UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-graduação da Engenharia Elétrica

Daniel Carvalho Lott

Design of 10 mA CMOS Current Driver Arrays for MEMS Actuators

Belo Horizonte
2025

Daniel Carvalho Lott

DESIGN OF 10 MA CMOS CURRENT DRIVER ARRAYS FOR MEMS ACTUATORS

Dissertation presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Supervisor: Prof. Dr. Dalton Martini Colombo

Federal University of Minas Gerais – UFMG
Graduate Program in Electrical Engineering – PPGEE
Laboratory for Optronics and Microtechnology Applications – OptMALab

Belo Horizonte - Brazil
2025

Lott, Daniel Carvalho.
L884d Design of 10 mA CMOS current driver arrays for MEMS actuators [recurso eletrônico] / Daniel Carvalho Lott. - 2025.
1 recurso online (110 f. : il., color.) : pdf.

Orientador: Dalton Martini Colombo.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Semicondutores complementares de óxido metálico - Teses. 3. Sistemas microeletromecânicos - Teses. I. Colombo, Dalton Martini. II.Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Reginaldo César Vital dos Santos CRB6/ 2165
Biblioteca Profº Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA - COLEGIADO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"DESIGN OF 10 MA CMOS CURRENT DRIVER ARRAYS FOR MEMS ACTUATORS"

DANIEL CARVALHO LOTT

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 06 de novembro de 2025.

Por:

Prof. Dr. Dalton Martini Colombo
DEE (UFMG) - Orientador

Prof. Dr. Kamal El-Sankary
Dep. ECE (Dalhousie University)

Prof. Dr. Hilton de Oliveira Mota
DEE (UFMG)

Prof. Dr. Paulo César Comasseto de Aguirre
(UNIPAMPA)

Documento assinado eletronicamente por **Dalton Martini Colombo**, **Professor do Magistério Superior**, em 06/11/2025, às 16:08, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Paulo César Comassetto de Aguirre**, **Usuário Externo**, em 06/11/2025, às 16:31, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Hilton de Oliveira Mota**, **Professor do Magistério Superior**, em 06/11/2025, às 17:24, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Kamal El-Sankary**, **Usuário Externo**, em 07/11/2025, às 15:06, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site

https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4693218** e o código CRC **B34F9F91**.

Acknowledgements

This project represents a major endeavor, involving multiple design instances, layout iterations, and chip versions. Its successful execution would not have been possible without the support, collaboration, and dedication of many individuals, institutions, and funding agencies.

First of all, I would like to express my deepest gratitude to my family and to my girlfriend, Maria Luiza, for the unconditional love, encouragement, and great emotional support during this entire journey. Their patience, understanding, and motivation were essential in helping me through the most difficult times during this work.

I would like to sincerely thank my supervisor Dr. Dalton Colombo for all his technical guidance, mentorship, and continuous support during the development of this research.

My gratitude extends to Pedro Sartori for the design of the SPI of the Chip V1, and Bruno Guimarães, and Bárbara Rocha Raimundo for designing the layout of the SPI for Chip V2. I am also deeply thankful to Felipe Magalhães, from Impinj, for his assistance and guidance during the layout development phase.

I would like to thank João Vitor Saade and Bárbara Rocha Raimundo for the testing and characterization of Chip V1. My appreciation also goes to Maria Luiza Fonseca and Bernard Andrade Lucas for their help in preparing and refining the figures and images used in this dissertation.

I would also like to acknowledge the National Research Council Canada (NRC) for funding the fabrication of the integrated circuit, which was crucial for the experimental validation of this work. In particular, I express my sincere gratitude to Dr. Kamal El-Sankary and Mr. Greg Burley, whose technical involvement and facilitation were instrumental in making this possible. Their support ranged from discussions during the first design version, contributions to the PCB development and component sourcing, coordination with the fabrication foundry, and most importantly, enabling the NRC funding that made the fabrication process feasible.

Regarding the Electronic Design Automation (EDA) tools, I would like to acknowledge SBMicro (APCI Program) for providing access to the Cadence design software tools. Additionally, I express my appreciation to Cadence Design Systems for the institutional support made available during the development of this research.

This work was also made possible through financial support from the following funding agencies: Instituto Serrapilheira (grant Serra–process 2211-42117); FAPEMIG

(APQ-05837-23 and APQ-04579-25); CNPq (404467/2024-5 and 308900/2025-1); and CAPES. These agencies provided funding for conference participation, acquisition of laboratory equipment (computers, UPS devices, consumables), software maintenance, and other essential research expenses.

RESUMO

Drivers de corrente são componentes eletrônicos fundamentais que possibilitam o controle e fornecimento de corrente de forma precisa, exata e ajustável para outros circuitos. Esses circuitos atuam como uma interface entre o mundo digital e os sistemas físicos, que são inerentemente analógicos. Um exemplo de aplicação nesse contexto é a tecnologia de óptica adaptativa, cujo objetivo é obter imagens nítidas de corpos celestes por meio de um telescópio e seu circuito de controle. Esse tipo de sistema é composto por um conjunto de atuadores mecânicos responsáveis por controlar e ajustar a posição de espelhos MEMS, a fim de corrigir distorções atmosféricas nas imagens. Controlando a corrente que percorre cada atuador mecânico, é possível ajustar a posição dos espelhos. O foco desta dissertação de mestrado é o projeto de drivers de corrente para esse tipo de sistema. Esta dissertação aborda duas gerações de drivers de corrente integrados. Ambas as topologias foram projetadas para fornecer correntes de saída de até ±10 mA utilizando tecnologia CMOS de 0,35 μm. A palavra digital, composta por 11 bits na versão 1 e 10 bits na versão 2, é fornecida ao driver por meio de um circuito de interface série-paralelo (SPI). A primeira versão é composta, de forma simplificada, por um conversor digital-analógico de tensão (DAC) e por um conversor de tensão-corrente, ambos operando com uma tensão de alimentação de ±3,3 V. A segunda versão é composta por um DAC de corrente segmentado, com os objetivos de: (i) aumentar a escalabilidade do número de drivers em um mesmo chip de silício; (ii) melhorar as especificações de linearidade da relação entre corrente de saída e dado digital de entrada, mais especificamente INL e DNL; e (iii) reduzir a tensão de alimentação. Esta segunda versão foi planejada e projetada para a operação simultânea de 40 a 50 drivers, integrados em uma área de silício de 20 mm². As simulações *pós-layout* indicaram um consumo de potência nominal de 17,6 mW, e a análise de linearidade baseada em 1000 pontos de Monte Carlo (incluindo variações de processo e descasamento) resultou em um INL médio de pior caso de 0,57 LSB e um DNL médio de pior caso de 0,27 LSB.

Palavras-chave: drivers de corrente, matriz de drivers, atuadores de MEMS, alta corrente, CMOS

ABSTRACT

Current drivers are fundamental electronic components that enable precise, accurate, and adjustable current control and supply to other circuits. These circuits provide an interface between the digital world and physical systems, which are inherently analog. An example of application in this context is adaptive optics technology, whose goal is to obtain sharp images of celestial bodies through a telescope and its control circuit. This type of system consists of a set of mechanical actuators responsible for controlling and adjusting the position of MEMS mirrors in order to correct atmospheric distortions in the images. By controlling the current flowing through the mechanical actuator, the mirrors' position can be adjusted. The focus of this master's thesis is the design of current drivers for such a system. This dissertation addresses two generations of integrated current drivers. Both topologies are designed to deliver output currents up to ±10 mA using 0.35 μm CMOS technology. The digital word, composed of 11 bits for version 1 and 10 bits for version 2, is sent into the driver through a serial-to-parallel interface circuit (SPI). The first version is composed, in a simplified way, of a voltage digital-to-analog converter (DAC) and a voltage-to-current converter, both operating with a supply voltage of ±3.3 V. The second version is composed of a segmented current DAC, with the goals of: (i) increasing the scalability of the number of drivers on the same silicon die, (ii) improving the linearity specifications of the output current versus input digital data relationship, more specifically INL and DNL, and (iii) reducing the supply voltage. This second version was planned and designed for the simultaneous operation of 40–50 drivers, integrated in a silicon area of 20 mm^2. Post-layout simulations indicate a nominal power consumption of 17.6 mW, and the linearity analysis based on 1,000-point Monte Carlo simulations (including process and mismatch variations) resulted in a mean worst-case INL of 0.57 LSB and a mean worst-case DNL of 0.27 LSB.

Keywords: current drivers, driver arrays, MEMS actuators, high current, CMOS

List of Figures

Figure 1 – Conceptual drawing of the conventional adaptive optics system. 17
Figure 2 – Block diagram of the proposed driver. 18
Figure 3 – Ideal output current versus input digital code. 23
Figure 4 – Local INL of the sampled output current as a function of the input digital code. 24
Figure 5 – Local DNL of the sampled output current as a function of the input digital code. 25
Figure 6 – Signal magnitude plot in the frequency domain illustrating SFDR measurement. 26
Figure 7 – Block Diagram of the proposed MEMS deformable mirror system from work [1]. 29
Figure 8 – Proposed chip V1 architecture. 33
Figure 9 – Proposed driver architecture for chip V1. 34
Figure 10 – a) PNLS and b) NPLS circuits for chip V1. 34
Figure 11 – Voltage-to-Current converter circuit for driver from chip V1. 35
Figure 12 – Output stage circuit of the driver from chip V1. 36
Figure 13 – Full current range variation of the driver for chip V1. 37
Figure 14 – Plot of regional current variation for the driver in chip V1, highlighting its resolution. 37
Figure 15 – AVDD voltage variation and corresponding output current response for PSRR measurement. 38
Figure 16 – Output current variation with increasing supply voltage magnitude for nominal 10 mA and -10 mA output currents, respectively. 39
Figure 17 – Histograms of Monte Carlo simulation results for the output current at nominal –10 mA and 10 mA, respectively. 39
Figure 18 – Corners simulation results for the output current at nominal 10 mA. . . 40
Figure 19 – Measured full output current range of the driver for chip V1, obtained from five fabricated samples. 41
Figure 20 – Measured a) positive DNL and b) negative DNL for the driver in chip V1 across five samples. 41
Figure 21 – Measured a) positive INL and b) negative INL for the driver in chip V1 across five samples. 41
Figure 22 – IR Drop DC results for output current of 10 mA. 43
Figure 23 – IR Drop DC results for output current of -10 mA. 43
Figure 24 – EM DC results for output current of 10 mA. 44
Figure 25 – EM DC results for output current of -10 mA. 44

Figure 26 – Diagram of voltage mode topology using buffers. 47
Figure 27 – Diagram of voltage reference circuit. 47
Figure 28 – Diagram of the modified output stage. 49
Figure 29 – Diagram of the fully binary topology. 50
Figure 30 – Diagram of the fully unary topology. 51
Figure 31 – Diagram of the segmented topology. 54
Figure 32 – SPI controller to SPI peripheral block diagram. 57
Figure 33 – Expected behavior of the SPI output signals for a specific 'MOSI' input. 58
Figure 34 – Expected behavior of the SPI output signals under changes in the cs_n signal. 59
Figure 35 – Results from SPI simulations under nominal operation. 60
Figure 36 – SPI layout. 62
Figure 37 – Segmented DAC topology developed. 63
Figure 38 – Unary blocks of the segmented DAC topology developed. 64
Figure 39 – Diode-connected transistors of the current mirror in the developed segmented DAC topology. 64
Figure 40 – Current cell of the NMOS sink block in the developed segmented DAC topology. 65
Figure 41 – Current cell of the PMOS source block in the developed segmented DAC topology. 66
Figure 42 – NMOS sink binary branches in the developed segmented DAC topology. 67
Figure 43 – PMOS sources binary branches in the developed segmented DAC topology. 67
Figure 44 – Switch architecture for the cascode in the NMOS sink blocks. 68
Figure 45 – Switch architecture for the cascode in the PMOS source blocks. 69
Figure 46 – Figure: 6-to-64-bit decoder block showing its input and output connections. 70
Figure 47 – Detailed block diagram of the 6-to-64-bit decoder architecture. 70
Figure 48 – Schematic circuits of the 3-to-8-bit decoder. 71
Figure 49 – Schematic circuits of the digital logic applied to the current direction bit to ensure correct activation of binary and unary branches. 72
Figure 50 – Alternative segmented DAC topology tested. 73
Figure 51 – 66-bit register block showing its input and output connections. 74
Figure 52 – Top-level schematic of the proposed chip with 50 drivers. 74
Figure 53 – Block diagram of the test circuits from the proposed chip with 50 drivers. 75
Figure 54 – Block diagram showing the implemented level shifters of the proposed chip. 76
Figure 55 – Schematic circuits of the PNLS implemented in the proposed chip. . . . 76
Figure 56 – Schematic circuit of the reference current generator. 77
Figure 57 – Block diagram of the connections between the developed SPI unit and the 50 drivers of the proposed chip. 78

Figure 58 – Layout of the proposed driver. 79
Figure 59 – Illustration of the proposed chip's usable area and the space occupied by the PAD ring. 80
Figure 60 – Floorplan of the chip, with 40 drivers. 82
Figure 61 – Histograms showing the variation of the worst-case INL from a) PMOS blocks and b) NMOS blocks across 1,000 Monte Carlo simulation points. The horizontal axis is expressed in LSB. 86
Figure 62 – SPI clock signal before and after the PNLS. 87
Figure 63 – Output current from transient simulation of the proposed driver with a full input sweep from 0 to 1023 in descending order. 88
Figure 64 – Zoomed-in view of the transient output current highlighting the binary and unary glitches, along with their respective glitch size values. 88
Figure 65 – Figure: Zoomed-in transient waveform of the output current during a unary bit transition, illustrating the worst-case driver's settling time. . 89
Figure 66 – Output current waveform of the proposed driver under sinusoidal excitation for SFDR analysis. 90
Figure 67 – 49-cycle output current waveform of the proposed driver under sinusoidal excitation used for SFDR analysis. 91
Figure 68 – Discrete Fourier Transform (DFT) of the 49-cycle output current waveform of the proposed driver used for SFDR measurement. 91
Figure 69 – Figure: Transient waveforms of the output current from three drivers after a full input sweep applied in specific order that results in ascending output currents from -10 mA to 10 mA. 92
Figure 70 – Zoomed-in transient waveforms of the output current from three drivers, illustrating their transition behavior. 93
Figure 71 – Initial DC EM results from proposed driver, with nominal output current of 10 mA. 94
Figure 72 – Initial DC EM results from proposed driver, with nominal output current of -10 mA. 94
Figure 73 – Initial Transient EM results from proposed driver, with full output current sweep. 95
Figure 74 – Final DC EM results from proposed driver, with nominal output current of 10 mA. 95
Figure 75 – Final DC EM results from proposed driver, with nominal output current of -10 mA. 95
Figure 76 – DC IR Drop results from proposed driver, with nominal output current of 10 mA. 96
Figure 77 – Transient IR Drop results from proposed driver, with full output current sweep. 96

List of Tables

Table 1 – Current Range Comparison Across Various Current Driver Applications 21
Table 2 – Specifications of the Current Driver proposed in [1] 30
Table 3 – Bit Sequences Used to Test the SPI and Their Expected Output Values 59
Table 4 – Binary to Thermometer Code Conversion (3-bit Example) 69
Table 5 – Estimated Silicon Area Breakdown for the Proposed Chip 81
Table 6 – Worst INL and DNL Metrics from Nominal DC Simulations 84
Table 7 – Worst-Case INL, DNL and Output Current Across PVT Corners 84
Table 8 – Monte Carlo simulation results based on 1,000 points. 85
Table 9 – Performance summary and comparison with other high-current driver implementations. 97

Contents

1 INTRODUCTION 16
1.1 Application 16
1.2 Defining the Architectural Challenge 17
1.3 Thesis Overview 19

2 LITERATURE REVIEW: CURRENT DRIVERS IN ADVANCED INTEGRATED CIRCUITS 20
2.1 Introduction, applications and design considerations of current drivers 20
2.2 Current range of current drivers 21
2.3 Current Drivers for MEMS Applications 21
2.4 Important concepts 22
2.4.1 Integral Non-Linearity (INL) 22
2.4.2 Differential Non-Linearity (DNL) 24
2.4.3 Spurious-Free Dynamic Range (SFDR) 26
2.4.4 Other parameters (SNR, SNDR and ENOB) 27
2.5 Chip V0: Voltage mode current driver [1] 28

3 FIRST DESIGNED PROJECT 32
3.1 Proposed Solution 32
3.2 Simulations and Performance Analysis 36
3.3 Driver EMIR Analysis 42
3.3.1 IR Drop Analysis results 42
3.3.2 Electromigration (EM) Analysis results 43

4 DESIGN EXPLORATION 46
4.1 Voltage Mode Topology Using Buffer Operational Amplifiers 46
4.2 V-to-I-Converter Topology Adapted to Reduce Supply Voltage . . . 48
4.3 Binary Current Steering DAC Topology 49
4.4 Unary Coded Current-Steering DAC 51
4.5 Segmented Current DAC Topology 53

5 THE DEVELOPED SERIAL PERIPHERAL INTERFACE 56
5.1 Basic concepts 56
5.2 Designed SPI 56
5.3 SPI simulations 59
5.4 Synthesis 61

5.5 Layout . . . 61

6 DESIGNED CURRENT DRIVER TOPOLOGY . . . 63
6.1 Schematic . . . 63
6.1.1 The current source and current sink cells . . . 65
6.1.2 The switch on CAS . . . 68
6.1.3 The thermometer decoder . . . 69
6.1.4 The current direction bit . . . 71
6.1.5 The bank of registers . . . 73
6.2 The chip . . . 74
6.2.1 Test Circuit . . . 75
6.2.2 Level Shifters . . . 75
6.2.3 Reference Current Generator . . . 76
6.2.4 SPI + Drivers . . . 77
6.3 Layout . . . 78
6.3.1 Full chip layout area estimation . . . 79
6.3.1.1 PADs area estimation . . . 79
6.3.1.2 SPI unit area . . . 80
6.3.1.3 Drivers area estimation . . . 80
6.3.1.4 Auxiliary circuits area estimation . . . 80

7 SIMULATIONS . . . 83
7.1 Optimizations . . . 83
7.2 Driver DC Simulations . . . 83
7.2.1 INL and DNL performance . . . 83
7.2.2 PVT Corner analyses . . . 84
7.2.3 MC analysis . . . 85
7.3 Driver transient simulations . . . 86
7.3.1 Level Shifter test . . . 86
7.3.2 Single Driver Transient Test . . . 87
7.3.3 Full Chip Transient Test: SPI + Drivers . . . 92
7.4 Power consumption analysis . . . 93
7.5 EMIR Analyses . . . 93
7.6 Comparison Table . . . 97

8 CONCLUSION AND FUTURE WORK . . . 99

BIBLIOGRAPHY . . . 101

ANNEX 105

ANNEX A – SPI VERILOG CODE 106

ANNEX B – SPI GENUS REPORTS 110

1 Introduction

1.1 Application

Adaptive optics (AO) is an evolving area that aims to allow for ground-based astronomy to have close quality to space-based instruments, facing the issues with Earth's atmosphere. The performance of the optical system is highly limited by the dynamic nature of this atmosphere, since several distortion effects happen due to the gases and turbulence from it, and therefore AO tries to correct these distortions.

A typical AO setup comprises several critical components: a telescope that collects the incoming light, a wavefront sensor (WFS) that measures the optical aberrations, a real-time control system that processes the WFS data, and a deformable mirror (DM) which is the main component responsible for physically correcting the distorted wavefront. The aberrations mentioned here are the distortion effects introduced when light passes through the Earth's atmosphere, during ground-based observations. The DM dynamically adjusts its surface shape, compensating for the optical distortions, restoring the coherence of the incoming light and enabling shaper and more detailed images.

The actuator used for this application is a Lorentz force MEMS actuator. This is the transducer for driving the deformable mirror, and it can be modeled as a 50-ohm grounded resistor. This solution has compelling advantages over more conventional deformable mirror technologies, being a good candidate for high-performance AO systems. Advantages such as its capability for low-voltage operation, which helps in reducing power consumption and having simpler driving electronic devices. Also the absence of magnetic hysteresis, ensuring highly repeatable and precise mirror positioning. Furthermore, the actuator has compact size facilitating the development of smaller and more integrated arrays of AO modules. The Lorentz actuators also enable bi-directional motion (push and pull motion), having greater control over the shape of the mirror's surface and better correction of surface flatness. [1]

The overall system architecture integrates the DM into the proposed adaptive optics loop, as shown in Figure 1. Incoming light, after passing through the atmospheric turbulence, is directed towards the deformable mirror via a beam-splitter. Part of this light is routed to a wavefront sensor, which analyzes the distortions. The information from this sensor is then used to adjust the DM, ensuring that its surface actively compensates for the aberrations detected, therefore creating a corrected wavefront for the science camera.

