating diffusion node, p. 46
M×N	Image size, p. 9
N_{intMax}	Number of clock periods for the maximum integration time, p. 66
SEL	Row select control signal, p. 16
V_m	Middle voltage of the control inverter operating range, p. 46
V_t	Threshold voltae, p. 76
m	Ratio between the capture photodiode and the control photodiode areas, p. 74
$\tau_{matrixMemWrite}$	Time necessary to write M×N pixel values in the memory, p. 23

List of Abbreviations

ADC	analog-to-digital converter, p. 2
APS	active pixel sensor, p. 3
CCD	charge coupled device, p. 1
CDS	correlated double sampling, p. 41
CIS	CMOS image sensors, p. 1
CMOS	complementary metal-oxide-semiconductor, p. 1
DDR	double data rate, p. 30
DoG	difference of Gaussian, p. 16
ENOB	effective number of bits, p. 26
FF	fill-factor, p. 73
FPN	fixed-pattern noise, p. 41
HDR	high dynamic range, p. 3
IoT	Internet of Things, p. 1
LoG	Laplacian of Gaussian, p. 16
MAC	multiply and accumulate, p. 20
MRMS	multi ramp multi slope, p. 24
MSSIM	mean structural similarity, p. 61
OSR	oversampling rate, p. 26
PE	processing element, p. 8
PSNR	peak signal-to-noise ratio, p. 61

QoE	quality of experience, p. 3
RGB	red, green, blue color model, p. 62
SAR	successive approximation register, p. 24
SIFT	scale-invariant feature transform, p. 3
SNR	signal-to-noise ratio, p. 1
SS	single slope, p. 24
VGA	video graphics array, p. 28

Chapter 1

Introduction

In the last decades, the use of CMOS (complementary metal-oxide-semiconductor) technology for image sensors has matured up to the point that it has overcome the charge coupled device (CCD), which was the most common imaging technology until the beginning of the twenty first century, for many applications [3]. The necessary investment to improve the image quality of the CMOS image sensors (CIS) came with the increase of the mobile devices industry market [3],[4]. The introduction of the pinned photodiode was an important milestone for the CMOS image sensors because it allowed for a significant increase in the signal-to-noise ratio (SNR). Furthermore, backside illumination, which increases the quantum efficiency, microlenses and color filters are possible features in most CIS technologies nowadays [4].

An important difference between CMOS and CCD technologies is that the latter allows for the design of processing circuits in the same chip of the pixel matrix, which enables the creation of entire systems inside chips [5]. Moreover, CIS, or imagers, are perfectly suitable in the context of smart camera design [6], where the output of the chip, or camera system, is not necessarily the image, but the processed image or even an action that has to be taken considering the information extracted from the image.

The Internet of Things (IoT) is an example of field which could highly benefit from CMOS imagers [7], [8]. Because of the amount of data that need to be stored, transmitted, processed and exchanged, vision systems represent a big challenge in the IoT paradigm [9]. By enabling processing inside the chip, CMOS image sensors can reduce the amount of data that need to be transmitted, which consequently enhances the throughput of the system.

For many applications, throughput enhancement and power consumption reduction are constant requirements [10]. Real-time applications, such as driver assistance and surveillance, require immediate response. The acceleration of vision algorithms in these cases is very important. The introduction of processing circuits

inside the pixel, namely focal-plane processing, helps for increasing throughput and energy efficiency of vision systems. Fabricating dedicated processing hardware in the pixel matrix opens many possibilities because it allows for concurrent image capture and processing. Many publications have explored focal-plane image processing advantages [11]-[19].

A drawback of this approach is image quality reduction, because of pixel pitch increase and fill-factor (ratio between photodiode area and pixel area) reduction. With the goal of taking the best of focal-plane processing but with the constraint of minimally affecting the image quality, we propose a pixel with only two extra transistors, thus requiring small area increase when compared to regular 4T pixel. The proposed 6T pixel computes the captured image Gaussian pyramid at the focal plane.

Gaussian filtering is a pre-processing technique used as first step for many image processing algorithms. Furthermore, by repeating the Gaussian filtering operation we can generate multi-scale representations of the image, such as the Gaussian pyramid. A simple way of performing Gaussian filtering is explained in [20], where neighboring pixels are connected into groups and charge redistribution operations are used to perform averaging among pixels values. Filtering is performed for the image with half the number of rows and half the number of columns of the original pixel matrix. We propose the implementation of this approach with only two extra transistors per pixel. These transistors act as switches and connect the nodes that store pixel values. By performing the filtering operation several times, it is possible to create a reliable scale space with the proposed hardware [21]. Details on how to perform the Gaussian filtering and creating a Gaussian pyramid will be explained in Chapter 2.

Main potential advantages of the 6T pixel proposed in Chapter 2 are conferred by parallel processing. To guarantee the true advantages of focal-plane image processing techniques, it is important to quantify the speed improvements yielded by those techniques. Actual advantages of the focal-plane approach are not granted by default. Exploring the conditions under which these advantages really occur, and benchmarking them, is an important contribution of this thesis. In other words, we perform comparative throughput and energy analyses of a non-conventional architecture based on a 6T pixel with a conventional hypothetical digital architecture, in which no pre-processing at all is performed in the sensor. We consider different kinds and numbers of analog-to-digital converters (ADCs) embedded in the sensor readout channel, and different numbers of processors in the digital approach. With these analyses we close the first part of the thesis by outlining the conditions in which the proposed focal-plane processing is advantageous.

Besides pre-processing, transistors can also be designed at the focal plane for

image quality improvement. The active pixel sensor (APS) source follower transistor, for example, was introduced in the pixel to reduce readout noise [22]. When compared to the passive pixel sensor, APS has improved image quality because of its higher SNR [23]. Capturing images of high dynamic range (HDR) is another quality improvement that can be achieved by introducing extra circuitry at the focal plane. The second part of this thesis will present the study of a pixel for HDR capture with autonomous control over the integration period.

The human eye is capable of instantaneously handling image signals with a dynamic range of 5 orders of magnitude and over 8 orders of magnitude with adaptation time. Conventional 8-bit linear sensors, on the other hand, capture only 3 orders of magnitude [24], [25]. Consequently, when taking a picture, the user may notice that details from the original scene were lost. Capturing the image and representing it as faithfully as possible to the real scene improves user visual *quality of experience* (QoE) [24]. Furthermore, applications in which there is no illumination control, such as surveillance cameras, also require HDR capture. HDR imaging and tone mapping have been largely addressed in recent years [3], [26]-[32].

Our focus will be on the implementation of a pixel matrix in which the integration time of each pixel depends on the matrix mean illumination and on the pixels local illumination [33]. This pixel will be studied and designed in a 180 nm CIS technology. Pixel design structural modifications will be proposed and simulated.

1.1 Objectives

This thesis is divided into two parts: a pixel for focal-plane Gaussian pyramid generation; and a HDR and tone-mapping pixel with control over the integration period.

For the first part we will present a pixel with only two extra transistors that allows for Gaussian filtering at the focal plane. By repeating the Gaussian filtering operation, multi-scale representations of the image are generated. We demonstrate how the proposed circuit can be used to generate a Gaussian pyramid. To validate the proposed circuit, we generate scale-space data for several images, using a system-level simulation of the same steps that generate the Gaussian pyramid in hardware. We then use the scale-invariant feature transform (SIFT) algorithm [34] to assess scale-space data quality. It is also important, though, to evaluate the advantages of performing such processing at the focal plane. Because of its high level of parallelism it is expected that the focal-plane method, when compared to a conventional digital approach, presents much higher values of throughput. We present a method to perform the evaluation of the focal-plane approach considering speed enhancement and energy savings. Using the proposed method, we show that these

advantages are not granted by default and are only realized through proper architectural design. The methodology presented for the comparison between focal-plane and digital approaches is a useful tool for imager design, allowing for the assessment of focal-plane processing advantages. Although the analysis results showed the focal plane advantage, it was expected that the focal plane approach would be even faster than the results indicate. For this reason we decided not to fabricate the circuit for Gaussian pyramid generation, which leads to the second part of the work.

The second part of the thesis focuses on HDR image capture and tone mapping. The circuit used in this part of the work and the one used in the first part of the work share the idea of connecting the floating diffusion nodes of neighboring pixels and performing charge redistribution. In both cases, n-channel transistors are used as switches to allow for the connection of the floating diffusion nodes. In the case of the second part of the work, the goal is to measure the average matrix illuminance and use this value to control the pixel integration time. The tone mapping response curve is adjusted according to this average value. The pixel presented in [33] is studied and designed in a 180 nm CIS technology. This pixel has two photodiodes, one for capture and one for the control of its own integration period. Thirteen 64×64 test matrices, which include small structural differences from the pixel from [33], were designed. Among these differences, pinned photodiodes are used with the goal of improving the noise performance of the pixel. It is expected that the noise will be improved based on pinned photodiode references [35]. To experimentally evaluate the pinned photodiode advantage, matrices with the standard photodidode were included in the chip layout. Future experimental tests will be performed to compare the noise figure of the matrix with the pinned photodiode to a matrix with the one of the standard photodiode. Different sizes of control photodiode were considered as well as a shared photodiode. Color filters were included in two test matrices. System-level simulations show the expected pixel response in each cases. We have little information about the pixel response and characteristics, so this thesis contemplates many possible scenarios for the pixel response. The results illustrate the advantages of the studied HDR pixel.

1.2 Thesis Organization

The first part of this thesis is composed by Chapters 2 and 3. Chapter 2 presents the 6T pixel and explains it operation. The steps necessary for Gaussian pyramid generation and system-level simulation results are also presented at Chapter 2. Chapter 3 compares the proposed focal-plane approach with a hypothetical and equivalent digital approach with the goal of assessing the focal plane advantages with respect to signal processing speed and energy consumption.

The second part of this thesis is composed by Chapters 4 and 5. The pixel description, analysis and system-level simulations are in Chapter 4. The chip design, organization and layout can also be found in this chapter. Chapter 5 shows schematic and post-layout simulation results. Chapter 6 presents the conclusions of this thesis and some topics for future research.

Chapter 2

A 6T Pixel for Focal-Plane Scale Space Generation and Gaussian Pyramid Implementation

Object recognition is a problem constantly faced in computer vision systems [36]. Many applications require object detection and recognition, such as optical character recognition, face detection and recognition, segmentation, location recognition, image search and recovery, and intelligent photo editing [36]. A recurrent challenge in the field of object recognition refers to scale invariance, which is the capacity of identifying and describing objects across multiple scales. Object recognition across different scales has been discussed in several references, starting with classic algorithms such as the SIFT [34], for object detection, or the Viola-Jones [37], for face detection, and going up to recent algorithms based on deep learning [38].

The problem arises because objects can be captured at different distances from the image sensor and will thus be represented with different sizes in the image. The concept of pyramid representation [39] thus arises as a multi-resolution scene representation composed by a sequence of filtered and subsampled copies of the input image. Each subsampled copy is a pyramid level. Formally, filtering followed by subsampling is the definition of the *reduce* operation, which can be described by:

$$f_l(i,j) = \sum_{m}^{M} \sum_{n}^{N} K(m,n) \cdot f_{l-1}(2 \cdot i + m, 2 \cdot j + n), \tag{2.1}$$

where K is the filtering kernel, and f_{l-1} is an input image with size M×N [39]. The pyramid is formed by levels, starting with the original image as Level 0. To generate pyramid Level 1, the original image is filtered with the kernel K, and the resulting image is subsampled. The process is repeated until a pre-defined resolution

is reached [40]. An example of pyramid is shown in Figure 2.1(a). The images are depicted subsampling with no subsampling in Figure 2.1(b) for better visualization of the applied filtering. The Gaussian pyramid [39], specifically, is formed by low-pass representations of the image. The generation of the pyramid using Gaussian filters is important because the Gaussian filters are linear, invariant to spatial shifts, and because no extrema points are generated as we proceed from finer to coarser scales, thus avoiding the creation of spurious key-points in the process [41], [42]. These characteristics of the Gaussian filter also make the Gaussian the only possible kernel for the generation of the scale-space representation of the image [41], [42].

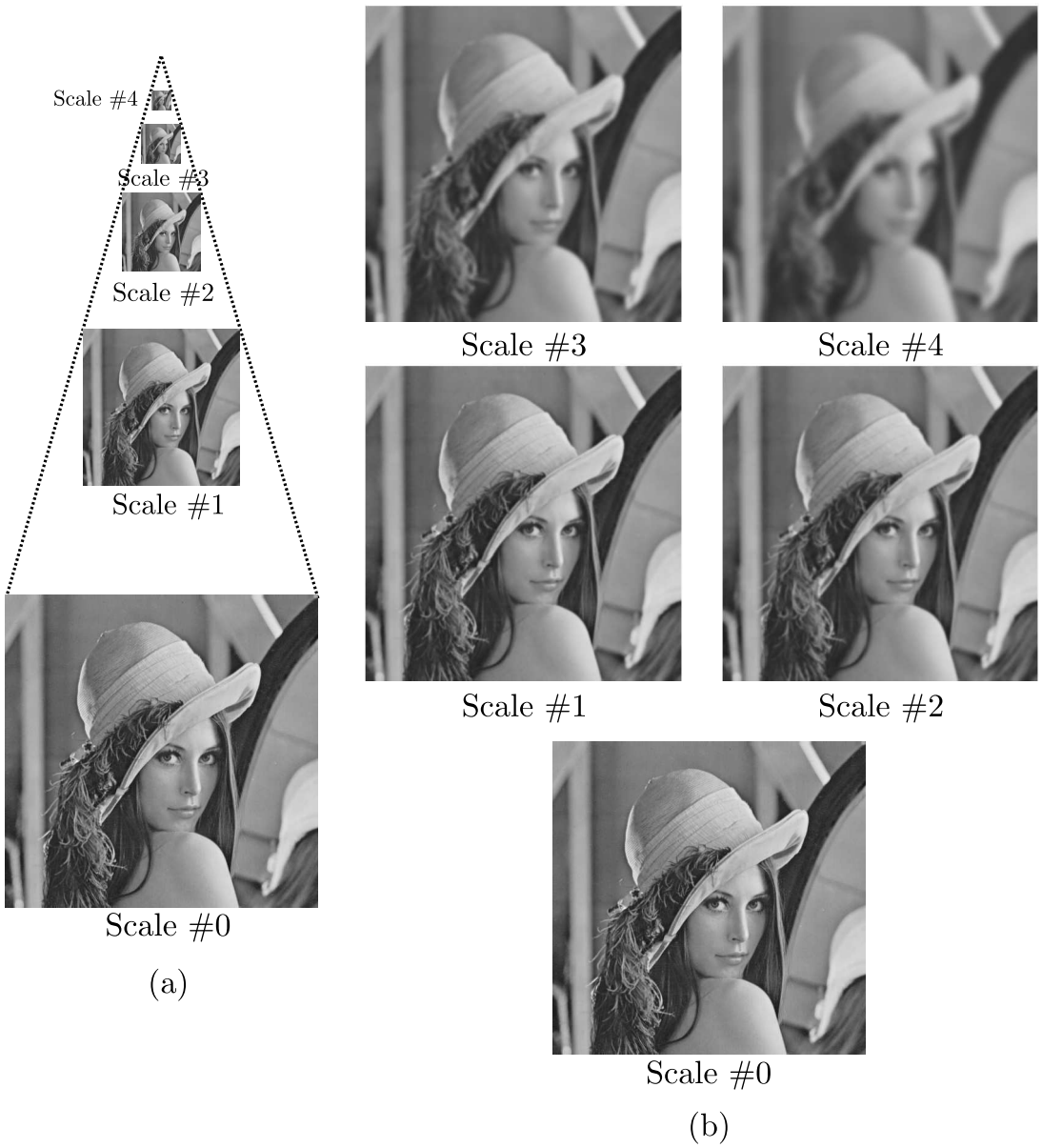


Figure 2.1: (a) example of pyramid representation, (b) pyramid in (a) with no subsampling in order to highlight the effect of the applied filtering.

For dedicated hardware approaches, the industrial standard OpenVX [43] defines the Gaussian pyramid generation based on the following 5×5 Gaussian kernel:

$$\mathbf{K_{5\times5}} = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \tag{2.2}$$

This kernel is also used in the OpenCV and Matlab Gaussian pyramid implementations. After filtering, OpenVX provides two possibilities for image subsampling, the half-scale pyramid, in which each image level has half the number of rows and half the number of columns of the anterior image level, and the "ORB scaled image" [43], in which each image level has $1/\sqrt[4]{2}$ rows and $1/\sqrt[4]{2}$ columns of the anterior image level. We will consider only the half-scale image. To increase processing speed, hardware-based Gaussian filtering implementations have been proposed in many papers for at least two decades [44]-[46].

A resistive mesh that performs the convolution between the input image and a variable width Gaussian filter inside the pixel matrix is presented in [44]. Considering neighboring pixels that are centered around a reference pixel, the voltage gains (from the reference pixel to the local pixel) are designed to decay in a Gaussian-like manner, which can be used for convolution. The convolution width depends on the ratio between two resistors of the network, which can be adjusted. The filtering is parallel. The disadvantage of this circuit is connection complexity. It impairs the circuit total size, resulting in large pixel pitch.

The image sensor presented in [12] not only provides a way of computing Gaussian filtering, but also a way for Gaussian pyramid generation. It is based on the fact that the Gaussian filtering effect is obtained from the solution of the heat-diffusion equation, which can be computed using resistive grids [11]. A switched-capacitor network is used to implement the resistive grid. The equivalent pixel value after an arbitrary number of clock cycles is proportional to the values of the neighboring pixels and to the node capacitance values. To change the pyramid level, the capacitors that store the pixel values are merged, thus reducing the resolution. Although the proposed circuit also requires significant pixel area, this architecture was conceived with 3-D stacking integration technologies in mind.

Digital solutions are also very common in the literature [46]-[48]. In [46], for example, an FPGA is used for Gaussian filtering implementation. That paper proposes an efficient way of performing floating-point multiplications, focusing at increasing speed while maintaining the precision.

Focal-plane sensing-processing [11], [12], [44], [45] allows for parallel image processing, as soon as capture ends, throughout the entire pixel matrix. Because of this high parallelization level, it has great potential for enhancing processing speed. The down side to this approach is that it typically suffers from large pixel pitch and low fill-factor, characteristics that reduce the image quality because of their impact on key parameters of image sensing like sensitivity, resolution and noise.

The proposed pixel, presented in Section 2.1, has the goal of performing processing while minimally affecting pixel pitch and fill factor. With the inclusion of only two additional transistors per pixel, we are able to perform Gaussian filtering and, by repeating the filtering operation, we are able to generate a Gaussian pyramid at the focal plane.

2.1 Proposed 6T Pixel Implementation

Figure 2.2 shows the block diagram of a conventional approach to generate a Gaussian pyramid compared to the block diagram of our proposed approach, using the pixel shown in Figure 2.3. In the conventional approach, Figure 2.2(a), the image sensed by an M×N pixel array is converted into digital and stored in memory. A prescribed number of processing elements (PEs) then access memory in order to process the image just captured and generate the corresponding pyramid. PEs operate in parallel.

In the proposed circuit implementation, shown in Figure 2.3, two n-channel transistors are used as switches, thus resulting in a pixel with six transistors [21]. Pixel operation starts as a regular 4T pixel, by resetting the floating diffusion (FD, in the

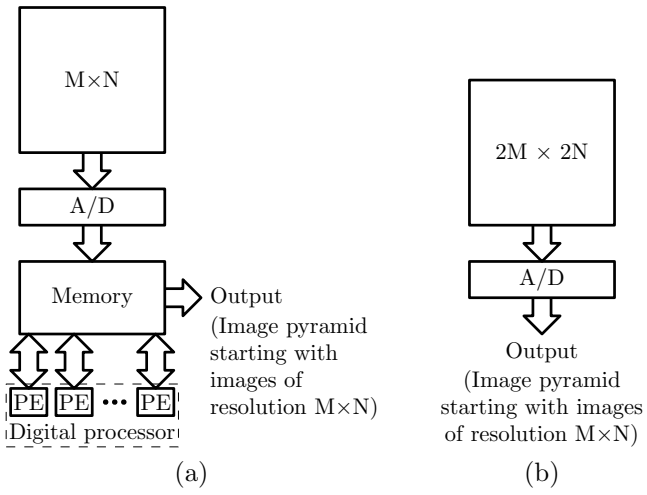


Figure 2.2: Two approaches for Gaussian pyramid hardware computation: (a) conventional digital approach, where PE stands for processing element. PEs operate in parallel; (b) focal-plane sensing-processing approach.

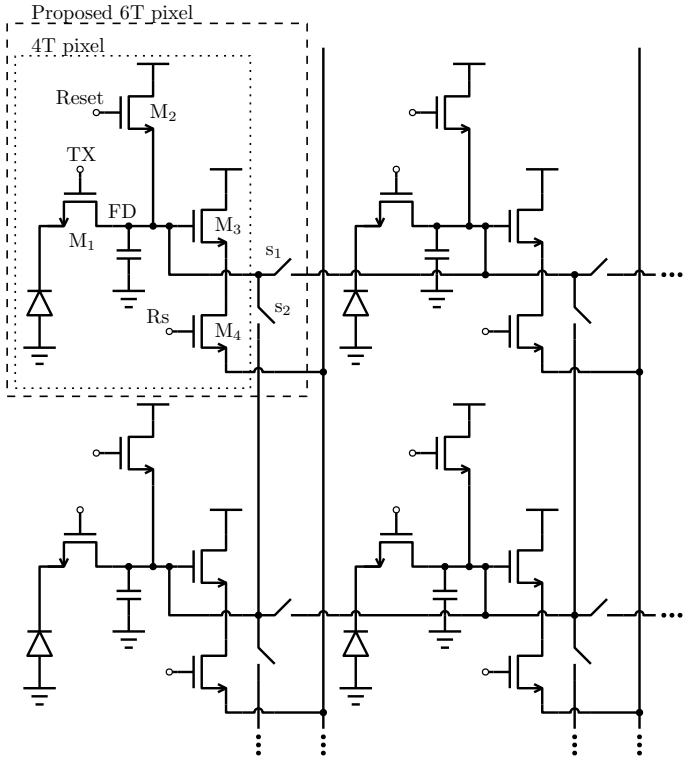


Figure 2.3: 2×2 section of a matrix with the proposed pixel for focal-plane Gaussian filtering computation. Two transistors acting as switches (s_1 and s_2) are included inside each pixel to connect the floating diffusion nodes of neighboring pixels.

figure) nodes. After the integration time, the charge accumulated at the photodiode cathode is transferred (according to TX) to the floating diffusion node. When the switches close, charge redistribution is performed among parasitic capacitors at the corresponding floating diffusion nodes. The average voltage after charge redistribution represents the mean luminance in the sub-matrix where the pixels were connected. This operation is lossy. Once pixels are interconnected in a sub-matrix, all parasitic capacitors end up holding the same voltage level.

To understand how filtering is performed with the proposed pixel, consider, as an example, the 8×8 matrix in Figure 2.4(a), where the pixel values encode an original image. The first step to generate the pyramid consists in connecting the pixels into 2×2 blocks, to perform an average operation inside each block, as shown in Figure 2.4(b). If we sample one pixel inside each block, the ones marked with a rectangle (or circle), then the resulting image has half the number of rows and half the number of columns of the original image. This is the image that will be filtered and whose Gaussian pyramid will be generated. That is why the block diagram of the proposed approach in Figure 2.2(b) starts with two times more rows and columns than in the digital approach from Figure 2.2(a). This first charge redistribution step is necessary to perform convolution in the proposed way, but

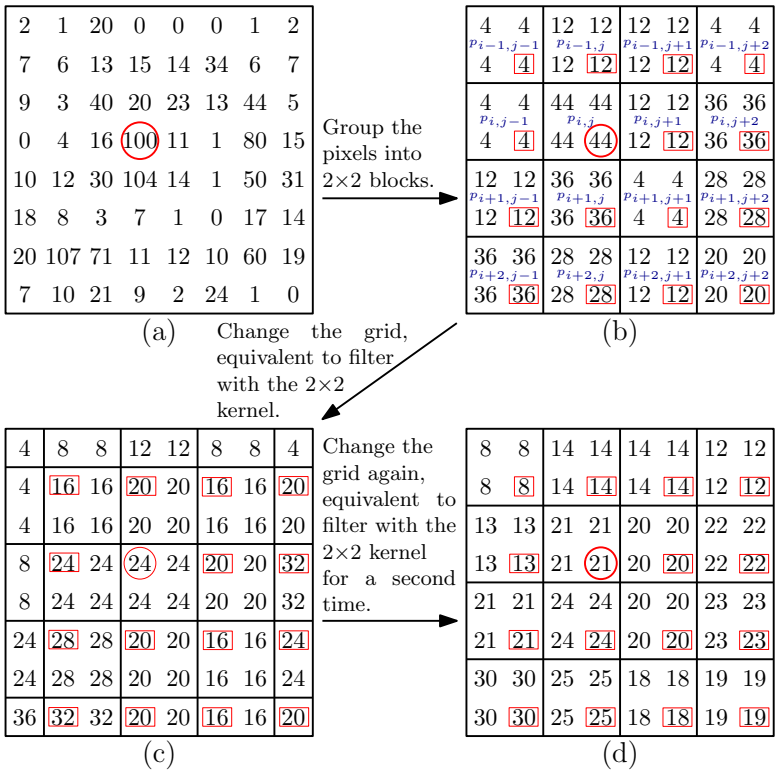


Figure 2.4: Gaussian filtering example.

it reduces the resolution of the image. Consider the pixel marked with a circle in Figure 2.4(a). After charge redistribution, this pixel value is the result of the averaging operation $p_{i,j} = \frac{40+20+16+100}{4} = 44$. Sub-sampled image pixel positions are written in $p_{i,j}$ format in the middle of each block in Figure 2.4(b). All subsequent steps perform Gaussian filtering on this sub-sampled matrix.

In the first subsequent step, we change the grid and, again, group the pixels into 2×2 blocks. This grid change and the result of the new charge redistribution step are shown in Figure 2.4(c). After the charge redistribution we have that $p'_{i-1,j-1} = (p_{i-1,j-1} + p_{i-1,j} + p_{i,j-1} + p_{i,j})/4$, $p'_{i-1,j} = (p_{i-1,j} + p_{i-1,j+1} + p_{i,j} + p_{i,j+1})/4$, $p'_{i,j-1} = (p_{i,j-1} + p_{i,j} + p_{i+1,j-1} + p_{i+1,j})/4$ and $p'_{i,j} = (p_{i,j} + p_{i,j+1} + p_{i+1,j} + p_{i+1,j+1})/4$, where p' represents the pixel values after the second charge redistribution. The result is equivalent to filtering the sub-sampled image from Figure 2.4(b) with the 2×2 binomial filter:

$$\mathbf{K_{2\times 2}} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{2.3}$$

Depending on how the pixels at borders of the image are treated, filtering results at image borders might be affected by zero-padding effects. For pixels that are

not at the pixel array borders, the computation results correspond exactly to the convolution with the previously mentioned binomial filter. The pixel marked with a circle in Figure 2.4(c) is the result of $p'_{i,j} = \frac{44+12+36+4}{4} = 24$.

If we change the grid again, back to the first grid, as shown in Figure 2.4(d), we perform the same filtering for a second time. Changing the grid back to the first one corresponds to filtering the sub-sampled image from Figure 2.4(c), which is formed by the pixels marked with a rectangle (or circle), with the 2×2 binomial, or equivalently, to filtering the sub-sampled image from Figure 2.4(b) twice with the 2×2 binomial, or once with the 3×3 kernel:

$$\mathbf{K_{3\times 3}} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{2.4}$$

The values from Figure 2.4(d) can be computed as:

$$\begin{aligned} p''_{i,j} =& \frac{1}{4}(p'_{i-1,j-1} + p'_{i-1,j} + p'_{i,j-1} + p'_{i,j}) = \\ =& \frac{1}{16}(1p_{i-1,j-1} + 2p_{i-1,j} + 1p_{i-1,j+1} + \\ & 2p_{i,j-1} + 4p_{i,j} + 2p_{i,j+1} + \\ & 1p_{i+1,j-1} + 2p_{i+1,j} + 1p_{i+1,j+1}), \end{aligned} \tag{2.5}$$

where $p''_{i,j}$ is the pixel from position (i, j) after subsampling the image in Figure 2.4(d). Comparing the multiplying factors from Equation (2.5) with the kernel shown in Equation (2.4), we see that the charge redistribution method is equivalent to filtering the subsampled image from Figure 2.4(b) with the kernel from Equation (2.5).

Using the same pixel position as before, the pixel within the red circle is computed as $p''_{i,j} = \frac{16+20+24+24}{4} = 21$ if we use the pixels from Figure 2.4(c), or as:

$$\begin{aligned} p''_{i,j} =& \frac{1}{16}(1 \cdot 4 + 2 \cdot 12 + 1 \cdot 12 + \\ & 2 \cdot 4 + 4 \cdot 44 + 2 \cdot 12 + \\ & 1 \cdot 12 + 2 \cdot 36 + 1 \cdot 4) = 21, \end{aligned} \tag{2.6}$$

if the pixels from Figure 2.4(b) are used.

According to the example in Figure 2.4 we conclude that, for every grid change, the sub-sampled image is filtered with the 2×2 binomial kernel presented in Equation (2.3). The size of the targeted kernel determines the number of times that the

grid must be shifted and charge redistribution enabled. The possible kernels that can be implemented with the proposed hardware are 2×2 binomial kernel cascade associations.

The OpenVX Gaussian pyramid kernel (Equation (2.2)) is implemented with the proposed hardware by performing four charge redistribution operations after the resolution change. Figure 2.5 presents the steps required for the generation of a three-level pyramid considering the kernel from Equation (2.2). Step (2) from Figure 2.5 is required for changing the image resolution. To generate Level 0, which is the Gaussian pyramid starting level, we sample one pixel inside each 2×2 block of the image generated after this charge redistribution. This image is then filtered through steps (4) to (7), resulting in the image that is subsampled to generate Level 1. To compute Level 2, we connect the pixels into 4×4 blocks, with the same goal of step (2), thus reducing the resolution. As in the calculation of Level 1, four charge redistribution operations are performed to filter the image, which is done in steps (10) to (13). By the end of these operations the result is subsampled, generating Level 2. To create a pyramid with four levels, the pixels are connected into 8×8 pixel blocks. The maximum number of levels that can be generated by the proposed hardware mainly depends on the leakage current of the fabrication technology and on the floating diffusion node capacitance.

2.1.1 Schematic Simulations with the 6T Pixel

The proposed pixel operation relies on charge redistribution between parasitic capacitance to perform the filtering operation. When opening and closing the switches to filter the captured image, charge injection and clock feedthrough can represent a problem because it affects the final voltage result in the averaging operation. Schematic simulations were performed using a 110 nm[1] CIS technology to understand how the average changes considering the circuit non-idealities.

The designed pixel has a photodiode of 3 μm × 3 μm and a total dimension of 6.28 μm × 6.28 μm, resulting in a fill-factor of 22.8%. Transistor M_4 (the select switch) and switches s_1 and s_2, both implemented with an n-channel transistor, from Figure 2.3, have minimum size (according to the technology used, for 3.3 V n-channel transistors): width equal to 180 nm and length equal to 340 nm. The reset transistor, M_2 from Figure 2.3, has 1 μm × 340 nm. To reduce the bottleneck effect during the charge transfer, the transfer gate transistor, M_1 from Figure 2.3, also has 1 μm width. The transfer gate length is equal to 450 nm, which is the minimum length defined by the technology for transfer gate transistors. The source

[1]The 6T pixel was designed and simulated using the 110 nm CIS technology, but for the HDR chip, presented in Chapters 4 and 5, an 180 nm CIS techology was used for the design, simulation and fabrication.

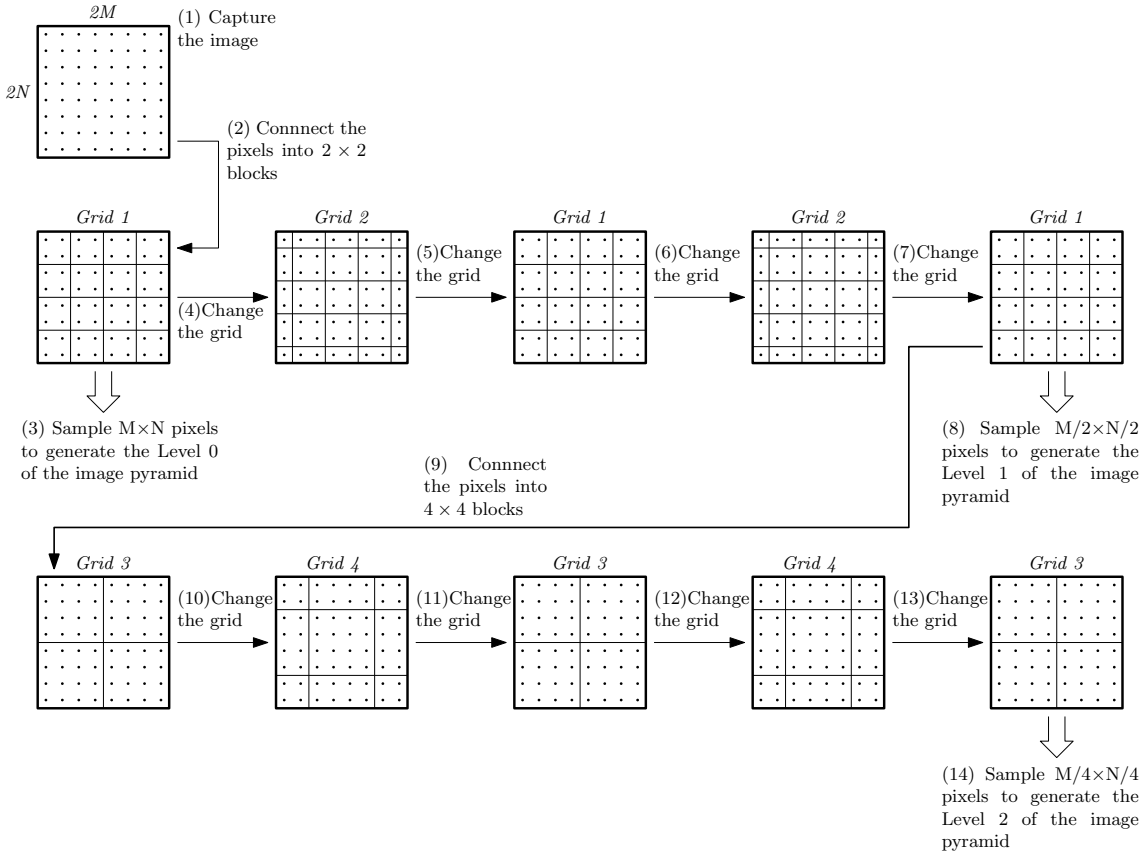


Figure 2.5: Example of Gaussian pyramid generation at the focal plane.

follower, M_3 from Figure 2.3, has 4 μm width and 340 nm length, being thus the largest transistor. These source follower dimensions have the goal of increasing the floating diffusion capacitance in order to reduce the vulnerability to charge injection and clock feedthrough effects. The layout can be seen in Figure 2.6.

Schematic and post-layout simulations were performed using a row of four pixels. Charge is performed between the two neighboring pixels from the middle. The pixels from the borders were added to guarantee that the floating diffusions had the same number of transistors connected and, consequently, have the same equivalent capacitance. These simulations were repeated for a column of four pixels.

For these simulations, the model presented in Figure 2.7 was used for the pinned photodiode and transfer gate. Consequently, there is an ideal charge transfer between the photodiode and the floating diffusion. Using this model, all the charge stored in the capacitor is transferred to the floating diffusion when the transfer gate switch is closed. A buffer is used to assure that the capacitor will have the same voltage at both terminals when the switches close, and that all the generated current will go to the floating diffusion.

Figures 2.8 and 2.9 show schematic and post-layout simulation results. In the top of these figures, the pixels belong to the same column and in the bottom to the

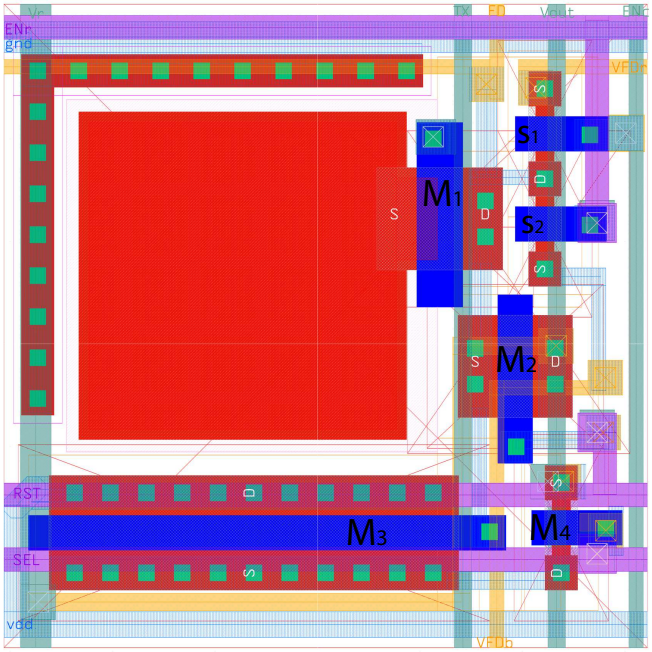


Figure 2.6: Proposed pixel for Gaussian pyramid generation layout.

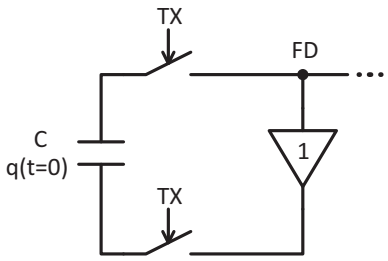


Figure 2.7: Pinned photodiode and transfer gate ideal model used for simulations.

same row. The dotted lines in these figures show the pixel control signals. The RST is responsible for resetting the floating diffusion voltage and the TX transfers the photodiode accumulated charge to the floating diffusion, thus reducing its voltage. The next signal is the charge redistribution enable. In Figures 2.8 and 2.9 top, two rows are connected together when EN_R is activated. EN_C connects two columns together, which is shown in Figures 2.8 and 2.9 bottom. After the charge redistribution is performed, the floating diffusion voltage is sampled with the row select signal, SEL. In the case of the pixels connected in the same column, two select signals are necessary to read each floating diffusion. For the pixels connected in the same row, only one select is necessary.

The floating diffusion voltages are represented with solid and dashed lines. There is no voltage error when TX is turned on or off because an ideal switch was considered for the transfer gate. When the EN signal is activated the floating diffusion voltages

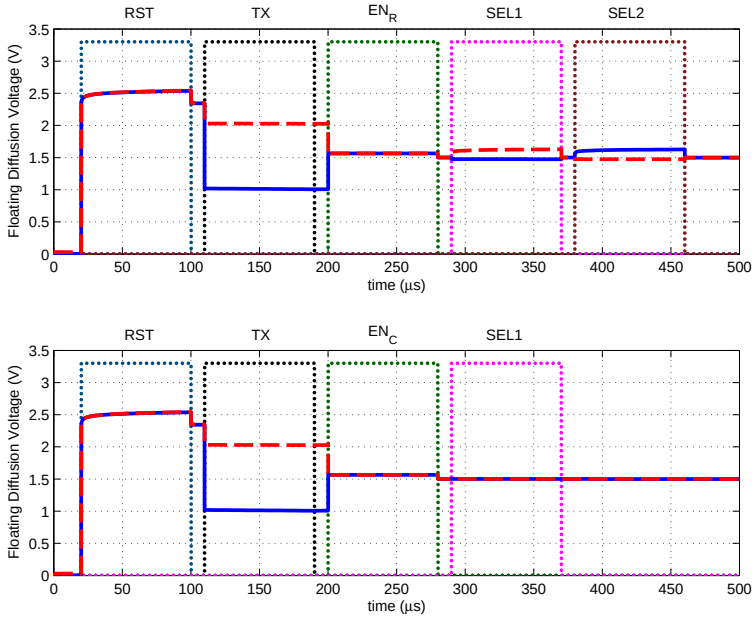


Figure 2.8: Schematic simulation results for two pixels from the same column (top) and from the same row (bottom). The floating diffusion voltages, which represent the pixel values, are shown in solid lines, for one pixel, and dashed lines, for the second pixel. The control signals are shown in dotted lines: RST defines a initial voltage for the floating diffusion nodes, TX transfers the photodiodes charges to the floating diffusion nodes, EN_R connects the pixels from the same column, performing charge redistribution between these pixels, EN_C connects the pixels from the same row, performing charge redistribution between these pixels, SEL1 and SEL2 are the row select signals for row different rows.

converge to the same value, ideally equal to 1.5 V. The maximum voltage error when SEL is activated is of 2% in the schematic simulation and of 5% in the layout simulation.

2.2 Case Study: SIFT

The proposed hardware can also be used to generate the scale space representation of the image, which can be applied to the SIFT algorithm. In order to evaluate the method implemented by hardware, we have generated the scale space of a dataset of images and used it with the OpenCV [49] implementation of the SIFT.

To generate the multi-scale representation used in the SIFT, the image is first filtered n times with Gaussian kernels of carefully chosen standard deviation, creating the first octave. The image from the middle of the octave is then copied and subsampled. The resulting image is filtered with the same kernels of the first octave, thus generating the second octave. The procedure is repeated until the desired number of octaves is created [34]. To increase the number of interest points the first octave

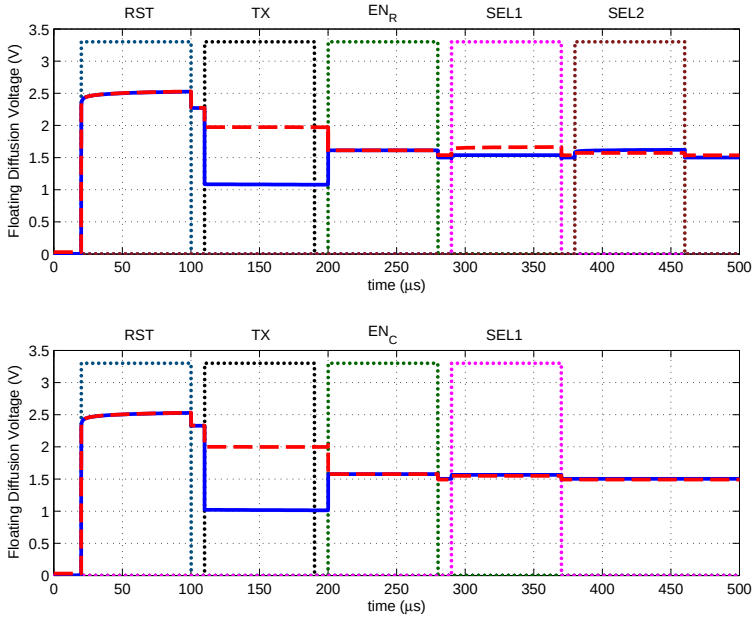


Figure 2.9: Extracted layout simulation results for two pixels from the same column (top) and from the same row (bottom). The floating diffusion voltages, which represent the pixel values, are shown in solid lines, for one pixel, and dashed lines, for the second pixel. The control signals are shown in dotted lines: RST defines a initial voltage for the floating diffusion nodes, TX transfers the photodiodes charges to the floating diffusion nodes, EN_R connects the pixels from the same column, performing charge redistribution between these pixels, EN_C connects the pixels from the same row, performing charge redistribution between these pixels, SEL1 and SEL2 are the row select signals for row different rows.

can be created with an expanded version of the original image [34]. A difference of Gaussian (DoG) is performed afterwards in order to create a scale-normalized Laplacian of Gaussian ($\sigma^2\nabla^2 G$) representation of the image. The points of interest are then searched throughout the scales of the Laplacian scale-space pyramid representation.

To show that the DoG can be used to generate the scale-normalized Laplacian of Gaussian (LoG), we note that approximation of the LoG by the DoG is given by $\sigma\nabla^2 G = \partial G/\partial\sigma \approx [G(x,y,k\sigma) - G(x,y,\sigma)]/(k\sigma - \sigma)$, where σ is the Gaussian filter standard deviation. The k constant is required for derivative approximation based on the finite difference method applied to nearby scales at $k\sigma$ and σ [34]. We can see that $G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G$, which shows that the DoG is a good approximation of the normalized Laplacian of Gaussian multiplied by a factor $(k-1)$. Each kernel must have a standard deviation equal to $k\sigma$ of the previous kernel, where k is a constant.

With the proposed hardware it is possible to generate a scale space that can be used by the SIFT without a significant performance drop [21]. First, we capture the

image and group the pixels into 2×2 pixel blocks. After sampling and quantization, the result is the first image from the first octave of the scale space. We then change the grid, which is equivalent to filtering the image with the 2×2 binomial filter, and obtain the second scale space image. This kernel is a good approximation of the Gaussian kernel with standard deviation $\sigma_{filter} = \sigma_1 = 0.5$. By changing the grid again, we perform a second filtering operation with the 2×2 binomial filter, which results in the third image from the scale space. The resulting standard deviation is thus equal to $\sigma_2 = \sqrt{\sigma_1^2 + \sigma_{filter}^2} = 0.707$, which is the standard deviation of the filter presented in Equation (2.4). These two filtering operations define the previously mentioned k constant: $k = \sigma_2/\sigma_1 = \sqrt{2}$. Consequently, the next image must be the result of filtering with a kernel with standard deviation equal to $k\sigma_2 = 1$. This is possible by using the binomial kernel twice: $\sqrt{\sigma_2^2 + \sigma_{filter}^2 + \sigma_{filter}^2} = 1$, resulting in the fourth image from the scale space. If we want a fifth image in this octave, we must perform grid changes until the equivalent standard deviation is equal to $\sqrt{2}$ [21]. The next octave is computed after all the images from the previous octave are generated, by grouping the pixels into 4×4 blocks and repeating the procedure.

The main differences between the proposed approach for generating the scale space and the approach presented in [34] are the filters, the octave change (based on the previous octave last image, instead of the middle one) and the fact that the image is not previously expanded to generate the first octave. The kernel, number of octaves and number of scales per octave also differ from the proposed ones. Even with these differences, system-level simulations show that the results achieved with the proposed hardware implementation are similar (in the sense of similar *repeatability* with respect to the OpenCV SIFT implementation, where *repeatability* is defined next) to those obtained with the original approach. These simulations were run using the database from [50][2] and OpenCV SIFT libraries [51]. By computing original image keypoints and comparing them with transformed image keypoints, we evaluate whether the proposed keypoint method is robust to those transformations. This evaluation measure is denoted as *repeatability*.

Table 2.1 shows the repeatability results for the original and the proposed method. The original method parameters are: three octaves, six scales per octave, 0.04 for contrast threshold, and 10 for edge threshold. The first threshold is used to filter weak features, while the second one is used to filter edge-like features. For

[2]The image database contains eight different images, which have the names that appear in Table 2.1. To generate distorted images (for the sake of diversity at the SIFT algorithm input), five different changes are applied to the original images, blur, in images " "Bikes" and "Trees", viewpoint, in "Graf" and "Wall", zoom and rotation, in "Bark" and "Boat", illumination, in "Leuven", and JPEG compression, in "UBC". These changes were applied to the images in five different transformations denoted as H1to2, H1to3, and so on, in Table 2.1. The key-points found in an reference image I1 can be related to the key-points found in a transformed image I2 by the homography matrix H1to2. All homography matrices are provided in the database website [50].

the proposed method, we also have three octaves, but four (instead of six) scales, 0.05 for contrast threshold (more selective), and 10 for edge threshold. As it can be seen in Table 2.1, the systems have similar results for most cases. Discrepancies occur when zoom transformations are applied (images "Bark" and "Boat"), which can be a consequence of the decrease in the number of scales per octave. The largest discrepancies correspond to a limitation of the proposed method. Even though, the average repeatability result indicates that the overall performance of the proposed method is similar to the original method performance, which validates the implementation of the proposed hardware scale-space implementation for SIFT. Results presented in [21] show that there is a small change in the repeatability results if errors of the same magnitude as the ones found with the circuit level simulations are included in the system-level simulations.

Table 2.1: Repeatability using the original and the proposed scale space.

Image	Bark		Bikes		Boat		Graf	
Transformation	Original	Proposed	Original	Proposed	Original	Proposed	Original	Proposed
H1to2	67.54%	65.77%	56.55%	76.57%	59.39%	69.11%	60.47%	55.76%
H1to3	62.76%	30.86%	57.06%	76.33%	60.06%	10.04%	48.15%	22.19%
H1to4	75.21%	23.70%	53.83%	73.40%	43.47%	36.59%	22.96%	8.37%
H1to5	73.09%	0.00%	55.29%	71.98%	41.26%	57.42%	0.00%	0.00%
H1to6	70.21%	9.82%	48.53%	67.78%	31.97%	5.87%	0.00%	0.00%

Image	Leuven		Trees		UBC		Wall	
Transformation	Original	Proposed	Original	Proposed	Original	Proposed	Original	Proposed
H1to2	63.99%	74.13%	51.47%	65.46%	67.86%	83.26%	61.82%	67.77%
H1to3	60.86%	75.34%	51.51%	64.54%	63.84%	77.46%	57.12%	62.57%
H1to4	60.34%	73.38%	44.17%	62.08%	57.54%	72.37%	52.95%	47.56%
H1to5	57.85%	71.92%	42.07%	66.28%	42.46%	64.84%	41.35%	34.91%
H1to6	52.49%	73.75%	38.49%	68.72%	40.44%	59.21%	10.10%	15.53%

Average repeatability: Original = 50.16%; Proposed = 51.07%

Chapter 3

Comparative Analysis of Gaussian Pyramid Hardware Architectures: Throughput and Energy Considerations

This chapter presents a method for evaluating focal-plane image processing advantages in terms of overall processing time and energy [52], which is one of the main contributions of this work. This evaluation is done by comparing the desired focal-plane approach with an equivalent hypothetical digital implementation. The digital realization is idealized as simple as possible, which would allow for it to be implemented (hypothetically) in the same chip of the pixel matrix. We will compare the reference digital implementation, depicted in Figure 2.2(a), with the focal-plane approach described in Section 2.1 and sketched in Figure 2.2(b).

Gaussian pyramid generation requires filtering the original image, by means of convolution with a fixed kernel, and subsampling. In our case, the filter used is the binomial kernel (Equation (2.3)). In the digital processor, the convolution is performed by sliding a binomial kernel across the image. The image pixels falling within the kernel window are multiplied pixelwise by the kernel elements, and the multiplication results are summed up. The hypothetical digital circuit that is considered for comparison purposes has a processor with a multiply and accumulate (MAC) unit, formed by one or more PE. The binomial kernel only requires addition and division by four, so the MAC unit is realized by simple digital circuitry (logic adders and shift registers) placed outside the pixel array. Filtering with a 2×2 kernel requires four pixel values for each kernel window, but two of these values are kept from the previous window operation, requiring only two memory-read accesses per window. Likewise, one MAC operation per window can be spared if we consider a partial

result from the previous window. After each window computation the memory is accessed for writing the result.

Based on these steps and the above proposed digital circuit, the total processing time and energy consumption were estimated. The next sections explain in details how these estimations were computed.

3.1 Time Analysis Comparison

For a numerical comparison, the flows of both architectures are broken into tasks, which are analyzed considering processing time and energy consumption. In the time analysis, each task is related to a variable τ that represents the time to perform a given task once. We then compute the number of times the task is executed. Overall time is equal to τ multiplied by the number of executions of that task. Parallel signal processing reduces overall processing time. In that case, we divide τ by the number of processors. After finding the processing time expressions for both approaches as functions of τ, each τ is associated with the clock period, τ_{Clk}, which leads to expressions with a single global variable. Capture and data output transmission are steps required for both digital and focal-plane approaches, taking approximately the same time in both cases. Image capture and transmission are thus not considered in the time comparison.

3.1.1 Focal-Plane Approach Time Analysis

The focal-plane approach steps are inferred from Figures 2.2(a) and 2.5. The two main steps are Gaussian pyramid generation and analog-to-digital conversion:

1. Gaussian pyramid generation: the time it takes to generate the Gaussian pyramid depends on the number of charge redistribution operations multiplied by the time it takes for a single charge redistribution. This operation is performed inside each pixel, in parallel across the entire matrix. Consequently, image size does not affect the Gaussian pyramid generation time. Kernel size determines the number of matrix charge redistributions per level. We need $n_k - 1$ charge redistribution operations to implement an $n_k \times n_k$ kernel. From Figure 2.5 we see that this operation is repeated at every level, except the last one. Finally, we sum the charge redistribution operations that take place when the pyramid level changes. The overall number of charge redistribution operations is $N_{CR} = (N_{Lev} - 1) \cdot (n_k - 1) + (N_{Lev} - 1) = n_k\,(N_{Lev} - 1)$, where N_{Lev} is the number of pyramid levels. Multiplying N_{CR} by the time required for performing one charge redistribution, τ_{CR}, we have the overall processing time $\tau_{FP_{Proc}} = n_k\,(N_{Lev} - 1) \cdot \tau_{CR}$.

2. Analog-to-digital conversion: after each pyramid level is computed at the focal plane, pixel values are read out and sent to an analog-to-digital conversion stage, which comprises one or more ADCs. The time required for one ADC to perform one sample conversion is τ_{ADC}. Overall data conversion time depends on the number of ADCs, N_{ADC}, and on the number of data samples converted, N_{conv}. To compute N_{conv}, we note that for every pyramid level the image size is reduced by a factor of 4: $N_{conv} = MN + MN/4 + \ldots + MN/2^{2(N_{Lev}-1)}$. Generalizing:

$$N_{conv} = \sum_{n=1}^{N_{Lev}} \frac{M \cdot N}{2^{2(n-1)}}. \tag{3.1}$$

Total conversion time is thus $N_{conv} \cdot \tau_{ADC}/N_{ADC}$:

$$\tau_{ADC_{Total}} = \sum_{n=1}^{N_{Lev}} \frac{M \cdot N}{2^{2(n-1)}} \cdot \frac{\tau_{ADC}}{N_{ADC}}. \tag{3.2}$$

Overall focal-plane processing time is obtained by adding up $\tau_{FP_{Proc}}$ and $\tau_{ADC_{Total}}$:

$$\tau_{FP_{Total}} = n_k \left(N_{Lev} - 1\right) \tau_{CR} + \sum_{n=1}^{N_{Lev}} \frac{M \cdot N}{2^{2(n-1)}} \cdot \frac{\tau_{ADC}}{N_{ADC}}. \tag{3.3}$$

3.1.2 Digital Implementation Time Analysis

The digital approach requires more steps than the focal-plane approach, as it can be seen in Figure 2.2:

1. Analog-to-digital conversion: the captured image is immediately converted to digital. This is the only data conversion required by this approach. The size of the converted data is equal to the pixel array size. Thus, $\tau_{ADC_{Total}} = M \cdot N \cdot \tau_{ADC}/N_{ADC}$.

2. Memory storage: the resulting $M \times N$ digital values are stored into an internal memory. Time taken by this step is $M \cdot N \cdot \tau_{Mem}$, where τ_{Mem} is the time required for accessing a single memory position. To consider simultaneous memory access, we introduce a new variable, N_{busMem}, that represents the number of possible parallel accesses. The total time required by this step is $\tau_{matrixMemWrite} = M \cdot N \cdot \tau_{Mem}/N_{busMem}$.

3. Gaussian pyramid generation: the digital processor accesses the memory to read the input values for the current pyramid level, performs multiply and accumulate operations and writes the result back into the memory. The number of times this operation is performed depends on image size and on the number of times the image is filtered by the binomial kernel inside each level. Image size changes at every level according to a series similar to the one given by Equation (3.1). We do not perform convolutions at the highest level. The number of operations is equal to:

$$N_{op} = (n_k - 1) \cdot \sum_{n=1}^{N_{Lev-1}} \frac{M \cdot N}{2^{2(n-1)}}. \tag{3.4}$$

As explained in the beginning of the chapter, at least two pixel values are necessary in every computation of the 2×2 binomial kernel convolution, so we define $\tau_{memRead} = 2\tau_{Mem}$. At least three multiply and accumulate operations are used in the 2×2 kernel. The time required for one MAC unit to perform these operations is $\tau_{convolutionWindow} = 3\tau_{op}$, where τ_{op} is the time required by a single MAC operation. The resulting value is written in the memory through a single access, and so $\tau_{memWrite} = \tau_{Mem}$.

The time needed by a single PE to perform the convolution is obtained by multiplying the number of operations by the sum ($\tau_{memRead} + \tau_{convolutionWindow} + \tau_{memWrite}$). Assuming that more than one PE is available, and that N_{busMem} simultaneous memory accesses are allowed, parallel convolution operations are carried out. The overall time required for performing the convolution operations is, then, $\tau_{convolution} = N_{op}(2\tau_{Mem}/N_{busMem} + 3\tau_{op}/N_{PE} + \tau_{Mem}/N_{busMem})$. If $N_{PE} > N_{busMem}$, then memory access collisions occur. To simplify the analysis, we ignore this issue by assuming that every PE may access the memory at any moment, with no additional hardware complexity. This simplification benefits the digital approach. Making advantageous assumptions for the digital approach leads to a conservative assessment of the focal-plane approach advantages with respect to throughput and energy. In the $\tau_{convolution}$ equation, N_{busMem} is thus substituted by N_{PE}: $\tau_{convolution} = N_{op}(2\tau_{Mem} + 3\tau_{op} + \tau_{Mem})/N_{PE}$.

By adding $\tau_{ADC_{Total}}$, $\tau_{MatrixMemWrite}$, and $\tau_{convolution}$, we have the digital approach overall time:

$$\tau_{digital_{Total}} = M \cdot N \cdot \frac{\tau_{ADC}}{N_{ADC}} + M \cdot N \cdot \frac{\tau_{Mem}}{N_{busMem}} + (n_k - 1) \cdot \sum_{n=1}^{N_{Lev-1}} \frac{M \cdot N}{2^{2(n-1)}} \left(\frac{2\tau_{Mem} + 3\tau_{op} + \tau_{Mem}}{N_{PE}} \right). \quad (3.5)$$

3.1.3 ADC Architectures Comparison

Before using the above equations to compare focal-plane and digital approaches, it is important to remember that it is common to work with the ADC at a clock period different from the one used for the other parts of the circuit. In our case, we define τ_{Clk} as the period of the clock signal that controls the pixel array, memory, and digital circuitry. The ADC clock period, on the other hand, is $K_{ADC} \cdot \tau_{Clk}$, where K_{ADC} depends on ADC type.

We consider five ADCs commonly used in CMOS image sensors: ramp (which can be implemented in many ways, such as single slope -SS- or multi ramp multi slope - MRMS), successive approximation register (SAR), sigma-delta ($\Sigma\Delta$), cyclic and pipeline [53]. Because of circuit limitations, it is not fair to compare these ADCs considering the same clock frequency. Thus to allow for the comparison of digital and focal-plane approaches using different ADC types, we first compare ADCs among themselves and find the appropriate clock period in each case. For the ADC comparison we use reported imagers in which the performance figures of the embedded ADCs are included [54]-[89]. Table 3.1 shows the conversion rate, power consumption and number of bits for each converter used for the comparison. Values from this table where extracted from the ADCs references [54]-[89]. For the energy analysis, we are interested in the energy per conversion, which is found by dividing the power consumption by the conversion rate. ADCs have already been compared by different authors [53], [90]. The present comparison only considers ADCs designed for image sensors, in the context of comparative time and energy analysis, including recently published works with experimental results.

The ramp converter, which is a linear approximation converter with simple architecture requiring low area and low power consumption [91], is probably the most used converter in image sensor applications [67]-[77]. It is suitable for working with high clock frequencies. We thus use it as a reference for other converter types: the ramp ADC clock period is set equal to the global clock, $\tau_{Clk_{Ramp}} = K_{Ramp} \cdot \tau_{Clk} = \tau_{Clk}$, so $K_{Ramp} = 1$. The data converters in the comparison were designed for different resolutions. For a fair comparison, we normalize the conversion rates and energy values for the same number of bits, which is set as $N_{bits} = 8$. The normalization is described next. The conversion rates before normalization is presented in the third column of Table 3.1. The energy before normalization is given by the fourth column of Table 3.1, power consumption, divided by the third column, conversion

Table 3.1: ADC properties that are used in performance comparisons based on time and energy analyses.

Ref	Architecture	Conversion Rate (kS/s)	Power (μW)	Number of bits
[67]	SS	81.54	1.36**	14
[68]	SS	19.23	6	8
[69]	SS	127*	99**	10
[70]	SS	40	0.53**	8
[71]	SS	18.87	77.5**	10
[72]	SS	81.54	93.8**	10
[73]	SS	135*	366.21**	12
[74]	SS	290*	302.08**	10
[71]	MRMS	78.13	95**	10
[71]	MRMS	62.5	95**	10
[72]	2-step SS	250	112.4**	10
[75]	2-step SS	119.05*	90	12
[76]	2-step SS	27.78	128	12
[77]	2-step SS	312.5	68	14
[54]	SAR	4000*	381	9
[55]	SAR	2500	38	10
[56]	SAR	16000*	330	8
[57]	SAR	526.3	14.4	12
[58]	SAR	830	209	8
[58]	SAR	830	229	8
[59]	SAR	588.2	41	14
[60]	SAR	66.67	1.25	9
[61]	SAR	561.8	41	10
[62]	SAR	411	56	12
[78]	Cyclic	500	120	12
[79]	Cyclic	434.78	300	13
[80]	Cyclic	344.8	149	12
[81]	Cyclic	2000	90	10
[82]	Cyclic	526.3	101	12
[83]	Cyclic	2000	430*	12
[84]	Cyclic	1000	118	12
[63]	$\Sigma\Delta$	200*	117**	12
[63]	$\Sigma\Delta$	400*	117**	12
[64]	$\Sigma\Delta$	100	200	13.5
[65]	$\Sigma\Delta$	434.78	55	12
[66]	$\Sigma\Delta$	69.2	376**	12
[85]	Pipeline	50000	28000	12
[86]	Pipeline	50000	192900	14
[86]	Pipeline	70000	184900	10
[87]	Pipeline	96000	70000	12
[88]	Pipeline	50000	17500	10
[89]	Pipeline	96000	400000	12

The references that appear more than once in the table indicate more than one ADC.

Acronyms: SS - single slope; MRMS - multi ramp multi slope; SAR - successive approximation register.

* Computed values based on information given in the paper.

** Power consumption per ADC computed using the given total power and divided by the number of ADCs.

rate. Although imagers with higher number of bits are common (particularly for HDR applications, mentioned in Chapters 4 and 5.1.1, in which 12-14 bits are necessary to convert the entire range that the pixel is capable of capturing), eight bits per pixel is more typical [92]. The conversion rate normalization depends on the number of clock cycles per bit each converter architecture requires. A single slope ramp converter, for example, requires $2^{N_{bits}} \cdot \tau_{Clk_{Ramp}}$ (maximum) for a conversion. The normalized conversion rate considering eight bits is $f_s' = 2^{N_{bits}} \cdot f_s/2^8$, where f_s and N_{bits} are the reported conversion rate and resolution, third and fith columns of Table 3.1. For the SAR and cyclic converters, the conversion time is $N_{bits} \cdot \tau_{Clk_{SAR,Cyclic}}$, so the normalization is $f_s' = N_{bits} \cdot f_s/8$. The $\Sigma\Delta$ conversion time depends on the oversampling rate (OSR). For second-order incremental $\Sigma\Delta$ converters, the number of bits is $N_{bits} = \log_2 [\mathrm{OSR} \cdot (\mathrm{OSR} + 1)] - 1$, where OSR is the reported oversampling rate. Using this equation and considering that we defined the number of bits equal to eight, we consider an oversampling rate equal to 25 because it yields a resolution equal to 8.3 bits. The normalization is $f_s' = \mathrm{OSR} \cdot f_s/25$. The pipeline converter conversion time is one $\tau_{Clk_{Pipeline}}$, with some latency, which does not depend on the number of bits, i.e. normalization is not required. Pipeline converters are not as common in image sensors as the other converter types (simulation results have been reported, as well as experimental results from ADC chips working together with imaging chips), but they are included in the comparison because of their improved speed.

To normalize energy figures, we assume that the power consumption doubles for every bit added [90]: $E = 2^8 \cdot P/(f_s' \cdot 2^{N_{bits}})$. Walden's figure of merit for ADCs [93] uses the effective number of bits (ENOB) instead of the resolution. The normalized energy values in Figure 3.1 are based on the resolution because some references do not report ENOB. Figure 3.1 shows the normalized energy versus normalized conversion rate for the five ADC types considered. The median conversion rate and energy (black markers in the figure) are chosen as representative values for each converter type. The median values suggest that, for eight-bit resolution, cyclic and SAR converters are approximately two times faster than ramp converters. The conversion times are related according to $\tau_{ADC_{Ramp}} = 2 \cdot \tau_{ADC_{SAR,Cyclic}}$ and $\tau_{ADC_{Ramp}} = 2^8 \cdot \tau_{Clk_{Ramp}}$, $\tau_{ADC_{SAR,Cyclic}} = 8 \cdot \tau_{Clk_{SAR,Cyclic}}$. So, the cyclic or SAR converters run at a clock which is approximately 16 times slower than the ramp converter clock. For the focal-plane and digital approaches comparison, we thus assume $K_{SAR,Cyclic} = 16$, where $K_{SAR,Cyclic}$ is the constant that multiplies the global clock period τ_{Clk} to yield $\tau_{Clk_{SAR,Cyclic}}$. The $\Sigma\Delta$ conversion time is 1.3 times smaller than the ramp ADC conversion time, so $\tau_{Clk_{\Sigma\Delta}} = 2^8 \cdot \tau_{Clk_{Ramp}}/(1.3 \cdot 25) \approx 8\tau_{Clk_{Ramp}}$. The multiplying constant is $K_{\Sigma\Delta} = 8$. For the pipeline converter, $\tau_{Clk_{Pipeline}} = [2^8/(\tau_{ADC_{Ramp}}/\tau_{ADC_{Pipeline}})]\tau_{Clk_{Ramp}}$ and $\tau_{ADC_{Ramp}}/\tau_{ADC_{Pipeline}} \approx 130$, $K_{Pipeline} = 2$.

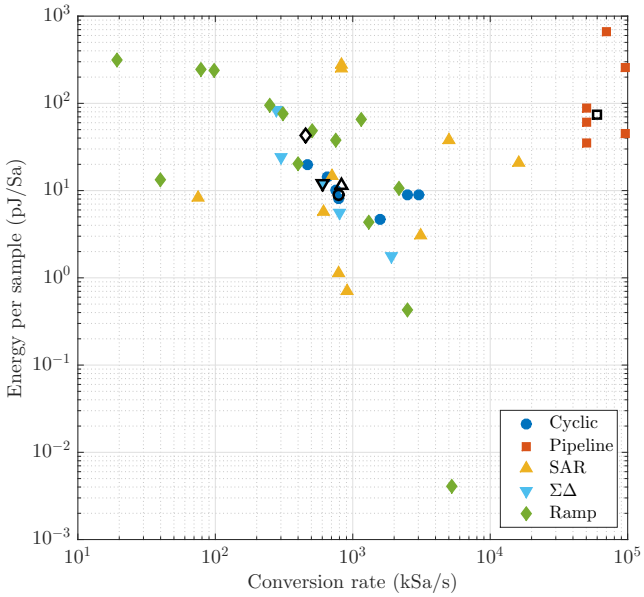


Figure 3.1: Eight-bit normalized conversion rate versus energy per sample, for five ADC types. The median values for each type of ADC are plotted using black unfilled markers with the respective shape.

Summarizing, we defined $K_{ramp} = 1$, since this converter is used as reference, and, using reported figures, found $K_{SAR,Cyclic} = 16$, $K_{\Sigma\Delta} = 8$ and $K_{Pipeline} = 2$. These constants define the ratio between the ADC clock period and the clock period τ_{Clk}, used for the other stages of the circuit.

3.1.4 Time Comparison Results

We now establish some default values for the parameters in Equations (3.3) and (3.5), and associate the overall times to a global clock period. As explained in Section 3.1.3, τ_{Clk} is the period of the clock signal that controls the pixel array, memory, and digital circuitry and $K_{ADC} \cdot \tau_{Clk}$ is the ADC clock period.

Assuming that charge redistribution is practically instantaneous, it is clear from Equation (3.3) that the bottleneck of the focal-plane approach is at the ADC, because of the amount of data to be converted. The digital approach bottleneck, on the other hand, is either at the ADC or at the processing stage, which depends on ADC type. For either approach, we explore different ADC types and N_{ADC} values. For the digital approach, we explore several N_{PE} values. We thus do not define default values for τ_{ADC}, N_{ADC}, and N_{PE}. The maximum N_{ADC} value is set to the number of columns at pyramid Level 0, since image sensors with one ADC per column are commonly found [94]. Although stacking technologies allow for the integration of one ADC per pixel [54], this is still an upcoming technology with high fabrication costs.

We use video graphics array (VGA, 640×480 pixels) standard for the pyramid Level 0 image size. Consequently, the pixel array size in the focal-plane approach is 1280×960. The time analysis does not change significantly if the resolution increases, but the bandwidth for the transmission of the generated data increases. Increasing the resolution and using one ADC per column also increases power consumption. The pyramid size can not be too large, because computation accuracy is limited by leakage currents. The operations can be performed as long as the capacitance voltages are not affected by these currents. Temperature variations have an influence in leakage currents, but temperature effects are not considered in this work. We set $N_{Lev} = 4$. Setting $N_{busMem} = 4$ yields a suitable balance between circuit complexity and speed. Choosing $N_{busMem} = 1$ would impair digital circuit performance, but increasing the number of simultaneous memory accesses increases digital circuit size and complexity.

Charge redistribution, memory access and MAC operation times (τ_{CR}, τ_{mem} and τ_{op}) are written as functions of the clock period τ_{Clk}. Charge redistribution itself is practically instantaneous, but the time required for driving the charge redistribution switches is considered, so $\tau_{CR} = 1\tau_{Clk}$. The time to access the memory, τ_{mem}, was defined as $2\tau_{Clk}$ considering that one clock period is necessary to define the position of the memory access and another to actually access that position. The time to perform a MAC operation, τ_{op}, was also defined as $2\tau_{Clk}$: division by four requires two clock cycles (two shift operations), and summation is performed by combinatorial logic, whose output does not depend on the clock. Table 3.2 summarizes the established parameter values. Applying the parameter values in Equations (3.3) and (3.5) yields:

Table 3.2: Time analysis equations parameters.