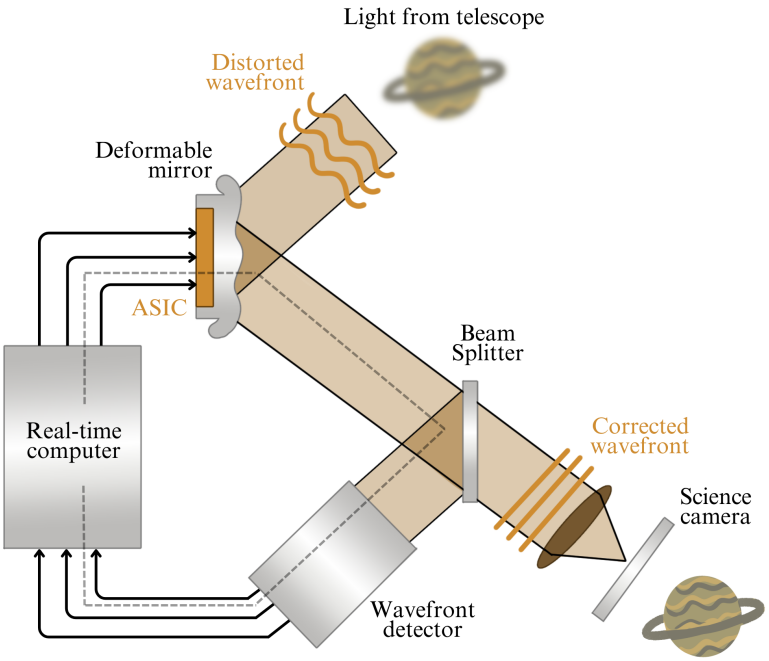

Figure 1 – Conceptual drawing of the conventional adaptive optics system.

1.2 Defining the Architectural Challenge

The motivation for the present work begins with the fundamental engineering challenge in next generation adaptive optics systems: the need for compatible drive electronics to support a high actuator count array. When the size of telescope primary mirrors increases, the complexity of the AO system increases too, and the need for more precise wavefront correction devices (DMs) also increases. A common limitation for all DM technologies is that scaling to high actuator counts results in proportional increase in the number of control signal wires, leading to the need for a scalable form of DM electronics driver topology. This is the core technological issue that requires efficient drive electronics, forming the context for the considerations made in this dissertation.

An important reference for contextualizing this dissertation is the research developed in the doctoral thesis “Next Generation Adaptive Optics” by Colin G. A. Ross (Dalhousie University, 2020) [1]. In particular, the driver solution described in Chapter 3 of that work, entitled “ASIC Driver Development”, presents a high-voltage architecture intended for MEMS deformable mirror actuation. Inspired by the concepts and challenges introduced in that work, the present dissertation proposes a completely new driver topology that operates at low voltage, emphasizing linearity, scalability and area efficiency. The insights and methodologies discussed in that prior research provided valuable context for subsequent research aimed at addressing the inherent limitations of these systems, particularly in the pursuit of higher performance and scalability.

The main requirements of the proposed current driver are as follows:

- Deliver output current values between -10 mA and 10 mA.
- Drive a resistive load with a nominal value of 50 ohms, subject to variations up to ±30%.
- Provide good current resolution of 10 bits.
- Provide good current robustness over corners (variations below 20%).
- Provide good linearity (output current x digital word). The linearity will be measured for each polarity (sink/source) separately.
- Be able to operate on all process corners and in temperatures that range from 0 to 100 ºC.
- Occupy a small silicon area, allowing better scalability, therefore being able to integrate dozens of drivers in a single chip. The ultimate goal is to integrate 50 drivers within an area of 20 mm².
- Offer easy digital control of the multiple driver inputs.
- Be implemented in 0.35 μm CMOS technology.
- Use a dual supply voltage of ±1.65 V.
- Each driver must be updated at a minimum data rate of 1.25 kHz.

Figure 2 illustrates the target current driver specifications.

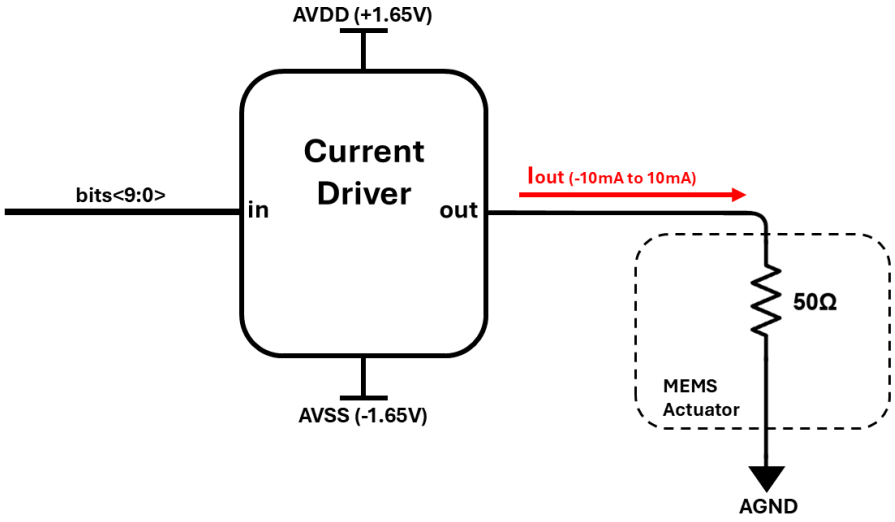

Figure 2 – Block diagram of the proposed driver.

As can be seen, the architectural challenge of creating a high-performance current driver revolves around rigid specifications regarding current amplitude, load tolerance and precise digital control, often under the constraints of integrated circuits. Designing a current driver that can source and sink currents up to 10 mA into a 50 ohm nominal load provides a unique intersection of high-output power and digital control precision. The operation must be simple and programmable, thus control signals should be transmitted through a Serial Peripheral Interface (SPI). The SPI is a synchronous serial interface standard for short-distance communication, which for this application can be used to transmit the digital input bits serially to the 50 drivers.

1.3 Thesis Overview

This master's thesis is organized into eight chapters, described below. Chapter 2 presents a literature review of the state-of-the-art current drivers, while Chapter 3 describes the first version of the current driver design, highlighting its limitations and points of improvement. Chapter 4 explores several different possible topologies for current drivers, which were designed and validated through simulations. Chapter 5 details the development of the Serial Peripheral Interface (SPI) unit for the digital input control of the drivers. Chapter 6 presents the chosen topology for the second version of the current driver and also the designed chip with arrays of this driver unit. Chapter 7 presents the simulation results for both the driver and the entire chip, covering several different simulation analyses. Finally, Chapter 8 concludes the work and outlines directions for future research.

2 Literature Review: Current Drivers in Advanced Integrated Circuits

2.1 Introduction, applications and design considerations of current drivers

Current drivers are a basic building tool that is used in many of today's electronic systems. Usually, the design of the current driver will support either a transition between the digital and analog domain, or it will simply provide an accurate and stable current source. The complexities of the design of current drivers are often related to the demanding performance requirements in several different areas of applications, including biomedical instrumentation, LiDAR sensors, Micro-Electro-Mechanical Systems (MEMS), and high-speed telecommunications [2] [3] [4] [5]. In real world applications, the effectiveness of these systems can be limited from issues with the current driver providing accurate, stable, and high-quality current under the required conditions [6].

The use of current drivers in complex systems means that a careful evaluation of the drivers performance ratings and topology is necessary. In telecommunication systems, current-steering architectures are frequently used to support wideband and high-speed communications [7]. DACs are crucial for converting the digital information into analog signals, demanding high linearity, good dynamic range, and low distortion. There are several design challenges associated with designing these drivers, which include reducing code dependent switching glitches, ensuring current source matching in order to reduce integral non-linearity (INL) or differential non-linearity (DNL), as well as addressing how finite output impedance and parasitic effects will influence the device's performance [8].

In biomedical applications, current drivers are used prominently for Electrical Impedance Tomography (EIT), bioimpedance spectroscopy and neural stimulation applications. The drivers for EIT systems, for instance, will need to accurately provide sinusoidal currents with low total harmonic distortion (THD) and high output impedance, all through a wide frequency range, in order to accurately characterize tissue properties [6] [9]. Neural stimulators require high compliance drivers that can source and sink accurate currents to account for variations in the electrode-tissue interface impedances and to ensure patient safety through features including high voltage (HV) protection and charge balance [4]. Power efficiency and miniaturization are also very important when used for wearable and implantable devices [10].

LiDAR sensors, commonly used in autonomous vehicles, robotics, and indoor

monitoring, rely on Vertical-Cavity Surface-Emitting Laser (VCSEL) drivers to emit light pulses [10]. These drivers need to supply precise modulation currents and stable bias currents with compact designs to reduce the chip area, and ensure VCSEL stable operation. There are some typical design challenges like having voltage headroom and being capable of dealing with the parasitic capacitance incurred from the bond-wires and ESD (Electrostatic Discharge) protection diodes, which are semiconductor devices placed at the I/O pads of ICs to protect their circuits from sudden high-voltage electrostatic discharge events [10].

2.2 Current range of current drivers

The specifications of the current range for the driver circuits are strongly dependent on the target application, which is exemplified in Table 1. A detailed analysis of them is beyond the scope of this work.

Table 1 – Current Range Comparison Across Various Current Driver Applications

Application	**Current Range**	**Resolution Step (LSB value)**	**Articles**
Neural Stimulator (High Current/High Voltage)	Maximum of ±15 mA	0.94 mA	[11]
Deep Brain Stimulator (DBS)	Up to 1.5 mA (complementary currents)	0.05 mA	[12]
Wearable Electrical Impedance Tomography (EIT)	6 mApp	NA (analog control)	[5]
Short-Range LiDAR Sensors	Up to 10 mApp	0.1 mA	[10]
Maskless Lithography Systems (Wafer Patterning)	5 mA	0.02 mA	[2]

2.3 Current Drivers for MEMS Applications

Current drivers are taking on an increasingly critical role in Micro-Electro-Mechanical Systems (MEMS), particularly in enabling advanced functionalities such as diagnostics and precise actuation control.

For electrostatic MEMS switches, smart high-voltage (HV) CMOS drivers are designed to not only provide the expected voltage to close the switch, but also to diagnose its operating status by determining if a switch is stuck or broken [13]. Constant current

drivers are also proposed to realize accurate power control of an array of devices using thermal heaters as actuators, which are designed to power electrothermal MEMS switches with time-controlled output current channels [3].

A low power MEMS mirror driver was developed as part of maskless lithography systems [2]. In this application, the driver is meant to precisely control the position of an array of MEMS mirrors with a matrix setup of 512 x 128. The system aims to dynamically project light patterns to a substrate so that very expensive optical masks could be replaced, while requiring high optical control and throughput [2].

In essence, current drivers are essential across several electronic fields, from telecommunications and biomedical devices to LiDAR sensors, in which balancing technical challenges like power and area consumption, linearity and output impedance is of prime importance.

2.4 Important concepts

This work presents and evaluates multiple Digital-to-Analog Converters (DACs) architectures in an attempt to address the challenges of the proposed driver design. There are a number of important figures of merit that can be used to describe the quality of the conversion from digital to analog signals when evaluating the performance of DACs. Some of the main figures of merit for characterizing both the static and dynamic performance of DACs are Integral Non-Linearity (INL), Differential Non-Linearity (DNL), and Spurious-Free Dynamic Range (SFDR). These 3 concepts will be explained in this section.

2.4.1 Integral Non-Linearity (INL)

Integral Non-Linearity (INL) is defined as the deviation of the DAC's actual transfer characteristic from its ideal linear behavior. The ideal transfer function of a DAC is a perfect staircase where all the steps are the same. For a current DAC, this means that for every binary increment of the digital word leads to the same change of current value. Therefore, INL is the error between the ideal position of the steps on the staircase and the actual position of the steps. INL can also be interpreted as the accumulated error of the DAC while one sweeps through the digital input codes - and reports the overall (or global) linearity of the converter. INL is typically reported in terms of the Least Significant Bit (LSB). For the ideal DAC, it is expected that the maximum INL is no greater than ± 0.5 LSB to a straight line between zero and full scale. If the DAC produces a non-linearity of ± 0.5 LSB or less, we can refer to the DAC as "monotonic", as the DAC output with increasing digital input never decreases. Large nonlinearities in DACs can primarily be attributed to inadequately matching the current sources within the converter, which can

further impact other figures of merit such as SFDR. The INL performance of a DAC can be enhanced with special circuit layouts, such as the random walk layout method [14].

The INL equation for output current (Iout) is given by:

$$\mathrm{INL}(n) = \frac{I_{\mathrm{out}}(n) - I_{\mathrm{out,ideal}}(n)}{\text{Ideal LSB}} \tag{2.1}$$

In which “n” refers to the binary input value, $I_{\mathrm{out,ideal}}$ refers to the expected Iout value for that input, and the Ideal LSB is given by:

$$\text{Ideal LSB} = \frac{I_{\mathrm{out,max}} - I_{\mathrm{out,min}}}{2^{\mathrm{N}} - 1} \tag{2.2}$$

In which “N” is the number of bits from each DAC. The "Ideal LSB" is commonly referred to as the converter’s resolution or quantization step. In this application, with 9 bits of magnitude and a current range of 0 mA to 10 mA (both positive and negative), the ideal LSB value can be calculated as follows.

$$\text{Ideal LSB} = \frac{10\,\mathrm{mA} - 0\,\mathrm{mA}}{2^9 - 1} = \frac{10\,\mathrm{mA}}{511} = 19.57\,\mu\mathrm{A} \tag{2.3}$$

The graph from Figure 3 shows the ideal sampled current values over the input digital code.

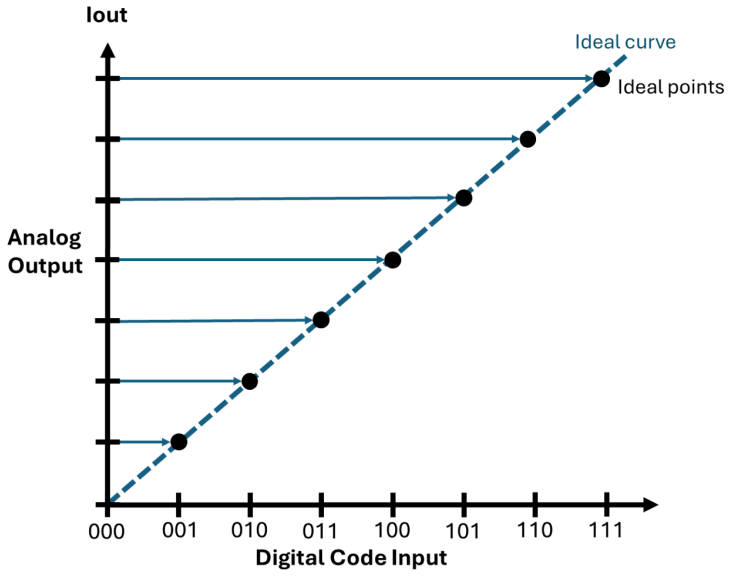

Figure 3 – Ideal output current versus input digital code.

Since all current values are matching with the ideal current curve, INL of all points is 0. And the graph from Figure 4 shows an example of INL values from a more realistic DAC output.

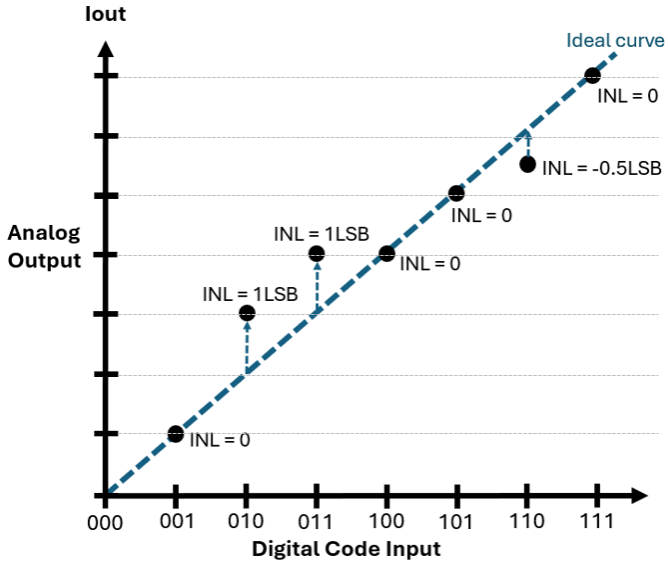

Figure 4 – Local INL of the sampled output current as a function of the input digital code.

As can be seen from the example, the INL value is calculated based on the distance between the output current of the specific digital codes and the ideal curve. Any point that matches the curve has INL=0. Since point 110 has value -0.5 LSB smaller than value from the ideal curve, its INL value is -0.5 LSB.

The INL of the proposed driver was evaluated using the best-fit line method. Traditionally, the INL can be computed using the straight-line (end-point) method, in which the reference line is defined by connecting the ideal output values at zero-scale and full-scale. This approach removes offset and gain errors by construction, but the resulting INL curve can be more sensitive to local deviations since the reference is constrained only by the end points. [15]

In contrast, the best-fit line method uses all the simulated data points for determining the reference line through a 'least-squares' regression. By fitting the entire transfer characteristic, this approach intrinsically compensates for offset and gain errors globally, reducing the impact of noise or localized variations. As a result, the extracted INL reflects only the intrinsic nonlinearity after removing the converter's best linear fit. [15]

2.4.2 Differential Non-Linearity (DNL)

Differential Non-Linearity (DNL) refers to the deviation of the size of individual output steps from the ideal LSB size. While INL represents a cumulative error across the full transfer curve, DNL is a local error that evaluates how much each individual step differs from the expected ideal 1 LSB. A DNL error of zero indicates that each adjacent output step is precisely equal to 1 LSB. If DNL is always less than 1 LSB, the DAC is guaranteed to be monotonic. The poor performance of DNL can happen due to mismatches in the

binary array segment of a DAC, replicating non-linearity after major code transitions. Therefore, the DNL performance is also highly influenced by current source matching [16].

While good DNL performance usually equates to good INL performance, the reverse may not be true. The INL itself is represented by a summation of all possible DNL errors in a transfer characteristic, thus the DNL errors contribute directly to the non-linearity of INL. For example, a large signal distortion in a DAC will be primarily defined by its INL maximum value, while a smaller DNL will help to limit the INL values. It is important to recognize that optimizing both INL and DNL is important for achieving high linearity, subsequently allowing for improved dynamic performance metrics such as SFDR.

The DNL equation for output current (Iout) is given by:

$$\mathrm{DNL}(n) = \frac{I_{\mathrm{out}}(n+1) - I_{\mathrm{out}}(n)}{\text{Ideal LSB}} - 1 \tag{2.4}$$

When analyzing a DAC's performance, it is usual to take the worst case positive DNL/INL and worst case negative DNL/INL from the entire curve as one of the key accuracy metrics.

The graph from Figure 5 shows an example of DNL values from a DAC output.

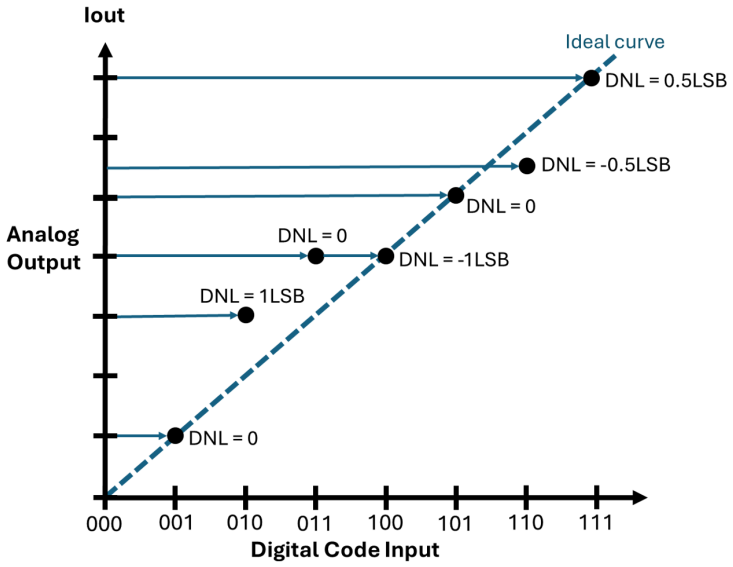

Figure 5 – Local DNL of the sampled output current as a function of the input digital code.