Parameter	Value
Pyramid Level 0 size (M×N)	640×480
Maximum number of ADCs ($N_{ADC_{Max}}$)	640
Equivalent kernel size (n_k)	5
Number of bits (N_{bits})	8
Number of levels (N_{Lev})	4
Number of memory accesses (N_{busMem})	4
Time to perform charge redistribution (τ_{CR})	$1\tau_{Clk}$
Time to access the memory (τ_{Mem})	$2\tau_{Clk}$
Time to perform a MAC operation (τ_{op})	$2\tau_{Clk}$
K_{ramp}	1
$K_{SAR,Cyclic}$	16
$K_{\Sigma\Delta}$	8
$K_{Pipeline}$	2

$$\tau_{FP} = 15\tau_{Clk} + \frac{640 \cdot 480 \cdot 1.33\tau_{ADC}}{N_{ADC_{FP}}}, \text{and} \tag{3.6}$$

$$\tau_{digital} = 640 \cdot 480 \left(\frac{\tau_{ADC}}{N_{ADC_{Dig}}} + \frac{\tau_{Clk}}{2} + \frac{63\tau_{Clk}}{N_{PE}} \right). \tag{3.7}$$

Equations (3.6) and (3.7) allow different N_{ADC} values for focal-plane and digital approaches. Charge redistribution time is not taken into account, because of its negligible contribution to Equation (3.6). The ratio between the expressions in Equations (3.7) and (3.6) is:

$$\frac{\tau_{digital}}{\tau_{FP}} = \frac{\left(\frac{\tau_{ADC}/\tau_{Clk}}{N_{ADC_{Dig}}} + \frac{1}{2} + \frac{63}{N_{PE}} \right)}{\frac{1.33\tau_{ADC}/\tau_{Clk}}{N_{ADC_{FP}}}}. \tag{3.8}$$

Using the K_{ADC} constants defined in Section 3.1.3, we replace τ_{ADC} in Equation (3.8) by an appropriate function of τ_{Clk}, which depends on the converter architecture. For the ramp converter we have $\tau_{ADC} = 2^8 \cdot K_{Ramp} \cdot \tau_{Clk} = 256\tau_{Clk}$. Considering that both the focal-plane and digital approaches use the ramp converter, the maximum advantage that the focal-plane approach achieves occurs when $N_{ADC_{Dig}} = 1$, $N_{PE} = 1$ and $N_{ADC_{FP}} = N_{ADC_{Max}} = 640$. The focal-plane approach is then 600 times faster than the digital approach. If $N_{ADC_{Dig}} = N_{ADC_{FP}} = N_{ADC_{Max}} = 640$, the focal-plane approach is 120 times faster. For ramp converters, the effect of increasing the number of PEs is shown in Figure 3.2, in dash-dotted line, where the ratio between digital and focal-plane total operation times is plotted. With only four PEs, the focal-plane approach is 31 times faster, so for ramp ADCs the focal plane advantage is modest.

For the SAR or cyclic converters, we have $\tau_{ADC} = N_{bits} \cdot K_{SAR,Cylic} \cdot \tau_{Clk} = 128 \cdot \tau_{Clk}$. These converters require fewer clock cycles to perform one conversion, but their operation frequency is limited, hence resulting in performance comparable to that of the ramp ADC. The maximum advantage the focal plane achieved with SAR or cyclic converters corresponds to 700 times faster. The dashed line in Figure 3.2 shows the evaluation of Equation (3.8) for the SAR converter when $N_{ADC_{Dig}} = N_{ADC_{FP}} = N_{ADC_{Max}} = 640$. To reduce the advantage of the focal plane to less than two orders of magnitude, three PEs are necessary. With ten PEs, the focal-plane approach is 28 times faster. The $\Sigma\Delta$ conversion time depends on the OSR, which is equal to 25, as explained in Section 3.1.3: $\tau_{ADC} = \text{OSR} \cdot K_{\Sigma\Delta} \cdot \tau_{Clk} = 200 \cdot \tau_{Clk}$. The dotted line in Figure 3.2 shows the comparison between focal-plane and digital approaches when the $\Sigma\Delta$ converter is used. The result lies in between the ramp converter and SAR converter results: only two PEs are necessary to reduce the advantage of the focal

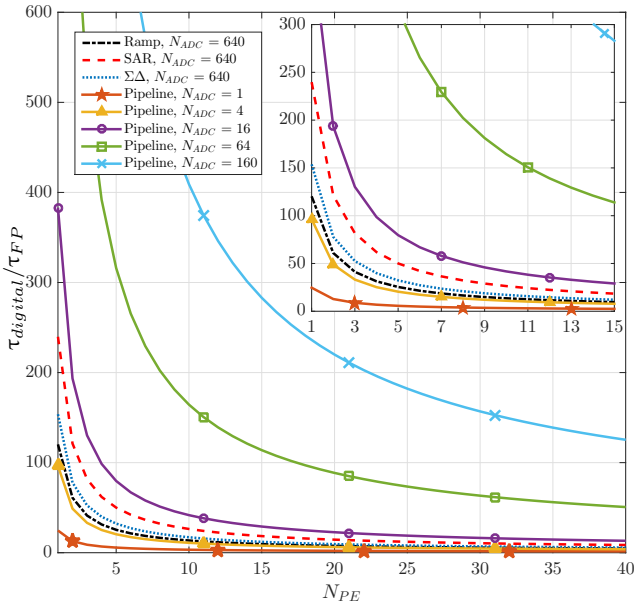


Figure 3.2: Ratio between digital and focal-plane processing times as a function of the number of PEs. Ramp, SAR, ΣΔ, and pipeline ADCs are shown, respectively, in dash-dotted, dashed, dotted, and solid lines. For better visualization, a zoom of the curves is presented at the top right part of the figure.

plane to less than two orders of magnitude. For the pipeline converter analysis, we assume that it is not possible to integrate 640 converters inside the chip, because an imager with one pipeline converter per column has not been reported, to the best of our knowledge. For this converter, $\tau_{ADC} = K_{Pipeline} \cdot \tau_{Clk} = 2 \cdot \tau_{Clk}$. The solid lines in Figure 3.2 correspond to results considering different numbers of pipeline ADCs. The focal-plane approach is highly advantageous when the number of ADCs is higher than 64. In this case, 18 PEs are necessary to reduce the focal plane advantage to less than two orders of magnitude.

The speed of the digital processor may be increased by using double data rate (DDR), which allows for memory access and shift operation (division by four) to be carried out in a single clock period. In order to perform timing comparisons between the focal-plane approach and generic digital circuits not having additional power or area requirements, we do not take the DDR into account in the analysis. Nevertheless, if $\tau_{mem} = \tau_{op} = \tau_{Clk}$, the processing time ratios presented in Figure 3.2 halve.

While focal-plane processing is being performed it is not possible to capture a new frame, which limits the frame rate. Still, the frame rate is much larger than 30 fps for VGA resolution: if we consider a 100 MHz global clock, and one ramp converter per column, then approximately 1600 μs are necessary for generating the Gaussian pyramid. Assuming that the image capture requires an additional 400 μs, then 2000 μs are necessary for image capture and Gaussian pyramid generation,

which yields a frame rate around 500 fps. If the image resolution is increased to 6400×4800 (a factor of 100), it is still possible to achieve 60 fps by keeping the same conditions, which are namely one ramp converter per column and a global clock frequency of 100 MHz.

The energy analysis is more complicated because it is highly dependent on the architecture, the technology parameters are also of major importance and there is no global parameter (as the clock period was global in the time analysis). Also, aside from the stages necessary for the Gaussian pyramid generation in each approach, both architectures must comprise the controlling circuits outside the pixel matrix, which are responsible for the interface between each stage shown in Figure 2.2. Although these circuits play an important part on the energy consumption, a proper energy analysis of the controlling circuitry requires a careful design of this stage, which is not under the scope of this work, so these circuits are not considered.

3.2 Energy Analysis Comparison

For the ADC stage, the energy consumption depends on the type of converter and architecture. A general empirical analysis on the energy efficiency of ADC architectures based on reported values from 1400 papers can be found in [95]. In that reference, the author plots the reported energy per sample versus the ENOB (where ENOB is the data converter effective number of bits) for different ADC types. Based on this plot, the author sets $2^{2(ENOB-9)}$ pJ/Sa as a lower bound for energy consumption per sample, and claims that setting the resolution to less than nine bits results in minor advantages regarding energy savings. The minimum energy per sample in our case, considering eight bits, would be thus equal to 1 pJ/Sa, which is given when ENOB = 9. Although it is important to have this lower bound limit, which was found by analyzing several reported measured data for different ADC architectures, it is also interesting to consider converters that have been designed specifically for image sensors. As mentioned in Section 3.1.3, several references were used for finding representative conversion rate and energy consumption values for each ADC architecture. The median energy consumption per sample for each type of ADC, which can be seen in Figure 3.1, is used in this section.

Aside from the ADC, the other sources of energy consumption can be divided in: DC consumption, E_{DC}, when there is a constant current flowing, usually for biasing circuits; dynamic consumption, $E_{Dynamic}$, as a result of the circuit activity, which requires charging and discharging capacitive nodes of the circuit; static consumption, E_{Static}, which is the energy that the transistor consumes even when it is off, depending on the leakage current I_{leak}; and short-circuit consumption, $E_{Shortcircuit}$, which is another sink of dynamic energy. The short ci-

rcuit happens when logic gate inputs are switched, at the moment when both n-channel and p-channel transistors are on, thus allowing for a short-circuit current to flow. The short-circuit current can be minimized by matching the rise/fall times of the input and output signals, reaching a maximum of 15% of the total dynamic consumption [96]. $E_{Shortcircuit}$ is computed as a portion of the dynamic energy: $E_{Shortcircuit} = 15(E_{Dynamic} + E_{Shortcircuit})/100 \rightarrow E_{Shortcircuit} = 15E_{Dynamic}/85$. In the following equations, C_n is the node capacitance, $V_{dd,Matrix}$ is the pixel matrix voltage supply and V_{dd} is the voltage supply outside the pixel matrix.

The dynamic power consumed by a digital circuit can be estimated by $P_{Dynamic} = N_d \cdot C_n \cdot V_{dd}^2 \cdot f_{0\rightarrow1}$, where N_d is the number of nodes and $f_{0\rightarrow1}$ is the frequency at which the nodes switch from 0 to 1 [96]. This equation assumes that every node in the digital circuit is capacitive and that the energy necessary to charge a capacitive node is equal to $C_n \cdot V_{dd}^2$ [91]. The switching frequency can be written as a function of the clock frequency: $f_{0\rightarrow1} = \alpha f_{clk} = \alpha/\tau_{Clk}$, where α is called switching activity factor and represents the probability of a node switching from 0 to 1, resulting in $P_{Dynamic} = \alpha \cdot N_d \cdot C_n \cdot V_{dd}^2/\tau_{Clk}$. The energy is given by $P_{Dynamic}$ multiplied by the time during which the circuit operates: $E_{Dynamic} = \alpha \cdot N_d \cdot C_n \cdot V_{dd}^2 \tau_{Total}/\tau_{Clk}$. For the circuits presented in this work, τ_{Total} can be computed according to the time analysis presented in Section 3.1.

The SRAM memory is considered for the energy analysis of the digital circuit. The schematic diagram of a one-bit cell of this memory is shown in Figure 3.3. The memory has the same size of the Level 0 image in the pyramid, M×N, and each pixel is represented with N_{bits}. In order to read a value from the memory, we need to select the memory row using the switch WL and read the result in the BL bus. Writing requires selecting a memory cell through the WL switches and setting $\overline{\text{Write}}$ to zero, which turns on transistor M_1 or M_2, depending on the bit that is being written, W_{bit}. If W_{bit} is logical zero, transistor M_2 is turned on and the bias current generated by V_{bias} discharges the bitline BL. If W_{bit} is logical one, transistor M_1 is turned on and the bias current discharges the bitline $\overline{\text{BL}}$ and thus charges BL.

3.2.1 Focal plane

In addition to the A/D conversion stage energy analysis, which was described in Section 3.1.3, the steps that were considered for the energy consumption estimation are described next. As opposed to the time analysis computation, here we have to consider the image capture and readout steps because the pixel matrix size has an influence in the consumption.

1. Image capture: this operation involves, for each pixel, charging the floating diffusion node and operating the Reset and TX switches, shown in Figure 2.3.

Dynamic: the energy for capturing a single pixel can be estimated as the one necessary for charging three capacitances, $E_{pixCapture} = (C_{FD} \cdot V^2_{dd,Matrix}) + (C_{Rst} \cdot V^2_{dd,Matrix}) + (C_{TX} \cdot V^2_{dd,Matrix})$. Since this operation happens for every pixel of the matrix, $E_{capture} = 2M \cdot 2N \cdot E_{pixCapture}$. The capacitances C_{FD}, C_{Rst} and C_{TX} can be replaced by the node capacitance C_n, thus $E_{capture} = 2M \cdot 2N \cdot (3 \cdot C_n \cdot V^2_{dd,Matrix})$. *Static*: transistors M_1 and M_2 from Figure 2.3 are off for most of the operation and contribute with static energy consumption, $E_{matrixStatic} = 2(2M \cdot 2N \cdot V_{dd,Matrix} \cdot I_{leak} \cdot \tau_{FP_{Total}})$, where $\tau_{FP_{Total}}$ is given by Equation (3.3).

2. Charge redistribution: this operation is passive, but energy is necessary to close the switches that connect the floating diffusion nodes. *Dynamic*: the energy that is needed to turn on (or to turn off) switches per pixel, $2C_n \cdot V^2_{dd,Matrix}$, must be multiplied by the number of times the charge redistribution is performed (from Section 3.1.1) and by the size of the pixel matrix, since the operation is performed throughout the entire matrix, $E_{CR} = (N_{Lev} - 1)n_k \cdot 2M \cdot 2N \cdot (2C_n \cdot V^2_{dd,Matrix})$, where n_k is the size of the filter.

3. Image readout: reading a pixel requires turning the row select switch on and enabling the current source that biases the source follower. This current flows for the time necessary to charge the pixel matrix column capacitance. *Dynamic*: the gate of transistor M_4, from Figure 2.3, is connected to a bus with every other select transistor of the same row of the matrix. The equivalent capacitance is estimated as $2M \cdot C_n$. The pixel matrix column capacitance, on the other hand, depends on the number of rows and is estimated as $2N \cdot C_n$. The dynamic energy is thus $E_{pixelReadDynamic} = (2M + 2N) \cdot C_n \cdot V^2_{dd,Matrix}$. The pixel matrix columns capacitances are charged whenever a pixel is read. The number of times a pixel is read is equal to N_{conv}, as defined in Section 3.1.1. The row select switch is activated every time the image

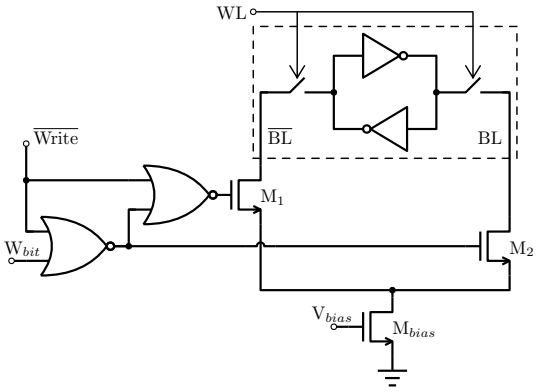


Figure 3.3: One-bit SRAM memory cell, inside the dashed box, and memory write control circuit.

is being read, once for each row, thus N_{conv}/M times. The total energy is $E_{readTotal} = [(N_{conv}/M) \cdot 2M + N_{conv} \cdot 2N] \cdot C_n \cdot V^2_{dd,Matrix}$.

3.2.2 Digital

For the digital approach, we have the following steps:

1. Image capture: following the same analysis as in the focal-plane case, but changing the image size, yields $E_{capture} = M \cdot N \cdot (3 \cdot C_n \cdot V^2_{dd,Matrix})$ and $E_{matrixStatic} = 2(M \cdot N \cdot V_{dd,Matrix} \cdot I_{leak} \cdot \tau_{Digital})$.

2. Image readout: also very similar to the focal plane, but for a different bus capacitance and for a single image readout, $E_{readTotal} = (N \cdot M + M \cdot N \cdot N) \cdot C_n \cdot V^2_{dd,Matrix}$.

3. MAC operation: the digital processor that is considered is a MAC unit composed by a logic adder and a shift register. *Dynamic*: the energy consumption of a digital circuit was explained in the beginning of Section 3.2. In the case of the MAC operation, the time during which the circuit operates is $N_{op} \cdot 3\tau_{op}$ (according to Section 3.1.2), so $E_{MACdynamic} = \alpha \cdot N_d \cdot C_n \cdot V^2_{dd}(N_{op} \cdot 3\tau_{op})/\tau_{Clk}$. *Static*: static energy consumption depends on the overall number of transistors inside the digital ports. Half of the transistors inside a common logic gate are off, so $E_{MACstatic} = N_{Off} \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit*: as explained in the beginning of the section, $E_{MACshortcircuit} = 15E_{MACdynamic}/85$.

4. Memory read: reading requires charging the WL bus capacitance C_{WL}, two switches per bit, and the BL or $\overline{\text{BL}}$ bus capacitance, represented by C_{BL}. *Dynamic*: $E_{readDyn} = (\alpha \cdot C_{BL} + C_{WL}) \cdot V^2_{dd} \cdot N_{op} \cdot 2\tau_{mem}/\tau_{Clk}$, where $N_{op} \cdot 2$ is the number of times the memory is accessed for reading, according to Section 3.1.2. The activity factor α is only necessary for the BL bus and represents the cases where the bus voltage does not change when closing WL. The WL switch remains closed while the reading is performed and opens right after, so there is no activity factor in this case. *Static*: from Figure 3.3, inside a one-bit memory cell, each inverter has one n-channel transistor and one p-channel transistor. Regardless of the state of the memory there is one p-channel transistor off and one n-channel transistor off. Besides, the WL switches can be formed by one n-channel transistor each, which are off most of the time. Thus, $E_{readStatic} = 4 \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit*: $E_{readShortcircuit} = 15(E_{readDyn})/85$.

5. Memory write: writing a single value in the memory requires more energy than reading a single position of the memory because the bias current is activated, and the writing controlling circuits are used. *Dynamic*: $E_{writeDyn} = (\alpha \cdot C_{BL} +$

$C_{WL} + \alpha \cdot C_{W_{bit}} + C_{\overline{\text{Write}}} + C_n) \cdot V_{dd}^2 \cdot N_{MemWrite} \cdot \tau_{mem}/\tau_{Clk}$, where $C_{W_{bit}}$ is the capacitance of the input W_{bit} of the controlling circuit, $C_{\overline{\text{Write}}}$ is the capacitance of the node $\overline{\text{Write}}$ and C_n is the gate capacitance of either M_1 or M_2, which are complementary nodes, so only one capacitance is considered. The number of times the memory is accessed for writing is $N_{MemWrite} = M \cdot N + N_{op}$, from Section 3.1.2. *Static*: memory write static energy consumption is due to the write control circuit alone. The cell circuit contribution to static energy consumption was provided in item (4). Transistors M_1 and M_2 are on only when a bit is written, so we assume that they contribute with static energy consumption during the entire operation. These transistors are necessary for every column of the memory matrix, so the contribution of these transistors to the static energy consumption is multiplied by $N_{bits} \cdot M$. Furthermore, inside the NOR gates there are always two transistors off. We can consider that this circuit is repeated for each bit and for, at least, each N_{busMem}, thus resulting in $E_{writeStatic} = (M \cdot 2 \cdot C_n + N_{busMem} \cdot 4 \cdot C_n) \cdot N_{bits} \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit*: $E_{writeShortcircuit} = 15(E_{writeDyn})/85$. *DC*: the bias current, that is activated whenever we need to swap a bit in the desired writing position, flows only for the time necessary to discharge the bus capacitance: $E_{MemDC} = \alpha \cdot N_{memWrite} \cdot V_{dd} \cdot I_{biasMem} \cdot \tau_{Clk}/10$, where τ_{Clk} is divided by ten to model capacitance discharge time. Capacitance discharge time is significantly shorter than the clock period. The activity factor is necessary to represent the cases where the cell bit that is being written does not change.

3.2.3 Energy Comparison Results

To compare focal-plane and digital approaches, we use the values shown in Table 3.3. Node capacitance, voltage supply, leakage and memory bias current were established by means of simulations with a 110 nm CMOS technology. To simplify the analysis, I_{leak} was estimated for a minimum size n-channel transistor and used for both n-channel and p-channel transistors. The clock frequency determines static energy consumption: 100 MHz is arbitrarily chosen, considering the clock frequency reported in some papers [69], [74]. The activity factor is $0 < \alpha \leq 1$ [96]. Two values, 0.2 and 0.8, were chosen for α to give an idea of how the energy changes according to it. An activity factor closer to one benefits the focal-plane approach. Data converter energy values are the median energy consumption values from Figure 3.1.

Aside from the values defined in the table, it is also necessary to estimate the number of nodes of the MAC unit circuit. An example of a two-bit adder with carry and an eight-bit shift register is shown in Figure 3.4. From the figures, we deduce that an N_{bits} adder requires at least $4 + 7(N_{bits} - 1)$ nodes, and the shift register

Table 3.3: Energy analysis equations parameters.

Parameter	Value
Node capacitance (C_n)	4 fF
Matrix voltage supply ($V_{dd,Matrix}$)	3.3 V
Voltage supply outside the matrix (V_{dd})	1.5 V
Leakage current (I_{leak})	2.6 pA
Memory $I_{biasMem}$	50 μA
Clock frequency	100 MHz
Activity factor (α)	0.2; 0.8
Ramp ADC energy	43 pJ/sample
$\Sigma\Delta$ ADC energy	12 pJ/sample
SAR ADC energy	11 pJ/sample
Cyclic ADC energy	9 pJ/sample
Pipeline ADC energy	74 pJ/sample

with N_{bits} bits requires at least N_{bits} nodes. Thus, a single PE of our MAC unit can be implemented with $(8 \cdot N_{bits} - 3)$ nodes. The flip-flop from Figure 3.4 actually requires more nodes, but we are assuming N_{bits} nodes as an optimistic estimation, which benefits the digital approach.

Determining the memory node capacitances is also necessary for the comparison. The capacitance of the node $\overline{\text{Write}}$, $C_{\overline{\text{Write}}}$, is equal to $2C_n$, since $\overline{\text{Write}}$ is connected to two logic gate inputs. For the bit capacitance, considering that it is connected to

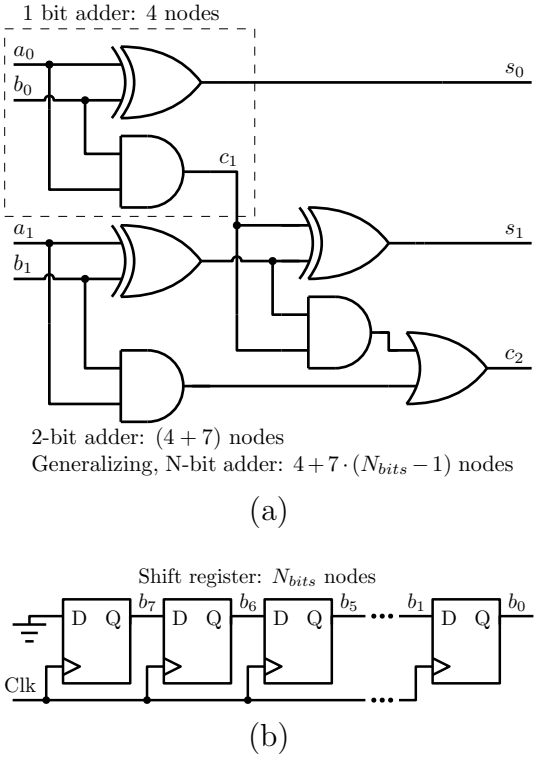


Figure 3.4: Circuits considered for the MAC energy estimation: (a) N_{Bits} adder and (b) shift register.

a column bus, $C_{W_{bit}} = N \cdot C_n$. The bitline capacitance also depends on the number of rows, $C_{BL} = N \cdot C_n$. The wordline capacitance depends on the number of the memory matrix columns: $2 \cdot N_{bits} \cdot M \cdot C_n$.

Considering the values from Table 3.3, $\alpha = 0.2$ and 640 converters for both focal-plane and digital approaches, the former requires 33 times less energy than does the latter approach when the ramp converter is being used. For the SAR, cyclic and $\Sigma\Delta$ converters, the focal plane is around 52 times more energy-efficient. For the pipeline converter, the focal-plane approach is 24 times more energy-efficient. Making $\alpha = 0.8$, there is a modest increase in the advantage of the focal plane: it is 34, 54 and 25 times more energy efficient for the ramp, SAR (also cyclic and $\Sigma\Delta$) and pipeline, respectively.

It is interesting to see the effect of the capacitance increase on the result. Since most of the nodes considered for the analysis are connected to metal input or output lines (namely the matrix reset control signal nodes, transfer gate nodes, row select nodes, output voltage node, memory bit-line and word-line capacitances,) the metal parasitic effects would probably result in capacitances higher than the ones considered. Figure 3.5 shows how the ratio between digital energy consumption ($E_{digital}$) and focal-plane energy consumption (E_{FP}) varies as the C_n of the nodes connected to metal lines increases. The activity factor used in this plot is 0.2.

Let us consider, for example, that we use the ADC presented in [61]. This is a column parallel SAR ADC that, normalized to eight bits, consumes 14.6 pJ per sample, with an ADC clock frequency of $\tau_{Clk_{SAR}} = 5.6$ MHz. Under these conditions, the focal-plane approach takes 911 μs to generate the Gaussian pyramid. If we use 10 PEs in the digital approach, then the focal plane is 26 times faster. The energy consumed with the focal-plane approach is around to 23 μJ, or, equivalently, 49 times more energy-efficient than the digital approach.

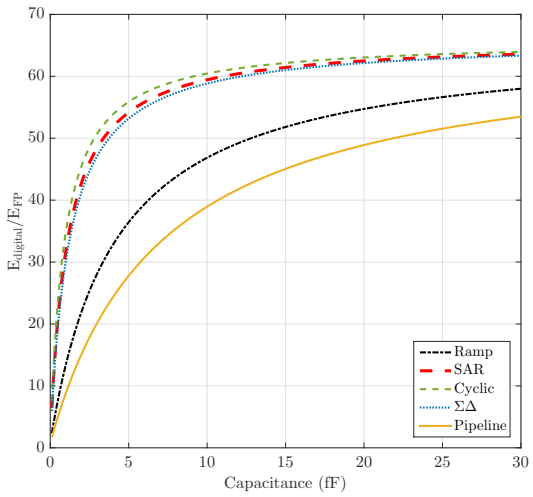


Figure 3.5: Node capacitance effect on energy consumption.

Chapter 4

A Pixel for Asynchronous HDR Acquisition through Adaptive Tone Mapping

In an image sensor, the dynamic range is defined by the ratio between the maximum and minimum luminance that the imager can sense. Due to the photoelectric effect, the incident light is converted to a photocurrent inside the photodiode. Since the photocurrent is proportional to the incident light, the dynamic range can be defined as:

$$\mathrm{DR[log]} = 20\log_{10}\left(\frac{i_{max}}{i_{min}}\right), \tag{4.1}$$

where i_{min} is the minimum current perceived, which will be determined by the image sensor noise, and i_{max} is the maximum current that the imager is capable of reading before saturation. The dynamic range of conventional CCD and CMOS cameras is around 65 to 75 dB [97]. The human eye, on the other hand, can reach reach up to 160 dB with long time adaptation [25], but, in the presence of bright light, the details in shadows are not well perceived [26]. A scene that includes sun light and shadowed areas can reach 170 dB of dynamic range [25]. When taking a picture of such scenes it is desirable that the captured image has details from both the brightest and the darkest areas so that the representation of the scene is as faithful to the real scene as possible. With an image that looks as close as possible the real scene, the viewers *quality of experience* (as defined in [24]) is improved. HDR imagers play an important role in applications such as vehicle safety, driver assistance, surveillance, field monitoring, or in any other application in which light conditions are not controlled. Real-time adjustment to the dynamic range of the scene is also necessary in some cases. Driver assistance, for instance, requires dealing with abrupt light changes: for example, when a vehicle enters a tunnel, when a

vehicle faces the headlights of oncoming vehicles, or when the vehicle faces road sign reflection lights [26]. Furthermore, many applications require more than the range visible to the eye, such as satellite and medical imaging, industrial welding or physically-based rendering [25], [98].

The bit depth and tone mapping concepts are related to the dynamic range concept. If the dynamic range of the image sensor is increased, the number of bits should also be increased so that the entire range can be digitally represented. A faithful representation of an HDR scene requires more than the usual eight bits depth. Professional cameras have the option of storing the raw data captured by the sensor with around 12 or 14 bits per pixel. When compared to an eight-bit representation of the same scene, the raw image has more details, being more faithful to the original scene. There are two problems with the use of 12 bits: the need for more storage space and, most important for our case, its incompatibility with most of the displays that are currently available. Tone mapping is necessary when we want to transform a twelve-bit image into an eight-bit image either for display or for storage. Simply clamping the exceeding bits results in significant image quality reduction (detail loss). A tone-mapping operator reduces the dynamic range while preserving scene characteristics and its most important visual details [98]. The most important details are defined as those to which the human visual system pays more attention depending on local contrast properties.

Figure 4.1(a) shows an example of a clamped eight-bit image which originally had 14 bits. It is not possible to see the original, 14-bit, image without an HDR display. In an eight-bit display, the original image is presented as in Figure 4.1(a). Figure 4.1(b) shows the same image tone mapped with a logarithmic function followed by a normalization. As it can be seen, Figure 4.1(a) is very dark, and it has undergone a large loss of details. Figure 4.1(b), on the other hand, has a better subjective quality and is capable of representing more details than the clamped image. There is also loss in the tone mapped image, but it is not as evident to the naked eye as in the clamped image. Two other examples of tone-mapping operators are presented in Figure 4.1. The tone-mapping operator in Figure 4.1(c) was introduced in [1]. This tone-mapping operator is based on a simplified computational model of the human visual system photoreceptors which considers the adjustment of the photoreceptor to the illumination. The tone-mapping function is $I/[I+\sigma(I_a)]$, where I is the pixel value and $\sigma(I_a)$ is a function that represents the photoreceptor long-term adaptation state, which depends on the adaptation level, I_a [1], computed using the pixel value and the matrix average pixel value. The last image, Figure 4.1(d), is the result of a system-level simulation of the tone-mapping function that the implemented pixel can generate. The pixel operation and the method used for system-level simulation will be further explained in Section 4.1. As in [1] our tone-mapping operator also

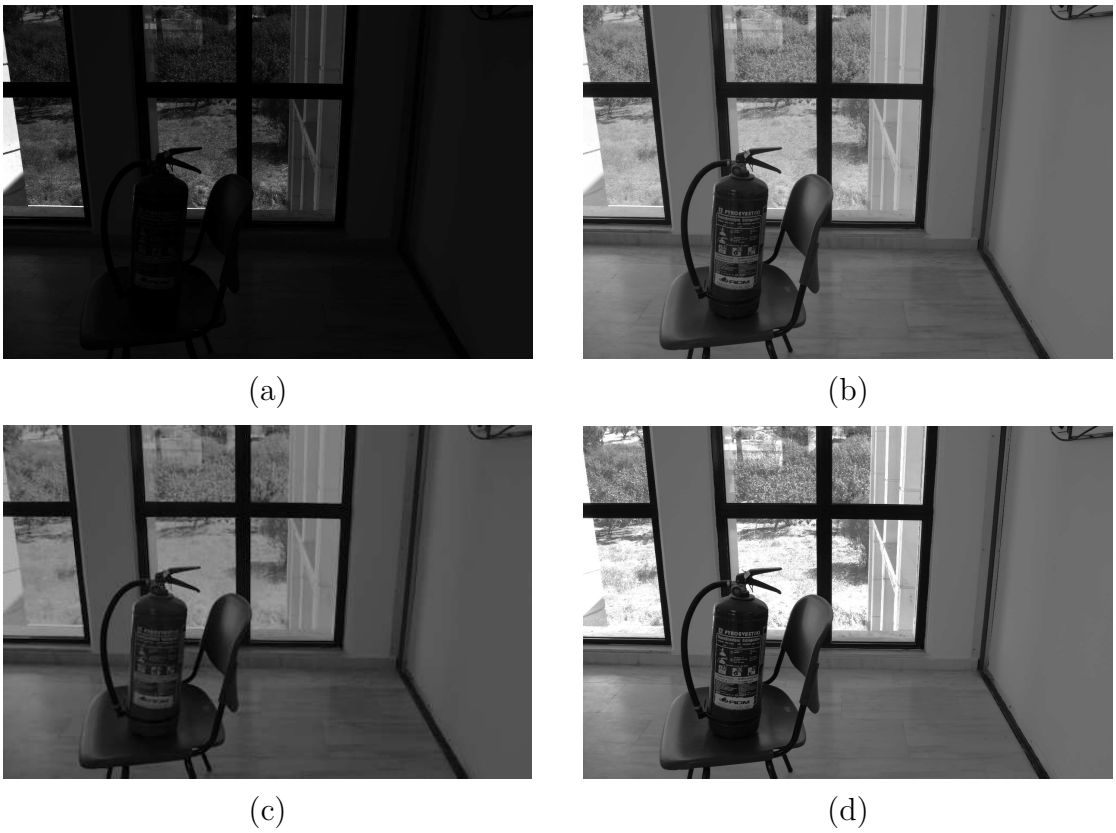


Figure 4.1: Example of an HDR image after (a) setting to zero the bits that are less significant than the eighth bit, (b) using a log tone-mapping curve, normalizing and clamping, (c) using the tone-mapping operator presented in [1], and (d) using the tone-mapping curve of the designed circuit.

depends on the pixel value and on the matrix average pixel value.

The use of a tone-mapping curve before converting an HDR signal to digital is also very beneficial [99]. Tone-mapping methods emphasize important segments (regions) of the image signal before the analog-to-digital conversion. Tone-mapping methods thus allow for the use of eight bits with less detail loss, which contributes to reduced bandwidth and storage requirements.

Tone-mapping operators may be classified as global or local. Global operators have mapping functions, with parameters that depend on image characteristics, which are applied to all pixels in the image. Local operators, on the other hand, depend on pixel neighborhood. Tone-mapping operator choice is typically application-dependent [25].

A common technique for capturing HDR is the use of multiple exposures. Shorter exposure times will benefit brighter areas, allowing for the capture of details in regions were pixels easily saturate. Longer exposure times, on the other hand, allow for the capture of details from darker areas, where pixels need more time to integrate. The images are then combined to generate a final image with extended

dynamic range. If the sensor response is linear and there is no movement in the scenes between the captures, generating the resulting HDR image can be as simple as normalizing each pixel value with its corresponding integration time, excluding the pixels under and over exposed, and computing the average image [98]. Most of the times, though, it is necessary to derive the camera response function to linearize the data before the normalization, treat ghost effects, which appear when the subject being captured moves, and align the images from different exposures, if the camera itself moves. After these corrections, tone mapping may also be necessary before displaying.