As explained before, the DNL of any point only compares its value with the previous point, in terms of step size. The ideal is that for each point, it increases by exactly 1 LSB in the output. In the example above, from point 001 to 010, the increase was of 2 LSBs, which means that point 010 has a DNL value of 1 LSB (1 LSB above the expectation). From point 101 to 110, the output increases by 0.5 LSB, which means that point 110 has a DNL value of -0.5 LSB (-0.5 LSB below expectation). From this perspective, it is easy to

observe that any value of DNL less than -1 LSB would cause non-monotonicity, because the output value of the point would be smaller than the output value of its previous input code.

When analyzing a DAC's performance, it is usual to take the worst case DNL and INL from the entire curve as one of the key accuracy metrics. Although the worst-case DNL and INL values are widely used as primary accuracy metrics, it is also useful to evaluate the Root Mean Square (RMS) DNL and INL. These RMS metrics provide a statistical view of the converter's linearity, capturing the expected deviation due to random mismatch rather than only the extreme points of the curve. This allows a more representative characterization of architectures where mismatch effects are distributed across many unit elements and facilitates fairer comparisons between different DAC structures and segmentation strategies [17].

2.4.3 Spurious-Free Dynamic Range (SFDR)

Spurious-Free Dynamic Range (SFDR) is a key dynamic specification that relates the ratio of the power of the fundamental output signal to the power of the largest spurious component (or distortion product) in the output spectrum of the DAC as exemplified in Figure 6. SFDR is normally expressed in decibels. SFDR is a key measurement for quantifying the dynamic performance of a DAC, especially in wide-band applications and high-speed communications [16].

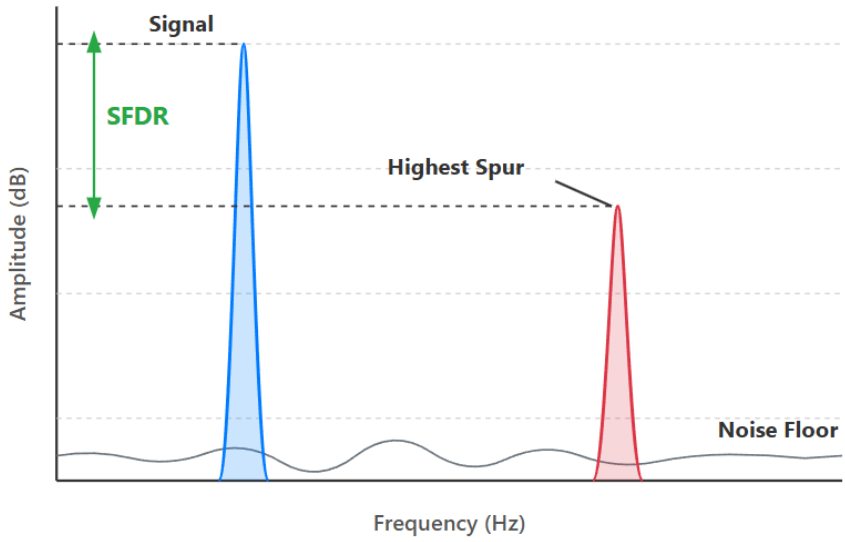

Figure 6 – Signal magnitude plot in the frequency domain illustrating SFDR measurement.

SFDR can be degraded due to several factors:

- Nonlinear Distortions: A substantial bottleneck to SFDR is the code-dependent timing glitches that cause nonlinear distortions. Techniques like Time-Relaxed

Interleaving Digital-Random-Return-to-Zero (TRI-DRRZ) have been developed to mitigate the glitches and improve SFDR [8].

- Current Source Mismatches: For low output frequencies, SFDR is primarily determined by current source mismatches that introduce non-linearity. As the output frequency increases, other degradation mechanisms become prominent.
- Finite Output Impedance: The finite output impedance of the current sources creates a code dependent error term that causes distortion and the subsequent degradation of SFDR.
- Switching Behavior: Non-ideal switches can create glitches and may introduce other dynamic errors that affect SFDR. Minimizing disturbances while transitions occur may reduce harmonic distortion, which indirectly improves SFDR.
- Timing Skew: Timing skews between current cells are important to SFDR and can cause distortion, especially if the skew is systematic and correlated to the digital code. Timing skews are especially important for high output frequencies [18].

In summary, INL and DNL represent the static accuracy of the DAC's voltage or current output as it maintains an ideally linear response. SFDR, on the other hand, measures the dynamic purity of the output signal as it relates to the signal being driven by a dynamic input, free from any unwanted spectral artifacts. These quantifiers are essential for measuring and optimizing DAC designs for a wide variety of technological implementations.

2.4.4 Other parameters (SNR, SNDR and ENOB)

In addition to INL, DNL, and SFDR, other important figures of merit that evaluate the dynamic performance of data converters are the Signal-to-Noise Ratio (SNR), Signal-to-Noise-and-Distortion Ratio (SNDR), and the Effective Number of Bits (ENOB). These parameters quantify how accurately the converter reproduces a time-varying analog signal while considering the impact of noise, nonlinearity, and distortion [19].

Signal-to-Noise Ratio (SNR) measures the ratio between the power of the desired fundamental signal and the total noise power, excluding harmonic distortion components. It provides an indication of the converter's ability to process signals in the presence of noise sources such as quantization noise, thermal noise, and clock jitter. For an ideal N-bit converter, the theoretical SNR is given by:

$$\mathrm{SNR}_{\mathrm{ideal}} = 6.02N + 1.76 \text{ dB} \tag{2.5}$$

A higher SNR indicates better purity of the reproduced signal and lower noise contribution.

Signal-to-Noise-and-Distortion Ratio (SNDR) extends the concept of SNR by including the harmonic distortion components in addition to noise. SNDR is defined as the ratio of the power of the fundamental signal to the combined power of all noise and distortion components (excluding DC):

$$\mathrm{SNDR} = 10 \log_{10} \left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}} + P_{\mathrm{distortion}}} \right) \tag{2.6}$$

Since it accounts for both random (noise) and deterministic (distortion) errors, SNDR is a more realistic and comprehensive metric for dynamic performance assessment compared to SNR. It is closely related to SFDR, but instead of focusing only on the largest spurious tone, it considers the total noise and all spurious components within the bandwidth [19].

Effective Number of Bits (ENOB) quantifies the real resolution of the converter under dynamic operation. It indicates how many bits of an ideal converter would be required to achieve the same dynamic performance as the actual converter. ENOB is computed from the measured SNDR value using the relation:

$$\mathrm{ENOB} = \frac{\mathrm{SNDR} - 1.76}{6.02} \tag{2.7}$$

ENOB allows translating the dynamic performance of converters to a more intuitive bit-resolution metric. While static parameters (INL and DNL) assess the linearity of the transfer function, ENOB represents the actual usable resolution under operating conditions including noise and distortion [19].

2.5 Chip V0: Voltage mode current driver [1]

The doctoral thesis [1] proposed a 5 x 5 array (25 elements) set of Lorentz force MEMS actuators. In order to operate these actuators, a mixed-signal CMOS driver module is then connected to this array. Each actuator is driven by voltages ranging between -15 V and +15 V to induce the necessary deflection of the mirror surface.

In order to achieve the precise control required by the MEMS DM, a dedicated ASIC driver was developed, specifically designed to provide the control signals for the MEMS actuator, enabling the level of wavefront correction necessary for high-resolution imaging. The system aims to achieve a bipolar ±15 V square wave output with current capability of ±5 mA in the presence of a 0.1 T magnetic field, which is fundamental to the Lorentz force actuation principle [1].

The developed chip contains individual drivers for each of the 25 actuators. It is created to interpret information from the wavefront sensor (WFS) transmitted via a serial peripheral interface (SPI). This digital data, representing the desired mirror shape, is then encoded into pulses using a pulse-width modulation (PWM) technique. Each actuator is charged or discharged by its driver with an 8-bit digital control signal. The digital logic part of the design uses low-voltage, later being converted into a high-voltage analog signal that drives the actuator. Figure below shows the block diagram of the proposed system.

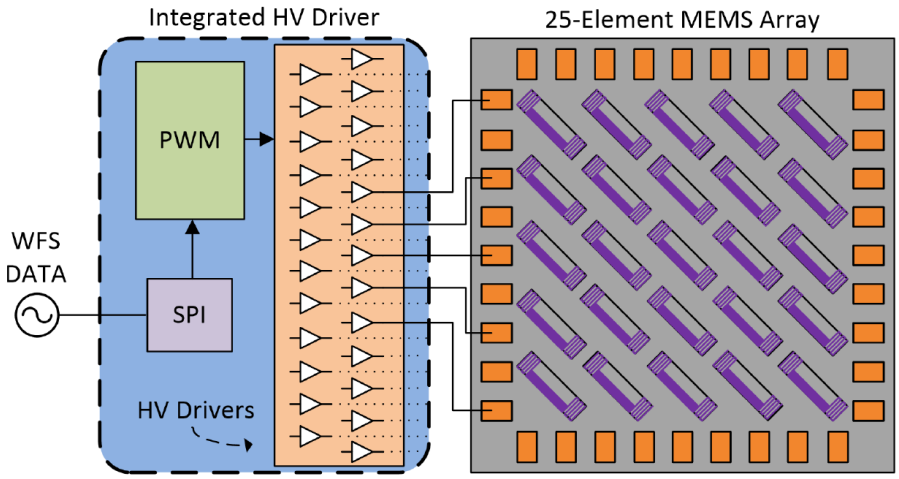

Figure 7 – Block Diagram of the proposed MEMS deformable mirror system from work [1].

The driver is based on an open-loop Class-D power amplifier architecture, due to its power efficiency, critical to minimize energy dissipation in multi-actuator systems. The PWM signal acts as a digital-to-analog conversion circuit, with a low-pass filter applied to the driver output in order to remove all the high-frequency components. This filter ensures that the resulting DC voltage values are directly proportional to the duty cycle from the PWM signal, therefore having a more precise control over the actuator deflection. The resulting signal controls the ON/OFF states of the power transistors, having highly efficient low-power operation by operating the transistors in either linear or cut-off regions, reducing power loss during transitions.

It was designed and fabricated using AustriaMicroSystems (AMS) 0.35 μm CMOS High-Voltage (HV) process technology. This process was chosen due to its suitability for mixed-signal designs, requiring both low-voltage digital logic and high-voltage analog output stages.

The testing confirmed the chip's ability to operate with external voltages of $\pm$3.3 V, $\pm$11.7 V and $\pm$15 V, generating a $\pm$15 V output square wave for a low-voltage PWM input. Linearity tests showed the correct behavior over a useful current range of $\pm$5 mA, while the step response tests indicated a 100 μs rise time with 140 mV ripple. Spectral analysis of the output yielded a signal-to-noise-and-distortion ratio (SNDR) of 51.4 dB

and an Effective Number of Bits (ENOB) of 7.68, with key operating parameters including a maximum output voltage of ±15 V, an operating frequency of 1 MHz, a bandwidth of 10 Hz - 1 kHz, and 150 mW power consumption for a 15 pF load.

The driver was also validated in a MEMS deformable mirror setup, in which a single actuator in a 5 x 5 array was driven with currents from -5 mA to +5 mA under a 0.1 T magnetic field, creating mirror deflections between -20 μm and +22 μm, confirming the driver's effectiveness for MEMS actuation. Table below summarizes the main specs from this proposed driver.

Table 2 – Specifications of the Current Driver proposed in [1]

Spec	**Value**	**Comments**
Technology Process	0.35 μm HV CMOS (AMS)	Chosen due to its high-voltage capabilities and availability.
VDD Supply Value	Bipolar ±15 V	Designed to produce a bipolar square wave to drive MEMS actuators in push-pull mode.
Power Consumption	150 mW per driver	Key parameter, since the design aims for a low-power solution applicable to high actuator count systems.
Occupied Silicon Area	0.24 mm^2 per driver	Compact size is crucial for integrating drivers with the MEMS array to create a scalable system.
Linearity (INL/DNL)	-	Not measured.
Current Range	±5 mA	-

Although the proposed solution was validated to be functional and has its advantages, this dissertation's work aims to improve the following specifications:

- Load value dependence

The main limitation of this driver's topology, discussed in thesis [1], is its dependency of the output current on load variations. This can lead to loss of precision and repeatability in actuator's movement, affecting the overall accuracy of the adaptive optics correction. Any change in the MEMS actuator properties can directly affect the current that it receives, causing inconsistent results.

- Scalability for future high-order wavefront correction

There is a limited number of actuators (25). Modern and future high-resolution AO systems require a lot more, hundreds or even thousands, to achieve high-order wavefront correction and to compensate for the fine spatial scales of atmospheric turbulence. High power consumption is also another limitation from the proposed driver (about 20 mW per

driver). Considering a chip with 1,000 actuators, that would lead to a total power value above 20 W, which would introduce significant challenges related to thermal management and operational costs.

- Linearity study not done

As shown in Table 2, the INL and DNL values that measure the current linearity according to the input code were not measured/simulated by the thesis' author.

- High voltage operation, leading to higher power

From the table, it can be observed that the driver consumes high voltage supply and, in consequence, a high power value. Since the target output current from this work can face current values up to 10 mA, it is clear that the voltage value for this operation would have to be very high. This work also aimed at reducing power consumption from [20], by majorly reducing the supply voltage level.

3 First designed project

This chapter provides an in-depth analysis and explanation of the first proposed topology for the ±10 mA Current Driver ("chip V1"). The device is designed using the AMS 350 nm CMOS process and utilizes a symmetric supply voltage of ±3.3 V. This design's output current value is controlled with an 11-bit digital word, in which 10 bits are used for the magnitude of the output current, and 1 bit (the MSB) is used for the current direction (positive or negative). For thoroughly evaluating and testing, four similar drivers were developed on one test chip to supply output currents simultaneously.

3.1 Proposed Solution

The implemented current driver uses an approach consisting of a digital-to-analog converter (DAC) and a voltage-to-current conversion circuit to operate together with a combination of high precision and robustness. The proposed topology provides output current that varies linearly in the standard 50-ohm load case from -10 mA to +10 mA with the current value programmed exactly from a digital input word. The circuit consumes between 33 mW and 86 mW of power while operating under the nominal ±3.3 V supply in the full scale operating mode (±10 mA).

The driver circuit architecture being proposed in this work is conceptually shown in Figure 8. The architecture consists of four identical current drivers, each specifically intended to drive four 50-ohm (and purely resistive) loads, independently.

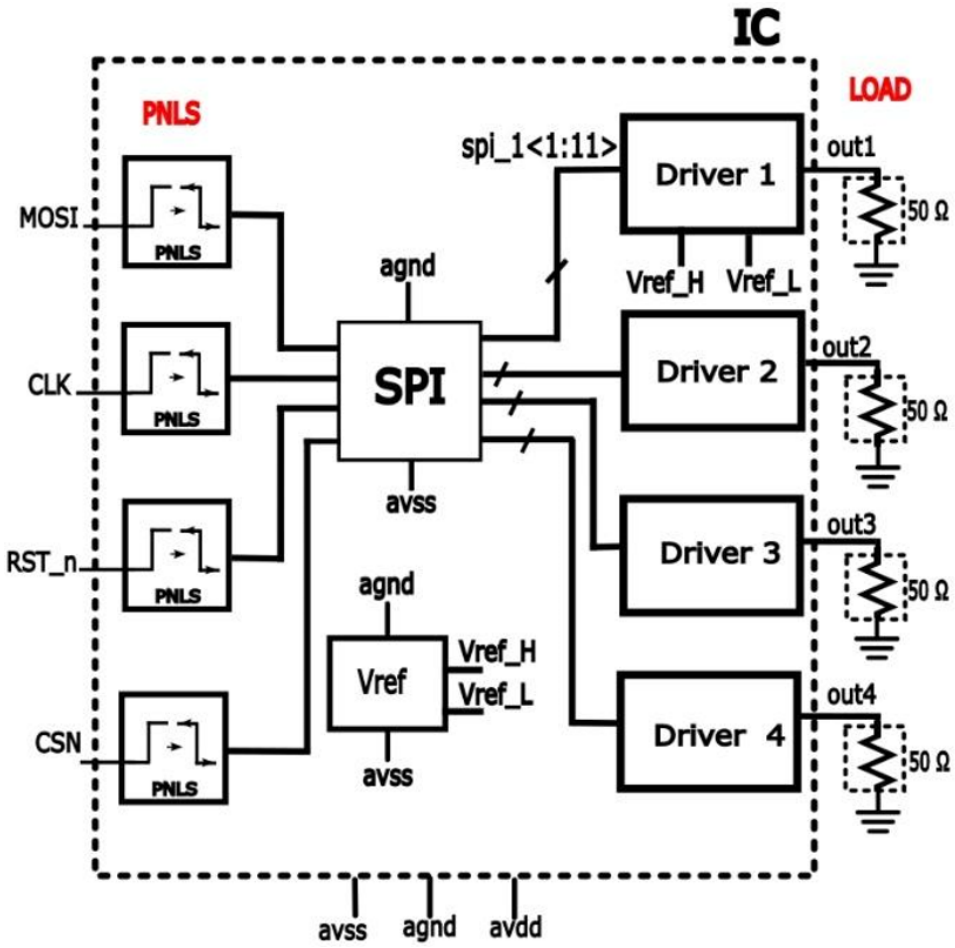

Figure 8 – Proposed chip V1 architecture.

The output currents of the four current drivers are, nominally, required to vary from -10 mA to 10 mA. The power supply consists of AVSS, AGND, and AVDD, with voltage values of -3.3 V, 0 V and 3.3 V, respectively. The control system consists of a Serial Peripheral Interface (SPI) circuit, which was used to create the appropriate 11-bit digital input signals to control current output.

In Figure 9 we can observe the architecture of each driver in the system. The current driver design includes the 11-bit digital input spi<1:11> for each driver. Of the 11-bits, the first 10-bits will be used for controlling the output current by varying an output from, nominally, 0 mA to 10 mA. The last bit, 'bit11', controls the output current direction, either negative (sink) or positive (source). This control mechanism enables the output current to precisely range from -10 mA to 10 mA through a 50-ohm load.

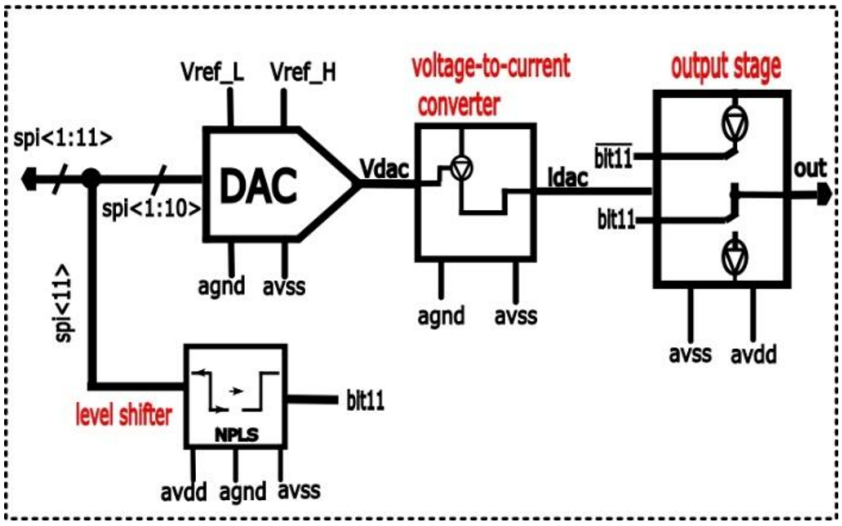

Figure 9 – Proposed driver architecture for chip V1.

The conversion from a 10-bit digital input signal into an analog voltage signal (Vdac) is done by the digital to analog converter (DAC) block. The DAC block converts the input based on two reference voltage values, Vref_L and Vref_H, generated with bandgap voltage reference circuits [21].

Since the dual supply (±3.3 V) integrated circuit operates on symmetric power supplies, the digital signals used internally in the integrated circuit operate with two separate voltage rails: 0/+3.3 V and -3.3 V/0 V. This means dedicated level converter circuits are required, called PNLS (Positive-to-Negative Level Shifter) and NPLS (Negative-to-Positive Level Shifter), as shown in Figure 10.