Although the multiple exposure approach results in good HDR image quality, the computational effort and necessary storage space may impose significant limitations. Besides, this approach is unsuitable for real-time applications due to the necessary time for the different captures and the necessary processing time. Extending the capture dynamic range directly inside the image sensor can be a way to overcome these drawbacks. This topic has been largely explored in the latest years. Operating the pixel in logarithmic mode, for example, is a simple technique that requires small pixel area and leads to 160 dB dynamic range [28]. Figure 4.2 shows the schematic diagram of this pixel. It consists of a classic 3T pixel whose reset transistor gate is connected to the drain during the entire operation. As a consequence, reset transistor drain current is the generated photocurrent, which forces subthreshold operation. In the subthreshold mode, the source voltage of the reset transistor will be proportional to the logarithm of the photocurrent. The response is thus a non-linear logarithmic function which will increase the contrast of the darker areas and reduce the luminance dynamic range at the brighter areas of the image, thus extending the dynamic range towards the brighter values.

The disadvantage of this pixel is its high FPN (fixed-pattern noise). Classic techniques to reduce noise, such as CDS (correlated double sampling), cannot be

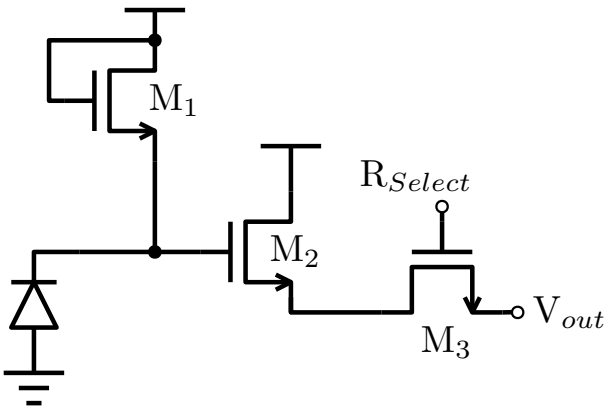


Figure 4.2: Schematic diagram of the logarithmic pixel.

used in this pixel because of the non-linear response, causing sensitivity loss towards low light [29]. The logarithmic pixel also has low signal swing, resulting in low SNR. This pixel operates continuously, having no integration period. In order to read the pixel it is necessary to wait for it to reach the logarithmic steady state, which depends on the amount of light. The settling time is long, thus resulting in low frame rate [28].

To take advantage of the benefits of the logarithmic pixel, but increasing sensitivity in the dark areas, it is possible to combine the linear operation of the regular 3T pixel, which is suitable for dark image regions, with the logarithmic operation of the log pixel, which is suitable for bright image regions [28]. The dynamic range is thus extended toward bright, but the low-light sensitivity is similar to the linear pixel sensitivity. This pixel starts operating in the linear mode and switches to log mode when the discharge voltage curve reaches a defined voltage level.

A technique to reduce FPN in the multi-mode linear-log pixel is presented in [28]. In this technique, the output signal and a reference signal, used to set a low voltage in the photodiode through the reset transistor, are sampled. Readout noise will be present in both samples. Thus, subtracting the samples will suppress part of the FPN. This pixel has a dynamic range of 111 dB before correction and of 115 dB after FPN correction.

Dynamic range may also be increased by using multiple gains for pixel value readout [3]. The device shown in [100] uses two column amplifiers: a high-gain amplifier and a low-gain amplifier. Both amplifiers work at the same time and CDS is performed for each case [100]. The higher gain amplifier has low readout noise, extending the dynamic range towards the lower signals. The low gain amplifier, on the other hand, captures signals related to larger local luminance values. For those large luminance values, the high-gain amplifier is expected to yield the same saturated (constant) output voltage. The final image is generated by combining the result of the high gain amplifier with the result of the low gain amplifier. The disadvantages in this case are the small increase in the dynamic range and the necessary post-processing.

Adjusting the capacity of the pixel to store the photo-generated charge is another way to capture HDR images. With this technique we create a multi-segment, or piecewise linear, pixel response [26]. A possible implementation for this method is presented in [30]: a lateral overflow integration capacitor is used to increase the capacitance of the floating diffusion node, storing charge values that exceed the floating-diffusion node capacity. This imager has a dynamic range of 93 dB.

An HDR image may also be captured by measuring the amount of time it takes for the pixel readout node voltage to reach a reference voltage level. A common way to address this problem is to count the number of times a pixel reaches saturation

during a period of time [101],[102]. Dynamic range is increased because the pixel that reaches saturation is reset and allowed to integrate again, thus better exploiting the full well capacity. The main problem of this technique lies on the implementation of the event counter, since counting inside the pixel would require large amount of area and counting outside the pixel creates a synchronization problem on how to send all the information to the hardware outside the pixel matrix.

The multiple capture technique uses global integration control [29]. Another possibility is to consider the scene illumination to adapt the integration time according to the exposure conditions. The integration time is a function of global variables, such as average luminance value or image histogram, and it is applied to all the pixels of the matrix [32]. A tone-mapping technique with content-aware compression is presented in [32]. A global operator is used, with the same function being used for every pixel. The image histogram is used to define the tone mapping function. This chip achieves a dynamic range of 151 dB, and only seven bits per pixel are necessary.

Autonomous integration time control is a technique in which each pixel is responsible for controlling its own exposure time, regardless of the other pixels integration time. The idea is to let the pixel choose its own integration time depending on the amount of light it receives. Dark pixels require more integration than brighter pixels. Hence, the dynamic range increases for as long as the integration time of the darkest pixels can be increased and the integration time of the brightest pixels can be reduced. In [103] an example of pixel with autonomous control over the integration time is presented. In this case the value of the pixel is compared at certain point of the integration time, which defines whether the pixel will saturate before the integration time. If it is estimated that it will saturate, then the pixel auto-resets. All pixel values are available at the same time at the end of the maximum integration time, but those that reset the integration have a smaller effective integration time. This circuit is able to reach 97 dB dynamic range.

In the circuit proposed in [33], the pixel controls its integration time based on its own luminance and the matrix average luminance value. This circuit was fabricated using a standard technology. In this work we introduce the design of pixels using the idea proposed in [33], but with the UMC 180 CIS technology. Since the technology is designed for image sensors it has improved sensitivity, reduced noise and the possibility of using the pinned photodiode, color filters and microlenses. The inclusion of microlenses is the only feature we do not explore in the present work. Thirteen matrices with 64×64 pixels were designed to study variations on the pixel proposed in [33] and to compare those variations with the regular 4T pixel. Section 4.1 explains the proposed pixel circuit and shows system-level simulations. Several test matrices and the differences among them are presented in Section 5.1.

4.1 Pixel for HDR Capture

In the proposed HDR scheme, the integration time of each pixel varies according to its own luminance and to the average luminance of the matrix. Pixels from darker areas will have more time to integrate and pixels in brighter areas will stop the integration sooner, ideally before reaching saturation. A detailed analysis of the proposed scheme is presented in this chapter. The next sections show the proposed HDR pixel schematic diagram, the resulting tone-mapping equation and system-level simulations.

4.1.1 Pixel Schematic Description

The schematic diagram of the proposed pixel is presented in Figure 4.3. The pixel has two photodiodes: one photodiode for the very image capture, and one photodiode for integration time control. We assume that the luminance signal at the control photodiode is approximately equal to the luminance signal at the closest adjacent capture photodiode.

The pixel circuit can be divided in two parts. The first part is the capture circuit, which is composed by a capture photodiode, ph, a reset transistor, M_2, a source-following amplifying stage (henceforth simply referred to as "source follower"), M_3, a row-select n-channel switch, M_4, and a transfer gate, M_1. Although this part of the pixel has the same structure of the usual 4T pixel, its operation is slightly different because the transfer gate is kept on during the entire integration period. The transfer gate is turned off only when the second part of the circuit (i.e. the control circuit) determines so. The consequence of operating the circuit in this way will be discussed later on in this chapter. Since the second part of the circuit controls the integration period, it will be called control circuit. It is composed by a second photodiode, ph_{ctrl}, transfer gate, M_5, and reset transistor, M_6, and also by two switches, s_1 and s_2, that connect the floating diffusion of the pixel control circuit to its neighbor from the right and from the bottom, and a digital circuit, with two inverters and a NOR logic gate, which turns the capture circuit transfer gate on or off.

Analyzing the logic of the digital circuit, it can be seen that it could be simplified using a NAND gate and a single inverter, but this simplification was not done because inverter I_1 acts as a comparator, to stop capture photodiode integration, when the control floating diffusion voltage reaches the middle of its own dynamic range. To yield more accurate control over the comparison threshold, it was decided to implement the desired logic with the two inverters and the NOR gate. The control circuit integration period begins with s_1 and s_2 turned on, thus connecting the entire matrix controlling circuitry. Considering that the control pixel floating

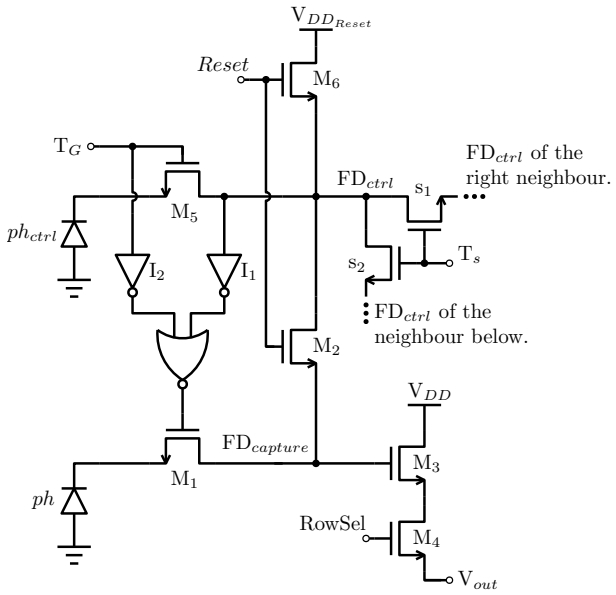


Figure 4.3: Schematic diagram of the studied pixel for HDR image capture.

diffusion nodes have approximately the same capacitance, the FD_{ctrl} discharge rate is proportional to the matrix controlling node (i.e. averaging node) discharge rate. Afterwards, s_1 and s_2 are turned off, and FD_{ctrl} starts to discharge according to its own luminance signal. When FD_{ctrl} reaches the middle of I_1 operating range, V_m, the NOR logic gate output goes to zero, thus turning M_1 off and stopping the capture circuit integration. Consequently, each pixel controls its own integration time according to the matrix average luminance value and to the local luminance value. A brighter pixel will reach V_m before a darker pixel, so the brighter pixels will integrate for less time than the darker pixels.

The pixel controlling signals and floating diffusion node timing diagram based on system level simulations are shown in Figure 4.4. The controlling signals are presented in dotted lines. The labels are placed at the instant in which each control signal is turned off. The solid and dashed lines represent the floating diffusion node voltages during the pixel operation for a dark (green and blue lines) and a bright pixel (black and red lines). The difference between the solid and dashed lines is the instant that T_s is turned off. Circuit operation begins with photodiode and floating diffusion reset. To do that, *Reset*, T_G and T_s are turned on. To understand what happens when these switches are closed, we consider a patent that presents a partially pinned photodiode [2]. The partially pinned photodiode cross section, reproduced from [2], is presented in Figure 4.5(a). In this cross section, we see a photodiode that has a pinned region and a floating unpinned region. The unpinned region works like a floating diffusion without a nearby transfer gate. The partially pinned photodiode operation is similar to the operation of the pixel in this

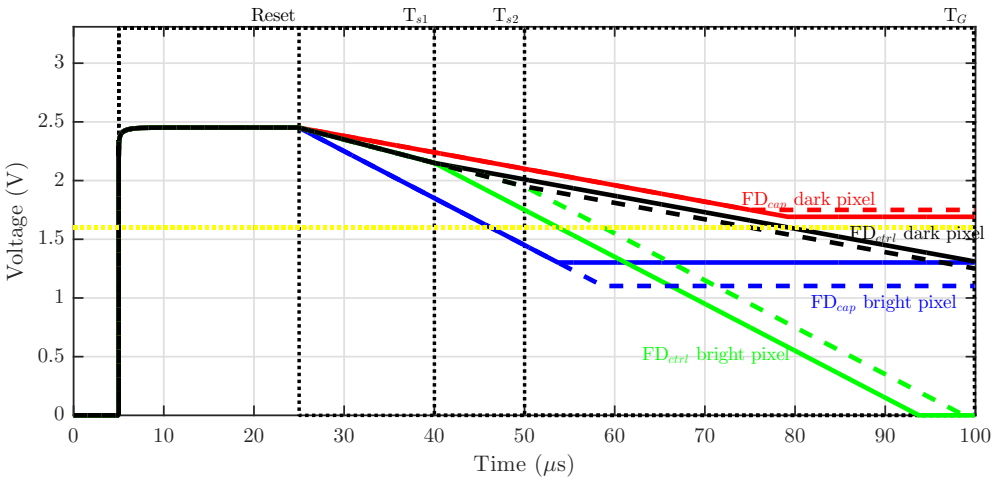


Figure 4.4: Pixel operation timing diagram. In solid and dashed lines, control (black and green) and capture (red and blue) floating diffusion nodes discharge. Discharge in the floating diffusion nodes for a pixel above (black and red lines) and a pixel below (green and blue lines) the average are considered. In dotted line, control signals. When T_{s1} is used, the floating diffusion nodes behave as presented in solid lines. When T_{s2} is used, the floating diffusion nodes behave as presented in dashed lines.

work, because the photodiode is constantly connected to the floating diffusion. The electrostatic potential analysis of this component shows that the pinned photodiode resets at the pinning voltage while the floating diffusion may reach a voltage level close to V_{reset} [2]. Furthermore, the pixel will have two discharge curves, as a consequence of the capacitance difference between the photodiode and the floating diffusion. The partially pinned photodiode response is presented in Figure 4.5(b). Figures 4.5(c) and 4.5(d) show the electrostatic potential of this component before and after reaching the pinning voltage. The electrons generated by the photoelectric effect are first stored only in the unpinned region. When the pinning voltage is reached, both the unpinned region and the and the pinned region store electrons. The capacitance thus increases, resulting in the second segment of the component response. We expect that the reset operation in the present work HDR pixel be similar to the partially pinned photodiode reset operation, and that the similarity also holds for their voltage discharge curves. For now, we also assume that the photodiode and the floating diffusion reset voltages are the same, and that the pixel discharge is linear (i.e. the pixel voltage decreases according to a ramp).

After the reset, T_s is kept on during a period that is adjusted by the user. While T_s is on, s_1 and s_2 are on, all control pixel floating diffusion nodes are connected and the discharge will depend on the photocurrents of every control photodiode. The overall photocurrent can be estimated as the sum of all pixels photocurrents and the equivalent capacitance as the sum of all pixels capacitance. Thus, the voltage

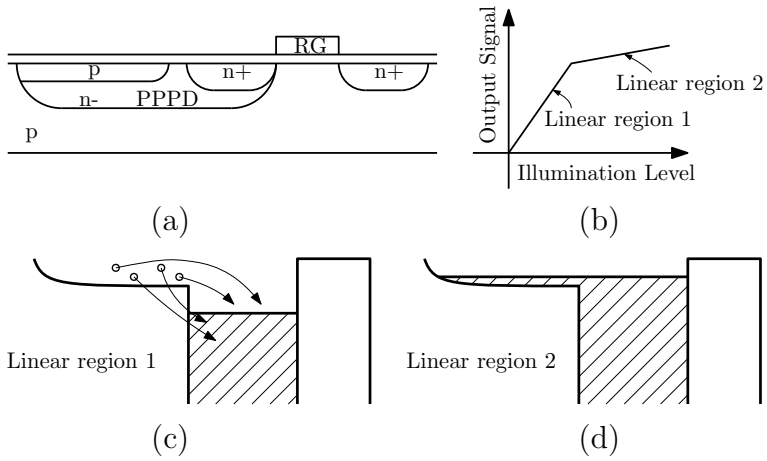


Figure 4.5: Partially pinned photodiode, reproduced from [2], (a) cross section, (b) response curve, and electrostatic potential considering two cases: (c) before reaching the pinning voltage, first linear response, and (d) after reaching the pinning voltage, second linear response.

discharge rate is proportional to the average luminance value. In a single pixel, the equivalent photocurrent can be estimated as $\sum_{\forall_{i,j}} I_{ph_{ij}}/M \times N = \bar{\mathrm{I}}_{ph}$. T_s is then turned off, thus changing the voltage discharge rate according to the pixel local luminance value. As can be seen in Figure 4.4, the integration stops when FD_{ctrl} reaches V_m, which is shown in the figure as the horizontal dotted line. In order to guarantee a minimum acceptable frame rate, we use another control, T_G, to stop integration for pixels whose local capture node voltage has not reached V_m. T_G controls the maximum integration period, T_{Max}. Figure 4.4 shows two possible T_s values which affect the final pixel voltage. If we change from T_{s1} to T_{s2} (increasing T_s), the operation is given by the dashed lines. As it can be seen in the figure, the bright pixels will integrate more and the dark pixels will integrate less when increasing T_s.

The control floating diffusion presents a non-linear response because it is composed of two segments of different voltage discharge rate: the average discharge rate and, after T_s, the local discharge rate. The capture photodiode, on the other hand, ideally presents a linear response, which depends on the local discharge rate. Nonetheless, the tone-mapping curve is non-linear. It is determined by the equations that describe circuit behavior. The capture floating diffusion voltage at a determined time t is given by:

$$\mathrm{V}_{FD_{capture}}(t) = \mathrm{V}_{rst} - \frac{t \cdot \mathrm{I}_{ph}}{C}, \tag{4.2}$$

where V_{rst} is the voltage of the floating diffusion right after the reset signal, I_{ph}

is the capture photocurrent of the analyzed pixel, and C is the total capacitance of the floating diffusion capture node. The floating diffusion capture node overall capacitance depends on the capture photodiode capacitance and on the parasitic capacitance of the node.

The control floating diffusion discharge rate, on the other hand, is determined by two slopes (one slope before T_s and one slope after T_s). As explained in the beginning of this section, it is considered that the luminance signal at the capture photodiode is equal to the control photodiode luminance signal. If the capture and control photodiodes have the same area, then the same photocurrent is generated. It is important, though, to consider what happens if the area of the control photodiode changes. It may lead to the pixel area reduction and fill factor increase. If the control photodiode area changes, then the photocurrent changes proportionally. The constant m_{ph} is used to model the difference between the control and capture node photocurrents. The difference exists only if the photodiode areas are different, which is represented by $\mathrm{I}_{ph_{ctrl}} = \mathrm{I}_{ph}/m_{ph}$. Another consequence of changing the photodiode area is capacitance change. The ratio between the equivalent capacitance seeing by the capture floating diffusion node and the control floating diffusion node is modeled by m_C: $C_{ctrl} = C/m_C$. In this case the capacitance difference is given not just by the area difference, but also by the parasitic difference. Considering these constants, the control floating diffusion voltage decreases according to:

$$\mathrm{V}_{FD_{control}}(t) = \mathrm{V}_{rst} - \frac{\mathrm{T}_s \cdot \bar{\mathrm{I}}_{ph}/m_{ph}}{C/m_C} - \frac{(t - \mathrm{T}_s) \cdot \mathrm{I}_{ph}/m_{ph}}{C/m_C}, \tag{4.3}$$

where $\bar{\mathrm{I}}_{ph}$ is the matrix average photocurrent. As explained in the beginning of the section, it can be computed, for a matrix of size $M \times N$, as $\sum_{\forall_{i,j}} I_{ph_{ij}}/M \times N$. To find out the final floating diffusion voltage we substitute t, from Equation (4.2), by T_{mid}, which is the time instant at which the floating diffusion of the control circuit reaches the middle of its own dynamic range, $\mathrm{V}_{FD_{control}}(\mathrm{T}_{mid}) = \mathrm{V}_m$, or substituted by T_{max} if the maximum integration period is exceeded. Using Equation (4.3):

$$\mathrm{T}_{mid} = \frac{m_{ph}}{m_C}\frac{C}{\mathrm{I}_{ph}}(\mathrm{V}_{rst} - \mathrm{V}_m) + \mathrm{T}_s - \mathrm{T}_s \cdot \frac{\bar{\mathrm{I}}_{ph}}{\mathrm{I}_{ph}}. \tag{4.4}$$

If T_{mid} is greater than the maximum integration time, then the pixel integration time is equal to T_{max}, otherwise, it is equal to T_{mid}. Figure 4.4 shows examples of how T_{mid} changes according to the pixel value and to T_s. As it can be seen in the figure, when T_s increases, pixels that discharge faster than the average discharge rate (pixels brighter than the average) will have more time to integrate, and thus

will appear brighter, and pixels that discharge lower than the average (pixels darker than the average) will have less time to integrate, and thus will appear darker. This means that increasing T_s increases the contrast between pixels below the average pixel value and pixels above the average pixel value. This effect is shown in Section 4.1.2, which presents the results of system-level simulations based on mathematics computational software. For the proper operation of the circuit for HDR capture, T_s must be turned off before the control floating diffusion voltage reaches V_m. In other words:

$$T_{sMax} = (V_{rst} - V_m) \cdot \frac{(C/m_C)}{(\bar{I}_{ph}/m_{ph})}, \tag{4.5}$$

which is the time it takes for the voltage of the control floating diffusion to drop from V_{rst} to V_m with a discharge equivalent to the mean photocurrent. If $T_s \geq T_{sMax}$, then all pixels have the same integration time equal to T_{sMax}, so no HDR operation is performed.

The pixel transistor sizes from Figure 4.3 were determined through schematic diagram simulations and are presented in Table 4.1. Pixel area was a constraint, but minimum length transistors were not considered because simulations with a single transistor showed significant short channel effects. Mismatch simulations also influenced the transistor size choices. The threshold voltage of the reset transistors, for example, determines the photodiode and floating diffusion reset voltage. To reduce error caused by threshold voltage variation, CDS may be used. Although we are still using small channel transistors for the sake of area, we avoided using minimum size. One of the test matrices include only minimum transistors, which will allow for experimental comparison. The transfer gate transistor sizes (i.e. the sizes of M_1 and M_5) were defined by the technologies rules. The size of the capture photodiode for all test matrices (considering the "sensor" layer from the technology) is 3 μm $\times$ 3 μm. To investigate pixel pitch reduction, we consider control photodiodes with three different sizes. The consequences of photodiode size variations are presented in Section 4.1.2. To set V_m approximately at the middle of the control photodiode voltage range, the inverter p-channel transistor width is equal to three times the inverter n-channel transistor width. For the NOR gate there was no such care, since it is not connected to the sensing node.

4.1.2 System-Level Simulations

To test the proposed approach, the circuit Equations (4.2), (4.3), (4.4) and (4.5), were implemented using mathematic computational software. The script used for

Table 4.1: HDR pixel transistors sizes.

Transistor	W (μm)	L (μm)
M_1	1.4	0.8
M_2	0.5	0.5
M_3	1.0	0.5
M_4	0.5	0.5
M_5	1.4	0.8
M_6	0.5	0.5
s_1	0.5	0.5
s_2	0.5	0.5

simulation can be seen in Appendix A. Images from an HDR database were used as inputs [104]. The capture photocurrent for each pixel was estimated using the pixel values and an arbitrary constant expressed in amperes. The arbitrary constant represents a photodiode sensitivity factor. Given an input image with pixel values $\mathbf{P}$, the capture photocurrent matrix is equal to $\mathbf{I_{ph}} = K_{ph} \cdot \mathbf{P}$, were K_{ph} is the sensitivity factor, set to 50 pA, and $\mathbf{P}$ is the raw pixel matrix. Once the capture photocurrent matrix is found, the control photocurrent matrix is $\mathbf{I_{ph_{ctrl}}} = \mathbf{I_{ph}}/m_{ph}$ or similarly, $\mathbf{I_{ph_{ctrl}}} = K_{ph} \cdot \mathbf{P}/m_{ph}$. The capture node capacitance, C, also chosen arbitrarily, is equal to 20 fF. Although the sensitivity constant and the capacitance were chosen arbitrarily, they only affect the choice of T_s and T_{Max}. For each pair K_{ph}, C with a given T_s and T_{Max}, there is an equivalent pair K_{ph}, C which gives the same result with different T_s and T_{Max}. Since T_s and T_{Max} are variables that can be adjusted outside the chip, there is no problem if K_{ph} and C are not estimated correctly.

The reset voltage, $V_{rst} = 2.44$ V, was based on circuit simulations using Cadence Virtuoso. V_m depends on the inverter I_1. For an inverter with p-channel transistor width three times greater than the n-channel transistor width, V_m is approximately equal to 1.6 V.

For the system-level simulations, the T_s value is select from the [100 μs, T_{sMax}] interval. The constants that model the difference between the control and capture photodiodes were established as $m_{ph} = m_C = 1$. This definition considers that both photodiodes have the same size and thus same sensitivity and capacitance, but the definition ignores the parasitic effect on the total capacitance. Disregarding the parasitics is adequate. Parasitic capacitance values are often negligible in comparison to the capture and control photodiode capacitance values, which correspond to the largest capacitors at the capture and control nodes. Section 4.1.3 explores non-ideal pixel response for conditions under which the parasitic effects must be taken into account.

The maximum integration time, T_{max}, was set to 30 ms. Substituting these

values in Equation (4.4) and limiting the maximum integration time to T_{max}, we find integration periods for each pixel. Substituting the integration periods in the variable t from Equation (4.2), we have an estimated output voltage for each pixel, V_{HDR}. The V_{HDR} values are then normalized to the $[0, 1]$ interval, to allow for display: $p_{HDR} = (V_{rst} - V_{HDR})/(V_{rst} - V_{min})$, where V_{rst} is the maximum floating diffusion voltage that can be detected. The maximum detectable floating diffusion voltage corresponds to pixel value equal to zero. The minimum detectable voltage is V_{min}. It is approximately equal to 0.7 V. Pixels that reach V_{min} are fully discharged and correspond to pixel value equal to one.

To generate the images in Figure 4.6, we use the method that has just been described. The input image is a patch of the HDR image presented in Figure 4.7(a). Four different values of T_s were considered: starting with 1 ms in Figure 4.6(a) and going up to 4 ms in Figure 4.6(d), with 1 ms step. From the sequence of figures we can see how the contrast increases with T_s. As shown in Figure 4.4, for the pixels with an integration time smaller than T_{Max}, when T_s increases bright pixels have more time to integrate and dark pixels less time to integrate, increasing the contrast. Overall contrast increases, but the local contrast among darker pixels (e.g. inside the flowers, or inside the leaves) does not increase. The small differences that exist among these pixels are kept approximately the same. For those pixels, integration period is determined by T_{max}, and so T_s variations have little influence on them.

Figure 4.7 shows the same effect in a larger image. Figure 4.7(a) is the original

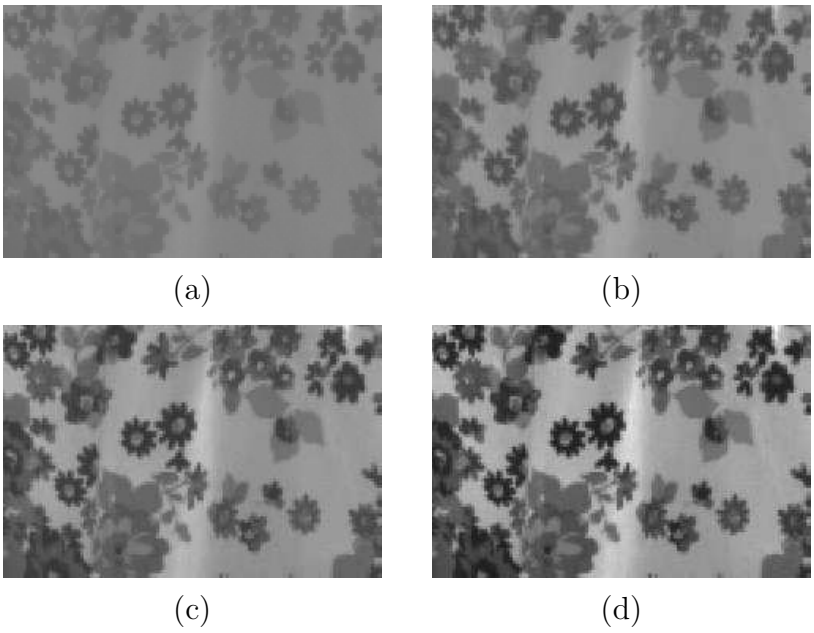


Figure 4.6: Tone-mapped image using the designed HDR pixel system-level model. T_s varies from (a) 1 ms to (d) 4 ms, with 1 ms step.

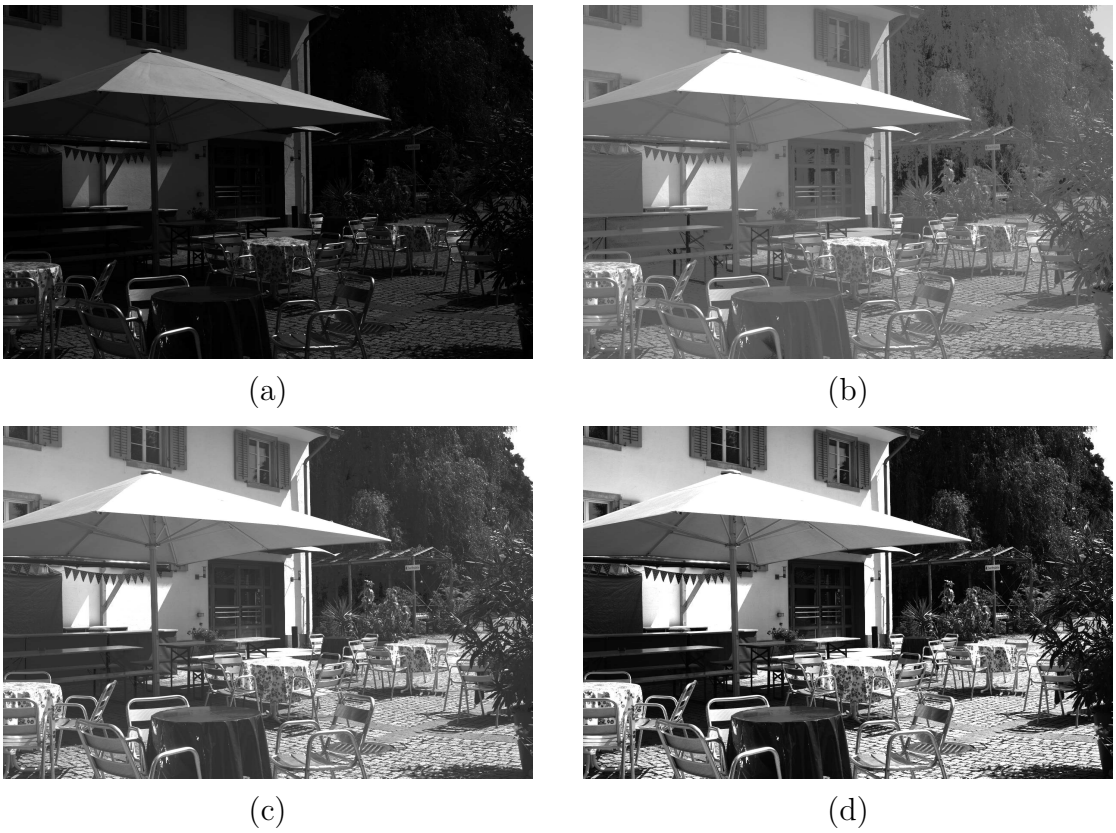


Figure 4.7: (a) Original image used as input for the system-level simulations, (b) image tone-mapped using $T_s = 1$ ms, (c) $T_s = 2$ ms, and (d) $T_s = 3$ ms.

image, without any correction. Figure 4.7(b) shows the result of a histogram correction of the image. Figures 4.7(d) until (f) show the results of the proposed HDR system-level simulation for increasing T_s values. In this figure we can see the disadvantage of defining a T_s too high, which is the saturation of bright areas, leading to detail loss in the white towel in the middle of the image.

It is interesting to see how the pixels from the raw image are being mapped to the final image. This analysis allows us to understand the tone-mapping function that is implemented by the proposed circuit. To do that, I_{ph} is replaced by $K_{ph} \cdot p$ in Equations (4.2) and (4.4). The new pixel value is obtained with $p_{HDR} = (V_{rst} - V_{HDR})/(V_{rst} - V_{min})$, where V_{HDR} is the result of Equation (4.2). Simplifying the equations we have:

$$T_{mid} = \frac{m_{ph}}{m_C} \frac{C}{K_{ph} \cdot p} (V_{rst} - V_m) + T_s - T_s \cdot \frac{\overline{p}}{p}, \tag{4.6}$$

where $\overline{p}$ is the mean of the pixel matrix. The integration period T_{int} is either equal to T_{mid} or equal to T_{Max}, and the final pixel value is:

$$p_{HDR} = \frac{\mathrm{T}_{int} \cdot K_{ph} \cdot p}{C \cdot (\mathrm{V}_{rst} - \mathrm{V}_{min})}. \tag{4.7}$$

Using Equations (4.6) and (4.7) we vary p from zero to one and plot the relation between p and p_{HDR} for different average pixel values and T_s values, as shown in Figures 4.8 and 4.9 respectively. These figures show that the tone-mapping function is piecewise linear. It is composed by two slopes and, depending on the conditions, a clamp. Furthermore, the output does not necessarily cover the entire eight-bit range. That happens because the circuit output voltage of the pixel whose discharge rate is equal to the average discharge rate is equal to V_m. In the system-level simulation, the pixel value that is equal to the average pixel value is mapped to the quantized eight-bit value that corresponds to V_m (which is approximately the center of the output eight-bit dynamic range, depending on the inverter I_1). Since V_m is not exactly at the middle of the voltage output range, it is equal to 1.6 V instead of $(\mathrm{V}_{rst} + \mathrm{V}_{min})/2 = 1.57$, the pixel value that is equal to the average pixel value is mapped to a tone-mapped pixel value slightly smaller than 0.5, as it can be seen in Figures 4.8 and 4.9. Changing V_m makes it possible to adjust the output range. A smaller V_m means that the average pixel value is mapped to a higher output value, which makes it possible to increase the output dynamic range of brighter (high-average) images. A possible circuit modification that would allow for V_m control consists in providing external (user) access to the inverter supply voltage. If the inverter supply voltage is less than 3.3 V, then V_m is reduced. An interesting circuit feature that can be seen in the figures is how the focal-plane tone-mapping operator adapts itself to make images with darker averages brighter and with brighter averages darker, not allowing for a saturated or dark result.

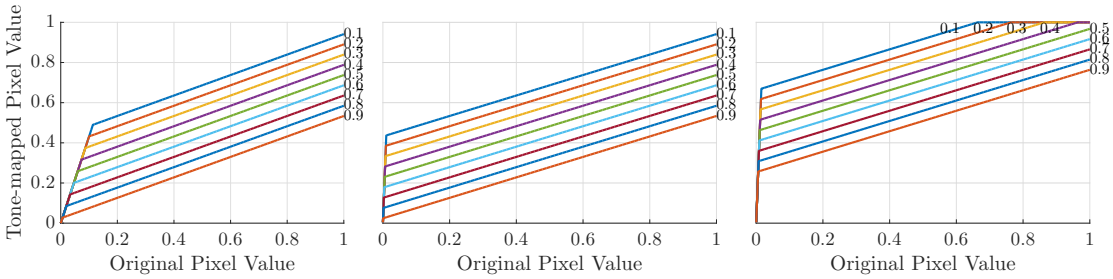


Figure 4.8: Tone-mapping functions for averages varying from 0.1 to 0.9 and (a) T_{Max} equal to 3 ms and V_m equal to 1.6 V, (b) T_{Max} equal to 30 ms and V_m equal to 1.6 V, and (c) T_{Max} equal to 3 ms and V_m equal to 1.2 V.

Substituting T_{int} in Equation (4.7), we can also see why arbitrarily choosing C and K_{ph} only affects the choice of T_s and T_{Max}. First, if $T_{mid} > T_{Max}$, then $T_{int} = T_{Max}$ and $p_{HDR} = (T_{Max} \cdot K \cdot p)/[C \cdot (V_{rst} - V_{min})$. If $T_{mid} < T_{Max}$, we substitute the result of Equation (4.6) in Equation (4.7):

$$p_{HDR} = \frac{V_{rst} - V_m}{V_{rst} - V_{min}} + \frac{K_{ph} \cdot p}{C} \cdot T_s \cdot (1 - \overline{p}/p). \tag{4.8}$$

As it can be seen from these equations, T_s or T_{Max}, depending on the final value of T_{mid}, multiply K_{ph}/C. Thus, even if K_{ph} and C are not equal to the values chosen, it is always theoretically possible to find the exact same response if T_s and T_{Max} change accordingly. The limitations are on generated signal precision, and on transfer gate control line charge time.

Figure 4.9 shows the effect of different T_s values for images with average pixel values equal to 0.2, 0.4, 0.6 and 0.8. T_s varies from $T_{sMax}/10$ to T_{sMax} with $T_{sMax}/10$ step, where T_{sMax} is given by Equation (4.5) and depends on the average value. As qualitatively observed before in Figures 4.6 and 4.7, T_s changes the contrast of the image. In Figure 4.9 we can see that this happens because T_s changes slope of the second segment of the tone-mapping function. Increasing T_s also increases the

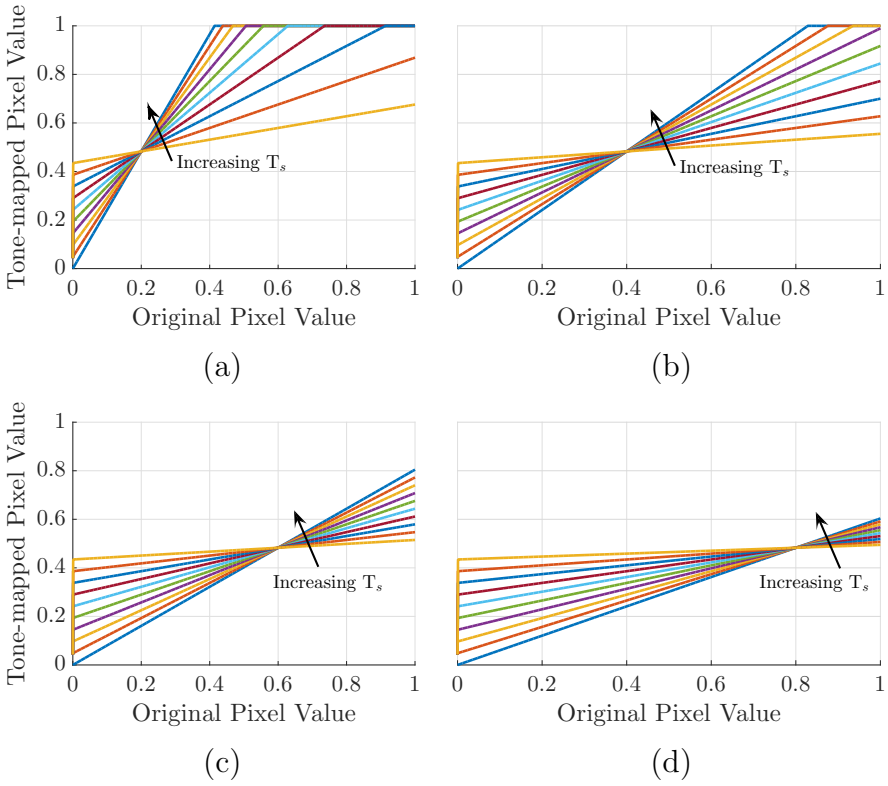


Figure 4.9: Tone-mapping functions for varying values of T_s considering averages equal to (a) 0.2, (b) 0.4, (c) 0.6 and (d) 0.8.

slope, resulting in the contrast enhancement. The idea of reducing V_m, observed in Figure 4.8(c), also has the advantage of allowing for contrast increase, since reducing V_m increases T_{sMax}, as it can be seen in Equation (4.5).

In Figures 4.8(a), 4.8(b) and 4.8(c), we consider images with average pixel values (at the input HDR domain) varying from 0.1 to 0.9. We also consider that T_s is the same for all images, i.e. for all average pixel values. T_s is set to 95% of the smallest T_{sMax}, which is the one computed for the highest average, 0.9. T_{Max} in Figure 4.8(b) is ten times larger than in 4.8(a). The effect of changing T_{Max} can be seen at the beginning of the response (i.e. at the leftmost part of the tone-mapping functions). With a smaller T_{Max} more pixels reach the maximum integration time limit, and will thus respond according to the leftmost segment of the piecewise linear function seen in Figure 4.8. In Figure 4.8(c) V_m was changed to 1.2 V. We can see in this figure that the curves are now higher, corresponding to a brighter image.

4.1.3 Non-Ideal Behavior

The operation of the pixel described in Section 4.1.2 does not consider some non-ideal effects that may appear in the fabricated circuit. Charge injection and clock feedthrough, for example, will affect the final pixel voltage value. Charge injection occurs because of the charge accumulated in the transistor channel, which flows to source and drain nodes when the transistor opens. Clock feedthrough, on the other hand, depends on parasitics between the switch control and the signal nodes. Although clock feedthrough can be reduced by avoiding signal and control line crosses in the layout, the transistor gate to source (or drain) capacitance couples these lines and affects the floating diffusion voltages. Although it is expected that they will appear in the fabricated circuit, these phenomena were not modeled for the system-level simulations.

In this section, we investigate two non-ideal effects and their impact on pixel tone-mapping functions: the non-linear discharge described in [2], and the floating diffusion parasitic capacitances effect.

Non-Linear Discharge

So far, we have considered that the floating diffusion and photodiode node capacitors discharge linearly. It is possible, though, for the floating diffusion node capacitance to discharge non-linearly (i.e. not according to a time-decreasing voltage ramp), when it keeps directly connected to the pinned photodiode [2]. This non-linear effect begins after the floating diffusion node voltage reaches the pinning voltage, because of the difference between the photodiode capacitance and the floating diffusion capacitance. Consequently, it affects the discharge of brighter pixels. The effect

on the tone-mapping curve can be seen in Figure 4.10 for three different average pixel values and for two different floating diffusion capacitances. The non-linearity is actually beneficial, because it may avoid the abrut clamp that exists at the original tone-mapping function. The bigger the total capacitance value with respect to the floating diffusion capacitance, the smaller is the response slope after the pinning voltage. Figure 4.11(b) shows this effect in an image. Compared to the ideal model, which is presented in Figure 4.11(a), Figure 4.11(b) shows the brighter regions not saturated: see for example the sky, the tent and the white towel, whose top is not completely white as in the ideal model result.

Floating Diffusion Parasitic Capacitances

Another non-ideal behaviour regards the effect of the floating diffusion parasitic capacitances in the pixel response. The overall capacitance seen by the floating diffusion node can be estimated as $C_{FD_{Total}} = C_{ph} + C_p$, where C_{ph} is the capacitance of the photodiode and C_p is the total parasitic capacitance. As can be seen in Figure 4.3, the floating diffusion of the control circuit is connected to more transistors than the floating diffusion of the capture circuit. The dominant parasitic capacitance of the capture circuit is given by the source follower gate capacitance, while the dominant parasitic capacitance of the control circuit is defined by the inverter input capacitance. Since the transistors of the inverter are bigger than the source follower, the parasitic capacitance of the control circuit is bigger than the capture circuit

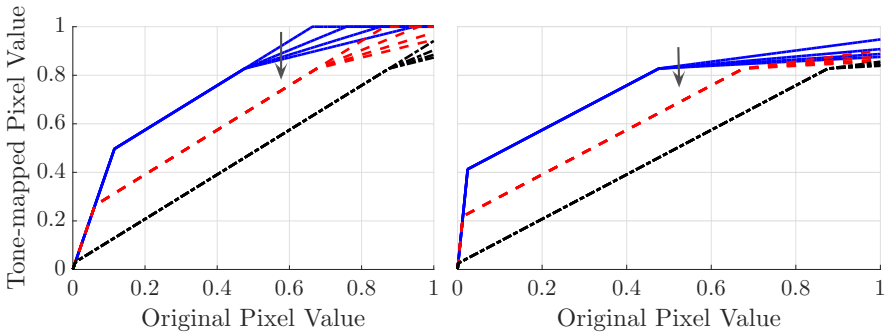
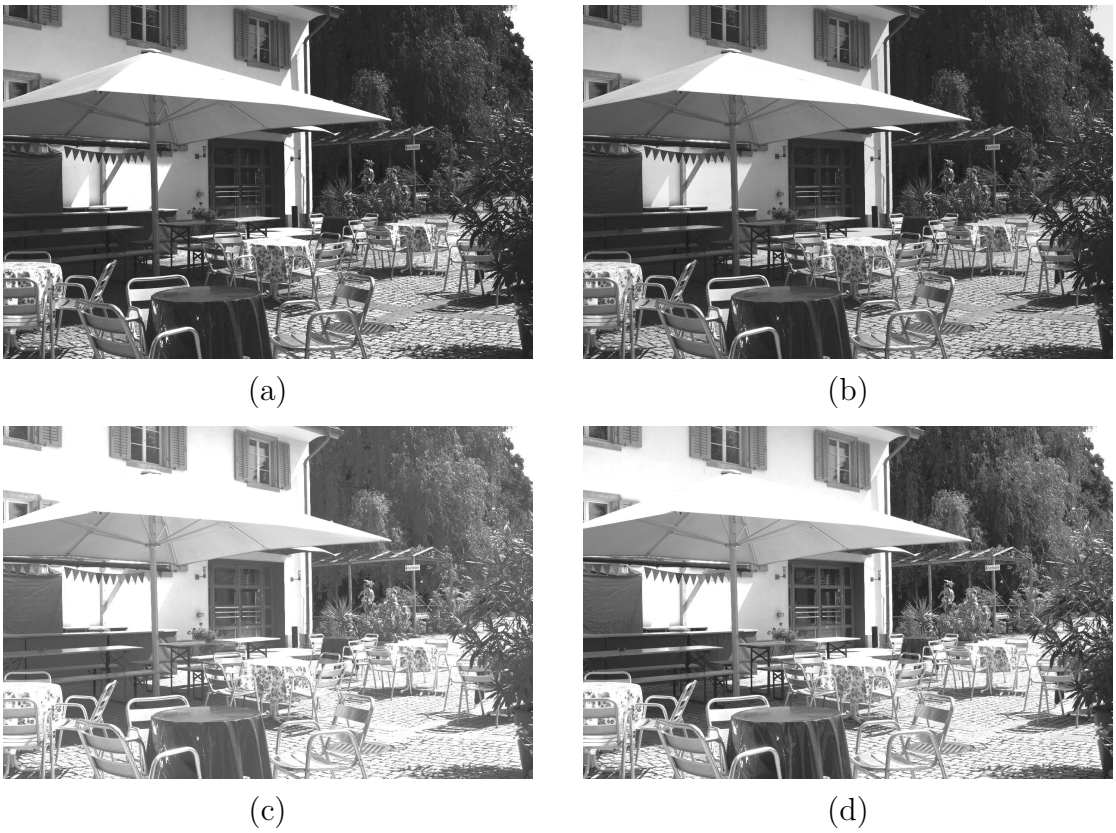


Figure 4.10: Tone-mapping functions considering the difference between photodiode and floating diffusion capacitances for input image average pixel values equal to 0.1, in solid line, 0.3, in dashed line, and 0.5, in dash-dotted line: (a) floating diffusion capacitance equal to 20 fF, and (b) floating diffusion capacitance equal to 5 fF. In either plot, the overall capacitance varies in the same way: from 20 fF to 50 fF with 10 fF step (for input values that are high enough for the pixel to reach output voltage equal to the pinning voltage). The arrow in the figure shows the direction in which the capacitance increases in the plotted curves.



(a) (b) (c) (d)

Figure 4.11: (a) ideal tone-mapping result; (b) tone-mapping result taking into account photodiode and floating diffusion capacitance difference; (c) tone-mapping result taking into account parasitic effects added to 20 fF nominal capacitance; (d) parasitic effects added to 80 fF nominal capacitance.

parasitic capacitance. Thus, if the capture photodiode has same size of the control photodiode, then $C_{ctrl} > C$ and $m_C < 1$. The smaller C_{ph} is, the more influence the parasitics will have in the result.

Schematic simulations of the circuit showed that the parasitic capacitance of the capture floating diffusion is equal to 8.8 fF and the parasitic capacitance of the control floating diffusion is equal to 15.4 fF. For a photodiode capacitance equal to 20 fF, we have $C \approx 28.8$ fF, $C_{ctrl} \approx 35.4$ fF and $m_C = C/C_{ctrl} \approx 0.8$. Figure 4.12(a) shows how the pixel response changes considering these parasitics, considering input image average pixel value equal to 0.4, and varying the photodiode capacitance in the same way for both capture and control circuits. The dashed line shows the response curve without the parasitic effect. Reducing the photodiode capacitance also reduces m_C, which has the effect of stirring the response curve upward and to the left. This is the same effect that was observed in Figure 4.8, when V_m changes. The tone-mapped image will thus appear brighter.

Figure 4.12(b) combines both the parasitic effect and the non-linear response discharge. It is considered that capture and control photodiodes have same size. The

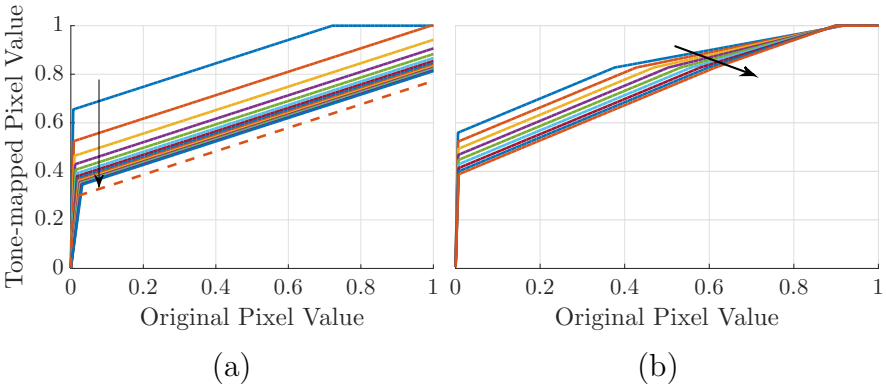


Figure 4.12: Pixel response considering input image average pixel value equal to 0.4: (a) in dashed line, disregarding the effect of different parasitics and with $m_C = 1$, and, in solid line, considering the effect of different parasitic in the capture and control floating diffusion nodes, and varying nominal photodiode capacitance (from 0 fF - just parasitics influend - to 70 fF with 6 fF step). The arrow represents the increase of the photodiode capacitance. The bigger the capacitance, the smaller is the parasitic effect and closer to the dashed response. (b) considering two non-linearity effects, the parasitic and the non-linear discharge for capture and control photodiodes of same size. The arrow represents the increase in the capacitance component (from 0fF to 8 fF with 1 fF step) that is equal to both control and capture floating diffusion nodes.

capacitance after the pinning voltage is fixed and the floating diffusion capacitance computed considering a fixed parasitic value and a variable component that changes in the same way for both photodiodes. By increasing this common component, both effects become less evident because the parasitic become less important and the floating diffusion capacitance become closer to the capacitance after the pinning voltage.

Figure 4.11(c) and (d) show the effect of introducing the parasitic capacitance in the system-level simulations. A 20 fF nominal capacitor is used in Figure 4.11(c), thus making parasitic effects more evident than they are in Figure 4.11(d), in which a 80 fF capacitor was used, when compared to the ideal model. As expected, the parasitic effects make the image brighter.

To allow for fill factor increase, it is interesting to reduce control photodiode area. If the area are reduced by a factor equal to m, both the sensitivity and photodiode capacitance reduce by the same factor. Thus, if the parasitic effect is not considered, there is no change in the pixel response because m_{ph} remains equal to m_C. Considering the parasitic effect, if the area of the control photodiode is equal to one half of the capture photodiode area, then $m_{ph} = 2$ and $C_{ctrl} = C_{ph}/2 + C_p$, which leads to $m_C = C_{ph}/(C_{ph}/2 + C_p)$. If $C_{ph}/2 >> C_p$, $m_C \approx 2$. The larger m is, the more evident will be the parasitic effect. In the chip design all capture photodiodes

have 3 μm×3 μm, but three different control photodiodes were implemented, 3 μm × 3 μm, $m = 1$, 2 μm × 2 μm, $m = 2.25$, and 1 μm × 1 μm, $m = 9$, each in a different test matrix.

Figure 4.13 shows how the parasitic effects change the tone-mapping functions. In this figure, T_{Max} was set arbitrarily high to allow for the analysis of the result without its influence. In the top row of this figure the input image average pixel value is equal to 0.2and in the bottom row it is equal to 0.8. In Figures 4.13(a), (b) and (c), C_{ph} is 20 fF, 50 fF and 80 fF respectively. The results in Figure 4.13(a) are more sensitive to the parasitics and to changing the control photodiode area. In solid line $m = 1$, in dashed line $m = 2.25$ and in dash-dotted line $m = 9$. This figure shows that, depending on photodiode capacitance and parasitic effects, the pixel tone-mapping function saturates more easily if the control photodiode area is significantly smaller than the capture photodiode area. There is thus a limit for the control photodiode to be reduced. Nonetheless, since we have no information on the photodiode capacitance and thus cannot determine this limit, a matrix with a control photodiode having 1/9 of its original area was laid out in the test chip, to investigate a possible fill-factor increase.

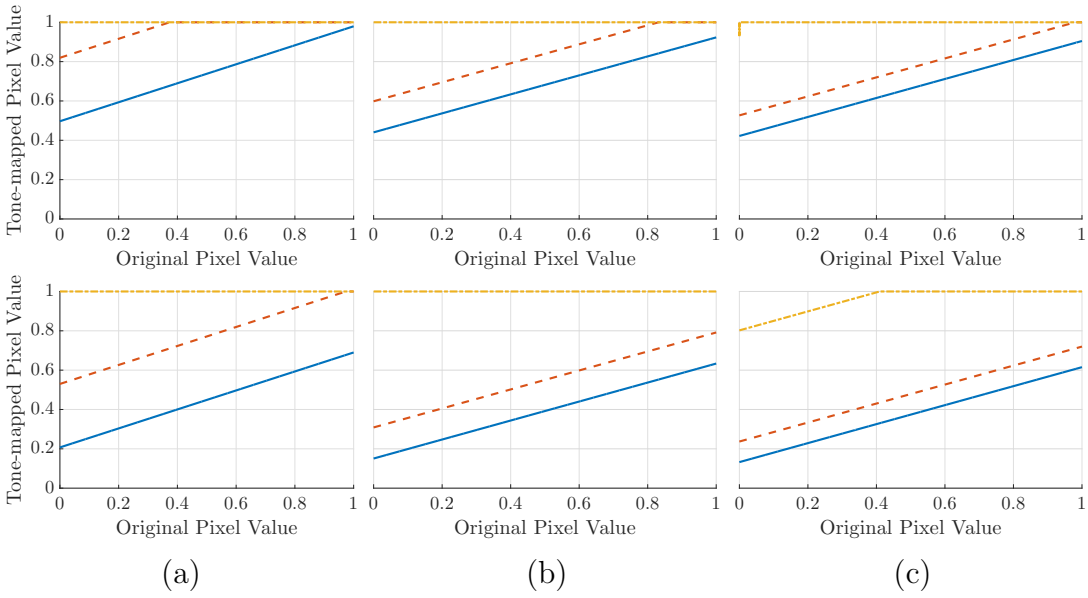
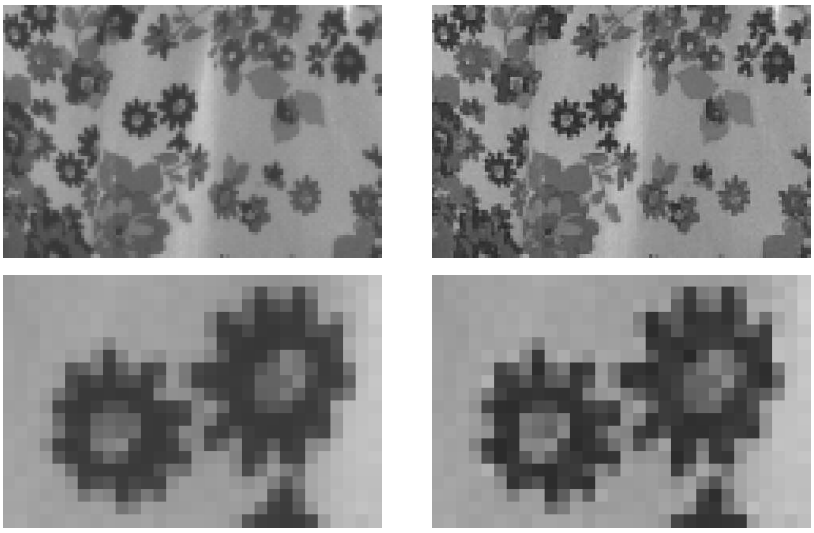


Figure 4.13: Pixel response curve considering parasitics and varying the photodiode size and capacitance. Photodiode capacitance equal to (a) 20 fF, (b) 50 fF, and (c) 80 fF. In the top row, the input image average pixel value is equal to 0.2. In the bottom row, it is equal to 0.8. In solid line both photodiodes have the same size. In dashed line the capture photodiode has area equal to 9 μm^2, and the control photodiode area is equal to 4 μm^2. In dash-dotted line, the capture photodiode area is equal to 9 μm^2, and the control photodiode area is equal to 1 μm^2.

4.2 Pixel with Shared Control Photodiode

Sharing the control photodiode and circuit among a group of capture photodiodes is a good solution for reducing the pixel area. The problem with this solution is that it might create artifacts on the image. We have considered in the previous analysis that the control and capture photodiodes have the same discharge rate. Although it is not expected that the discharge is exactly the same, this is a reasonable assumption, since the control and capture photodiodes are designed to be close to each other in the layout. If a single control photodiode is shared by a (possibly large) group of pixels, then it is no longer expected that all photodiodes will receive the same luminance input value. To minimize this effect, the control photodiode is shared for a small group of 2×2 pixels and the layout, which is shown in Section 5.1.1, was designed so that these photodiodes are close to each other.

To investigate possible artifacts, system-level simulations were performed considering that the shared control photodiode input is equal to the average luminance input of the 2×2 block. The result is shown in Figure 4.14(b). Figure 4.14(a) is included for comparison purposes. It is the regular model result. In the regular model, there is one control photodiode per pixel, and the capture and control photodiodes have the same discharge rate. Comparing the images at the top of Figures 4.14(a) and 4.14(b), top, we note that Figure 4.14(b) is more noisier. This observation is confirmed by comparing the images at the bottom of Figures 4.14(a) and 4.14(b)



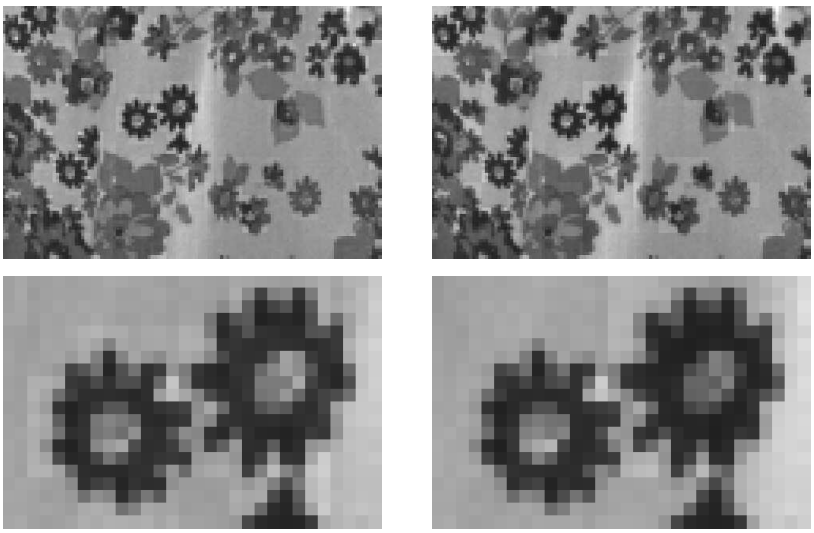
(a) (b)

Figure 4.14: (a) image tone-mapped using the regular pixel model, which has one control photodiode per pixel, and (b) image tone-mapped considering one control photodiode per 2×2 pixel block. The images at the bottom contain details from the images at the top.

bottom, which show a small detail of the top images.

Using Figure 4.14(a) as reference, Figure 4.14(b) presents an MSSIM (mean structural similarity) [105] equal to 0.97. The closest MSSIM index is to 1, more similar are the images. To reaffirm the MSSIM result, PSNR (peak signal-to-noise ratio) is equal to 33.4 dB. These results show that although there is noise introduced, quality of the image is similar to the reference of one control circuit per pixel.

Increasing the number of pixels that share the control photodiode makes the noise artifacts more evident. Figure 4.15 shows the results of 4×4 pixels blocks and 8×8 pixels blocks. MSSIM in these cases are 0.95 and 0.94, and PSNR are 29.6 dB and 28.1 dB. In Figure 4.15(b) we can clearly see blocking artifacts. These artifacts are also present in Figure 4.15(a): in the bottom image detail, the edges between the flowers and the background contain blocking artifacts. As the block size increases, artifacts appear in the image and layout becomes more and more complex. Sharing control photodiodes among 2×2 pixels corresponds to a suitable trade-off between noise and pixel are, because the noise effects are not as evident, the layout can guarantee that the pixels are placed close to each other, and the area reduction increases the pixel fill-factor.



(a) (b)

Figure 4.15: Blocking artifacts appear in the tone-mapped images, when a single control photodiode is shared by pixels within pixel blocks that are larger than 2×2: (a) 4×4 pixel block, and (b) 8×8 pixel block. The images at the bottom contain details from the images at the top.

4.3 Proposed Circuit for Color Implementation

The inclusion of color in imaging systems involves not only acquisition and compression but also white balance and demosaicing. White balance consists of a color adjustment procedure that aims at presenting color information as faithfully to the original scene as possible, by multiplying pixels with constants which depend on the camera response to each color and on the light source that is being used. White-balance constants are defined through experimental tests. For each camera, there exists a table with constant multiplying factors that are selected according to the estimated light conditions. White balance is usually performed before demosaicing. In the raw data yielded by the camera sensor, which is usually covered by a color filter array, each pixel contains information regarding a single color. Demosaicing is the procedure that maps the raw data (with incomplete color information) into three separate and complete color channels, by using neighboring pixels to estimate the missing color information at a given pixel.

In a purely digital tone-mapping scheme, white balance is usually performed right after image capture, and it is immediately followed by demosaicing. Depending on the tone-mapping operator, the tone-mapping procedure is either independently performed in each color channel, or it is performed in the luminance channel that is computed from the demosaiced RGB (red, green, blue color model) image. In the luminance-channel case, each color pixel is then linearly adjusted depending on the tone-mapped luminance, on the original luminance, and on the color pixel value itself. To show the tone-mapping result without taking into account the focal-plane complexity constraints and non-ideal effects, Figure 4.16(a) was used as input for a purely digital implementation of the proposed tone-mapping operator. In this implementation we first apply tone mapping to reduce the luminance dynamic range, and then adjust the color using white balance. The result is shown in Figure 4.16(b).

In the circuit implementation, because of complexity constrains, white balance and demosaicing are not performed at the focal plane. Differently from the digital approach, at the focal plane approach the white balance and demosaicing operations are performed after tone mapping. To include color in the proposed focal-plane tone-mapping circuit, we consider two approaches. These approaches were also tested using system-level simulations to guarantee the viability of the implementations [106]: i) The first approach consists in simply including Bayer pattern color filters [5] in the same circuit of the gray scale implementation, which was presented in Figure 4.3. This implementation is equivalent to tone-mapping each channel independently, using the matrix global average pixel value as a common parameter. Since white balance is performed after tone mapping, it is necessary to evaluate the effect of the tone-mapping curve in this operation. De-

Figure 4.16: (a) input RGB image, (b) digital implementation of the proposed tone-mapping operator (luminance channel only), (c) system-level focal-plane simulation with each pixel controlling its own integration period (tone mapping in all three channels), and (d) system-level focal-plane simulation with green pixels controlling blue and red channel tone mapping.

tails of this evaluation is presented in [107]. In this reference we can see that if tone-mapping is applied in each channel after white balance we have that the tone mapped pixel value is equal to $p_{HDR} = [\mathrm{T}_{int} \cdot K_{ph} \cdot WB \cdot p]/[C \cdot (\mathrm{V}_{rst} - \mathrm{V}_{min})]$, as presented in Equation (4.7), but including the white-balance constant WB, which depends on the pixel color and ilumination. Because of the non-linear pixel response described in Section 4.1.2, it is not possible to simply use WB after the tone-mapping. The new white balance factor can be found by resolving $[\mathrm{T}_{int} \cdot K_{ph} \cdot WB \cdot p]/[C \cdot (\mathrm{V}_{rst} - \mathrm{V}_{min})] = WB' \cdot [\mathrm{T}'_{int} \cdot K_{ph} \cdot WB \cdot p]/[C \cdot (\mathrm{V}_{rst} - \mathrm{V}_{min})]$, where WB' is the desired value that varies according to the pixel value, integration time of the pixel with tone-mapping before white-balance (T'_{int}) and integration time of the pixel with tone-mapping after white-balance (T_{int}). The integration periods can be computed using Equation (4.6), which depend on circuit constants (capacitace, sensitivity, V_{rst}), on the pixel value and matrix average value. Thus, it is necessary to invert the tone-mapping function to find a suitable multiplying factor for each pixel. Demosaicing is performed afterwards. The result can be seen

in Figure 4.16(c). The color difference between Figures 4.16(b) and 4.16(c) is a consequence of performing, in Figure 4.16(c), tone-mapping in each channel separately, which changes the, for each pixel, ratio between the color channels, and ii) the second approach uses the green information for red and blue channel tone mapping. The circuit necessary for this approach is presented in Figure 4.17. The green photodiode immediately below the red pixel controls both green and red capture circuits, and the green photodiode immediately above the blue pixel controls both green and blue capture circuits. This idea was proposed based on the digital implementation, in which tone mapping is not performed separately for each color channel. The green is used as reference, in this case, because of the Bayer pattern itself: where the green pixels are more abundant. Another motivation for this color implementation is area reduction. The previous implementation requires two photodiodes and control circuit for each pixel. When we use the green pixels to control the other colors pixels, we need three photodiodes and a single controlling circuit for every two pixels. The white balance for this case is simple: only the regular white-balance constants are necessary. This can be demonstrated by inverting the tone-mapping curve, considering the neighbor green as reference [107]. Figure 4.16(d) shows system-level

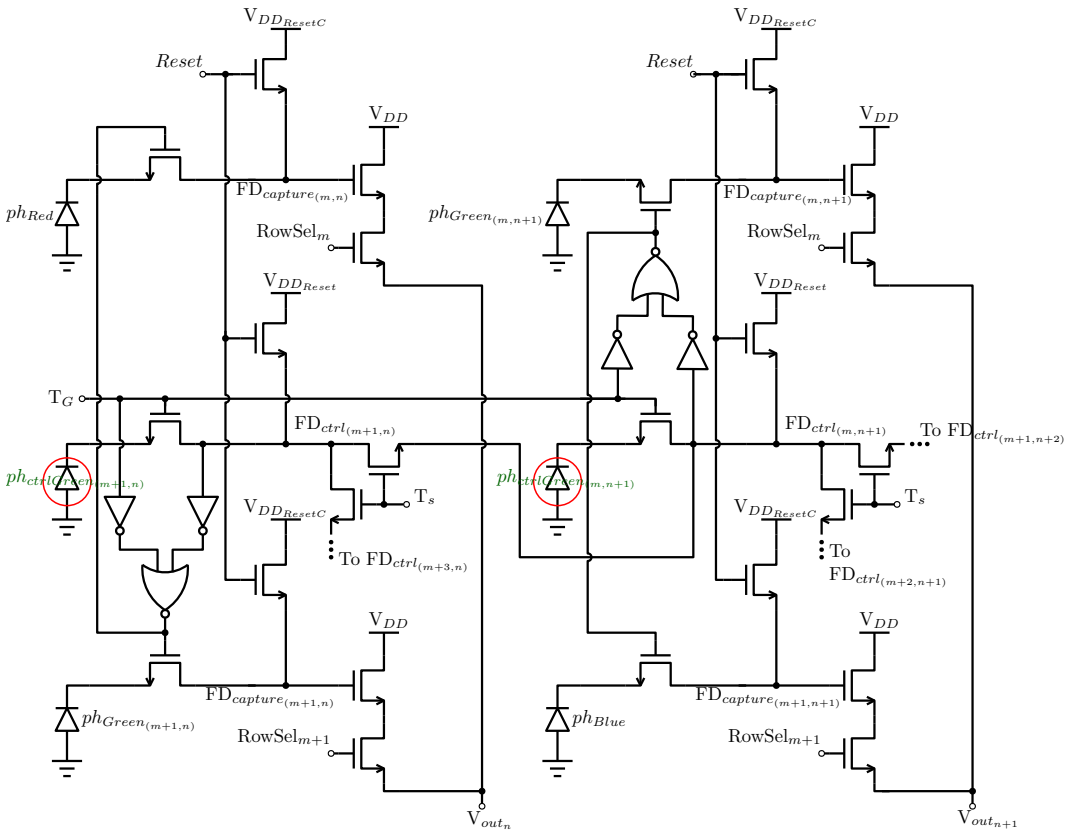


Figure 4.17: Schematic diagram of color 2×2 HDR pixel blocks in which the green pixels control the red and blue channel tone mapping. The circled photodiodes are the ones used for control.

simulation results for the green control approach. The color in this image is similar to the digital implementation because, on the contrary to the first approach, the second approach is able to preserve the ratio between the color channels for most of the pixels (if there is saturation in any of the color channels the ratio is not preserved). In either case, these results are subjectively good and validate the color circuit implementation.