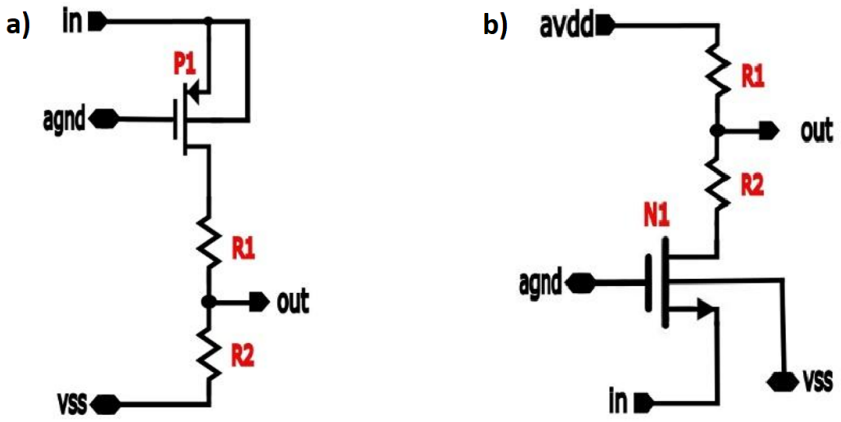

Figure 10 – a) PNLS and b) NPLS circuits for chip V1.

The operation of both voltage level converters is similar. In the PNLS, when the In pin sees a voltage level of 3.3V, transistor P1 becomes active, meaning that current can flow through resistors R1 and R2. This turns the combined resistance into a voltage divider

circuit, with an output voltage of approximately 0 V. When the In pin sees a voltage of 0 V, P1 enters cut-off region, thus the output voltage equals AVSS (or -3.3 V). The same operation occurs in the NPLS, converting input values of -3.3 V and 0 V into 0 V and +3.3 V, respectively. As shown in Figure 9, the NPLS is responsible for converting the spi<11> output signal (range -3.3 V - 0 V) to the bit<11> signal (range 0-3.3 V). This is important since the output stage of this driver uses symmetric supply voltages (±3.3 V).

The voltage-to-current converter, illustrated in Figure 11, is used to convert the analog output voltage produced by the DAC into a proportional current value that can then be referenced/mirrored to the output stage of the driver [22]. In doing so, an operational amplifier is configured to replicate the voltage at the positive input to the negative input, effectively placing the voltage Vdac across resistor R1, producing a reference current through transistor N1 that remains constant and proportional to the Vdac value. Then the reference current is mirrored by transistors P1 and P2, producing the current Idac defined in Equation 3.1:

$$I_{\mathrm{DAC}} = \frac{V_{\mathrm{DAC}}}{R_1} \tag{3.1}$$

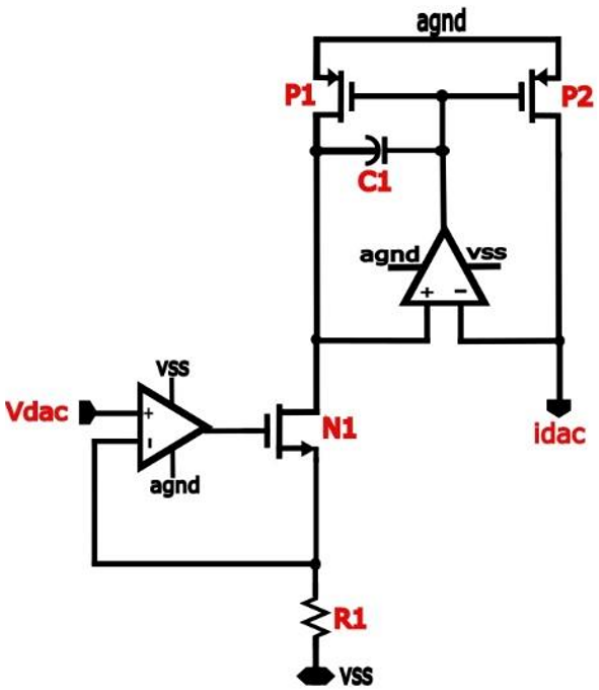

Figure 11 – Voltage-to-Current converter circuit for driver from chip V1.

Finally, the output stage, shown schematically in Figure 12, is responsible for conditioning the current signal which is delivered to the 50-ohm load at the end of this stage. This stage was also developed to be able to produce both negative and positive current values. Additionally, the output stage uses current mirrors to achieve further

current gain. The transistors were designed to operate in the strong inversion region to provide optimum matching characteristics and performance.

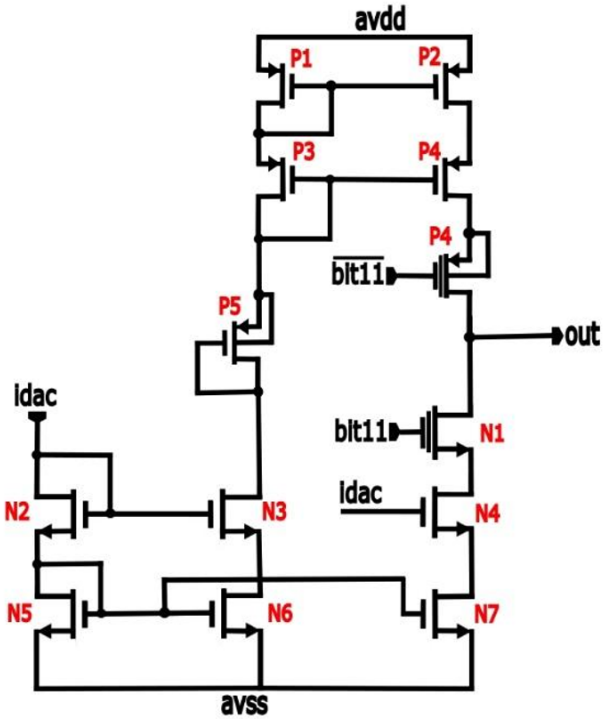

Figure 12 – Output stage circuit of the driver from chip V1.

The bit11 signal simply controls the direction of the output current. Specifically, it controls whether the output current flows out from the upper branch (powered by AVDD) or in through the lower branch (powered by AVSS). It must be noted that, by the nature of idac, the magnitude of the output current value is ideally the same for both positive and negative directions. Also worth noting that transistors N1 and P4 are 5 V and all other transistors in the proposed system are 3.3 V. Those two transistors are made for high voltage swings, being more robust and having special placement/routing rules.

3.2 Simulations and Performance Analysis

The first evaluation of the developed block was performed using a transient simulation, and results are presented in Figure 13. This simulation cycled through all the 2048 possible values of the 11-bit digital input word. In the instantaneous current shown from this simulation, the current waveform varies from approximately 0 mA to -10 mA, and then returns from +10 mA back toward 0 mA.

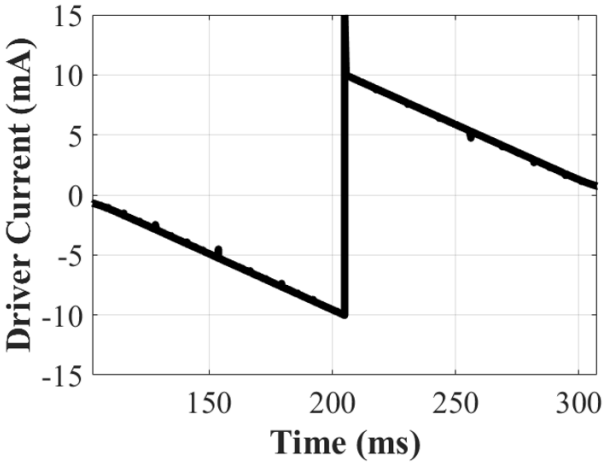

Figure 13 – Full current range variation of the driver for chip V1.

One important note is that there is a slight offset of about ±700 μA of current when the digital input word is set to zero. It is worth mentioning that the offset of this topology, due to the use of the operational amplifier in Figure 11, can be easily removed by optimizing its design. During the design phase, it was decided to reduce the number of different blocks to be implemented in the layout.

There is also a considerable amount of glitching in the current output when a transition from -10 mA to +10 mA occurs (unusual scenario), and other minor glitches at other transitions. For the fabricated circuit, it was expected that these glitches would exhibit smaller amplitudes due to the intrinsic capacitance from its internal nodes, and from the output node itself. Figure 14 is from the same transient simulation from Figure 13, but in a reduced temporal window, and therefore only showing the output responses of the circuit for five different digital input word states. From this detailed view, a current resolution value of approximately 10 μA is observed.

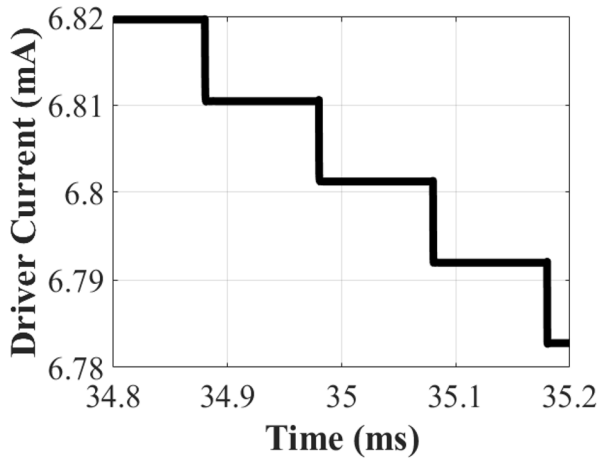

Figure 14 – Plot of regional current variation for the driver in chip V1, highlighting its resolution.

The other significant simulation performed was the power supply rejection ratio (PSRR) test, which measures the ability of the circuit to reject noise from the power supply. For this simulation, a worst-case scenario was considered in which periodic digital noise was injected, in the form of a 0.2 Vpp (peak-to-peak) square wave, into the AVDD supply voltage. The noise signal's effect on the output current was observed, particularly for the case of the highest positive current output at (+10 mA).

The AVDD simulated voltage stimulus and the effect of injecting this noise into the driver's output current are shown in Figure 15. A significant conclusion from this analysis is that the noise signal injected into the power supply had little effect on the driver's output current. Specifically, it was less than 0.2% with regards to the output current average value (< 2 LSBs), despite a large magnitude variation of about 6% in the power supply voltage. This good robustness could largely be attributed to the effective use of a cascode structure in the current mirrors.

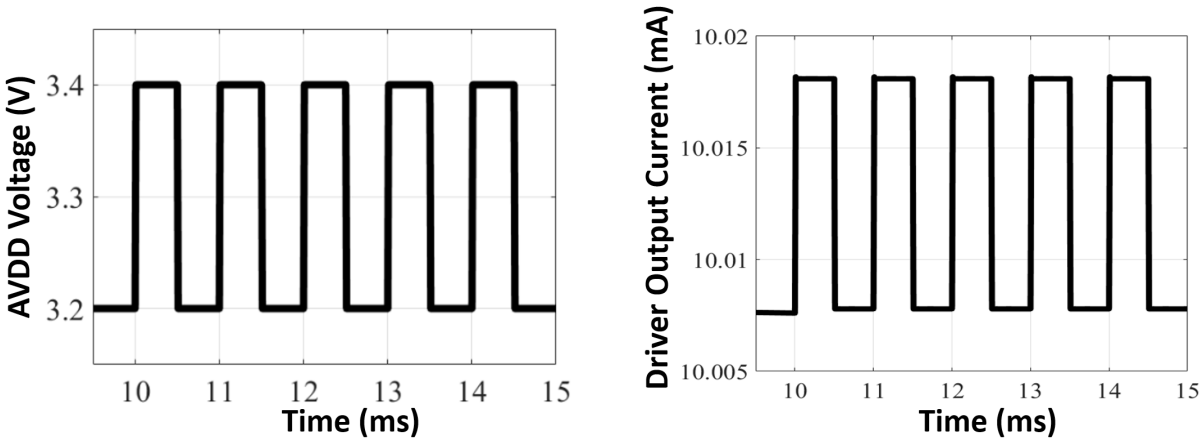

Figure 15 – AVDD voltage variation and corresponding output current response for PSRR measurement.

Further simulations were performed to study the circuit performance while varying the supply voltages. The magnitudes of the AVDD and AVSS supply voltages varied systematically from 0 to 3.3 V, and the output currents were monitored, as shown in Figure 16. The results indicated that the minimum operating voltage for the circuit is ±2.6 V, and the dependence of the output current on the supply voltages was determined to be 0.3 μA/mV (in the functional region).

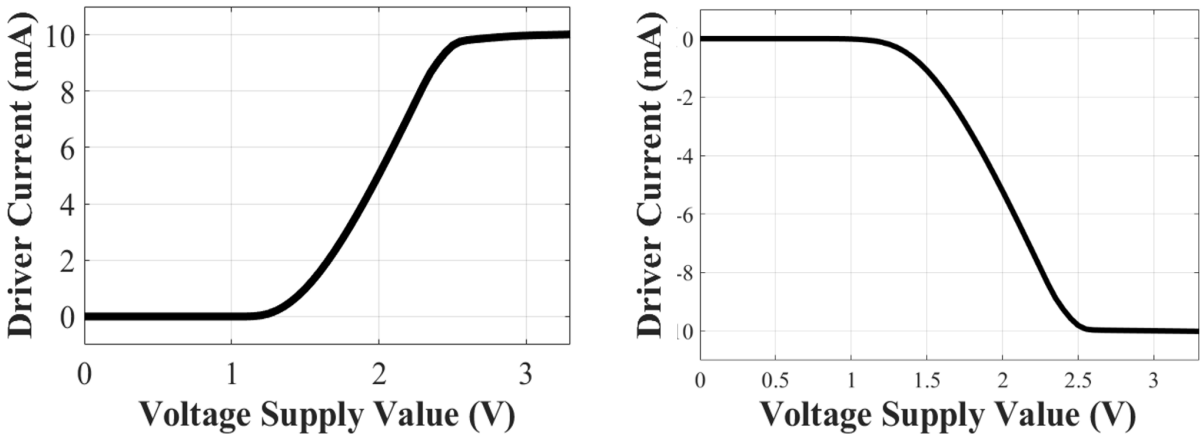

Figure 16 – Output current variation with increasing supply voltage magnitude for nominal 10 mA and -10 mA output currents, respectively.

Lastly, to estimate the impact of the process variation, attributed to different process conditions of the chip fabrication, on the driver's output current, two different simulations were performed using the Monte Carlo (MC) run mode, targeting the extreme nominal output current cases of +10 mA and -10 mA. 1,000 sampling MC points were performed for both cases. The results are displayed in the histograms in Figure 17.

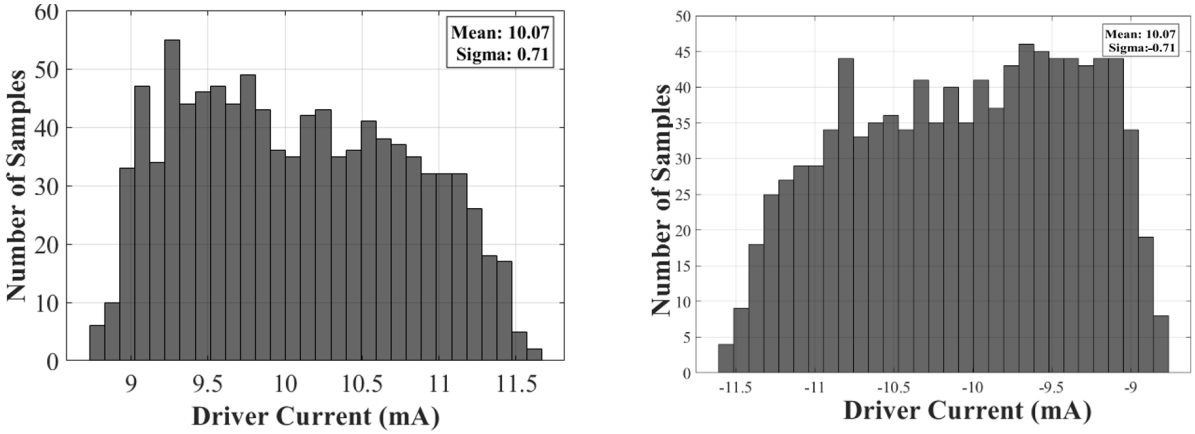

Figure 17 – Histograms of Monte Carlo simulation results for the output current at nominal –10 mA and 10 mA, respectively.

By reviewing the histograms of both positive and negative current cases, it was observed that the average output current associated with each case had the same magnitude of 10.07 mA. This is indicative of the strong correlation between the positive and negative conditions measured on the designed circuit. The calculated sigma for both cases was determined to be 0.71 mA. In this context, sigma denotes the standard deviation of the simulated data distribution. It characterizes the statistical dispersion of the analyzed

parameter, indicating the extent to which the Monte Carlo sample values deviate from the mean. From this information, it was determined that the driver's current output magnitude ranges between 7.94 and 12.2 mA for 99.7% of the measured cases (corresponding to mean ±3 sigma). These current variations are attributed to the current mirror and to the voltage-to-current conversion block. Also, to a lesser extent, to the reference voltages (Vref_L and Vref_H) used in the DAC.

A set of processes (WP, WS, WO and WZ), temperature (0 and 80 ºC) and voltage supply (VDD +/- 5%) corner simulations were also performed to see the variation of max output current values for these scenarios. The figure below shows the results obtained.

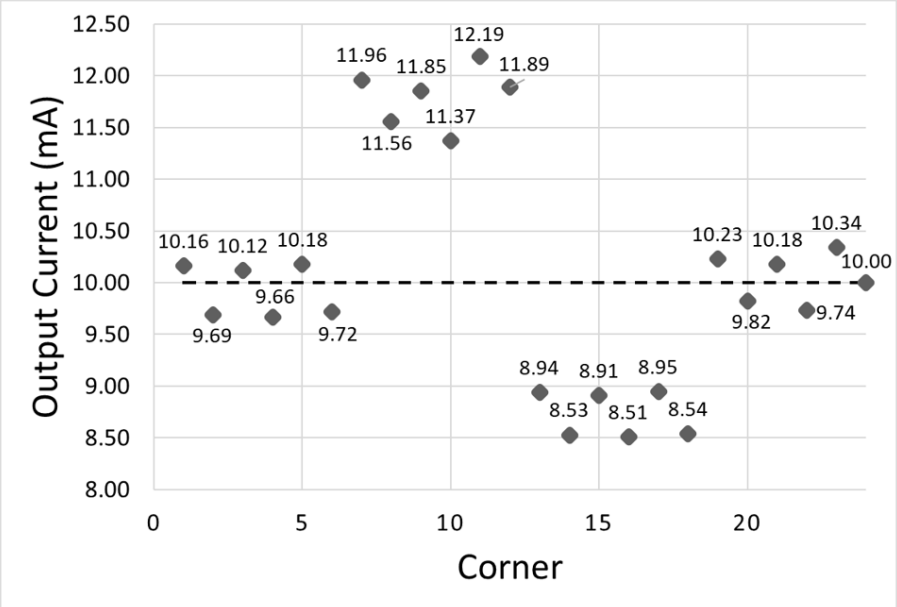

Figure 18 – Corners simulation results for the output current at nominal 10 mA.

From these results, it is observed that the current varies with a minimum value of 8.51 mA (−14.9 %) and a maximum value of 12.19 mA (+21.9%) between the process corners.

The following figures present the measurement results obtained from five samples of the same driver unit fabricated in chip V1.

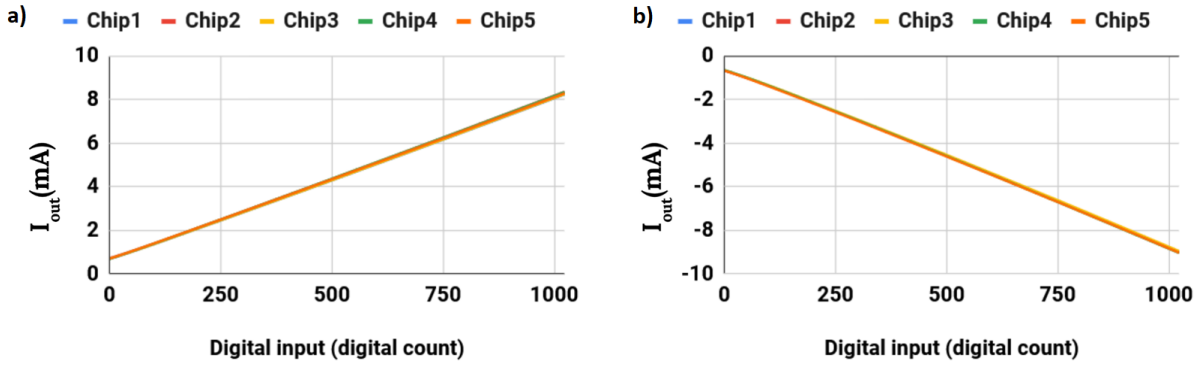

Figure 19 – Measured full output current range of the driver for chip V1, obtained from five fabricated samples.

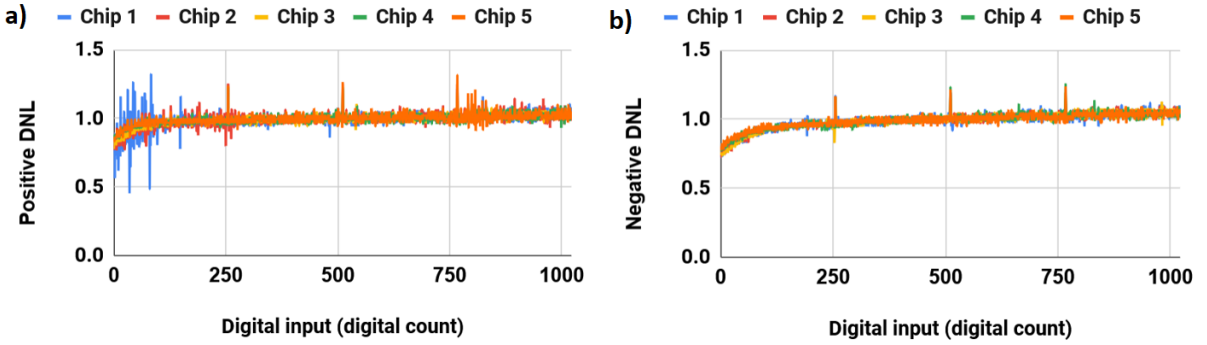

Figure 20 – Measured a) positive DNL and b) negative DNL for the driver in chip V1 across five samples.

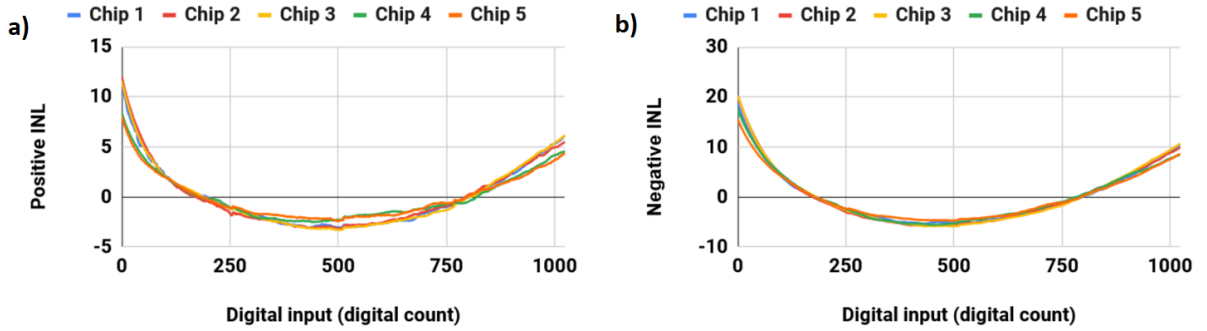

Figure 21 – Measured a) positive INL and b) negative INL for the driver in chip V1 across five samples.

Figure 19 shows the measured output current range for both positive and negative current regions. Across all samples, the driver was able to generate currents starting at

approximately ±0.7 mA and reaching a maximum of about ±9 mA. This behavior indicates that the implemented architecture was not capable of delivering the full intended current sweep, revealing a limitation in the achievable output range.

Figures 20 and 21 present the key linearity metrics obtained from the measurements. Regarding the differential nonlinearity (DNL), the results were satisfactory: all measured values remained close to 1 LSB across the entire input code range, demonstrating that the step-to-step uniformity of the DAC behavior was well preserved in silicon.

However, the integral nonlinearity (INL) measurements revealed a more critical shortcoming of this first implementation. As shown in Figure 21, the INL error exhibited substantial deviations, especially near the extremes of the input code range, both at very low and very high digital codes. In these regions, the INL exceeded 10 LSB for several samples, highlighting a significant intrinsic nonlinearity of the initially proposed architecture. This issue ultimately motivated the architectural revisions discussed in the subsequent sections of this dissertation.

3.3 Driver EMIR Analysis

After the driver layout was finished, it was important to run Electromigration and IR Drop (EMIR) Analysis in the design to make sure the currents provided, specially the ones from supply rails, were not causing any issues in the wires. This is a critical step in IC physical verification to ensure power integrity and reliability of the power distribution of the designed chip. For this flow, Cadence Voltus-XFi tool was used, running Spectre EMIR simulation in Tungsten view and using Voltus-XFi Results Browser menu to analyze the output results.

In order to assess the main scenarios, these 2 simulations were performed:

- DC EMIR run with nominal output of 10 mA for the driver.
- DC EMIR run with nominal output of -10 mA for the driver.

The results from each run are explained in the following subsections.

3.3.1 IR Drop Analysis results

The main reason for performing IR Drop Analysis is to verify that every instance receives the sufficient amount of necessary supply voltage. I.e., to make sure that the parasitic effects from the supply rails are not enough to cause relevant IR Drop which could impact the correct transistor operation. Figures 22 and 23 show all IR Drop DC current results from this chip for output current values of 10 mA and -10 mA.

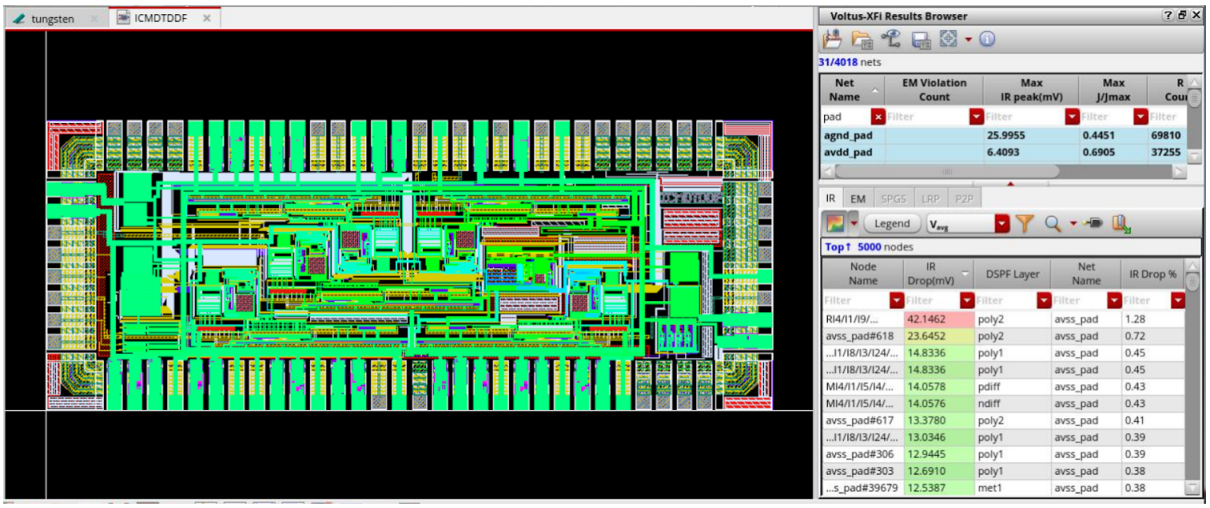

Figure 22 – IR Drop DC results for output current of 10 mA.

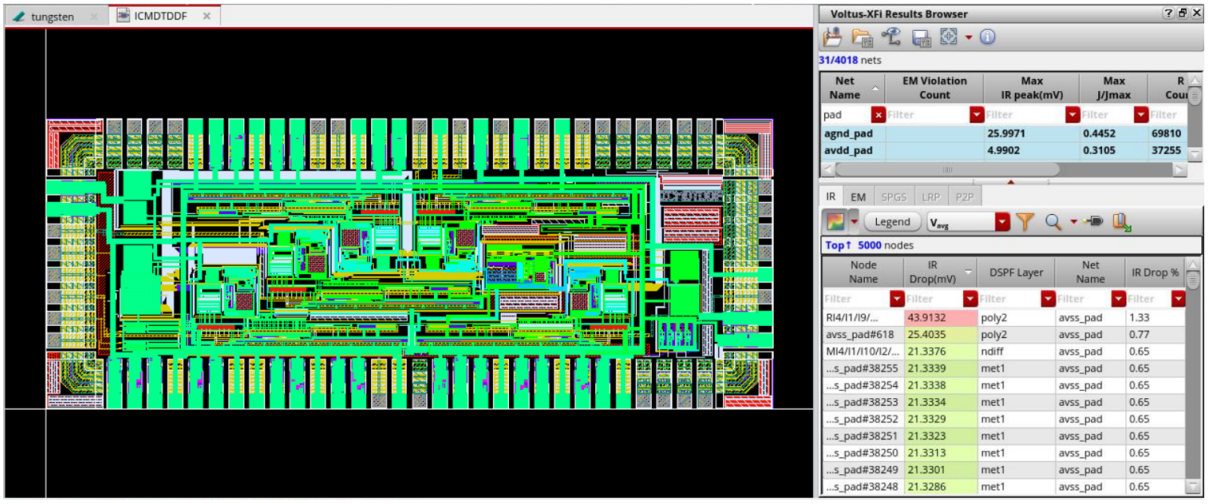

Figure 23 – IR Drop DC results for output current of -10 mA.

This analysis was performed only on the supply rails, which are: AVDD, AVSS and AGND nets. It can be seen that the maximum IR drop for both scenarios show values of less than 1.5%. And besides this one outlier result in poly2 layer, all other results are even better: smaller than 0.8%. Therefore, it can be concluded that for this chip, the IR drop is small and shouldn't be cause of concern.

3.3.2 Electromigration (EM) Analysis results

The main reason for performing EM Analysis is to assess the metal reliability over time. High current density on metal wires can cause metal atom movement, which can

cause wire degradation. Following images show all EM DC current results from this chip for output current values of 10 mA and -10 mA.

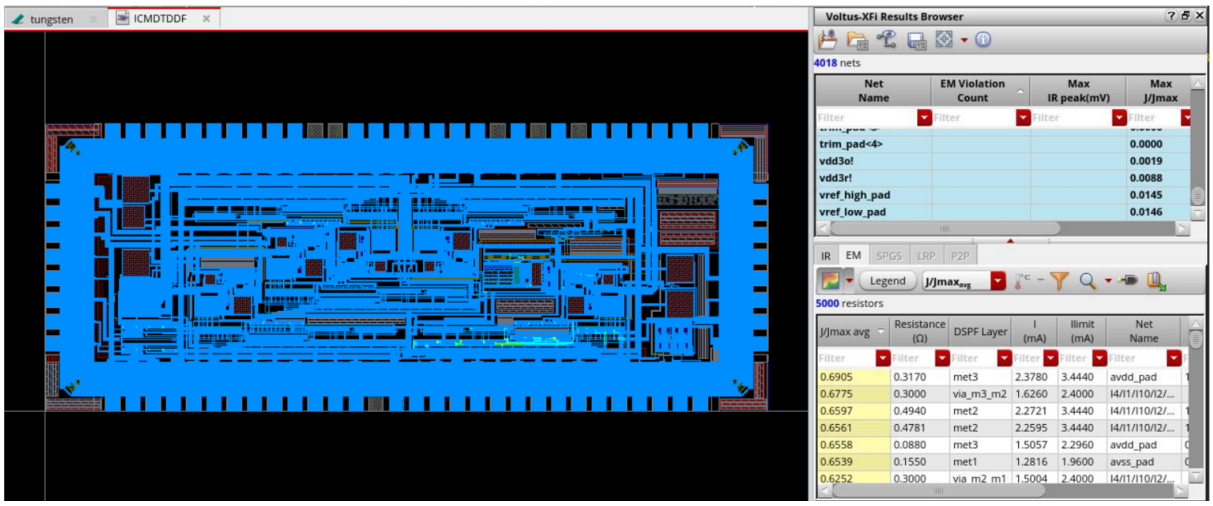

Figure 24 – EM DC results for output current of 10 mA.

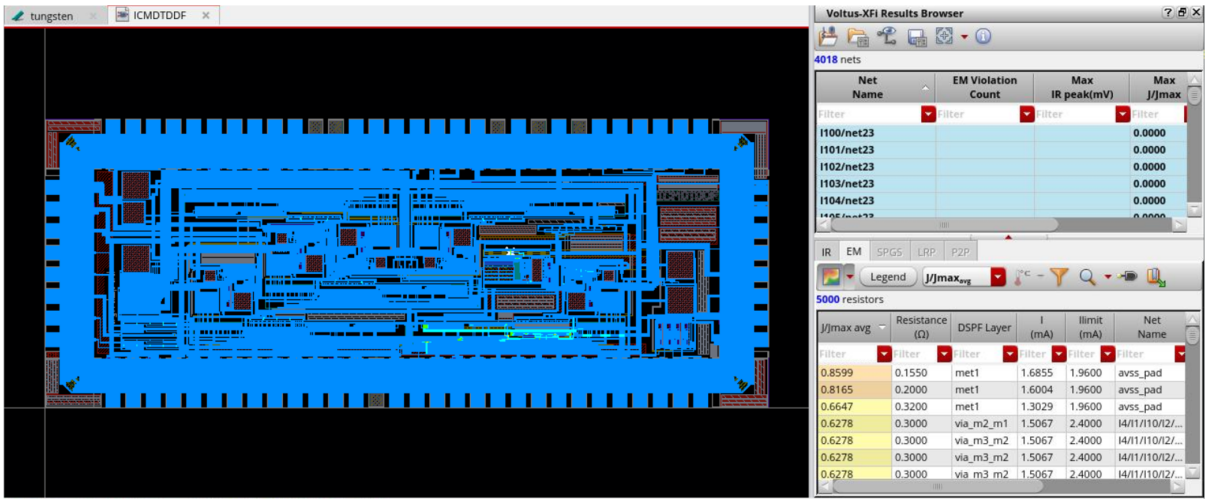

Figure 25 – EM DC results for output current of -10 mA.

In these figures, "J/Jmax avg" means average measured current density value divided by the max allowed current density value. Therefore, any value above 1 represents an electromigration issue. Since all values from both cases were below 1, the chip was free from EM issues.

Although the EMIR analyses for the chip V1 produced satisfactory results without demanding many modifications in the layout, these analyses are of particular importance when designing the layout of the 50 drivers' chip, since the magnitude of currents involved

in this other chip is much higher, both to the supply pins and to the output pin of each individual driver (Iout current).

In addition to the linearity limitations observed, this first version also presented an important constraint in terms of silicon area. The layout of chip V1 required approximately 0.46 mm^2 per driver, which poses a significant scalability challenge for systems requiring large driver arrays. Reducing the area consumption therefore became a critical design objective, motivating the exploration of alternative topologies capable of offering improved linearity together with substantially smaller area footprints. More details about the developed chip V1 design flow can be found in the paper [23].

4 Design exploration

This chapter presents the architectural study conducted after the completion of Chip V1, with the main objectives of improving the driver's linearity and reduce its silicon area. The investigation focused on analyzing and implementing various Digital-to-Analog Converter (DAC) design topologies that could achieve the same proposed goal, without having the need to use the voltage-to-current converter from Figure 9. The main topologies assessed as part of this work are discussed in detail in the following sections.

4.1 Voltage Mode Topology Using Buffer Operational Amplifiers

This topology will directly convert a binary input signal into an analog voltage signal. The voltage is applied to a 50-ohm load in order for the required output current to be generated. The 10-bit DAC module used in the implementation corresponds to one of the unit cells provided by the AMS foundry for this PDK.

The voltage DAC requires specific high and low reference voltages. Considering that the target output current is defined as between -10 mA and +10 mA, the simplest method of defining the reference voltages for the 50 ohm load is as follows:

$$V_{\text{high}} = 10\,\text{mA} \times 50\,\Omega = 500\,\text{mV} \tag{4.1}$$

$$V_{\text{low}} = -10\,\text{mA} \times 50\,\Omega = -500\,\text{mV} \tag{4.2}$$

In theory, the calculated reference voltage could be used and connected directly at the output from the DAC to the load, thus generating an output current from -10 mA to 10 mA. However, when running simulations, it became clear that the particular DAC unit used was incapable of providing such high current values to the load, which was actually the expected behavior. In order to solve this problem, an operational amplifier configured as voltage follower was designed to buffer the DAC to the driver's output. The diagram below shows the full circuit employed in this topology.

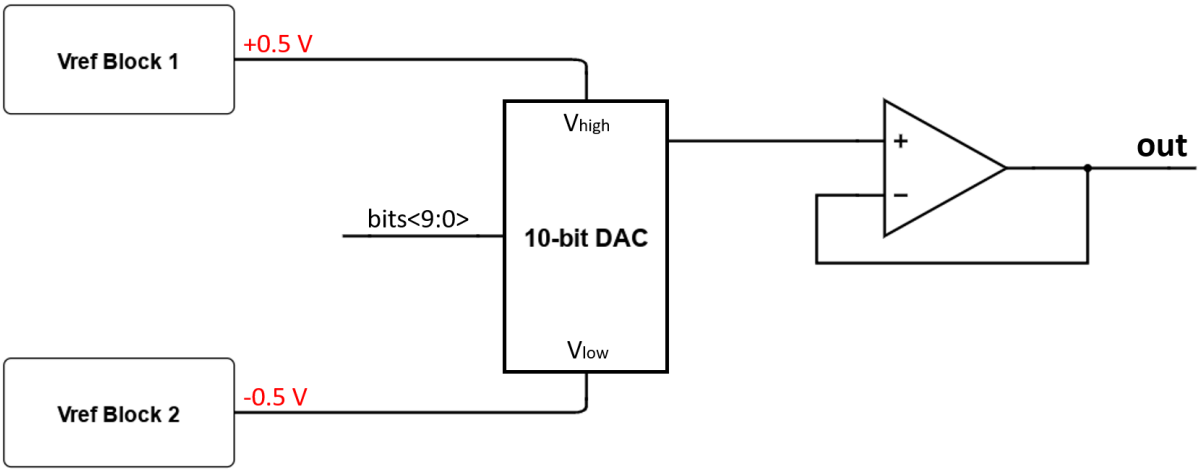

Figure 26 – Diagram of voltage mode topology using buffers.

Two simple resistive voltage dividers provided the high and low reference voltage values needed. Also, based on minimizing load effects on the reference voltages, buffer operational amplifiers were included as well. Figure 27 shows the full circuit employed in this topology.

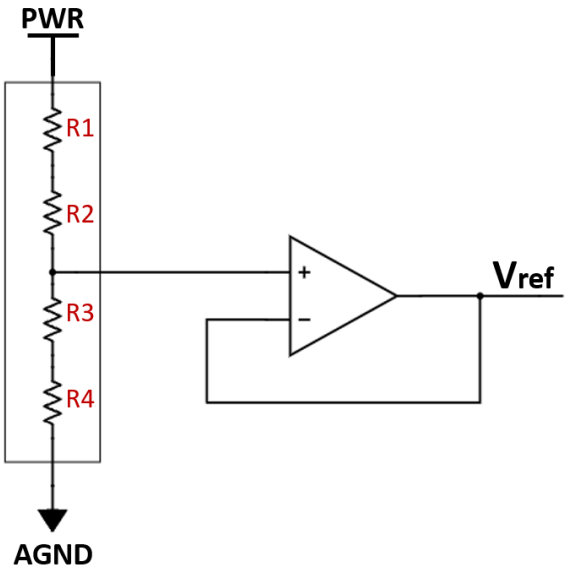

Figure 27 – Diagram of voltage reference circuit.

"PWR" pin in this figure can be AVDD (for Vhigh value generation) or AVSS (for Vlow value generation). For these reference buffer operational amplifiers, the output currents do not have to be large, so the silicon area they occupy is considerably smaller than the power amplifier that was used to provide the large output current for the 50 ohms load.

The main advantages of this topology are its high DNL and INL values and its simplicity. Due to a simple implementation circuit, without a large amount of transistors used, it is easier to achieve good linearity without using too much area. The main disadvantage of this approach is related to its resistance load value dependence. If the load varies slightly, the output current value will vary in the same proportion. For instance, a 50 ohms load receiving 500 mV will have the desired nominal current value of 10 mA, if the load is 20% smaller in resistance value (i.e. 40 ohms), the current value will be of 500 mV/40 ohms = 12.5 mA, which can be considered too high variation. Given that the MEMs actuators can have such high load resistance variations, this limitation becomes relevant.