4.4 Time Analysis of the Tone-Mapping Algorithm

The method proposed in Section 3.1.1 was used to compare the throughput of the designed HDR circuit with a hypothetical digital solution. The focal-plane solution steps for the HDR chip are capture, which occur concurrently with the tone mapping, and 8-bit analog-to-digital conversion. The time necessary for capture is equal to the maximum integration time, T_{Max}. T_{Max} is also expressed as a function of the clock period: $\mathrm{T}_{Max} = \mathrm{N}_{intMax}\tau_{Clk}$. The time for the analog to digital conversion is equal to the time of one conversion, $\tau_{ADC_{8bits}}$, multiplied by number of conversions, which is equal to the number of pixels M×N, and divided by the number of ADCs N_{ADC}. The focal-plane processing time is thus:

$$\tau_{FP_{Total}} = \mathrm{T}_{Max} + M \cdot N \cdot \tau_{ADC_{8bits}/N_{ADC}}. \tag{4.9}$$

For the digital solution, we consider that 12 bits are necessary to represent the value of the uncompressed pixel. The digital solution is limited by a single integration period. Consequently, details that are perceived in the HDR pixel because of the different integration periods in a single capture may not be perceived using the digital solution. This is another factor that needs to be assessed when comparing these solutions. This section focus only on the throughput comparison. Schematic simulations presented in Chapter 5 compare the details between the results of a regular 4T pixel matrix and the designed HDR matrix. Furthermore, future experimental tests with the fabricated chip, described in Chapter 5, will provide better insights in this matter. For the time comparison, the digital solution is composed by the capture step, 12-bit analog-to-digital conversion, memory write access and a digital processor which implements Equation (4.6), a comparison of the results with T_{Max} and Equation (4.7). The digital solution needs to access the memory for reading and writing one time for each pixel. The capture time was arbitrarily defined as half the maximum integration time from the focal-plane solution, assuming that it is the least necessary integration period to preserve the details, $\tau_{cap} = (\mathrm{N}_{intMax}/2)\tau_{Clk}$. For the analog-to-digital conversion, M×N pixels are converted by

a 12-bit converter whose conversion time takes $\tau_{ADC_{12bits}}$. The number of ADCs is also N_{ADC}. The processing time necessary to compute T_{mid} using Equation (4.6) is given by:

$$\tau_{Dig_{Tmid}} = M \cdot N \left(2\tau_{mult} + \tau_{div} + 3\tau_{sum}\right) + \tau_{div}, \tag{4.10}$$

where the division operation is necessary to compute $1/p$, which is multiplied and added in two parts of the equation. The third sum and the separate division are necessary to compute the matrix average. The M×N T_{mid} values are then compared with T_{Max}: $\tau_{Dig_{Tint}} = M \cdot N \cdot \tau_{comp}$. T_{int} is then substituted in Equation (4.7). The time necessary is $M \cdot N \cdot 2\tau_{mult}$, because of the multiplication between each T_{int} by its corresponding pixel value and by the constant given by the reset voltage, minimum voltage, photodiode sensitivity and capacitance. Memory access is necessary three times, to write the original value, read this value and write the tone-mapped value: $M \cdot N \cdot 3\tau_{mem}$. These four equations are divided by the number of processing elements, N_{PE}, to allow for parallel digital processing.

Summing all the equations, the total time the digital approach takes is given by:

$$\begin{aligned} \tau_{Dig_{Total}} = {} & T_{Max}/2 + M \cdot N\tau_{12bits} + \\ & \left(\tau_{div} + 3\tau_{sum} + \tau_{comp} + 3\tau_{mem}\right)/N_{PE} + \tau_{div} \end{aligned} \tag{4.11}$$

To compare Equations (4.9) and (4.11), the values from Table 4.2 were defined. The multiplication and division were defined as $3\tau_{Clk}$ and $4\tau_{Clk}$ to represent the fact that that these operations are more complex than a simple summing operation.

The ADCs constants from Table 3.2 were considered for the 8-bit case and were recomputed for the 12-bit case. The values for 12 bits were close to the 8 bits, as expected, since these constants define the clock ratios and should not change significantly with the number of bits. These 12-bit constants are: $K_{ramp} = 1$, also used as reference, and, using reported figures, found $K_{SAR,Cyclic} = 14$, $K_{\Sigma\Delta} = 13$

Table 4.2: HDR time analysis equations parameters.

Parameter	Value
M×N	640×320
Clock frequency	100 MHz
τ_{Clk}	10 ns
T_{Max}	1 ms
Time for a comparison operation (τ_{comp})	$1\tau_{Clk}$
Time for a summing operation (τ_{sum})	$1\tau_{Clk}$
Time for a multiplication operation (τ_{mult})	$3\tau_{Clk}$
Time for a division operation (τ_{div})	$4\tau_{Clk}$
Time for a memory access (τ_{mem})	$2\tau_{Clk}$
Number of ADCs (N_{ADC})	640

and $K_{Pipeline} = 2$. Figure 4.18 shows the ratio between the digital approach total processing time and the focal plane total processing time. Even for increasing values of number of processing elements, the focal-plane approach is approximately ten times faster than the digital approach with the ramp converter and five times faster with the $\Sigma\Delta$ converter. If we increase the digital processing clock frequency to 4 GHz, then the focal plane is nine times faster than the digital approach with the ramp converter and three times faster with the $\Sigma\Delta$ converter.

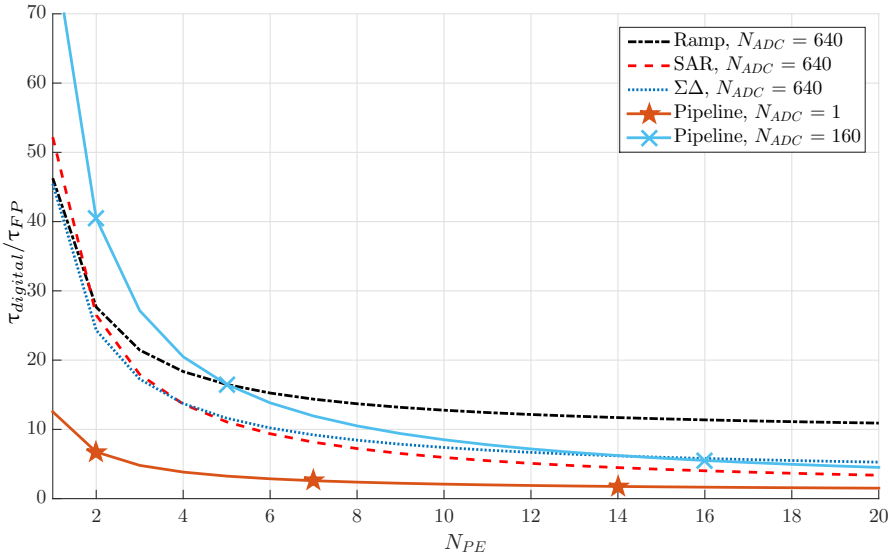


Figure 4.18: Ratio between digital and focal-plane tone-mapping processing times as a function of the number of PEs. Ramp, SAR, $\Sigma\Delta$, and pipeline ADCs are shown, respectively, in dash-dotted, dashed, dotted, and solid lines.

Chapter 5

HDR Chip Design and Simulation Results

The designed chip has 13 test matrices, each with 64×64 pixels. The test matrices and their layouts are described in Section 5.1. Pixels with control and capture photodiodes with different sizes, pixels that share the control photodiode, and color matrices were included. Regular 4T pixels were also included for comparison purposes.

Schematic and post-layout simulations using spectre models were performed, to evaluate the reliability of the proposed approach. A simulation with the complete number of pixels (each test matrix has 64 × 64 pixels) takes a few days, for a single schematic diagram simulation. To reduce the simulation time, most tests were performed using a 16×16 pixel matrix. These tests are presented in Sections 5.2 and 5.4.

5.1 Integrated Circuit Specifications

The 13 test matrices are organized in a higher level array of four rows by three columns of matrices and a single separate matrix. Figure 5.1 shows this higher level array, where the matrices are labeled according to their positions. From this figure, we can see that the rows of the matrices are connected together: that is, the row-select signal lines of matrix (1,1) are connected to matrix (1,2) and (1,3), row-select signal lines of matrix (2,1) are connected to matrix (2,2) and (2,3), and so on. Similarly, but regarding column connections, the outputs of matrix (1,1) are connected to the outputs of matrix (2,1), the outputs of matrix (1,2) are connected to the outputs of matrix (2,2), and the outputs of matrix (1,3) are connected to the outputs of matrix (2,3). These connections are mirrored at the two bottom matrices rows, but there is no connection between the outputs of matrices from the second

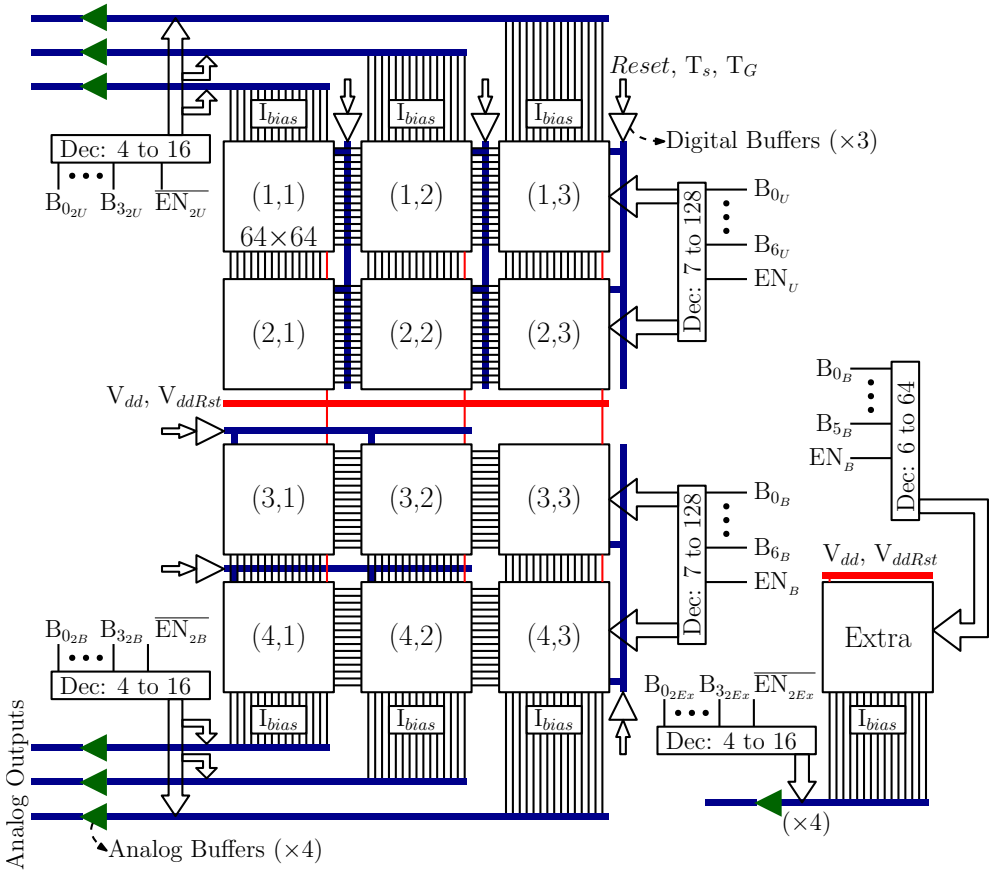


Figure 5.1: Chip block diagram.

and third rows. Two decoders with seven-bit inputs and 128 outputs are used to control the row-select signal lines: one for the two three-matrix rows at the top, and another one for the two three-matrix rows at the bottom.

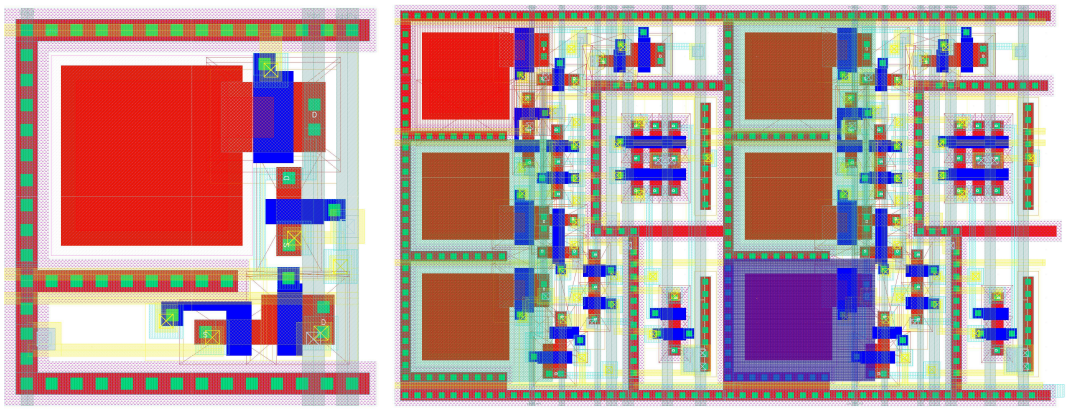
The extra matrix rows and columns are independent from the others, but the inputs of the decoder that controls the extra matrix row-select inputs are connected to the inputs of the decoder from matrices (3,1), (3,2) and (3,3). Each column output is connected to a current mirror with a bias current of 500 nA, leading to seven current mirror circuits, each with an input pin. The bias current choice was based on circuit power constraints per capture and on pixel simulations.

To connect the matrix outputs to the chip outputs (pads), analog buffers are used. There are four buffers for each set of 64 outputs, that is, one buffer for every group of 16 column outputs, and they are connected using simple switches. The buffer selection is done by a decoder with four inputs and 16 outputs. The buffer outputs are directly connected to output pins. Consequently, there are 28 voltage output pins: $4 \times 6 = 24$ pins for the matrices connected together and 4 pins for the extra matrix.

5.1.1 Test Matrices

In all matrices, the capture photodiode has an area equal to 3×3 μm^2. Matrices (4,1) and (4,2) from Figure 5.1 are regular 4T pixels, which were included for comparison and characterization purposes. The layout of the 4T pixel is presented in Figure 5.2. The 4T pixels have are of 7.1×7.1 μm^2 and fill-factor of 17.85%. Color filters were included in matrix (4,2), according to a Bayer pattern arrangement. The remaining eleven matrices implement the proposed HDR imaging circuit with small structural variations.

Matrices (3,3) and (4,3) are the HDR implementations with color. Matrix (3,3) uses the regular HDR pixel, with Bayer pattern filters, as discussed in Section 4.3. To allow for a better adjustment of the image quality through the control of V_m, the inverter supply voltage is separate from the matrix supply. Four pixels from matrix (3,3) are presented in Figure 5.3. Each pixel has 13.7 μm × 13.7 μm and a fill-factor of 4.8%, which is the lower fill-factor from all test matrices. For proper white balance, as explained in Section 4.3, it is necessary to measure the matrix average luminance value. For this reason it was included in this matrix a single source follower connected to one of the boarder switches that allow the the sampling of the average. Matrix (4,3), whose pixel layout is presented in Figure 5.2(b), is the color matrix in which the green pixels control the pixels assigned to the other two colors. This matrix also has a separate supply voltage for the inverters. In Figure 5.2(b) we can see the color filters layout layers. The read pixel is controlled by the green control photodiode that is immediately below it. The blue pixel is controlled by the green control photodiode that is immediately above it. The layout of the pixels red and green in Figure 5.2(b) have 13.4 μm of width and 15.7 μm of height. A separate



(a) (b)

Figure 5.2: (a) layout of a 4T pixel and (b) of four color pixels with the green pixels controling the intregration period.

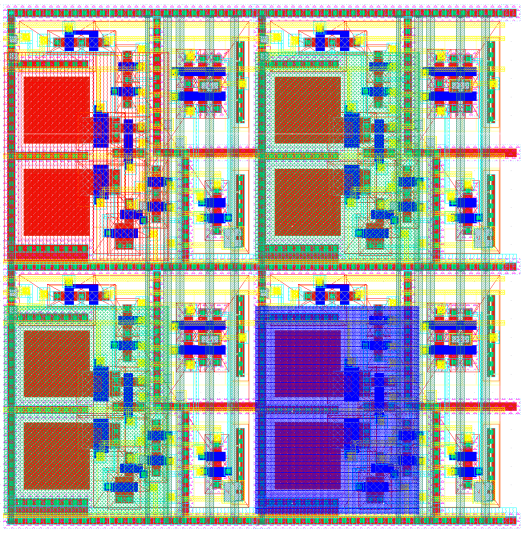


Figure 5.3: Four color pixels with independent integration period control.

pixel have thus 13.4 μm of width and 15.7/2 μm = 7.8 μm of height, yielding in 8.56% of fill-factor.

The pixel from matrix (1,3) has the capture and control photodiodes of same size and equal to 3 μm × 3 μm. The pixel layout is presented in Figure 5.4(a). This pixel has area equal to 13.5 μm × 13.5 μm and a fill-factor of 4.94%. Matrix (1,2) and (1,1) differ from matrix (1,3) just in the size of the control photodiode. The former has a control photodiode of 2 μm × 2 μm and the latter of 1 μm × 1 μm. These three matrices capture gray scale images using pinned photodiodes for both capture and control. Figure 5.4(b) presents matrix (1,2) pixel layout, with area of 12.5×12.5 μm^2 and 5.76% fill-factor. Figure 5.4(c) presents matrix (1,1) pixel layout, with area of 12×12 μm^2 and 6.25% fill-factor. The reduction of the control photodiode allowed of the pixel area reduction, but the control circuitry limits this reduction: there is not the same rate of reduction from matrix (1,3) to matrix (1,2) and from matrix (1,2) to matrix (1,1).

Matrices (2,1), (2,2) and (2,3) are equal to the ones above except for the fact that they do not use the pinned photodiode. Experimental comparison between pinned photodiode matrices and their standard photodiode counterparts will allow for an assessment of pinned photodiode advantages.

Matrix (3,2) is formed by pixels with minimum length, to reduce the area as much as possible. The matrix (3,2) pixel layout is presented in Figure 5.4(d). The area is equal to 10.9 μm × 10.2 μm, 8.65% of fill-factor.

The pixel presented in Figure 5.5(a) belongs to matrix (3,1). In this pixel, four capture photodiodes share a single control circuit. As it can be seen in the figure, photodiodes were laid out as close as possible to each other, to make control photodiode discharge similar to the capture-only pixels discharge. For this reason

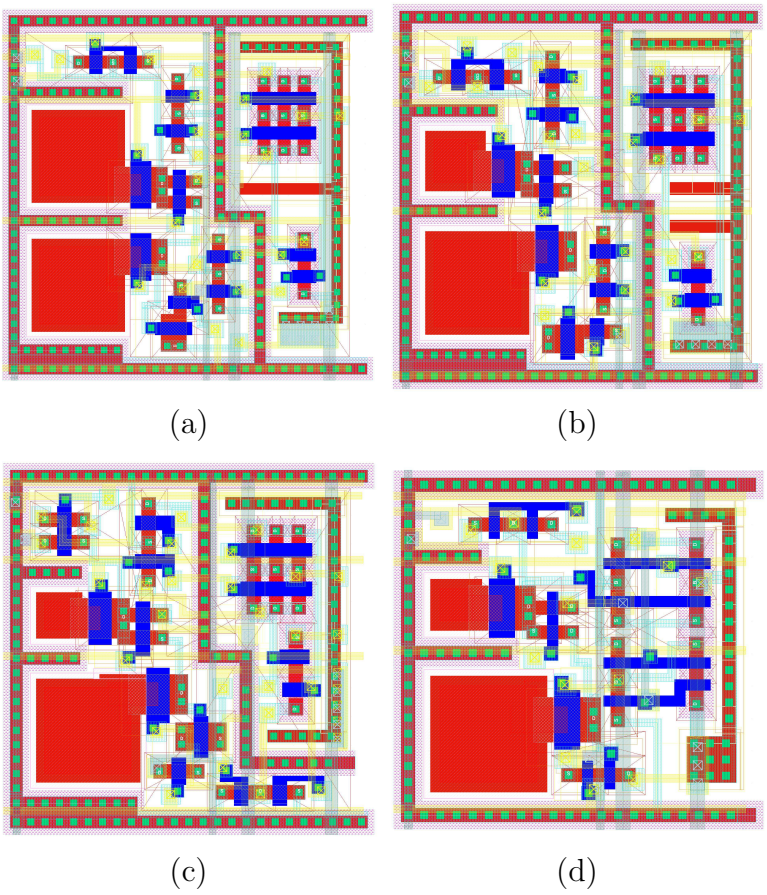
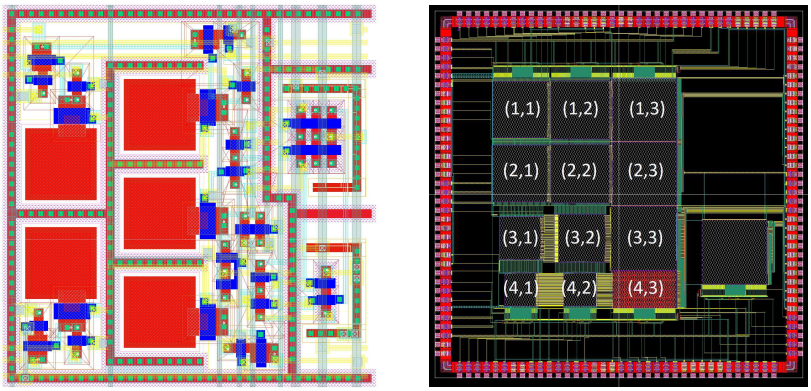


Figure 5.4: Different HDR pixel layouts, with different control photodiode sizes: (a) matrix (1,1) pixel, with capture and control photodiodes size equal to 3 μm $\times$ 3 μm; (b) matrix (1,2) pixel, with control photodiode size equal to 2 μm $\times$ 2 μm; (c) matrix (1,1) pixel, with control photodiode size equal to 1 μm $\times$ 1 μm; (d) matrix (3,2) pixel, which has minimal pixel area.

the photodiodes are not equally distributed in the matrix. The area in Figure 5.5(a) is 18.4 μm $\times$ 18.4 μm, which corresponds to four pixels. The effective pixel area is thus 9.2 μm $\times$ 9.2 μm, yielding 10.6% of fill-factor.

Figure 5.5(b) shows top cell layout. Although it is not possible to see the details, this figure shows how the matrices are organized, in agreement with Figure 5.1, and how they are connect to the die pads.

The extra matrix pixel layout presented is the same as the one presented in Figure 5.3, but without the color filters. This matrix implements the HDR pixel with capture and control photodiodes of same size, and with separate supply voltage for the inverters. The area difference between the pixel from Figure 5.3 (extra matrix) and 5.4(a) (matrix (1,3)) is given by the removal of the separate supply voltage for the inverters. The cost of this extra control is an increase of 3% in the total pixel area. Experiments will be conduced to evaluate the whether this area increase was actually beneficial.



(a) (b)

Figure 5.5: (a) layout of pixels sharing control photodiodes; (b) top cell layout view.

Table 5.1 summarizes the characteristics of all designed pixels. The HDR pixel that shares the control circuit among four capture circuits is the one with highest fill-factor, equal to 10.6%, which is a low fill-factor when compared to linear-log pixel implementations, but is comparable to other HDR implementations at the focal plane. The comparison between HDR implementation is shown in Section 5.6.

5.2 Schematic Diagram Simulation Results

Figure 5.6(a) shows a normalized 16×16 patch used in the schematic diagram simulations. The brightest pixel value is equal to one and the darkest pixel value equal to zero. This patch is part of an HDR image from [104], which presents a dark area in the form of a leaf and a bright area in the background. The pixels in this image were linearly mapped to photocurrents varying from 10 pA minimum to 30 nA maximum, which corresponds to 70 dB dynamic range (above 11 bits). The order of magnitude of the photocurrents used in electric simulations were chosen to avoid long simulation times while guaranteeing a dynamic range above 8 bits. In practice T_s and T_{Max} must be adjusted according to the incident dynamic range. In all simulations, the photodiode capacitance was set to 50 fF.

Figures 5.6(b) to (d) show the schematic simulation results for different T_s values and T_{Max} set to 200 μs. The simulation results show that the schematic behavior is similar to the system-level simulations presented in Section 4.1.2, that is, T_s increase enhances the contrast.

Corner simulation results are presented in Figure 5.7. This simulation considers the worst and best cases for n-channel and p-channel transistors operation. The models provided in the design kit consider cases in which the pixel response is slow,

Table 5.1: Designed pixels characteristics.

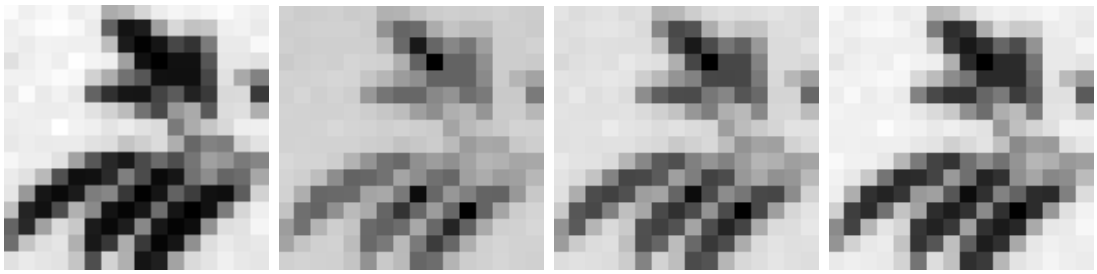
Matrix	PPD*	HDR	Area(ph_{ctrl}) (μm^2)	Pixel Area (μm^2)	FF†	Remarks
(1,1)	Yes	Yes	1×1	12×12	6.25%	m‡ = 9.
(1,2)	Yes	Yes	2×2	12.5×12.5	5.76%	m = 2.25.
(1,3)	Yes	Yes	3×3	13.5×13.5	4.94%	m = 1.
(2,1)	No	Yes	1×1	12×12	6.25%	Standard photodiode, m = 9.
(2,2)	No	Yes	2×2	12.5×12.5	5.76%	Standard photodiode, = 2.25.
(2,3)	No	Yes	3×3	13.5×13.5	4.94%	Standard photodiode, m = 1.
(3,1)	Yes	Yes	3×3	$(18.4 \times 18.4)/4$	10.6%	Shared control photodiode, separate inverter supply.
(3,2)	Yes	Yes	1×1	10.9×10.2	8.65%	Minimum length transistors, m = 9.
(3,3)	Yes	Yes	3×3	13.7×13.7	4.80%	Color filters, separate inverter supply., m = 1
(4,1)	Yes	No	-	7.1×7.1	17.85%	Regular 4T matrix
(4,2)	Yes	No	-	7.1×7.1	17.85%	Regular 4T matrix, color filters.
(4,3)	Yes	Yes	3×3	$13.4 \times 15.7/2$	8.56%	Color filters, separate inverter supply, m = 1.
Extra	Yes	Yes	3×3	13.7×13.7	4.80%	Separate inverter supply, m = 1.

All capture photodiodes are equal to Area(ph_{cap}) = 3 μm $\times$ 3 μm.

*Pinned photodiode.

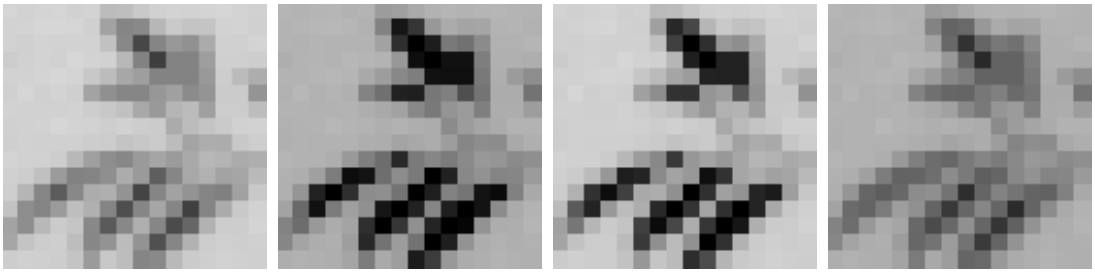
†Fill-factor = Area(ph_{cap})/Pixel Area.

‡m = Area(ph_{cap})/Area(ph_{ctrl})



(a) (b) (c) (d)

Figure 5.6: (a) 16 × 16 pixel image for schematic diagram simulations, (b) tone-mapping result for $T_s = 1$ μs, (c) tone-mapping result for $T_s = 1.5$ μs, and (d) tone-mapping result for $T_s = 2$ μs. The best result seems to be obtained for $T_s = 2$ μs.



(a) (b) (c) (d)

Figure 5.7: Schematic corner simulations (a) n-channel and p-channel slow, (b) n-channel and p-channel fast, (c) fast n-channel and slow p-channel and (d) slow n-channel and fast p-channel.

fast and typical. Simulations presented in Figure 5.7 consider the typical case. In Figure 5.7(a), both transistors are modeled for the worst case scenario. In this case, the transistors operations are slow. As it can be seen in the figure, some cases present more noise than others, but the overall result is good, showing that the circuit is working properly. Two main effects explain the differences among the images in these figures: V_m change and reset voltage change. By changing the transistors models, the response speed of the inverter I_1 will be affected, since the models change the transistors response speed, resulting in a shift on V_m. The maximum voltage that the floating diffusion can reach is also affected, because the threshold voltage changes as the transistor model changes. Figure 5.7(a) corresponds to the highest V_t and thus the lowest reset voltage, followed by Figures 5.7(d), (c) and, finally, (b), which has the lowest V_t, thus reaching the highest reset voltage. The influence of the V_t change can be mostly seen in darker pixels, in which T_{Max} is reached, because these pixels have the same time to discharge in all the models, but the maximum reset voltage change will result in pixels in Figure 5.7(a) being brighter

than pixels in Figure 5.7(b). The shift on V_m affects pixels that do not reach T_{Max}, which is easier to see in Figures 5.7(c) and (d). In Figure 5.7(c), when we make the n-channel transistor fast and the p-channel transistor slow, the inverter takes longer to change its output from zero to one. V_m in this case is approximately 1.4 V. From Figure 4.8(c) we can see that reducing V_m shifts the response curve up because pixels have more time to integrate, which is why the sky in Figure 5.7(c) is brighter than the sky in Figures 5.7(a), 5.7(b) and 5.7(d). The opposite effect is shown in Figure 5.7(d). When the n-channel transistor is slow and the p-channel transistor is fast, V_m increases to approximately 1.7 V, resulting in grayer sky region. Figure 5.7(d) corresponds to highest V_m, followed by Figures 5.7(a), 5.7(b) and 5.7(c).

The image presented in Figure 5.8 was also used for schematic diagram simulations. The goal of using this image was to observe how the circuit reacts when the image is mostly dark, but with a few bright details. For best visualization of pixel differences, this figure is plotted using a cold-to-hot colormap. This image is a 16 × 16 patch that was cropped from Figure 4.7(a) (table towel). It was then artificially modified, to have a high dynamic range. From simulations using this HDR 16 × 16 image, we can see advantages of the proposed circuit. Pixels at the first and second rows of the image were set to high current values, but the overall image remains mostly dark (i.e. with a low average pixel value). With respect to schematic diagram input stimulus, the first half of the first row is set to 60 nA current, the second half was mapped to 30 nA, and the beginning of the second is set to 15 nA. The other currents are below 4 nA and the minimum current is equal to 10 pA. The dynamic range is 76 dB (above 12 bits).

Comparative simulation results for the linear-capture (conventional 4T) pixel and for the tone-mapping pixel are presented in Figure 5.9. Figures 5.9(a) to 5.9(d) are the results of schematic simulations using regular 4T pixels, with increasing integration period, in which case the photodiode was modeled as a capacitor in parallel with a current source. The transfer gate of the 4T pixel operates as usual: by the end of the integration period, a transfer-gate control pulse is turned on for

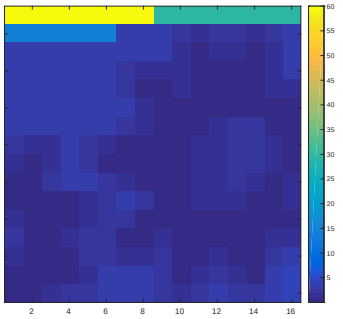


Figure 5.8: Image with 76 dB dynamic range, for schematic diagram simulations.

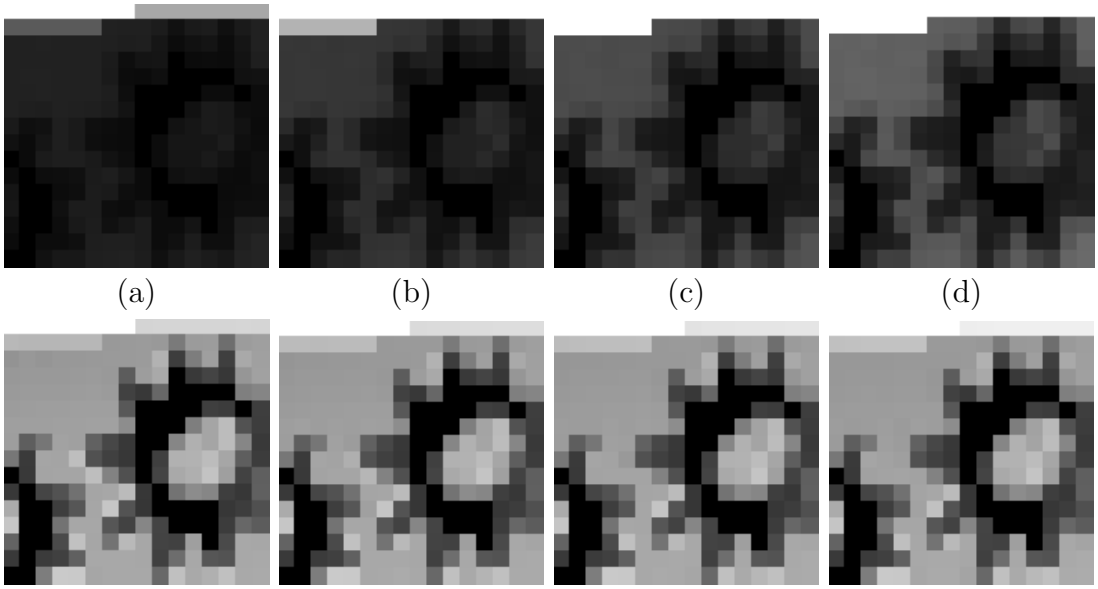


Figure 5.9: Simulation results (conventional 4T pixel and HDR pixel schematic diagrams, luminance only) for a 76 dB dynamic range input image: in (a) to (d), conventional 4T pixel with integration periods increasing from 1.5 μs (a) to 7.5 μs (d); in (e) to (h), HDR pixel with T_s increasing from 500ns (e) to 800 ns (h).

just enough time. The simulation results shown in Figures 5.9(a) and 5.9(b), which correspond to smaller integration periods, show low contrast in the darker areas, but we are able to see the details in the bright pixels of the image, image first and second rows are not saturated, that is, not completely white. When we increase the integration period, Figures 5.9(c) and 5.9(d), the result is the opposite: visualization improves at the darkest areas, but the differences between the brighter pixels are lost.