4.2 V-to-I-Converter Topology Adapted to Reduce Supply Voltage

This section explores a possible adaptation of the topology used in the previous chip (chapter 3). The design goals of this adaptation were to reduce the complexity of the overall design, and hence the total amount of silicon used. Ultimately, it is thought to be a more robust solution than the previous voltage-mode DAC solution, considering that the load resistance in many applications can vary widely. In these situations, the driving device must be capable of supplying accurate values of current for a wide range of load impedances.

It was noted that the output stage of the last chip required special transistors to accommodate a relatively high supply voltage range, larger than 3.3 V. Such 5-V transistors are typically defined by their large silicon area and strict place and route rules. Adapting the original design range to operate with a reduced range (±1.65 V) removed the need for special transistors rated for higher voltage operation, thus reducing the complexity of the design. The following diagram shows the newly designed output stage.

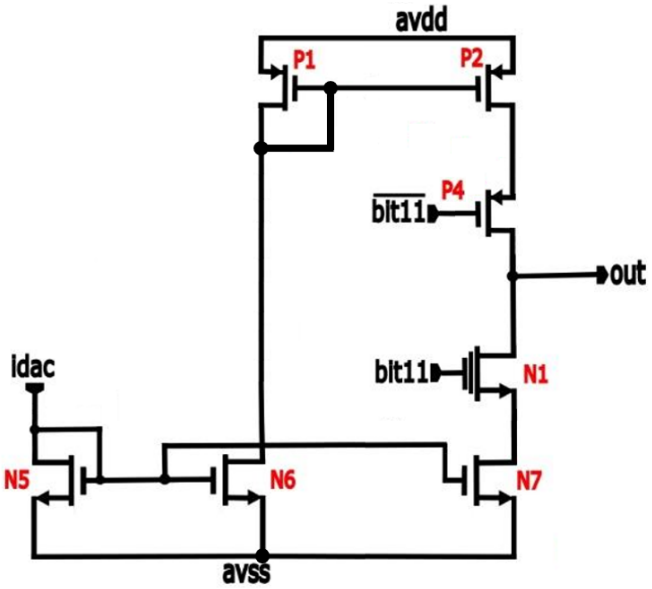

Figure 28 – Diagram of the modified output stage.

The other improvement was to implement a new operational amplifier that doesn't have the "dead-zone" previously explained in chapter 3, therefore causing major linearity improvement for low output current values.

All other topologies used in previous chip were maintained, only the transistors' dimensions were modified in order to better adapt to the new supply voltages.

4.3 Binary Current Steering DAC Topology

When we refer to current steering Digital-to-Analog Converters (DACs), the fully binary topology is the most basic and simple example of this type of design. This architecture directly uses the binary input signal to control an array of current sources, with a current scaled for binary weighting [24]. For example, digital input bit D0 (LSB) would correspond to a unit current (I_0), D1 would correspond to twice I_0, D2 would represent four times I_0, and so on. Binary weighting is often accomplished within each current mirror by providing the transistors of the current mirror with a different multiplicity value [18].

The binary DAC will ultimately have a binary-weighted current (or resistor) array, so each bit source is a current source (or resistor) scaled by a power of two ($2\hat{n}$). If a bit has logic level '1', the associated current sources send its current to the output load; if '0', it is typically directed to ground, then controlling the state of the associated steering switches. A diagram of the 10-bit fully binary DAC, responsible for providing only the sink current, designed in this work is shown below.

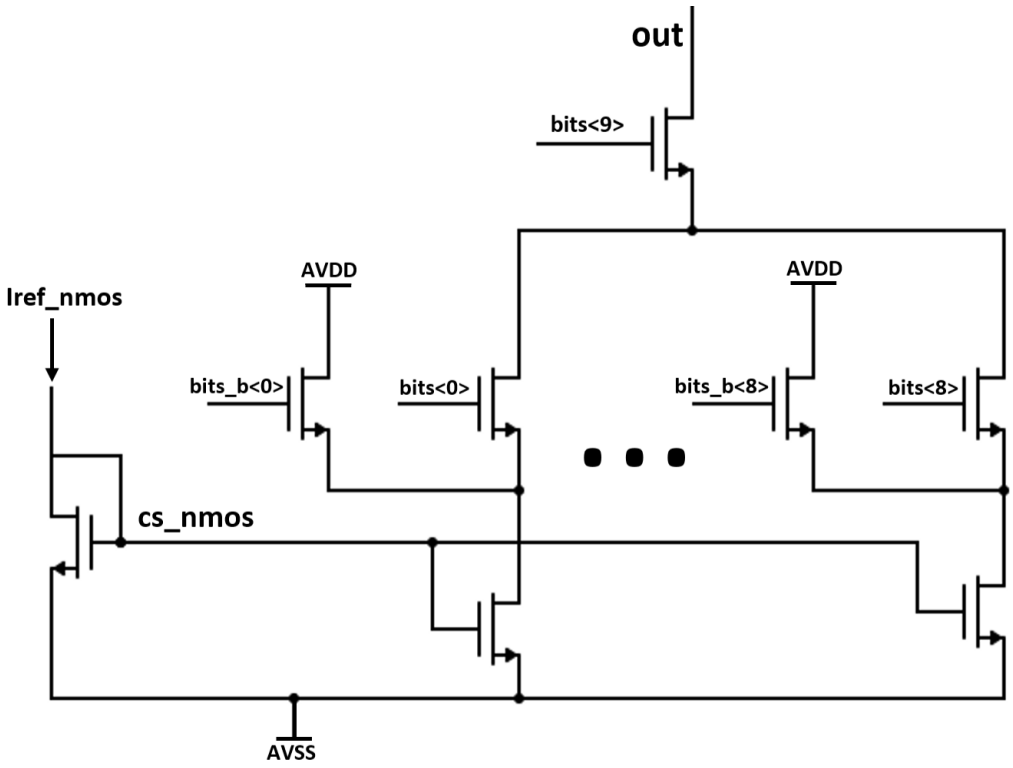

Figure 29 – Diagram of the fully binary topology.

All "bits_b" signals in the figure correspond to the inverted versions of their respective "bits" signals. This topology involves N current sources for N bits, which drive their respective switches' state, and the current from the array is summed at the output port to convert the digital signal into an analog signal: the output current that is sent to the driver's corresponding MEMS' actuator.

The benefits of a fully binary topology are mostly related to its smaller silicon area and lower power requirements. For higher bit DACs, it has also been reported to offer particularly fast transition speeds because there are less switching events involved than the ones that could often occur for more complex architectures. As such, it is suitable for certain usages in high speed applications [18].

Nonetheless, a significant drawback of this architecture is its vulnerability to mismatches in current branches, which can result in significant non-linearities and, most importantly, possible non-monotonic output [24].

In fully binary DACs, the matching and sizing of the current sources can become very difficult across a large dynamic range (from I_0 to $2^{(N-1)}I_0$) as the number of bits (N) is increased. Transistor mismatch can arise from many imperfections: lithographic variations, parametric spread or spatial gradients [18]. Mismatches at specific digital input transitions (e.g., from 011...1 to 100...0) are especially troublesome. In these transitions, many smaller current sources turn off while adding a new larger current source. If the currents are not

perfectly matched, it can produce a large error in output or a non-monotonic jump, in which the output can actually decrease while the input code increases. This non-monotonic behavior can be very common in fully binary DACs [18]. Therefore, fully binary DACs are usually not the best choice for high-resolution and high precision applications where monotonicity and linearity are considered relevant.

4.4 Unary Coded Current-Steering DAC

An unary current-steering DAC (also known as thermometric-coded DAC) works by activating a certain number of identical unit current sources based on the digital input signal. This architecture utilizes $2^N - 1$ identical current sources for N-bit resolution, and first converts the binary input code to a thermometer code, where the count of logic level 'ones' directly relates to the input value. For example, if the input signal is representing the decimal number 10 (binary code '1010'), then ten individual identical unary current source branches are activated, and a total of 10 times the reference unit current (Iu) is applied to the output load [18]. Figure 30 shows a diagram of the implemented 10-bit unary coded current-steering DAC.

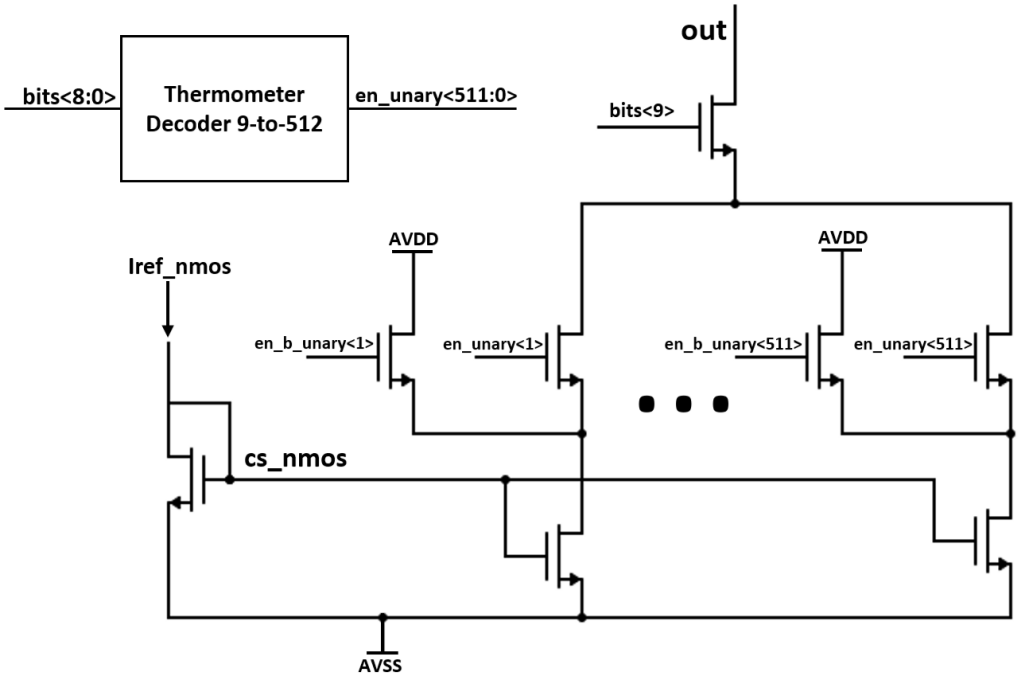

Figure 30 – Diagram of the fully unary topology.

All "en_b_unary" signals in the figure correspond to the inverted versions of their respective "en_unary" signals. In this topology, the first thermometer decoder output bit "en_unary<0>" is not used. One major merit of this approach, compared to the fully binary DAC, is that from the very beginning, it guarantees monotonicity [24]. It does

not allow for large jumps (non-monotonicity) at major input signal transitions, since it doesn't turn off large groups of scaled currents like the binary DAC. For the fully unary DAC, the subsequent input code simply activates another unit current branch of the same magnitude.

In addition, fully unary DACs tend to have better DNL characteristics, with most observed improvement being attributed to the "averaging" effect that occurs, in which you can balance the differences between currents from different branches by selectively activating branches spatially far from each other. That way, if one branch has higher current, the next to be turned on may have lower current and the average will be close to the ideal value. This effect is then explored with a good physical layout design, in conjunction with either certain dynamic element matching (DEM) strategies, or implementation of switching sequences [25]. Random variations between unit current branches can be spatially averaged by using arrangements like a common centroid layout or any number of other arrangements, as long as activation is done selectively [26]. Careful strategic spatial distribution minimizes gradients and thus the total INL error, whereas spatial distribution also improves distortion, as larger gradients yield larger signal distortion [18].

Conversely, the principle of spatial arrangement also means there is more complexity in implementation and a larger silicon area is needed. As said before, an N-bit DAC will utilize $2^N - 1$ individual unit elements. This can become quite large for high-resolution DACs. A 10-bit Digital-to-Analog Converter (DAC) would utilize 1,023 cells, which can easily result in complex routing and large area, especially for applications requiring high output current values. Additionally, the necessity of a thermometer decoder implementation also contributes to the increase in design complexity.

It should also be noted that the expected DNL and INL behavior depends greatly on the kind of DAC architecture. In binary-weighted converters, mismatches in the units depend much more strongly on the most significant bits, where larger DNL variations may end in larger INL distortion. Unary architectures use equally weighted unit elements. Hence, mismatch is spread uniformly in the array. Because of that, unary implementations generally show smaller DNL variations and a smoother INL profile, making them more suitable when linearity is a key requirement.

Statistically, the RMS DNL and INL are also strong functions of the DAC architecture: fully binary-weighted DACs have the largest RMS linearity errors since mismatch in the most significant elements dominates the overall distortion. Unary DACs attain the lowest RMS DNL/INL since the mismatch is averaged out over a large number of equally weighted unit elements. Segmented DACs fall somewhere in between: by thermometer-coding the upper bits and using binary weights on the lower bits, their RMS DNL and INL can be substantially reduced compared to a pure binary architecture, yet still maintain a significantly more compact area than a full thermometer array.

4.5 Segmented Current DAC Topology

The segmented current-steering Digital to Analog Converter (DAC) topology is a highly preferred hybrid architecture that integrates the functionality of binary-weighted DACs and unary (thermometric) DACs. The main focus on hybrid architecture is to create the suitable balance between linearity and area [18]. This configuration is made up of two separate sections:

- Binary-weighted branches for the Least Significant Bits (LSBs): In this section, a binary scaling scheme is used. As expected, the linearity errors related to the LSBs will have less impact on the INL and DNL values from the DAC output, since these carry less current. For instance, even a 50% error on a LSB is equivalent to just 25% error for the second bit, and half that value for the third bit, being less relevant for the output compared to the MSBs. Moreover, the matching of the LSBs current sources doesn't need to be as tightly controlled, which means that these bits can be easily implemented and occupy less silicon without a big drawback of precision.
- Unary (thermometric) branches for the Most Significant Bits (MSBs): For the MSB section, a thermometric encode style is implemented (i.e. turning current sources on and off in unit current source steps). This approach was chosen for these bits as it provides better linearity and monotonicity with greater mismatch tolerance than the binary approach.

The appropriate distribution of the DAC into these thermometric and binary bits is a design choice which depends heavily on linearity requirements and the available silicon area. For example, as a general practice for typical 10-bit designs, binary weighting generally applies to the first three or four LSBs, and thermometric segmentation is usually applied to the remaining higher order bits. This strategy allows designers to benefit from the area efficiency of the binary approach for the less critical bits while ensuring high linearity for the most relevant contributors to the analog output (the MSBs). Diagram below shows the designed 10-bit segmented current steering DAC, which uses 3 binary branches for the LSBs and 63 unary branches for the MSBs (bits 8 to 3).

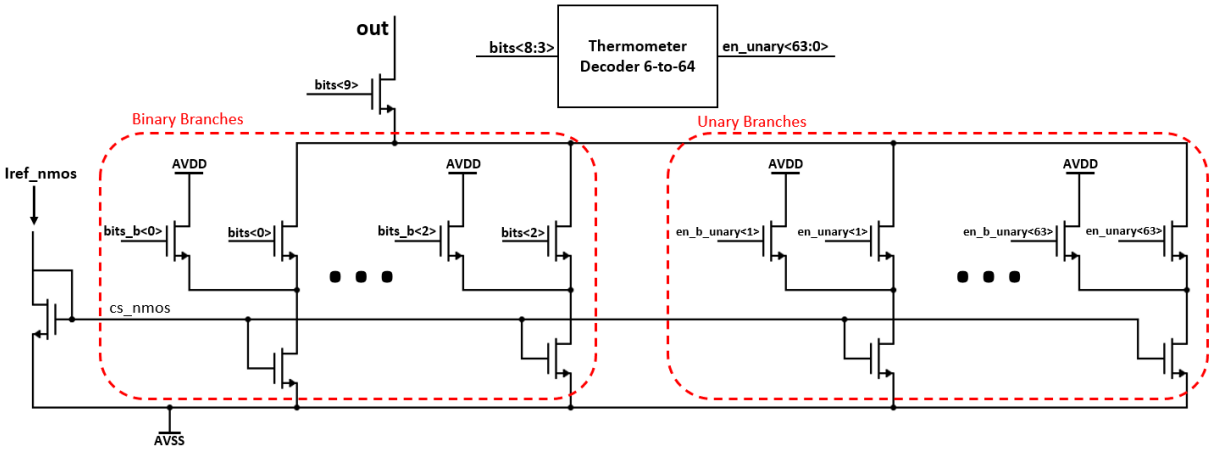

Figure 31 – Diagram of the segmented topology.

A significant downside of the segmented architecture is how complex the design is. The two different switching methodologies employed, plus the added variations in DAC's underlying circuits, lead to additional complexities when comparing with previously presented approaches. In particular, a challenge of this approach is how the two segments respond differently to the input digital signals; nominally the propagation delay is longer to the unary switches of the MSB section, because the binary MSB signal must go through the thermometer decoder circuit to get its thermometric equivalent before it can enter the individual current-steering cells. This additional electronic processing time results in longer propagation delays for the MSB than the LSB.

The disparity in signal arrival times, called timing skew, heavily contributes to glitches during transitions between different input codes. These are bad for a DAC's dynamic performance as they cause a substantial amount of distortion, especially for higher frequency operations [18]. One of the most effective methods to help reduce glitches and timing skew is to use registers (latches) for all the input signals for controlling all the switches in both unary and binary sections [24]. By clocking all the registers at the same time with a common clock signal, then the input signals to all the switches being controlled will come from the same synchronized instance (register). From a control standpoint, the goal is to transition all the current-steering cells at the same instant. Another useful approach for reducing glitches is to use clock trees, for clock distribution, in order to construct the clock signal and route it to all destinations equidistantly, mitigating the timing skew [18].

However, even with the implementation of registers and clock trees, this isn't enough to guarantee perfectly zero delay differences. Residual timing variations can still arise due to layout effects. For instance, each switch control signal will follow a shorter or longer physical path from the register output to its specific transistor switch of its

respective branch. These minor differences in the interconnect lengths will introduce varying parasitic capacitances and resistances. These layout dependent effects can be characterized very accurately by running an RC extraction from the layout design, and simulating the circuit again taking account of these parasitic capacitances and resistances. This extensive post-layout verification is an important step to realize the high linearity and dynamic performance ultimately required for advanced DAC designs.

5 The developed Serial Peripheral Interface

5.1 Basic concepts

A Serial Peripheral Interface (SPI) is a very common synchronous serial communication protocol used for short distance communications, especially when working with embedded systems, connecting microcontrollers to peripheral integrated circuits (IC), such as sensors, analog-to-digital converters (ADC), digital-to-analog converters (DAC), shift registers, etc.

The SPI protocol allows for sets of data to be sent from two devices: the controller (or initiator) and one or more peripheral devices (or targets). The SPI peripheral from this work was designed to receive digital information from the controller, convert this serial data into a parallel format, and then feed these data into the current drivers, serving as input data that guides these drivers to provide the desired output current values.

5.2 Designed SPI

The main steps for executing the design of this digital SPI unit started off using the Verilog language to write the SPI block description and running initial simulations using Cadence Xcelium Logic Simulator in Cadence Virtuoso Analog Design Environment (ADE) to make sure the unit performed as expected [27]. The Verilog code is included in the Annex A.

The RTL code was logically synthesized with the Cadence Genus tool, which converted RTL to a gate-level netlist, with power and area reports generated from this process. Then, the logical netlist was physically implemented in Cadence Innovus, while also performing project setup and configuring a Multi-Mode Multi-Corner (MMMC) environment for robust analysis under multiple operating conditions. The work in Innovus then moved onto power distribution and floorplanning, which included defining the core area, generating power rings and putting in power stripes for good power coverage. Still in Innovus, standard cell placement was done with area optimization, followed by I/O pin placements where pins for outputs were distributed on the edges of the design. A Clock Tree Synthesis (CTS) was also performed to distribute the clock signals with no transition violations and no skew errors. Next, a detailed routing was performed using the NanoRoute engine of Innovus [28]. For last, the physical verification steps of Design Rule Check (DRC) and Layout Versus Schematic (LVS) were performed in both Innovus and Virtuoso (OpenAccess database format granted the SPI library's compatibility between

these two tools), completing the flow with parasitic extraction using Quantus QRC [27]. It is important to add that the generated layout from Innovus must be imported by the Virtuoso tool. The simplified flow described above is also discussed in [27].

The original SPI unit, which was made for chip V1 for initial validation, had only 4 drivers with 16 bits of input data on each. The new version, designed to operate 50 drivers, is shown below.

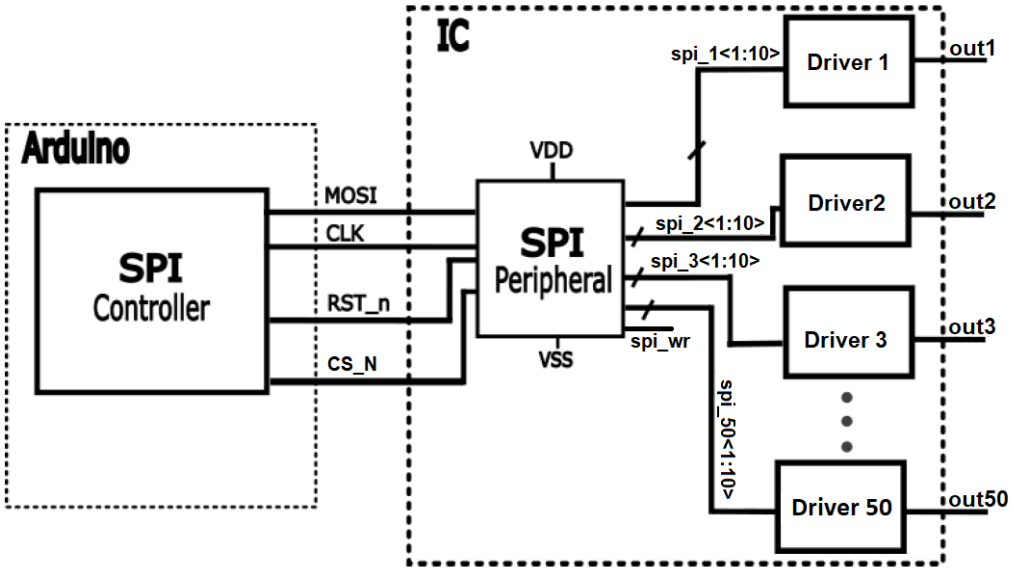

Figure 32 – SPI controller to SPI peripheral block diagram.

The SPI input signals are explained below:

- mosi: provides the serial input binary data that the user wants to write to the drivers input bits.
- clk: the clock signal that regulates the speed in which the input serial data ('mosi') will be processed for writing the SPI internal registers. The nominal clock frequency value is 2.5 MHz.
- cs_n: the enable signal. When it is 0, the SPI registers are being written with the serial data provided by the mosi signal.
- rst_n: resets all output data to logic value '0' if its value is '0'. Therefore, for correct SPI operation, rst_n must be set to '1' during the drivers' normal operation.

The output signals are the output bits for the drivers ("spi_n" bits) and the "spi_wr" signal, which informs if the SPI is currently writing new data into the dat_o bits. During every clock cycle, 1 bit is transmitted. After 16 bits are sent, these 16 bits are simultaneously updated at the spi_n output for one of the 50 drivers, and the new

set of 16 bits for the next driver is received, in a continuous process. It is important for this SPI unit to be designed to receive a set of 16 bits per driver since this is the official pattern from the SPI Controller.

During simulations of the initial SPI unit, used for chip V1, it was observed that, after the third full writing cycle, the output data from the original SPI was no longer overwritten, even though the bit sequence continued normally in the mosi input. This was a limitation from the original SPI driver which was studied and improved for the new SPI unit. Another main target for enhancing the SPI unit was to allow it to write to a maximum of 50 drivers, instead of just 4. However, some other enhancements were applied, as detailed below:

- Removed redundancies in the original code, reducing total number of registers per output.
- Parameterized the code to allow the implementation of any number of output drivers, not necessarily 50.
- Modified the Verilog code for easy setting of the number of desired drivers, by simply changing the parameter 'number_drivers' and recompiling the Verilog code.
- Changed the output of the SPI unit to send only the 10 LSBs from the 16 bits received for each driver, since the implemented driver has only 10 input bits, therefore reducing digital circuit 'waste'. Although the SPI unit needed to receive 16 bits as the official bit pattern from the SPI controller, the output didn't need to have this same pattern.

Therefore, for the new SPI unit, 50 drivers with 16 input bits on each were considered. Therefore, 500 output data bits for the SPI, plus the spi_wr signal, as shown in Figure 32. The expected behavior of the SPI signals is shown in figure below.

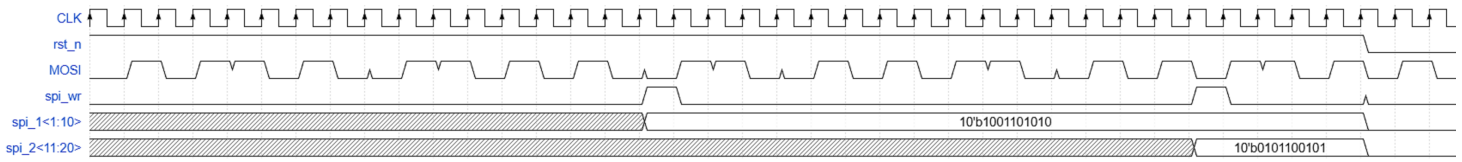

Figure 33 – Expected behavior of the SPI output signals for a specific 'MOSI' input.

In this figure, it can be observed that after every group of 16 clock cycles, one of the driver input data lines is updated sequentially, starting with spi_1, followed by spi_2, and so on. However, only the least significant 10 bits of the 16-bit MOSI sequence are effectively stored, while the first 6 bits are discarded. Additionally, the spi_wr signal is

asserted high for one clock period immediately after each driver input register captures the new data, thereby indicating that a valid data update has been recorded.

The figure below illustrates the behavior of the cs_n signal in this SPI unit.

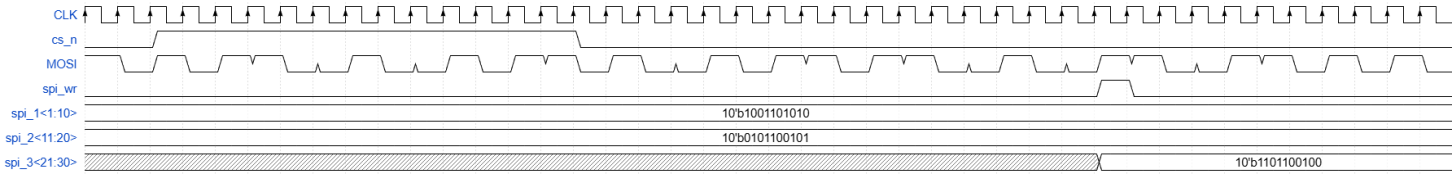

Figure 34 – Expected behavior of the SPI output signals under changes in the cs_n signal.

As shown, while cs_n remains at logic level '1', the incoming MOSI sequence is ignored and no data is written to the driver registers. Once cs_n returns to logic '0', the writing process resumes, and the data word 10'b1101100100 is correctly stored into the third driver (spi_3<21:30>).

5.3 SPI simulations

In order to validate the correct functioning of the SPI unit (described in Verilog code), 56 random bit sequences were chosen for mosi's input data. Considering that this sequence should repeat itself indefinitely in mosi's input after it finishes, the expected output data is described in Table 3.

Table 3 – Bit Sequences Used to Test the SPI and Their Expected Output Values

Driver	**16-bit Binary Input Value**	**10-bit Binary Output Value**	**Expected Decimal Output Value**
1	0000010111110000	0111110000	496
2	0000001001101110	1001101110	622
3	0000000011110011	0011110011	243
4	0000010110010010	0110010010	402
5	0000011011110111	1011110111	759
6	0000000010100100	0010100100	164
7	0101110000101000	0000101000	40
...			
50	0101010101010110	0101010110	342
1	1000001110100100	1110100100	932
2	1000101000010000	1000010000	528
3	0011011011010000	1011010000	720
4	1001011110110100	1110110100	948
5	0010110110101101	0110101101	429
6	1011011101101010	1101101010	874
7	0000010111110000	0111110000	496

From the table, it can be seen that, since 56 code patterns were tested, the 6 initial drivers will receive new codes, while the seventh will receive the same code that driver 1 had in the first place: 0111110000, which corresponds to 496.

To evaluate the functionality of the SPI Verilog code, three distinct test scenarios were considered:

1. Nominal SPI operation: with cs_n always in logic level '0' and rst_n starting as '0' and changing to level '1' after 1us.
2. Enable test: cs_n signal starting with logic level '0' and going to logic level '1' after some period, interrupting the SPI internal register writing.
3. Reset test: rst_n signal set to '0' after some time, resetting all outputs from the SPI block.

After all tests were performed, it was concluded that the SPI Verilog code worked as expected. Figure 35 shows the nominal operation output results, with each of 10 bit output binary words converted into unsigned decimal format for easier visualization.

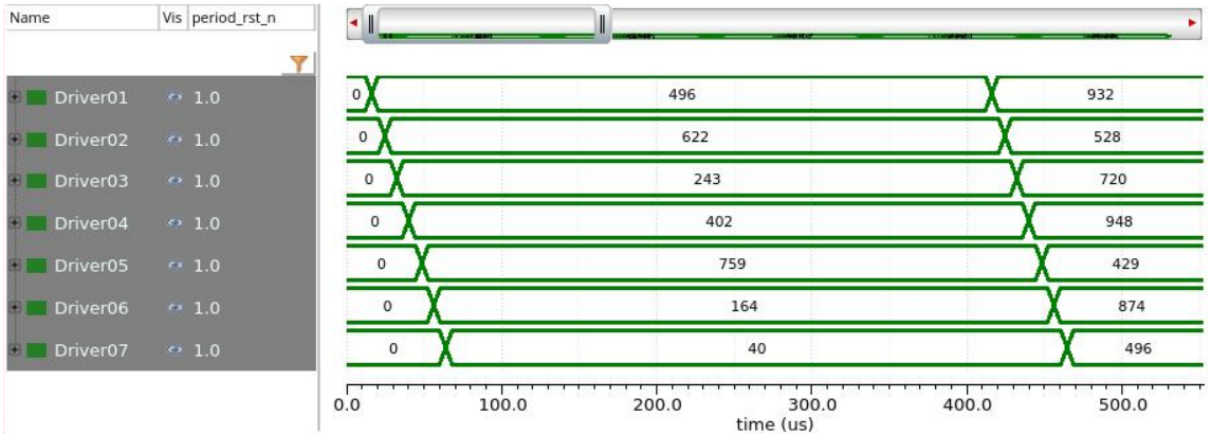

Figure 35 – Results from SPI simulations under nominal operation.

Under nominal conditions, with 'cs_n' equals to '0' and rst_n equals '1', the output data from each driver is sequentially written based on mosi's input signal, having the exact values predicted in Table 3. Also as expected, Driver 7 had a second value of 496, since all 56 input code tests had finished and the sequence restarted. Note that digital word of each driver (10 bits) is updated every 16*(1 / 2.5 MHz)= 6.4 μs.

5.4 Synthesis

After the Verilog code was implemented and tested, the Cadence Genus tool was used to synthesize the logic circuit from that code.. Its main role is to translate a RTL code (like Verilog or VHDL) into a gate-level netlist, which can then be used for physical design (placement, routing, etc.).

After synthesis was complete, the area and power reports were generated. They are included in the Annex B and are explained below in detail.

The total predicted area for the design indicates $243,517\ \mu m^2$, which is equivalent to 0.243 mm^2. The area includes 0.191 mm^2 for the standard cell instances (logic gates, registers, etc.) and 0.052 mm^2 for routing interconnects between the cells. The high proportion of cell area (around 78.6%) compared to the routing net area (21.4%) indicates a design with a fairly dense cell placement, with the design well-optimized in terms of area usage.

The power report shows that the design consumes a total of 0.29 mW, in which about 75.18% of the total power consumed by the design is internal power, mainly from short-circuit and charging/discharging internal nodes when switching. About 24.82% of the power consumed is switching power (caused by capacitance on nets) and leakage power is negligible. The major consumer of power is the registers, which account for 75.72% of total power consumption, followed by clock network at 20.64% and by logic gates at 3.64%.

More details about the developed SPI design flow can be found in the paper [27].

5.5 Layout

The SPI unit layout, developed by Bárbara Mariana Rocha Raimundo during her final engineering project using Cadence Innovus software, is shown below.

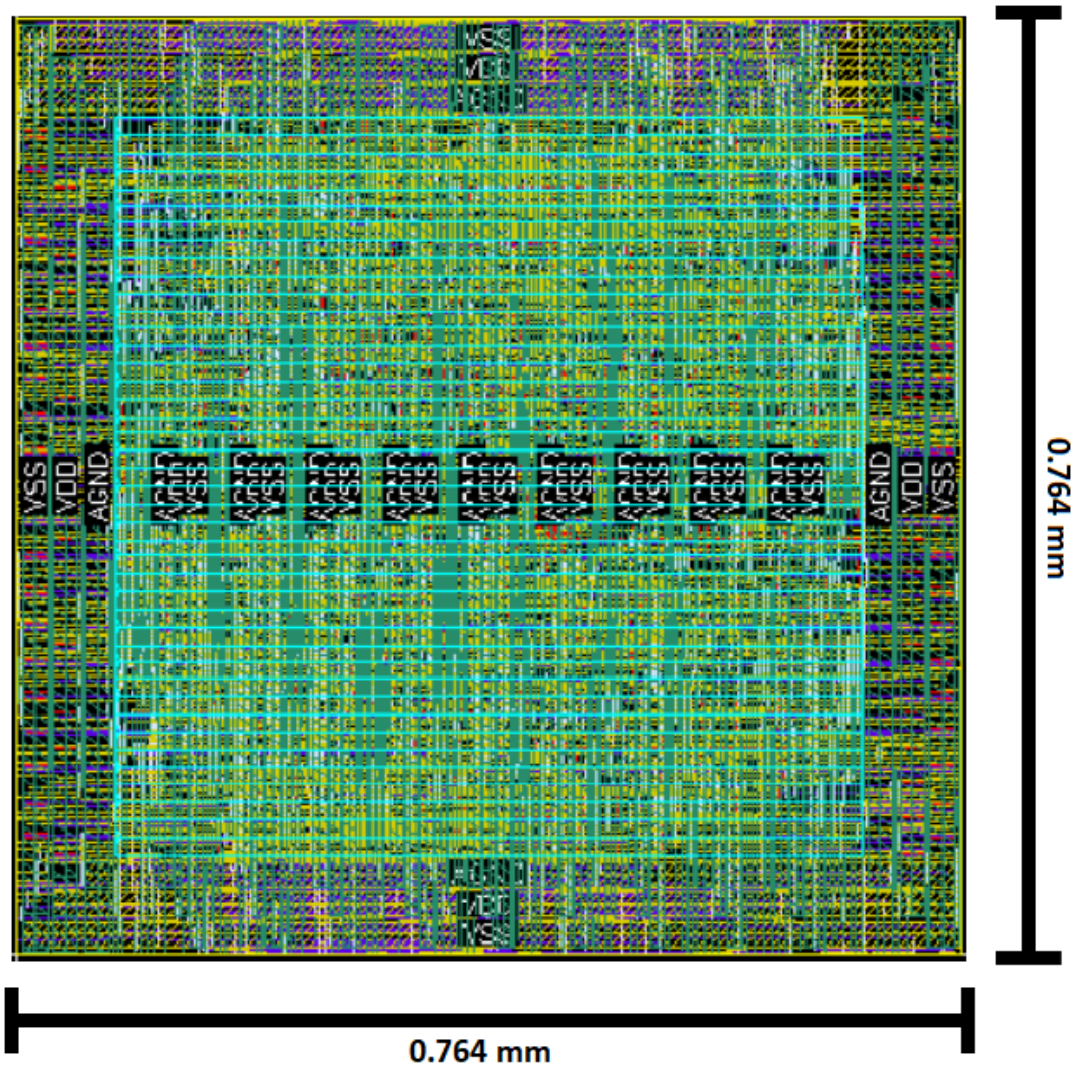

Figure 36 – SPI layout.

The actual area, after all placing and routing performed, was of 0.584 mm².

6 Designed Current Driver Topology

6.1 Schematic

Based on several different tests, focused mainly on the linearity vs area tradeoff, the chosen topology was the segmented DAC with 3 binary bits for the LSBs (bit<0>, bit<1> and bit<2>) and 6 unary bits for the MSBs (bit<3> to bit<8>). This approach is used to balance the complexity, large area and higher power consumption associated with the fully unary DAC, while maintaining good performance characteristics, assured with the binary structure for the 3 least significant bits, in which accuracy is less critical. Figure 37 shows the driver schematic.

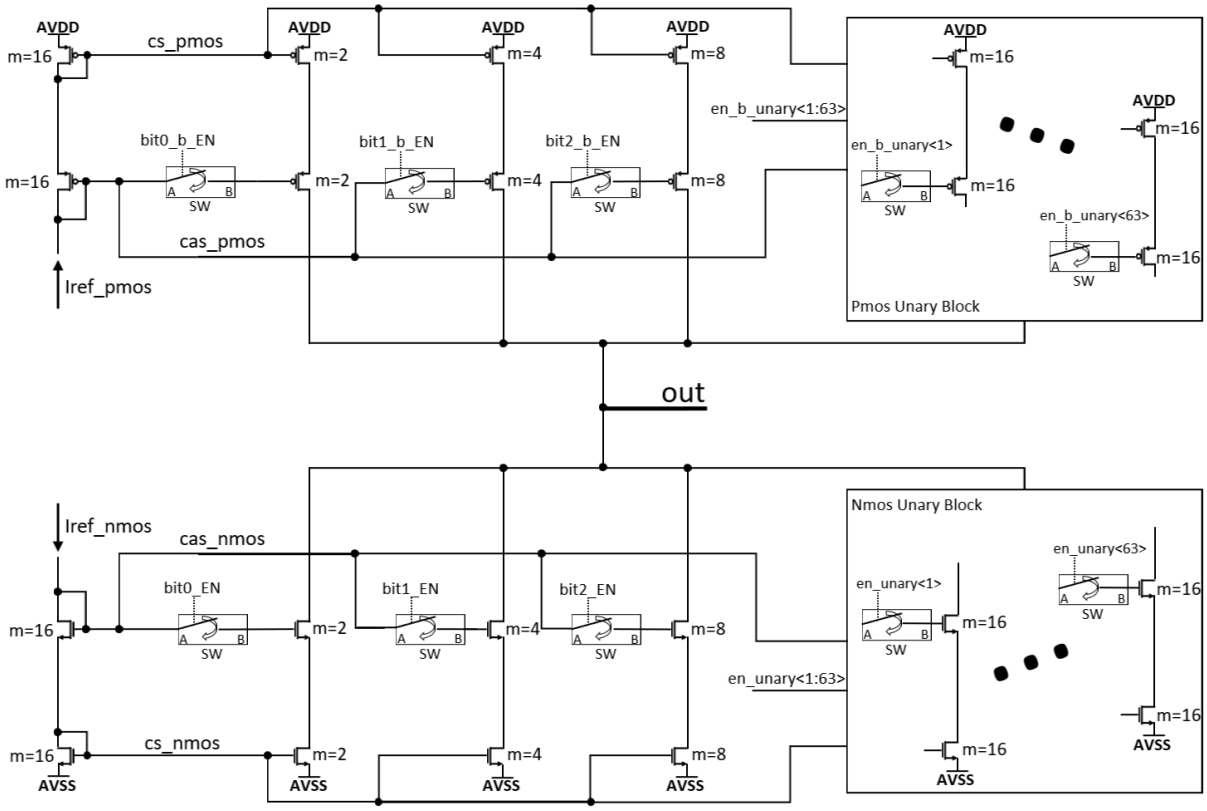

Figure 37 – Segmented DAC topology developed.

The tenth bit (called bit<9>) acts on control signals that choose if the current comes from the PMOS instances or from the NMOS instances, therefore being responsible for the current signal. If bit<9> has logic level “1”, current is negative (sink) when referenced to the grounded load, coming from the NMOS instances. If bit<9> has logic level “0”, current is positive (source), coming from the PMOS instances. The figure shows

that we have 3 binary branches for each side, representing the current that gets activated with bits bit<0>, bit<1> and bit<2>. The unary branches are represented by the iterated instances, as shown below.