Figures 5.9(e) to 5.9(h) show the results of the schematic simulation with the tone-mapping pixel. The contrast is significantly larger in the darker areas, whereas is still possible to see the brighter pixel luminance differences. Figures 5.9(e) to 5.9(h) show the effects of increasing T_s increasing from 500 ns to 800 ns. The input image average pixel value is 0.08, so it is expected that the circuit response curve be similar to the one seen in Figure 4.9(a). These figures show that increasing T_s increases the value of the brighter pixels, and eventually lead to a clamp in these pixels, which means that the details are the first pixel row disappear, if T_s is too high. The sequence of images in Figures 5.9(e) to 5.9(h) respond as expected, the contrast increases slightly, since there is a small change in T_s value. In Figure 5.9(h), the luminance differences between pixels in the first pixel row is very subtle, which means that those pixels are very close to the clamp.

To observe the schematic simulation results considering an image with large

average pixel value, the pixels values from Figure 5.8 were inverted (dark pixels were transformed to bright pixels and bright pixels to dark pixels) and the absolute difference between the pixels from first and second rows was reduced. The schematic diagram input image stimulus also has 10 pA minimal current and 60 nA maximal current, thus keeping the dynamic range at 76 dB. The first half of the first pixel row was mapped to 10 pA. The second half of the first pixel row was mapped to 3 nA, and the beginning of the second pixel row was mapped to 9 nA. The other currents are above 56 nA. This is not a good situation for the designed pixel. As observed in the system-level simulations (Figure 4.8), an input image with high average pixel value yields an output images that does not use the entire (eight-bit) dynamic range. The result is presented in Figure 5.10. The linear result, in Figures 5.10(a) and 5.10(c), saturates for small integration periods (1.2 μs). The proposed pixel is capable of not saturating the image. It also keeps the details in the first line, but the image does not have a good contrast, which can be seen in Figures 5.10(e) and 5.10(g). In comparison to the linear pixel, the advantage is that the information was not lost because of saturation. Figures 5.10(b), 5.10(d), 5.10(f) and 5.10(h) show the results

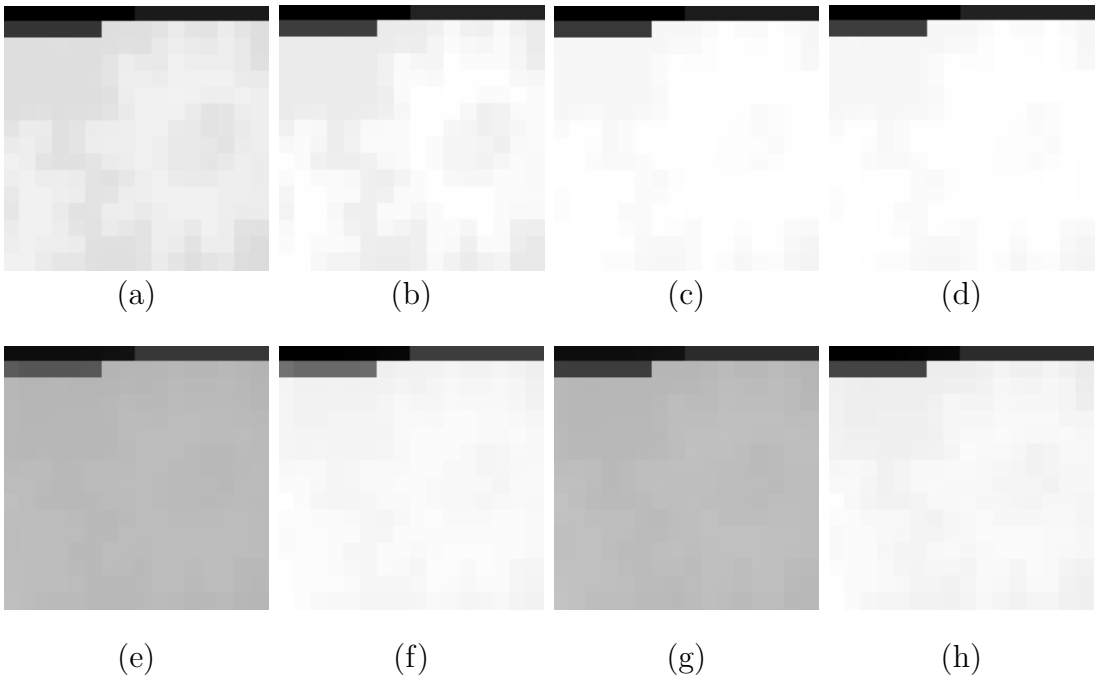


Figure 5.10: Simulation results (conventional 4T pixel and HDR pixel schematic diagrams, luminance only) for a 76 dB dynamic range input image with high average pixel value. In (a) to (d), conventional 4T pixel results: (a) integration period equal to 1.1 μm; (b) linear normalization of (a) to the $[0, 1]$ interval; (c) integration period equal to 1.2 μs, and (d) linear normalization of (c) to the $[0, 1]$ interval. In (e) to (h), HDR pixel results: (e) $T_s = 0.5$ μs, (f) linear normalization of (e) to the $[0, 1]$ interval, (g) $T_s = 0.7$ μs, and (h) linear normalization of (g) to the $[0, 1]$ interval.

linearly adjusted to a range of 0 to 1, which can be beneficial in this case, but there is still low contrast in the final image.

As proposed in Section 4.1.2, a possible solution to improve the HDR circuit result for high average images is the reduction of V_m, and V_m can be indirectly controlled by changing the supply voltage of the inverter connected to the control floating diffusion. Figure 5.11 shows the simulation results when the inverter supply voltage is equal to 2.7 V, which yields a V_m of approximately 1.3 V, and T_s varying from 800 ns to 1.1 μs. On the contrary to the HDR results presented in Figure 5.10, we are now capable of covering the entire (8-bit) dynamic range, as it can be seen from Figure 5.11(d), normalization is thus no longer necessary. In these results, we can see that there is a trade off when choosing T_s: Figure 5.11(a) shows better contrast in darker areas while Figure 5.11(d) shows better contrast in bright areas. Figure 5.11(c) seems to be the better compromise between details in the dark and bright areas.

Figure 5.12 shows plots of the pixel values from rows four (bright pixel values) and one (dark pixel values) of Figures 5.10(a), 5.10(c), 5.10(f), 5.10(h), 5.11(a) and 5.11(c). These figures correspond to 16 $\times$ 16 matrix simulations considering 4T pixels with 1.1 μs and 1.2 μs integration periods (Figures 5.10(a) and 5.10(c)), HDR normalized results with V_m = 1.6 V, T_s = 0.5 μs and V_m = 1.6 V, T_s = 0.7 μs (Figures 5.10(f) and 5.10(h)), and HDR result with V_m = 1.3 V, T_s = 0.8 μs and V_m = 1.3 V, T_s = 1.1 μs (Figures 5.11(a) and 5.11(c), normalization was not applied in this case). Because of the low contrast in Figures 5.10(e) and 5.10(g), HDR results without normalization when V_m = 1.6 V were not plotted. To make a simple evaluation of contrast enhancement in these images, we consider the difference between maximum and minimum pixel values in rows four and one. The human eye perception depends not only on the difference itself, but also on the neighboring pixels values, so the difference might not be perceived. We use the difference as a way to understand the circuit behavior in each case. From Figure

(a) (b) (c) (d)

Figure 5.11: HDR pixel results with V_m equal to to 1.3V and T_s varying from 0.8 μs to 1.1 μs.

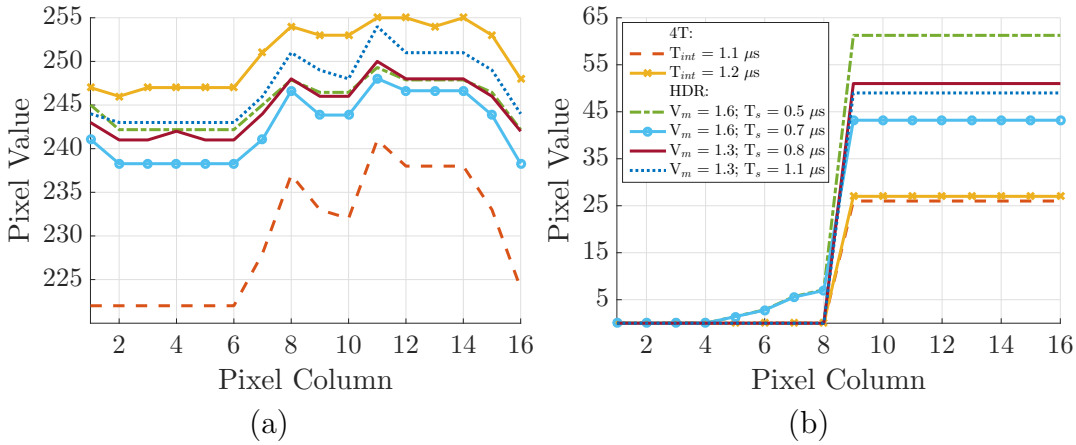
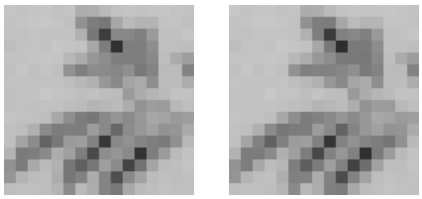


Figure 5.12: Pixel values in HDR simulations based on bright (high-average) images: (a) fourth pixel row, and (b) first pixel row. 4T pixel response considering 1.1 μs and 1.2 μs integration periods, HDR pixel normalized results with V_m = 1.6V and T_s = 0.5 μs and T_s = 0.7 μs, and HDR pixel with V_m = 1.3V and T_s = 0.8 μs and T_s = 1.1 μs.

5.12(a) we can see that highest contrast in the fourth image row is given by the 4T pixel when the integration time is equal to 1.1 μs, with a difference between maximum and minimum values in this row equal to 19. When the integration time changes to 1.2 μs, the fourth row has many saturated pixels and the contrast of the first row, Figure 5.12(b), is only modestly increased. In the case of the normalized HDR response, the difference between maximum and minimum values in row four is equal to 7 when T_s = 0.5 μs and to 10 when T_s = 0.7 μs. The highest contrast in the dark row, Figure 5.12(b), is given for the normalized HDR results when T_s = 0.5 μs, with a difference between maximum and minimum values equal to 61. With the change of V_m to 1.3 V, we are capable of increasing modestly the contrast in the bright areas, when compared to the 1.6 V HDR normalized response, while having a response better than the 4T pixel in darker areas. For the HDR pixel with V_m = 1.3 V, the differences between maximum and minimum values in Figure 5.12(a), are equal to 9, when T_s = 0.8 μs, and 11, when T_s = 1.1 μs. In Figure 5.12(b), are 51, when T_s = 0.8 μs, and 49, when T_s = 1.1 μs. Although the contrast increase in the bright row was small when compared to the normalized result, changing V_m represents an overall advantage, since the normalization is no longer needed and the dark row of pixels has better contrast than the 4T result.

5.3 Monte Carlo Simulation Results

Monte Carlo schematic simulations considering 100 runs with process and mismatch variations were also performed. Figure 5.13(a) shows the nominal result, simulated



(a) (b)

Figure 5.13: (a) nominal output image result, for comparison with Monte Carlo simulation results, and (b) average output image, computed from 100 Monte Carlo runs.

using the same conditions as the Monte Carlo analysis, namely T_s, T_{Max} and photodiode capacitance. Figure 5.13(b), shows the Monte Carlo average image result. Both images are very similar, the maximum difference between these images is equal to 11.5, which corresponds to 4.5% of the dynamic range. Considering the nominal image as the ideal image, the maximum average error is equal to 13.8, which corresponds to approximately 5.4% of the output image dynamic range. If we consider the average image in Figure 5.10(b) as the ideal reference image, then these numbers change to 12.3 and 4.8%. These results show that the circuit is robust to process and mismatch variations, with an error below 10% of the dynamic range. Figure

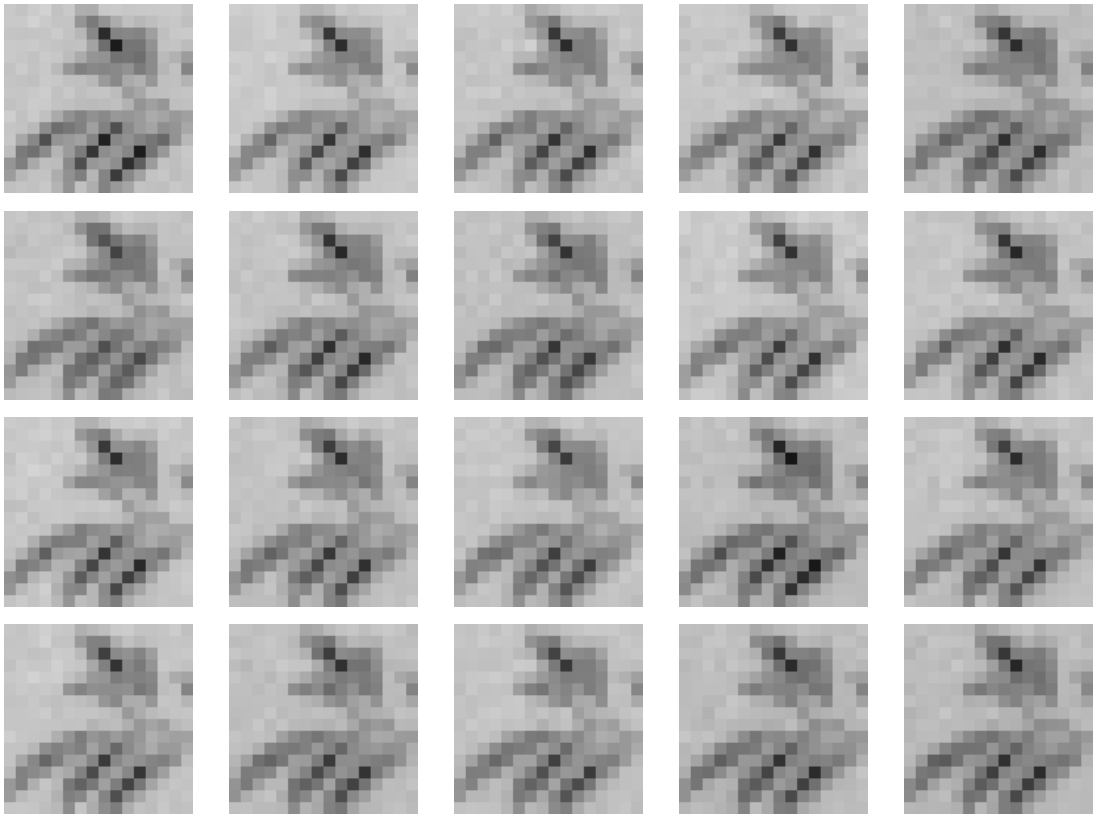


Figure 5.14: Twenty Monte Carlo simulation results.

5.14 shows 20 Monte Carlo simulation image results. As the 12.3 maximum average error suggests, all results are similar to each other, and also similar to the nominal image.

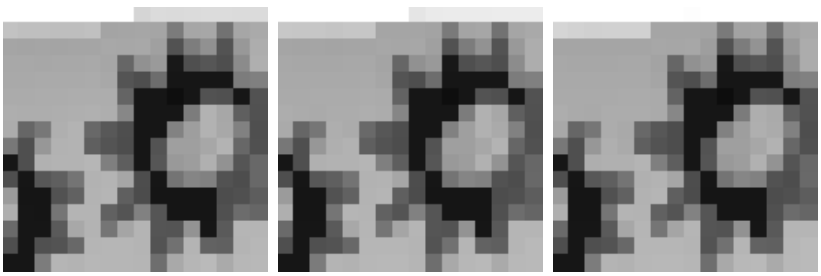
5.4 Layout Simulation Results

Figure 5.15 shows the result of post-layout simulation using the 16×16 image presented in Figure 5.8 as input. A 16×16 layout cell was extracted to allow for this simulation. The reference current source necessary for biasing all pixel source-following transistors was generated by a current mirror. This simulation includes the imprecision of this non-ideal current source.

This simulation result is very similar to the one presented in Figure 5.9. The MSSIM index considering Figures 5.9(a) and 5.15(a) is equal to 0.98 and the PSNR is equal to 28 dB. The low PSNR is explained by the fact that small circuit modifications change the final pixel value, which contributes to increasing the mean square error. The MSSIM is a better index for this case, reaffirming the figure similarities. The maximum absolute difference between these figures is equal to 32, 12.5% of the dynamic range. The absolute difference average is 2.2, 0.86% of the dynamic range. The results suggest that layout non-idealities (e.g. parasitics) have little influence on the overall result.

5.5 Simulation Results for 114 dB Dynamic Range Input Image

To observe the circuit response to dynamic ranges wider than 76 dB, another schematic diagram simulation was performed using an input range of 114 dB. The linear-log pixel reported in [28] presents a dynamic range of 111 dB to 115 dB, which was the inspiration for the simulated range. The image used for this simulation is a modified



(a) (b) (c)

Figure 5.15: Post-layout simulation results using a 76 dB dynamic range input image: (a) $T_s = 500$ ns, (b) $T_s = 700$ ns, and (c) $T_s = 1$ μs.

version of Figure 5.8, where the differences in the darker region were further reduced to verify the circuit capacity of detecting such small differences. The differences in the dark area of the modified version of the image vary from $1/2^{20}$ (corresponding to ranges even wider than 114 dB) to $1/2^{12}$. Because of these small differences, it is not possible to visualize this image without tone mapping. To allow for a better understanding of the experiment, the pixel values were normalized to a range of 0 to 255 and printed in Table 5.2. The values from Table 5.2 were mapped to photocurrents in a range of 100 fA to 50 nA and used as the schematic diagram inputs. The simulation result considering T_s = 750 ns and T_{Max} = 2.7 ms is presented in Figure 5.16.

To evaluate if the details of the original image were maintained, Figures 5.17 and 5.18 show comparisons, row by row, of the input image values and the schematic diagram simulation result. *Except for the second row of the image, Figure 5.17(b)*, the rows of the input and output images were normalized one by one and plotted. The normalization of the input rows do not cause the loss of details because the rows alone do not have high dynamic range (except for the second row). Thus, plotting the normalized result allows for the visualization of the differences between the pixels in a row, namely, the details. The second row begins with bright pixel value and end with dark pixel values, containg thus a wide range. A simple normalization results in detail loss. For this row, first the logarithm of the input image pixel values was computed. Then, the results were normalized. The schematic diagram simulation result of the second row was just normalized, since tone mapping was applied by the circuit. As it can be seen in Figures 5.17 and 5.18, the circuit outputs follows the input changes in most cases which means that the circuit is capable of perceiving and representing most of the details.

Figure 5.16: Schematic simulation result for an input image of 114 dB with T_s = 750 ns and T_{Max} = 2.7 ms.

Table 5.2: Input image pixel values used for the 114 dB dynamic range simulation

	Col 01	Col 02	Col 03	Col 04	Col 05	Col 06	Col 07	Col 08	Col 09	Col 10	Col 11	Col 12	Col 13	Col 14	Col 15	Col 16
Row 01	255.0000	255.0000	255.0000	255.0000	255.0000	255.0000	255.0000	255.0000	204.0000	204.0000	204.0000	204.0000	204.0000	204.0000	204.0000	204.0000
Row 02	127.5000	127.5000	127.5000	127.5000	127.5000	127.5000	0.0623	0.0623	0.0623	0.0156	0.0010	0.0156	0.0156	0.0005	0.0156	0.0623
Row 03	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0311	0.0623	0.0039	0.0005	0.0010	0.0019	0.0005	0.0019	0.0623
Row 04	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0156	0.0005	0.0019	0.0019	0.0002	0.0005	0.0005	0.0005	0.0019	0.0623
Row 05	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0156	0.0005	0.0005	0.0005	0.0002	0.0002	0.0002	0.0002	0.0010	0.0010
Row 06	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0623	0.0005	0.0002	0.0002	0.0000	0.0002	0.0005	0.0005	0.0002	0.0005
Row 07	0.0623	0.0623	0.0311	0.0623	0.0623	0.0623	0.0156	0.0010	0.0002	0.0002	0.0002	0.0010	0.0156	0.0078	0.0005	0.0005
Row 08	0.0156	0.0005	0.0010	0.0623	0.0156	0.0005	0.0005	0.0005	0.0002	0.0002	0.0010	0.0039	0.0156	0.0078	0.0010	0.0005
Row 09	0.0010	0.0005	0.0019	0.0623	0.0078	0.0005	0.0005	0.0005	0.0002	0.0005	0.0039	0.0039	0.0039	0.0156	0.0005	0.0005
Row 10	0.0002	0.0005	0.0156	0.0623	0.0623	0.0156	0.0010	0.0005	0.0002	0.0005	0.0019	0.0019	0.0078	0.0019	0.0005	0.0005
Row 11	0.0005	0.0002	0.0005	0.0005	0.0005	0.0156	0.0623	0.0039	0.0005	0.0002	0.0010	0.0010	0.0010	0.0005	0.0005	0.0005
Row 12	0.0019	0.0002	0.0002	0.0005	0.0005	0.0156	0.0156	0.0005	0.0005	0.0002	0.0002	0.0002	0.0002	0.0005	0.0005	0.0005
Row 13	0.0039	0.0002	0.0002	0.0010	0.0311	0.0156	0.0005	0.0005	0.0010	0.0005	0.0002	0.0002	0.0002	0.0005	0.0005	0.0005
Row 14	0.0005	0.0002	0.0002	0.0005	0.0078	0.0311	0.0010	0.0039	0.0156	0.0005	0.0005	0.0019	0.0002	0.0005	0.0311	0.0311
Row 15	0.0005	0.0002	0.0005	0.0005	0.0005	0.0311	0.0623	0.0623	0.0156	0.0005	0.0005	0.0156	0.0005	0.0005	0.0623	0.0623
Row 16	0.0002	0.0002	0.0010	0.0078	0.0039	0.0623	0.0623	0.0623	0.0311	0.0005	0.0078	0.0623	0.0156	0.0039	0.0623	0.0623

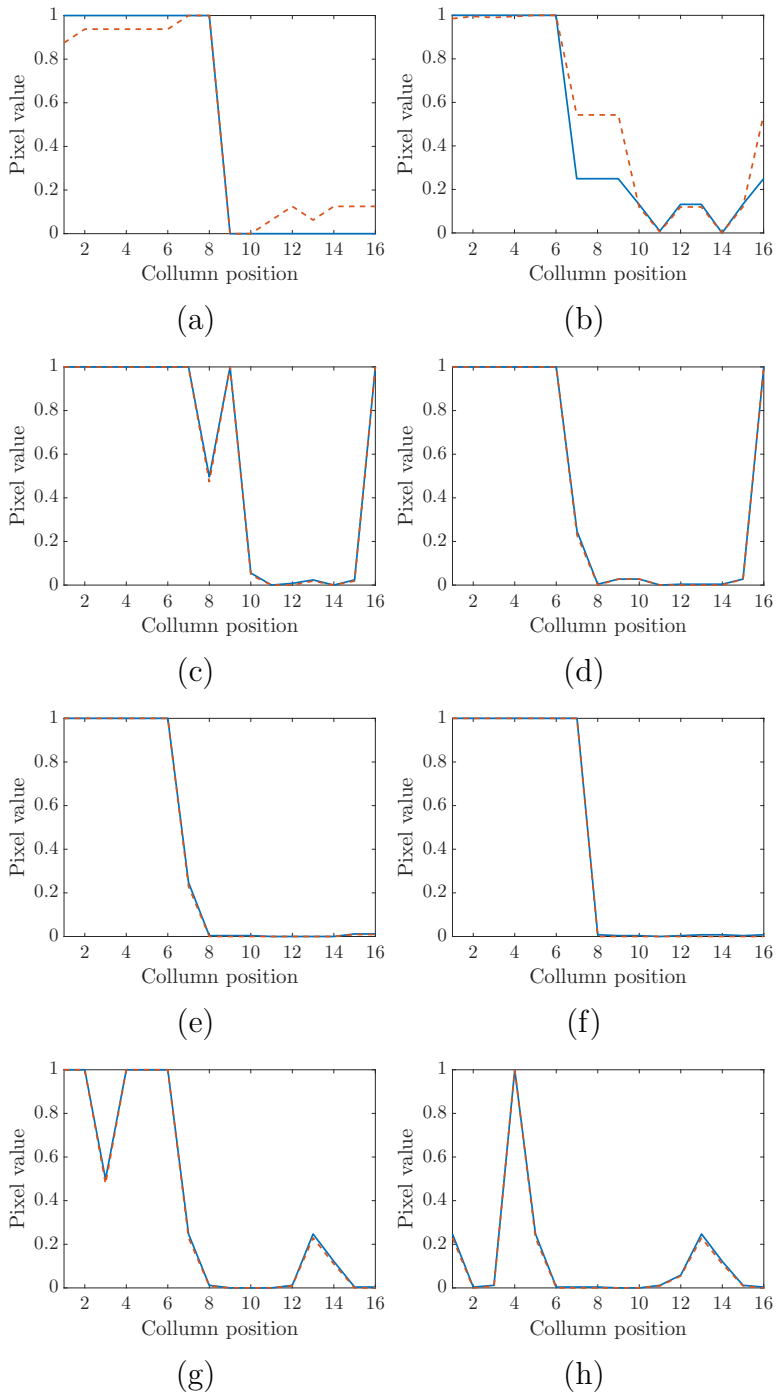


Figure 5.17: Comparison between input image rows details, in solid line, and the circuit output image rows, in dashed line, from row (a) one to row (h) eight. Except for (b), second image row, input and tone mapped pixel values where normalized row by row to a range of $[0, 1]$. In (b), the logarithm of input values was computed to enhance the details, then normalized to $[0, 1]$.

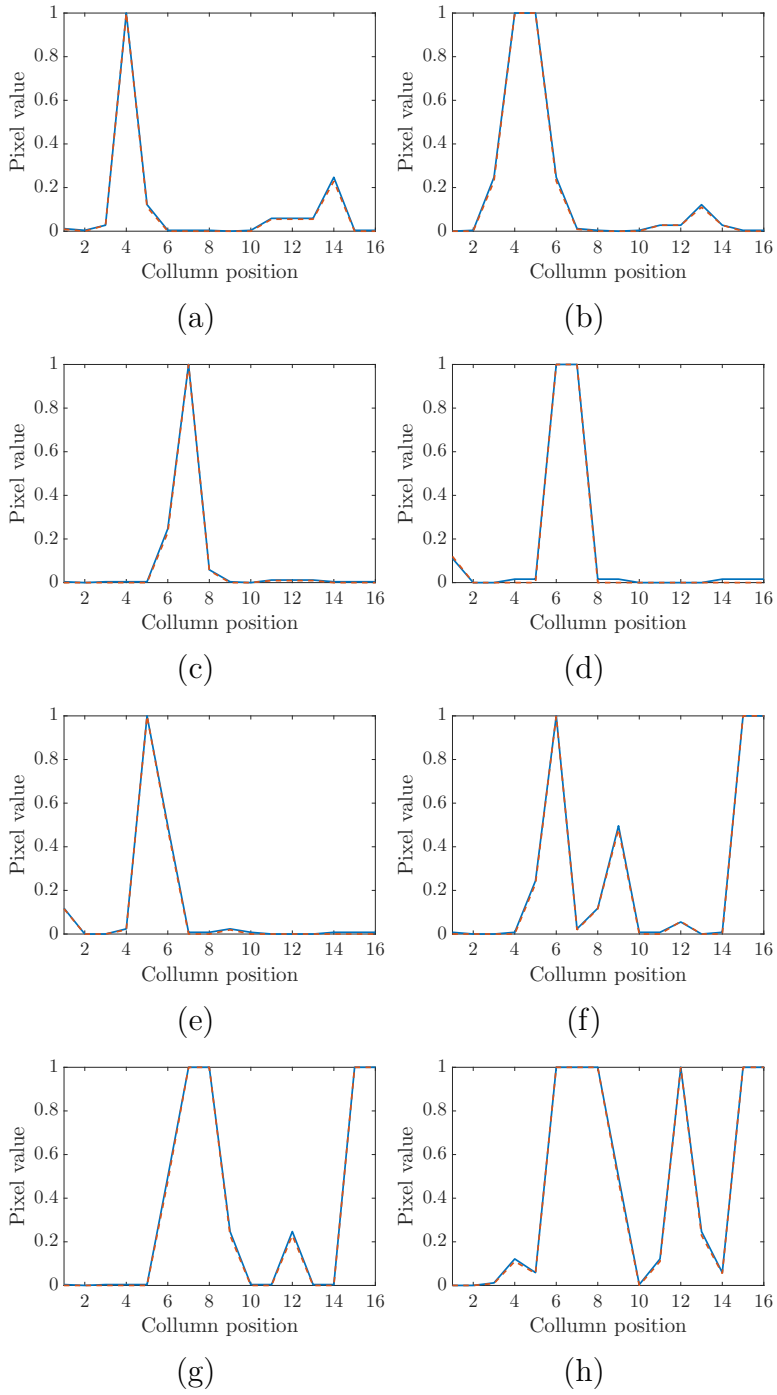


Figure 5.18: Comparison between input image rows details, in solid line, and the circuit output image rows, in dashed line, from row (a) nine to row (h) sixteen. Input and tone mapped pixel values were normalized row by row to a range of $[0, 1]$.

5.6 Discussion

Table 5.3 presents a comparison between the HDR designed pixel and six other HDR imagers considering different approaches. All references used for this table present experimental results, so the comparison with the studied pixel might not represent the real scenario since the chip has not been tested yet.

Linear-log pixels are presented in [108] and [109]. In both references, FPN reduction techniques are proposed. In [108], FPN is reduced by including a hard reset structure inside the pixel. After reading the pixel values, the pixels are reset by a p-channel transistor (which allows for the hard reset) and sampled. Then, the pixel and reset values are subtracted, allowing for partial FPN reduction in the log region and complete FPN reduction in the linear region. In [109], a usual linear-log pixel is implemented, which has the same structure as the 3T pixel, but with control over reset transistor drain voltage. Subtraction between the pixel value and a reference voltage is also performed. The reference voltage in [109] must be set to a value lower than the lowest detectable signal voltage. The linear-log pixels are the pixels with highest fill-factor, and they achieve significantly large dynamic range. The solutions present simple schematic diagrams, which yields in high fill-factor.

The proposed HDR scheme presented in [110] takes place outside the pixel matrix. A digital logarithmic single-slope analog-to-digital converter is proposed. The

Table 5.3: Comparison between HDR chips.

Ref.	Year	Array Size	Pixel Size (μm^2)	FF	DR (dB)	Remarks
[108]	2011	100×100	9.4×9.4	30%	112	Lin-log pixel and FPN reduction.
[110]	2012	320×240	2.25×2.25	-	80	Logarithmic single-slope ADC outside the pixel matrix.
[111]	2013	256×256	15×15	17%		Analog adjustment of the pixel value after capture.
[109]	2014	8×8	10×3	56%	115	Lin-log pixel and FPN reduction.
[32]	2015	180×148	33×33	0.8%	151	Time to reach a reference and output code adjustment.
	2016	128×96	25×25	10%	120	Number of events each pixel generates inside a frame.
This work		64×64	13.7×13.7 - 9.2×9.2	4.8% - 10.6%	70 - 114*	Designed pixels from worst to best fill-factor.

*Expected dynamic range from simulation results.

operation of the pixel is linear, but the values are tone mapped directly in the conversion. Different conversion modes are available. Although the fill-factor was not reported, this imager uses a regular 4T pixel, so it is expected to have a high fill-factor. This pixel does not achieve a dynamic range as wide as the linear-log pixels do. It is expected that the studied pixel can achieve a wider dynamic range, since it can capture pixels with different integration periods, which is not available with the technique proposed in [110].

HDR is also performed outside the pixel matrix in [111]. As in the designed pixel, this imager uses average pixel value information for tone mapping. Captured pixel values are adjusted using 2×2 pixel-block average values, rather than global average values. The pixel value adjustment is performed using analog circuits outside the pixel matrix. The tone-mapping function is $Y=X/(X+X_0)$, where X is the pixel value, X_0 is the 2×2 average value and Y is the adjusted value. As in [110], a single capture is performed and all the pixels integrate at the same time.

The pixel from [32] achieves the highest dynamic range among the references used for comparison. As in the designed pixel, the dynamic range depends on the maximum time the pixels are allowed to integrate. A dynamic range of 151 dB is reached for 0.125 fps and 127 dB for 30 fps. The pixel from this reference computes the time it takes for the integration node to reach a threshold voltage. To limit the frame rate, the reference voltage is linearly increased after a defined period. Tone mapping is performed by a variable output bit code. Using the histogram information of a previous frame, more bits are assigned to the most populated luminance ranges. The next frame histogram is thus adjusted directly in the analog-to-digital conversion. This approach has the lowest fill-factor from Table 5.3(0.8%).

In [112], HDR is achieved by counting the number of times each pixel reaches a threshold voltage. When the threshold voltage is reached, the pixel sends a signal that represents that an event has occurred to a circuitry outside the matrix and resets the photodiode. A time window defines the frame. Pixel event counts inside that time window represent pixel values. If this window is short, dark pixels might not have enough time to generate an event. Thus, the dynamic range depends on the frame rate. For a frame rate of 3 fps, the dynamic range is equal to 120 dB. The disadvantage of this approach is the pixel size and circuit complexity. The fill-factor is equal to 10%, comparable to the fill-factor of the designed pixel with shared control circuit.

In the proposed circuit, HDR capture is achieved by controlling the integration period of each pixel separately. The circuit necessary for controlling the integration period is fabricated inside the pixel matrix, which reduces the fill-factor. The best fill-factor was achieved by sharing the control photodiode among a block of pixels, thus yielding a fill factor of 10.6%, which is comparable to the fill factor from [112].

It is expected, based on simulation results, that the dynamic range achieved by the proposed pixel is 114 dB at least. These simulations showed that, in comparison to the regular 4T pixel, the proposed pixel presents contrast increase when the image average pixel value is low. However, if the average pixel value is high, then the proposed pixel has a limited contrast response.

Chapter 6

Conclusions

In the first part of this thesis, a pixel that is capable of generating a Gaussian pyramid at the focal plane was presented. System-level simulations demonstrated that this pixel generates input data suitable for the SIFT algorithm. This pixel only requires two extra transistors when compared to a regular 4T pixel. The additional circuitry has low impact on pixel area and fill factor.

The great motivation for using imagers with focal-plane processing is their potential for enhancing throughput, because of parallel signal processing, and for reducing power consumption, because analog processing typically has larger energy efficiency than digital for applications with moderate SNR requirements. To the best of our knowledge, focal-plane potential advantages had not been previously assessed, in comparison to an equivalent digital circuit. The analyses that were carried out in Chapter 3 show that the focal-plane processing advantages are case-specific. When comparing the focal-plane approach to Gaussian pyramid generation with the use of a conventional sensor followed by a digital processor, the number of analog-to-digital conversions necessary for the focal-plane approach represent a significant bottleneck. To find conversion rate and energy consumption values that can be used for comparison purposes, different image sensor ADCs were considered. Regarding processing time, results show that the focal-plane architecture requires fast ADCs, ideally one ADC per column, to report significant advantages. Regarding energy savings, the focal-plane approach yields best results with SAR, cyclic or $\Sigma\Delta$ topologies, as presented in Section 3.1.4. To reach that conclusion, we consider state-of-the-art experimental median figures regarding ADC energy consumption. Considering specific cases, the best case for energy savings is when the single-slope data converter from [67] is used. To cite one example using a specific converter, analysis using a column-parallel SAR ADC with 14.6 pJ/sample shows that the architecture with pre-processing sensor can be 26 times faster and 49 times more energy-efficient than the digital approach with 10 PEs. The methodology presented allows for a quantitative estimation of the advantages that focal-plane processing

might bring about. This is an interesting tool for imager designers to understand, before implementation, the strengths of the proposed focal-plane processing techniques. This technique will be used to evaluate future designs, but it will also be applied to previous works, such as [113], to quantify their advantages with respect to throughput and energy consumption. Although the focal-plane approach has presented advantages, it was expected that these advantages would be higher than the ones indicated in the analysis. Because of these results we have decided not to fabricate this circuit. This work has not considered an area comparison between the approaches. Area is an important parameter in integrated circuit design because it influences the circuit cost. This parameter was not considered in the proposed analysis because with the time and energy results we were able to decide not to fabricate, so the area comparison would not contribute to the final conclusion. It is interesting, though, to include this type of comparison in future works.

The second part of this work was dedicated a to tone-mapping imager analysis and implementation. This imager concept was presented in [33]. System-level simulations with 12-bit images were presented, indicating the proposed method viability for dynamic ranges above 70 dB. Variations of the proposed architecture were proposed in this work, considering: different control photodiode sizes, control photodiode sharing, and color filters. Section 4.1 indicated that there is a limit for reducing the control photodiode size, because of the circuit parasitic capacitances. Sharing the control photodiode, which was the best option for reducing pixel area among the implemented pixels, also has a limitation: blocking artifacts appear if 4×4 (or larger) pixel blocks share the control photodiode. Considering this limitation, 2×2 pixel blocks with shared control circuit were implemented. The resulting layout has a 10.6% fill-factor, which was the best fill-factor among the implemented HDR matrices. System-level simulations also validated the color implementation considering two different approaches: each pixel controls its own integration time; and the green pixels control the integration times of their neighboring color pixels. The second approach is also a case in which a control photodiode is shared. Aside from system-level simulations, schematic diagram and post layout simulations were presented. Electric simulations were performed with 76 dB dynamic range and compared to 4T pixel schematic diagram simulations. In these simulations, the HDR sensor was able to represent more dark area details than the regular 4T sensor. For high average value simulations, the sensor was not able to represent contrast as large as the 4T sensor simulations, which indicates a limitation in the proposed scheme. A simulation with an input of 114 dB was also presented. The 114 dB simulation result was compared with the original input image, and it was possible to see that the chip was able to preserve image details. All HDR circuits were designed and laid out using 180 nm CIS technology parameters.

The next immediate step of the project is to experimentally test the designed chip and compare the results with the model predictions. A test bench is being designed, where an FPGA will be used to generate the chip control signals. Analog to digital converters need to be connected to the chip outputs. We will use 12-bit converters, allowing for the assessment of using either 8 bits (most significant bits of the conversion) or 12 bits. The 4T matrix 12-bit results can be used for comparing the HDR results with the linear results, but also for the implementation of the operator digital approach. Furthermore, the total processing time between the two approaches can be experimentally measured and compared with the theoretical model. Image quality comparison between these two approaches must also be performed considering both HDR situations and low dynamic range situations. The proposed chip has a linear response when $T_{Max} < T_s$ and $T_s < T_{sMax}$, which can be interesting in low dynamic range situations. Even using the linear response of the sensor, high average value situations still represent a limitation because the entire output range can not be covered. Reducing V_m helps in increasing the output range in high average value cases, which can make the HDR sensor linear output similar to the regular 4T sensor response.

The techniques presented in [114] can be applied to the regular 4T matrix for pinned photodiode characterization. These techniques allow for several measurements: pinning voltage, pinned photodiode capacitance, pixel well capacity, transfer gate threshold voltage, and transfer gate channel potential at a given gate voltage. The operation is based on controlling the reset transistor drain voltage and use this voltage to inject charge directly in the pinned photodiode through the floating diffusion. The circuit modification is thus very simple, and was implemented in the test matrices: the reset transistor drain has a separate control bus. The extraction of these characteristics is important for future designs using the same technology.

It is also important to measure the noise profile of the 4T matrix in two situations: when operating as commonly seen in the literature, where a pulse activates the transfer gate by the end of the integration period [35]; and when operating as the designed HDR pixel operates, with the transfer gate kept on during the integration period and turned off only by the end. By comparing these two measurements, we can estimate the amount of noise being introduced in the designed HDR pixel because of the capture photodiode operation mode, in which the transfer gate is on during the integration period. Overall sensor noise will also be measured. The matrices on which standard photodiodes were integrated will allow for the assessment of the pinned photodiode advantages with respect to noise and light sensitivity. The extraction of the response curves for different averages, as the ones presented in Figure 4.8, for both standard and pinned matrices, can help understanding the pixel operation in practice and verifying the model fidelity. It will be challenging,

though, to control the average variation and pixel value as described in Section 4.1 for system-level simulations. Guaranteeing the same illumination conditions in each matrix can also be challenging.

An automatic control of T_s, T_{Max} and V_m is also an interesting research topic. This investigation and experimental tests can be carried out simultaneously. V_m control can depend simply on image average. As observed in Section 5.2, the circuit output benefits from a V_m reduction for high average value conditions. Reducing V_m shifts the response curve, allowing for output range increase, and resulting in a brighter image. T_s, T_{Max} can be computed using not only the average, but also the image histogram.

A possible circuit modification regards the inclusion of a pixel output signal to indicate the end of the integration period of a pixel. This allows for asynchronous pixel reading and can improve overall system throughput. Extra circuitry, such as the one presented in [112], would be necessary for the address event representation which would impact the fill-factor. The circuit would also benefit from well-capacity adjustment techniques, to avoid the clamp in bright pixels. Investigating whether the technique presented in [27] can be applied to the proposed scheme is also very interesting.

Bibliography

[1] REINHARD, E., DEVLIN, K. "Dynamic range reduction inspired by photoreceptor physiology", *IEEE Transactions on Visualization and Computer Graphics*, v. 11, n. 1, pp. 13–24, Jan 2005.

[2] LEE, T.-H., GUIDASH, R. M., LEE, P. R. "PARTIALLY PINNED PHOTODIODE FOR SOLID STATE IMAGE SENSORS". `http://www.google.it/patents/US4741207`, 2000. US Patent 6,051,447.

[3] KRISS, M. *Handbook of Digital Imaging*. 1 ed. , John Wiley & Sons, 2015.

[4] BOUKHAYMA, A. *Ultra Low Noise CMOS Image Sensors*. 1 ed. , Springer International Publishing, 2010.

[5] NAKAMURA, J. *Image Sensors and Signal Processing for Digital Still Cameras*. 1 ed. EUA, CRC Press, Talyor & Francis Group, 2006.

[6] BELCHAIR, A. N. *Smart Cameras*. New York, Springer, 2010.

[7] AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys Tutorials*, v. 17, n. 4, pp. 2347–2376, 2015.

[8] FONTAINE, R. "The state of the art of mainstream CMOS image sensors". In: *Int. Image Sensor Workshop*, Jan 2015.

[9] FERNÁNDEZ-BERNI, J., CARMONA-GALÁN, R., DEL RÍO, R., et al. "Focal-Plane Sensing-Processing: A Power-Efficient Approach for the Implementation of Privacy-Aware Networked Visual Sensors", *Sensors*, , n. 14, 2014.

[10] RAMESH, B., GEORGE, A. D., LAM, H. "Real-time, low-latency image processing with high throughput on a multi-core SoC". In: *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, Sept 2016.

[11] FERNÁNDEZ-BERNI, J., CARMONA-GALÁN, R., CARRANZA-GONZÁLEZ, L. "FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing", *IEEE Journal of Solid-State Circuits*, v. 46, n. 3, pp. 669–680, March 2011.

[12] SUÁREZ, M., BREA, V. M., FERNÁNDEZ-BERNI, J., et al. "CMOS-3D Smart Imager Architectures for Feature Detection", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, v. 2, n. 4, pp. 723–736, Dec 2012.

[13] RUSH, A., WOOD, S. L. "Focal plane processing for HOG detection with Bayer pattern sensors". In: *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 1603–1607, Nov 2016.

[14] LUO, Y., MIRABBASI, S. "A CMOS pixel design with binary space-time exposure encoding for computational imaging". In: *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, April 2017.

[15] DE MORAES CRUZ, C. A., DE LIMA MONTEIRO, D. W., COTTA, E. A., et al. "FPN Attenuation by Reset-Drain Actuation in the Linear-Logarithmic Active Pixel Sensor", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 61, n. 10, pp. 2825–2833, Oct 2014.

[16] ILLADE-QUINTEIRO, J., BREA, V. M., LPEZ, P., et al. "Time-of-flight chip in standard CMOS technology with in-pixel adaptive number of accumulations". In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1910–1913, May 2016.

[17] NERO BARDALLO, J. A. L., CARMONA-GALN, R., RODRGUEZ-VZQUEZ, . "A Wide Linear Dynamic Range Image Sensor Based on Asynchronous Self-Reset and Tagging of Saturation Events", *IEEE Journal of Solid-State Circuits*, v. 52, n. 6, pp. 1605–1617, June 2017.

[18] LIN, Z., HOFFMAN, M. W., SCHEMM, N., et al. "A CMOS Image Sensor for Multi-Level Focal Plane Image Decomposition", *IEEE Trans. Circuits and Systems*, v. 55, n. 9, pp. 2561–2572, 2008.

[19] CHEN, S., BERMAK, A., WANG, Y. "A CMOS image sensor with onchip image compression based on predictive boundary adaptation and memoryless QTD algorithm", *Very Large Scale Integration Systems (VLSI), IEEE Transactions on*, v. 19, n. 4, pp. 538–547, 2011.

[20] FERNÁNDEZ-BERNI, J., CARMONA-GALÁN, R., RODRÍGUEZ-VÁZQUEZ, A. "Image filtering by reduced kernels exploiting kernel structure and focal-plane averaging". In: *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*, pp. 230–233, Aug 2011.

[21] OLIVEIRA, F., GOMES, J. G., CARMONA-GALÁN, R., et al. "Focal-Plane Scale Space Generation with a 6T Pixel Architecture". In: *IS&T Electronic Imaging*, 2016.

[22] YADID-PECHT, O., ETIENNE-CUMMINGS, R. *CMOS Imagers: From Phototransduction to Image Processing*. 1 ed. EUA, Kluwer Academic Publishers, 2004.

[23] OTHA, J. *Smart CMOS Image Sensors and Applications*. 1 ed. EUA, CRC Press, Talyor & Francis Group, 2008.

[24] NARWARIA, M., PERREIRA DA SILVA, M., LE CALLET, P. "High Dynamic Range Visual Quality of Experience Measurement: Challenges and Perspectives". In: *Visual Signal Quality Assessment - Quality of Experience (QoE)*, Springer International Publishing, pp. 129–155, 2015. doi: 10.1007/978-3-319-10368-6_5.

[25] HOEFFLINGER, B. *High-Dynamic-Range (HDR) Vision*. 1 ed. , Springer-Verlag Berlin Heidelberg, 2007.

[26] DARMONT, A. *High Dynamic Range Imaging - Sensors and Architectures*. 1 ed. Bellingham, Washington 98227-0010 USA, SPIE Press, 2012.

[27] DECKER, S., MCGRATH, D., BREHMER, K., et al. "A 256 × 256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output", *IEEE Journal of Solid-State Circuits*, v. 33, n. 12, pp. 2081–2091, Dec 1998.

[28] DE M. CRUZ, C. A. "Simplified Wide Dynamic Range CMOS Image Sensor with 3T APS Reset-Drain Actuation", *Ph.D. Thesis, UFMG*, 2014.

[29] SPIVAK, A., BELENKY, A., FISH, A., et al. "Wide-Dynamic-Range CMOS Image Sensors – Comparative Performance Analysis", *IEEE Transactions on Electron Devices*, v. 56, n. 11, pp. 2446–2461, Nov 2009.

[30] ADACHI, S., LEE, W., AKAHANE, N., et al. "A 200-μ Ve$^-$ CMOS Image Sensor With 100-ke$^-$ Full Well Capacity", *IEEE Journal of Solid-State Circuits*, v. 43, n. 4, pp. 823–830, April 2008.

[31] LEÑERO-BARDALLO, J. A., CARMONA-GALÁN, R., RODRÍGUEZ-VÁZQUEZ, A. "A High dynamic range linear vision sensor with event asynchronous and frame-based synchronous operation". In: *IS&T Electronic Imaging*, 2016.

[32] VARGAS-SIERRA, S., LIÑÁN-CEMBRANO, G., RODRÍGUEZ-VÁZQUEZ, A. "A 151 dB High Dynamic Range CMOS Image Sensor Chip Architecture With Tone Mapping Compression Embedded In-Pixel", *IEEE Sensors Journal*, v. 15, n. 1, pp. 180–195, Jan 2015.

[33] FERNÁNDEZ-BERNI, J., OLIVEIRA, F. D. V. R., CARMONA-GALÁN, R., et al. "Image Sensing Scheme Enabling Fully-Programmable Light Adaptation and Tone Mapping With a Single Exposure", *IEEE Sensors Journal*, v. 16, n. 13, pp. 5121–5122, July 2016.

[34] LOWE, D. "Distinctive Image Features from Scale-Invariant Keypoints", *Int. J. Comput. Vision*, v. 60, n. 2, pp. 91–110, nov. 2004.

[35] FOSSUM, E. R., HONDONGWA, D. B. "A Review of the Pinned Photodiode for CCD and CMOS Image Sensors", *IEEE Journal of the Electron Devices Society*, v. 2, n. 3, pp. 33–43, May 2014.

[36] SZELISKI, R. *Computer Vision: Algorithms and Applications*. 1st ed. New York, NY, USA, Springer-Verlag New York, Inc., 2010.

[37] VIOLA, P., JONES, M. "Robust real-time face detection", *International Journal of Computer Vision*, v. 57, pp. 137–154, 2004.

[38] VAN NOORD, N., POSTMA, E. O. "Learning scale-variant and scale-invariant features for deep image classification", *Pattern Recognition*, v. 61, pp. 583–592, 2017.

[39] ADELSON, E., ANDERSON, C., BERGEN, J., et al. "Pyramid methods in image processing", *RCA Engineer*, v. 29, n. 6, pp. 33–41, 1984.

[40] GONZÁLEZ, R. C., WOODS, R. E. *Digital Image Processing*. Upper Saddle River, NJ, USA, Prentice-Hall, Inc., 2006.

[41] LINDEBERG, T. "Scale-space theory: A basic tool for analysing structures at different scales", *Journal of Applied Statistics*, v. 21, n. 2, pp. 224–270, 1994.

[42] BABAUD, J., WITKIN, A. P., BAUDIN, M., et al. "Uniqueness of the Gaussian Kernel for Scale-Space Filtering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-8, n. 1, pp. 26–33, Jan 1986.

[43] "OpenVX". https://www.khronos.org/openvx/, . Accessed: 2016-11-02.

[44] KOBAYASHI, H., WHITE, J. L., ABIDI, A. A. "An active resistor network for Gaussian filtering of images", *IEEE Journal of Solid-State Circuits*, v. 26, n. 5, pp. 738–748, May 1991.

[45] NI, Y., ZHU, Y. M., ARION, B., et al. "Yet another analog 2D Gaussian convolver". In: *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, pp. 192–195 vol.1, May 1993.

[46] KABBAI, L., SGHAIERY, A., DOUIK, A., et al. "FPGA implementation of filtered image using 2D Gaussian filter", *Advanced Computer Science and Applications, International Journal of*, v. 7, n. 7, 2016.

[47] RAJAN, B., RAVI, S. "FPGA Based Hardware Implementation of Image Filter With Dynamic Reconfiguration Architecture", *Computer Science and Network Security, International Journal of*, v. 6, n. 12, 2006.

[48] ZHANG, H., XIA, M., HU, G. "A Multiwindow Partial Buffering Scheme for FPGA-Based 2-D Convolvers", *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 54, n. 2, pp. 200–204, Feb 2007.

[49] "OpenCV". opencv.org/, . Accessed: 2018-02-23.

[50] "Affine Covariant Features". http://www.robots.ox.ac.uk/~vgg/research/affine/, . Accessed: 2018-01-29.

[51] "OpenCV SIFT Documentation". http://docs.opencv.org/2.4/modules/nonfree/doc/feature_detection.html. Accessed: 2018-02-23.

[52] OLIVEIRA, F. D. V. R., GOMES, J. G. R. C., FERNÁNDEZ-BERNI, J., et al. "Gaussian Pyramid: Comparative Analysis of Hardware Architectures", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 64, n. 9, pp. 2308–2321, Sept 2017.

[53] LEÑERO-BARDALLO, J. A., RODRÍGUEZ-VÁZQUEZ, A. *Book chapter: ADCs for Image Sensors: Review and Performance Analysis in Analog Electronics for Radiation Detection*. CRC Press., 2016.

[54] KIYOYAMA, K., LEE, K. W., FUKUSHIMA, T., et al. "A very low area ADC for 3-D stacked CMOS image processing system". In: *3D Systems Integration Conference (3DIC), 2011 IEEE International*, pp. 1–4, Jan 2011.

[55] KIM, H. J., HWANG, S. I., KWON, J. W., et al. "Delta readout scheme for image-dependent power savings in a CMOS image sensor with multi-column-parallel SAR ADCs". In: *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 1–4, Nov 2015.

[56] LIN, J. Y., CHANG, K. H., KAO, C. C., et al. "An 8-bit column-shared SAR ADC for CMOS image sensor applications". In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 301–304, May 2015.

[57] KIM, M. K., HONG, S. K., KWON, O. K. "A Small-Area and Energy-Efficient 12-bit SA-ADC With Residue Sampling and Digital Calibration forCMOS Image Sensors", *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 62, n. 10, pp. 932–936, Oct 2015.

[58] CHEN, D. G., TANG, F., BERMAK, A. "A Low-Power Pilot-DAC Based Column Parallel 8b SAR ADC With Forward Error Correction for CMOS Image Sensors", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 60, n. 10, pp. 2572–2583, Oct 2013.

[59] MATSUO, S., BALES, T. J., SHODA, M., et al. "8.9-Megapixel Video Image Sensor With 14-b Column-Parallel SA-ADC", *IEEE Transactions on Electron Devices*, v. 56, n. 11, pp. 2380–2389, Nov 2009.

[60] CHEN, D. G., TANG, F., LAW, M. K., et al. "A 64 fJ/step 9-bit SAR ADC Array With Forward Error Correction and Mixed-Signal CDS for CMOS Image Sensors", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 61, n. 11, pp. 3085–3093, Nov 2014.

[61] SHIN, M. S., KIM, J. B., KIM, M. K., et al. "A 1.92-Megapixel CMOS Image Sensor With Column-Parallel Low-Power and Area-Efficient SA-ADCs", *IEEE Transactions on Electron Devices*, v. 59, n. 6, pp. 1693–1700, June 2012.

[62] XU, R., NG, W. C., YUAN, J., et al. "A 1/2.5 inch VGA 400 fps CMOS Image Sensor With High Sensitivity for Machine Vision", *IEEE Journal of Solid-State Circuits*, v. 49, n. 10, pp. 2342–2351, Oct 2014.

[63] LE-THAI, H., XHAKONI, A., GIELEN, G. "A column-and-row-parallel CMOS image sensor with thermal and 1/f noise suppression techniques". In: *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, pp. 221–224, Sept 2016.

[64] XHAKONI, A., LE-THAI, H., GIELEN, G. G. E. "A Low-Noise High-Frame-Rate 1-D Decoding Readout Architecture for Stacked Image Sensors", *IEEE Sensors Journal*, v. 14, n. 6, pp. 1966–1973, June 2014.

[65] CHAE, Y., CHEON, J., LIM, S., et al. "A 2.1 M Pixels, 120 Frame/s CMOS Image Sensor With Column-Parallel $\Delta\Sigma$ ADC Architecture", *IEEE Journal of Solid-State Circuits*, v. 46, n. 1, pp. 236–247, Jan 2011.

[66] OIKE, Y., GAMAL, A. E. "CMOS Image Sensor With Per-Column $\Sigma\Delta$ ADC and Programmable Compressed Sensing", *IEEE Journal of Solid-State Circuits*, v. 48, n. 1, pp. 318–328, Jan 2013.

[67] OIKE, Y., AKIYAMA, K., HUNG, L. D., et al. "An 8.3M-pixel 480fps global-shutter CMOS image sensor with gain-adaptive column ADCs and 2-on-1 stacked device structure". In: *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pp. 1–2, June 2016.

[68] SPIVAK, A., BELENKY, A., YADID-PECHT, O. "Very Sensitive Low-Noise Active-Reset CMOS Image Sensor With In-Pixel ADC", *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 63, n. 10, pp. 939–943, Oct 2016.

[69] LIM, Y., KOH, K., KIM, K., et al. "A 1.1e- temporal noise 1/3.2-inch 8Mpixel CMOS image sensor using pseudo-multiple sampling". In: *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 396–397, Feb 2010.

[70] KLEINFELDER, S., LIM, S., LIU, X., et al. "A 10000 frames/s CMOS digital pixel sensor", *IEEE Journal of Solid-State Circuits*, v. 36, n. 12, pp. 2049–2059, Dec 2001.

[71] SNOEIJ, M. F., THEUWISSEN, A. J. P., MAKINWA, K. A. A., et al. "Multiple-Ramp Column-Parallel ADC Architectures for CMOS Image Sensors", *IEEE Journal of Solid-State Circuits*, v. 42, n. 12, pp. 2968–2977, Dec 2007.

[72] LIM, S., LEE, J., KIM, D., et al. "A High-Speed CMOS Image Sensor With Column-Parallel Two-Step Single-Slope ADCs", *IEEE Transactions on Electron Devices*, v. 56, n. 3, pp. 393–398, March 2009.

[73] TOYAMA, T., MISHINA, K., TSUCHIYA, H., et al. "A 17.7Mpixel 120fps CMOS image sensor with 34.8Gb/s readout". In: *2011 IEEE International Solid-State Circuits Conference*, pp. 420–422, Feb 2011.

[74] NITTA, Y., MURAMATSU, Y., AMANO, K., et al. "High-Speed Digital Double Sampling with Analog CDS on Column Parallel ADC Architecture for Low-Noise Active Pixel Sensor". In: *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, pp. 2024–2031, Feb 2006.

[75] LEE, J., PARK, H., SONG, B., et al. "High Frame-Rate VGA CMOS Image Sensor Using Non-Memory Capacitor Two-Step Single-Slope ADCs", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 62, n. 9, pp. 2147–2155, Sept 2015.

[76] LYU, T., YAO, S., NIE, K., et al. "A 12-Bit High-Speed Column-Parallel Two-Step Single-Slope Analog-to-Digital Converter (ADC) for CMOS Image Sensors", *Sensors*, , n. 14, 2014.

[77] BAE, J., KIM, D., HAM, S., et al. "A Two-step A/D Conversion and Column Self-Calibration Technique for Low Noise CMOS Image Sensors", *Sensors*, , n. 14, 2014.

[78] TANG, F., WANG, B., BERMAK, A., et al. "A Column-Parallel Inverter-Based Cyclic ADC for CMOS Image Sensor With Capacitance and Clock Scaling", *IEEE Transactions on Electron Devices*, v. 63, n. 1, pp. 162–167, Jan 2016.

[79] PARK, J. H., AOYAMA, S., WATANABE, T., et al. "A High-Speed Low-Noise CMOS Image Sensor With 13-b Column-Parallel Single-Ended Cyclic ADCs", *IEEE Transactions on Electron Devices*, v. 56, n. 11, pp. 2414–2422, Nov 2009.

[80] MASE, M., KAWAHITO, S., SASAKI, M., et al. "A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic A/D converters", *IEEE Journal of Solid-State Circuits*, v. 40, n. 12, pp. 2787–2795, Dec 2005.

[81] LIM, S., CHEON, J., CHAE, Y., et al. "A 240-frames/s 2.1-Mpixel CMOS Image Sensor With Column-Shared Cyclic ADCs", *IEEE Journal of Solid-State Circuits*, v. 46, n. 9, pp. 2073–2083, Sept 2011.

[82] KITAMURA, K., WATABE, T., SAWAMOTO, T., et al. "A 33-Megapixel 120-Frames-Per-Second 2.5-Watt CMOS Image Sensor With Column-Parallel Two-Stage Cyclic Analog-to-Digital Converters", *IEEE Transactions on Electron Devices*, v. 59, n. 12, pp. 3426–3433, Dec 2012.

[83] FURUTA, M., NISHIKAWA, Y., INOUE, T., et al. "A High-Speed, High-Sensitivity Digital CMOS Image Sensor With a Global Shutter and 12-bit Column-Parallel Cyclic A/D Converters", *IEEE Journal of Solid-State Circuits*, v. 42, n. 4, pp. 766–774, April 2007.

[84] PARK, J. H., AOYAMA, S., WATANABE, T., et al. "A High-Speed Low-Noise CIS with 12b 2-stage Pipelined Cyclic ADCs". In: *2011 International Image Sensor Workshop*, Jun 2011.

[85] CHOI, M. H., AHN, G. C., LEE, S. H. "12b 50 MS/s 0.18 μm CMOS ADC with highly linear input variable gain amplifier", *Electronics Letters*, v. 46, n. 18, pp. 1254–1256, September 2010.

[86] CHO, S.-H., PARK, J.-S., AHN, G.-C., et al. "A 14—10 B Dual-mode Low-noise Pipeline ADC for High-end CMOS Image Sensors", *Analog Integr. Circuits Signal Process.*, v. 80, n. 3, pp. 437–447, set. 2014.

[87] ZHANG, S., XIAOKANG, L., REN, G., et al. "A 12-Bit 96Msample/s double-data-rate (DDR) pipeline ADC with speed and noise optimization for CMOS image sensors". In: *2014 International Conference on Information Science, Electronics and Electrical Engineering*, v. 3, pp. 1798–1803, April 2014.

[88] PARK, J. S., AN, T. J., KIM, Y. M., et al. "A 10b 50MS/s 90nm CMOS skinny-shape ADC using variable references for CIS applications". In: *2013 International SoC Design Conference (ISOCC)*, pp. 080–082, Nov 2013.

[89] CHO, K. B., LEE, C., EIKEDAL, S., et al. "A 1/2.5 inch 8.1Mpixel CMOS Image Sensor for Digital Cameras". In: *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pp. 508–618, Feb 2007.

[90] MURMANN, B. "Trends in Low Power, Digitally Assisted A/D Conversion", *IEICE Transactions on Electronics*, v. 93, n. C(6), pp. 718–729, June 2010.

[91] PELGROM, M. *Analog-to-Digital Conversion*. 1 ed. , Springer, 2010.

[92] SULLIVAN, G. J., OHM, J. R., HAN, W. J., et al. "Overview of the High Efficiency Video Coding (HEVC) Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1649–1668, Dec 2012.

[93] WALDEN, R. H. "Analog-to-digital converter survey and analysis", *IEEE Journal on Selected Areas in Communications*, v. 17, n. 4, pp. 539–550, Apr 1999.

[94] ELDESOUKI, M., JAMAL DEEN, M., FANG, Q., et al. "CMOS Image Sensors for High Speed Applications", *Sensors*, v. 9, n. 1, 2009.

[95] JONSSON, B. E. "An empirical approach to finding energy efficient ADC architectures". In: *2011 International Workshop on ADC Modelling, Testing and Data Converter Analysis and Design and IEEE 2011 ADC Forum*, July 2011.

[96] RABAEY, J. *Low Power Design Essentials*. 233 Spring Street, New York, NY 10013, USA, Springer, 2009.

[97] YADID-PECHT, O. "Wide-dynamic-range sensors", *Optical Engineering*, v. 38, n. 10, pp. 1650–1660, 1999.

[98] REINHARD, E., HEIDRICH, W., DEBEVEC, P., et al. *High Dynamic Range Imaging*. 1 ed. Bellingham, Washington 98227-0010 USA, Morgan Kaufmann Publishers, 2006.

[99] SICARD, G., ABBAS, H. O., AMHAZ, H., et al. "A CMOS HDR Imager with an Analog Local Adaptation". In: *Intl. Image Sensor Workshop*, 2013.

[100] FOWLER, B., LIU, C., MIMS, S., et al. "A 5.5Mpixel 100 frames/sec wide dynamic range low noise CMOS image sensor for scientific applications". In: *Proc. of SPIE-IS&T Electronic Imaging*, v. 7536, 2010.

[101] CARMONA-GALÁN, R., LEÑERO-BARDALLO, J. A., FERNÁNDEZ-BERNI, J., et al. "Pixel-wise parameter adaptation for single-exposure extension of the image dynamic range". In: *International Conference on Distributed Smart Cameras*, 2016.

[102] BELMONTE, P. N. A., CHAVES, L. M., DE LIMA MONTEIRO, D. W. "A pixel concept that simultaneously enables high dynamic range, high sensitivity and operation in intense backgrounds". In: *2017 30th Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug 2017.

[103] BELENKY, A., FISH, A., SPIVAK, A., et al. "A Snapshot CMOS Image Sensor With Extended Dynamic Range", *IEEE Sensors Journal*, v. 9, n. 2, pp. 103–111, Feb 2009.

[104] "ETH Zürich, EMPA Media Technology, HDR Database". http://empamedia.ethz.ch/hdrdatabase/, . Accessed: 2017-07-19.

[105] WANG, Z., BOVIK, A., SHEIKH, H., et al. "Image quality assessment: from error visibility to structural similarity", *Image Processing, IEEE Transactions on*, v. 13, n. 4, pp. 600–612, abril de 2004.

[106] NUNES, G. M. S., OLIVEIRA, F. D. V. R., GOMES, J. G. R. C., et al. "Color Tone-Mapping Circuit for a Focal-Plane Implementation". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS) - Accepted paper*, 2018.

[107] NUNES, G. M. S. "A COMPARISON BETWEEN TONE-MAPPING ALGORITHMS FOR A FOCAL-PLANE IMPLEMENTATION", *Dissertação de Mestrado, COPPE/UFRJ*, 2018.

[108] VATTERONI, M., VALDASTRI, P., SARTORI, A., et al. "Linear-Logarithmic CMOS Pixel With Tunable Dynamic Range", *IEEE Transactions on Electron Devices*, v. 58, n. 4, pp. 1108–1115, April 2011. ISSN: 0018-9383.

[109] DE MORAES CRUZ, C. A., DE LIMA MONTEIRO, D. W., COTTA, E. A., et al. "FPN Attenuation by Reset-Drain Actuation in the Linear-Logarithmic Active Pixel Sensor", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 61, n. 10, pp. 2825–2833, Oct 2014.

[110] KIM, D., SONG, M. "An Enhanced Dynamic-Range CMOS Image Sensor Using a Digital Logarithmic Single-Slope ADC", *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 59, n. 10, pp. 653–657, Oct 2012.

[111] SICARD, G., ABBAS, H., AMHAZ, H., et al. "A CMOS HDR Imager with an Analog Local Adaptation". In: *International Image Sensor Workshop (IISW'13)*, pp. 1–4, Snowbird, United States, jun. 2013. Disponível em: <https://hal.archives-ouvertes.fr/hal-00968779>.

[112] LEÑERO-BARDALLO, J. A., CARMONA-GALÁN, R., RODRÍGUEZ-VÁZQUEZ, A. "A high dynamic range linear vision sensor with event asynchronous and frame-based synchronous operation", *Electronic Imaging*, v. 2016, n. 12, pp. 1–7, 2016.

[113] OLIVEIRA, F., HAAS, H., GOMES, J., et al. "CMOS Imager With Focal-Plane Analog Image Compression Combining DPCM and VQ", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, v. 60, n. 5, pp. 1331–1344, 2013.

[114] GOIFFON, V., ESTRIBEAU, M., MICHELOT, J., et al. "Pixel Level Characterization of Pinned Photodiode and Transfer Gate Physical Parameters in CMOS Image Sensors", *IEEE Journal of the Electron Devices Society*, v. 2, n. 4, pp. 65–76, July 2014.

Appendix A

HDR System-Level Scripts

Script used for system-level simulations:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HDR pixel with autonomous integration
% period system-level simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Constants:
Vdd = 3.3;
Vt=0.86;
Vrst = Vdd - Vt;
C = 20e-15;
Vmin = 0.7;
Vm = 1.6;
Tmax = 30000e-6;

mph = 1;
mc = 1;

% Reading the image and transforming
% the pixel values to photocurrents

% Camera information:
black = 2048; % For Canon 1100D, from dcraw
saturation = 15280;

% Pattern: RGGB
% Image 'img0.tiff' is the one presented in Figure 4.7(a)
```

```
I = imread('img0.tiff');

% Only the green information was considered
% for tone mapping:
Im1 = I(1:2:end,2:2:end);

% Normalization
p = (double(Im1)-black)/(saturation-black);
p(p<0) = 0;
p(p>1) = 1;

% Transforming the pixel values to
% photocurrents:
iph = K*p;


% Control capacitance
Cctrl = C/mc;

% Control local photocurrent
iphCtrl = iph/mph;

% Average photocurrent values
IphCtrlMedio = mean(mean(iphCtrl));
IphMedio = mean(mean(iph));

% The maximum value Ts can assume, TsMax,
% is given by the instant the contage in
% FDctrl crosses Vmid:
TsMax = (Vrst - Vm)*Cctrl/IphCtrlMedio;

% Varible that stores all generated images:
ImAll = zeros(size(iph,1),size(iph,2),2);


TsRef = (round(0.98*TsMax*1e6))*1e-6;

for i=TsRef/4:TsRef/4:TsRef
```

```
Ts = i;

if(Ts > TsMax)
    texto = 'Ts value is higher than the maximum value
       it can assume. Please define Ts < %f us.\n';
    fprintf(texto,TsMax*1e6);
    return;
end;

% To find out each pixel integration
% period, is necessary to know the voltage
% in the floating diffusion control node
% at the instant Ts. Before Ts all control
% floating diffusion nodes are connected
% together, and the discharge depends on
% the average value. After Ts each control
% floating diffusion node discharges
% according to the local photocurrent.

VctrlEmTs = Vrst - Ts*IphCtrlMedio/Cctrl;


% Tmid a matrix that holds each
% pixe integration period value.
% The integration period is defined when
% the voltage at the control floating
% diffusion crosses the middle of the
% dynamic range Vm. Tmid is defined by the
% control circuit:
Tmid = Cctrl*(VctrlEmTs + Ts*iphCtrl/Cctrl - Vm)./
   iphCtrl;
Tmid(find(Tmid > Tmax)) = Tmax;

% Once each pixel integration period
% was found, the caputre photodiode
% voltage (which is directly proportional)
% to the voltage output) is found using
% the pixel discharge equation. This
% valure depends on the capture circuit:
```

```
        VphHDR = Vrst - Tmid.*iph/C;
        VphHDR(VphHDR < Vmin) = Vmin;


        % Defining the output range of [0,255]:
        ImFromVphHDR = 255*(Vrst - VphHDR)./(Vrst-Vmin);

        % Displaying the image
        figure; imshow(uint8(kron(ImFromVphHDR,ones(4,4))));

        % Saving all images
        ImAll(:,:,ind) = uint8(ImFromVphHDR);
        ind = ind +1;

end;
```