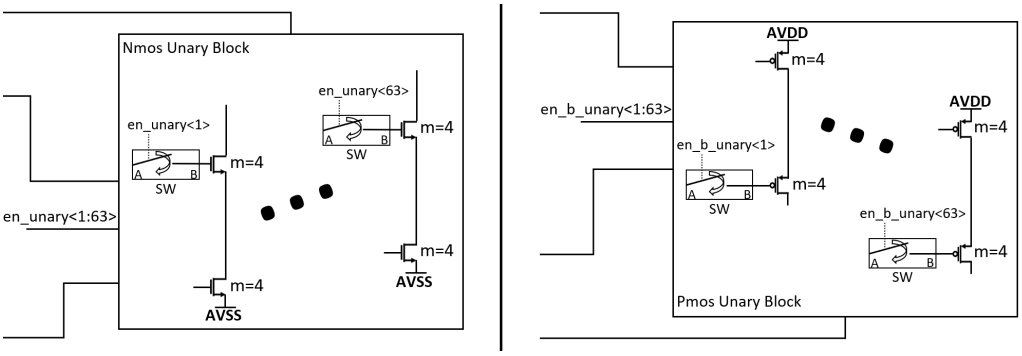

Figure 38 – Unary blocks of the segmented DAC topology developed.

For both PMOS and NMOS part, we have 63 unary branches. As explained in chapter 4, for the unary branches a conversion of the bit signals into thermometer code needs to be done. In this case, bits bit<3:8> are converted into unary<63:1>. The method used for this conversion will be explained in a later section.

Besides the unary and the binary branches, the other important parts of this driver's design are the diode-connected transistors of the current mirror for the PMOS and NMOS branches, which are shown in Figure 39.

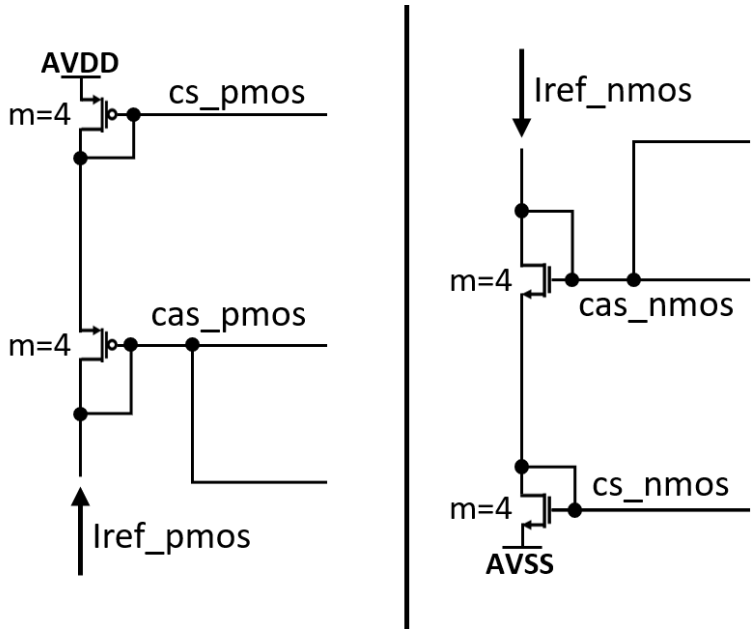

Figure 39 – Diode-connected transistors of the current mirror in the developed segmented DAC topology.

The current value received on each branch is mirrored to all the binary and unary

branches. The composition of the current cell for these branches will be explained in the next section.

6.1.1 The current source and current sink cells

The designed current cell for the NMOS sink block is shown in Figure 40.

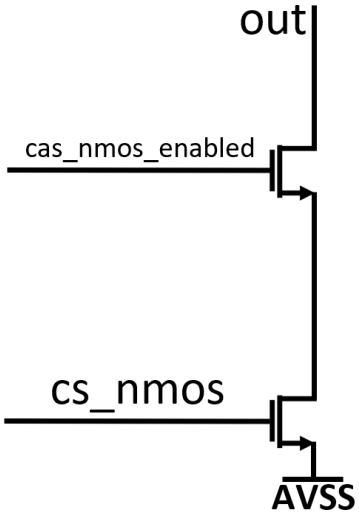

Figure 40 – Current cell of the NMOS sink block in the developed segmented DAC topology.

The chosen topology uses cascode transistors mainly to improve the current linearity. The main advantage of this approach is the considerable increase in the circuit's output impedance. The high output impedance ensures that the output current is more independent of variations in the drain voltage [7]. So it is more robust, including for temperature and process variations [10].

While the voltage in the drain and gate terminals of the transistor in the current mirror "MNO_CS" feeds the "MN2_CS" transistor gate's voltage directly, the gate's voltage value of the cascode transistor MN2_CAS is driven by a signal that comes from a switch unit, which will be explained in detail in the next section. The current cell for the source part (made of PMOS transistors) has the same format from the NMOS sink part, as shown below.

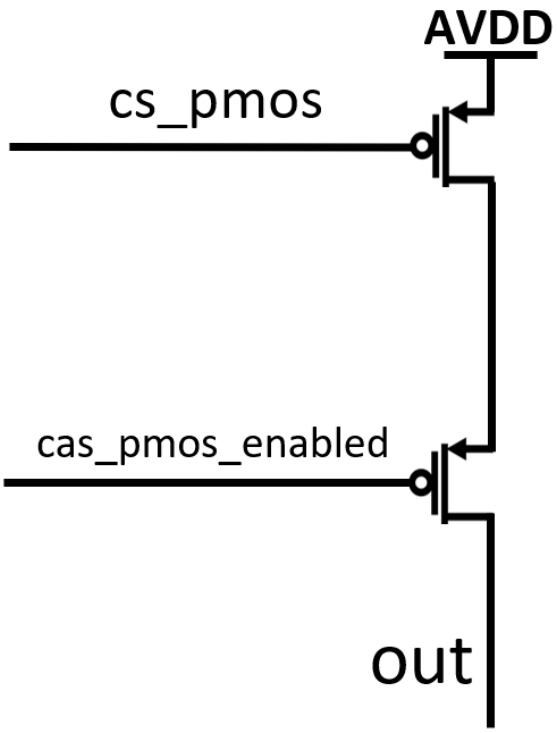

Figure 41 – Current cell of the PMOS source block in the developed segmented DAC topology.

For all the unary branches, the cascode and the current source transistors have a multiplicity of 4, which matches the multiplicity of the reference branch, meaning the same amount of current will flow through them. With regards to the 3 binary branches, we have following requirements:

- Binary branch of bit<2>: must receive $\frac{1}{2}$ of the current value from the reference branch.
- Binary branch of bit<1>: must receive $\frac{1}{4}$ of the current value from the reference branch.
- Binary branch of bit<0>: must receive $\frac{1}{8}$ of the current value from the reference branch.

The implementation of binary branches which ensure these current values is shown below, for the NMOS sink part.

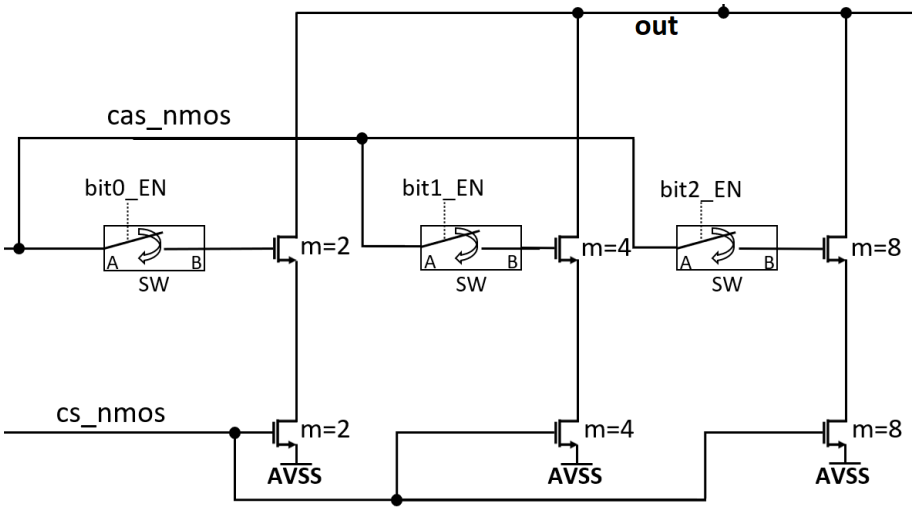

Figure 42 – NMOS sink binary branches in the developed segmented DAC topology.

For both the cascode and the current sources, the design strategy applied was to reduce the multiplicity of the transistors based on each desired current value. So the binary branch of bit<2> has transistors with multiplicity 2, and the binary branch of bit<1> has transistors with multiplicity 1. Due to the impossibility of setting a multiplicity value of 0.5 to bit<0>, two identical transistors are put in series for both the cascode and the current source units. That way, this arrangement is equivalent to setting a double length value to these transistors (2 W/L transistors in series correspond to 1 W/2L transistor). The same scheme is used for the PMOS source part, as shown in Figure 43.

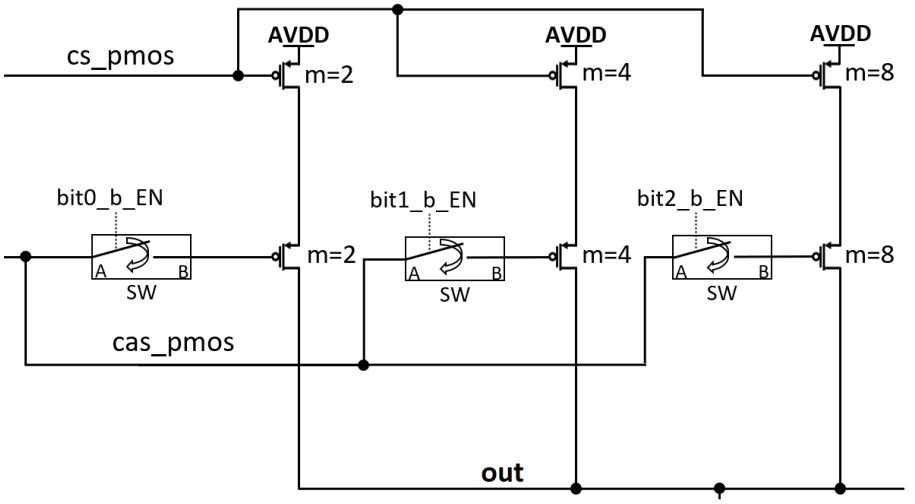

Figure 43 – PMOS sources binary branches in the developed segmented DAC topology.

Therefore, in the end, we have these current values for each branch:

- 63 unary branches with the full current value from the reference branch on each of them.

- 1 binary branch with $\frac{1}{2}$ of the current value from the reference branch.
- 1 binary branch with $\frac{1}{4}$ of the current value from the reference branch.
- 1 binary branch with $\frac{1}{8}$ of the current value from the reference branch.

This means that we have a total amount of current given by following equation:

$$63 \cdot I_{\text{ref}} + \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8}\right) I_{\text{ref}} = \left(63 + \frac{7}{8}\right) I_{\text{ref}} \tag{6.1}$$

In which Iref is the current magnitude that feeds the reference branch. Considering that we want a maximum output current value magnitude of 10 mA, it is easy to calculate the necessary Iref value:

$$\left(63 + \frac{7}{8}\right) I_{\text{ref}} = 10\,\text{mA} \quad \Rightarrow \quad I_{\text{ref}} = 156.56\,\mu\text{A} \tag{6.2}$$

Due to observed second order effects, the reference current had to be increased to 156.86 μA.

6.1.2 The switch on CAS

Considering that this design is intended for a low frequency application, in which dynamic errors are not as critical for proper operation, an alternative to the use of the current steering was found in [3]. It consists basically of using a switch in the cascode transistor's gate, in which its voltage will be either set to the supply voltage ("AVSS" for NMOS case, "AVDD" for PMOS case), when the switch is off, or to the expected voltage from the reference branch cascode transistor ("MN0_CAS") drain and gate voltages, when the switch is on. The switch can be placed in the gate of the cascode transistor from the current cell in order to avoid glitches and dynamic errors which would occur when the switches were on the transistor drain. This approach is shown in figures below, for both NMOS and PMOS cases.

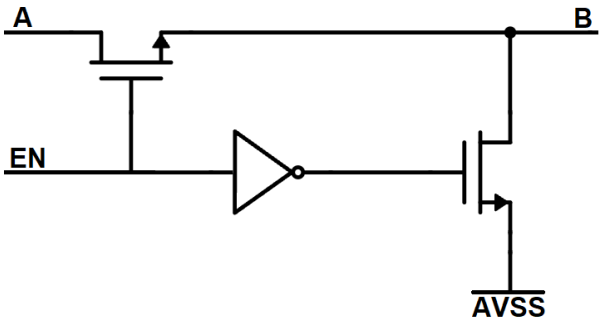

Figure 44 – Switch architecture for the cascode in the NMOS sink blocks.

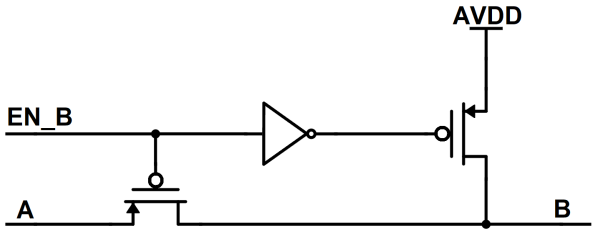

Figure 45 – Switch architecture for the cascode in the PMOS source blocks.

For PMOS switch, the enable is called “EN_B” because it enables the signal from A to B when EN_B is on logic level 0, not 1.

The main limitation of this approach is its speed: the time to charge and discharge the cascode voltages makes this approach unsuitable for high-frequency operation. Also, the alternative of using current steering switches is more proper to drive loads differentially, due to their complementary outputs, which is not the case for this application, in which the load needs to be grounded.

The main advantage of this approach is that it helped to significantly reduce the driver total area while still guaranteeing proper switching, considering this is a low frequency application (2 kHz).

6.1.3 The thermometer decoder

As explained in chapter 4, in order to send the correct control signals to the unary elements, a thermometer decoder needed to be designed. A thermometer decoder translates an N-bit binary input into a thermometer coded output, which is a sequence of bits where we have all the logic levels “0” appearing first, followed by logic levels “1” after. Table 4 gives an example of this decoder “truth table” for the case of 3-bit binary input.

Table 4 – Binary to Thermometer Code Conversion (3-bit Example)

Binary Input	**Thermometer Code Output**
000	0000000
001	0000001
010	0000011
011	0000111
100	0001111
101	0011111
110	0111111
111	1111111

Considering that we have 63 unary elements and 6 binary input bits, a 6-to-64 decoder was implemented, with the first output bit (“thermometer_state<0>”) being

discarded. Figure below shows the decoder block.

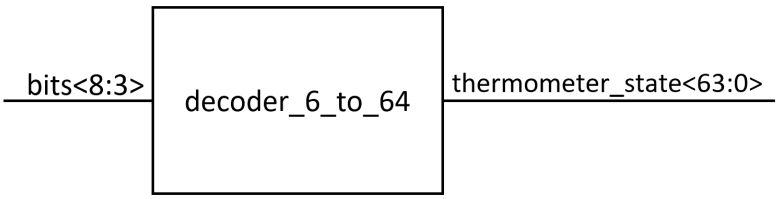

Figure 46 – Figure: 6-to-64-bit decoder block showing its input and output connections.

In order to implement this decoder, a set of 3-to-8 decoders were implemented first, as shown in Figure 47.

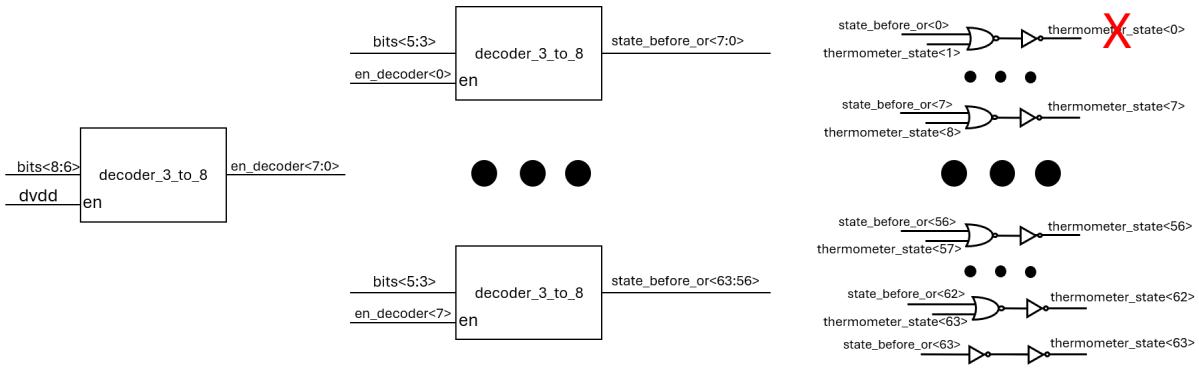

Figure 47 – Detailed block diagram of the 6-to-64-bit decoder architecture.

The first 3-to-8 decoder receives the 3 MSBs (bits 6, 7 and 8) and uses them to decide which ones of the 8 other decoder instances to enable. The LSBs (bits 3, 4 and 5) decide how many states from these decoders to put on logic level 1. At these 3-to-8 decoder outputs, a set of NOR+NOT logic gates (which correspond to OR gates) ensure that the thermometer_state<i> will receive logic level 1 either if state_before_or<i> is 1 or if the thermometer_state<i+1> is 1. This is the expected behavior for the 6-to-64 thermometer decoder, because if a specific output bit is 1, all other output bits of a lower magnitude will necessarily be 1 too. Therefore, all the logic values “0” will happen first in the bit sequence.

Each one of these 3-to-8 decoders have the following structure from Figure 48.

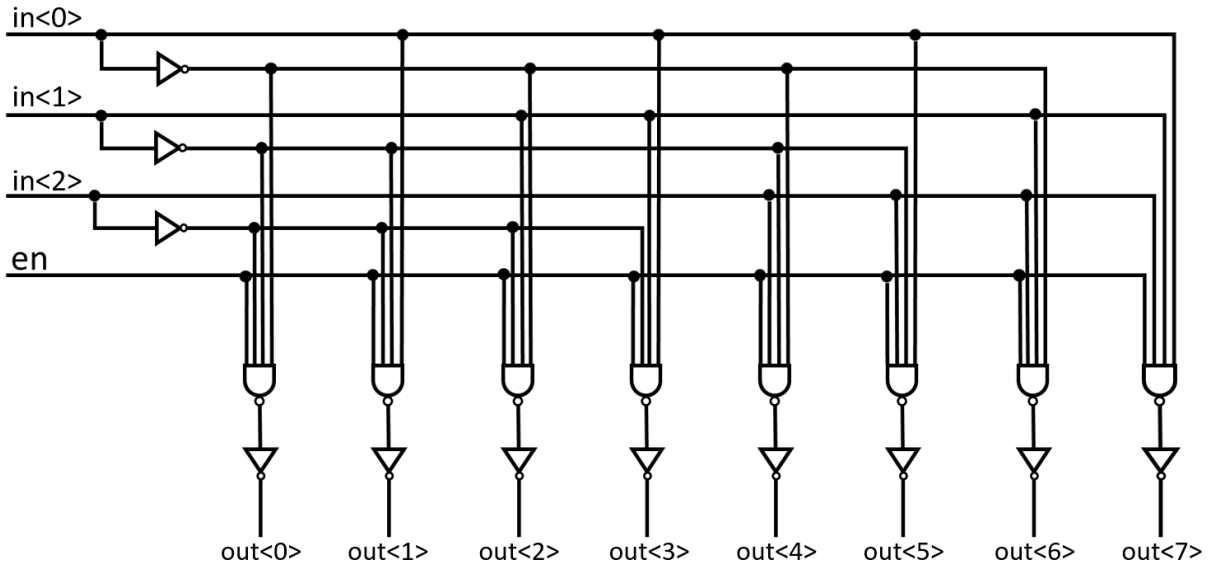

Figure 48 – Schematic circuits of the 3-to-8-bit decoder.

From the circuit inspection, it can be easily assessed that its truth table corresponds to the one from Table 4.

This thermometer decoder design is also useful to reduce glitches, because it ensures that only one current cell transition will happen at a time during major input code changes, reducing the risk of transient spikes that would degrade the SFDR [7].

6.1.4 The current direction bit

Since this driver requires both negative and positive current values in order for it to have proper operation, bit<9> (the MSB) acts as a switch that activates the NMOS sink currents, when the desired output current is negative, and the PMOS source currents, when the desired output current is positive. In order to better implement this logic without having to use actual switches between the current sources and the output current net, a digital circuit was implemented to control the enable (or "S") signal that goes to each switch from each binary or unary branch. The digital circuit is shown below.

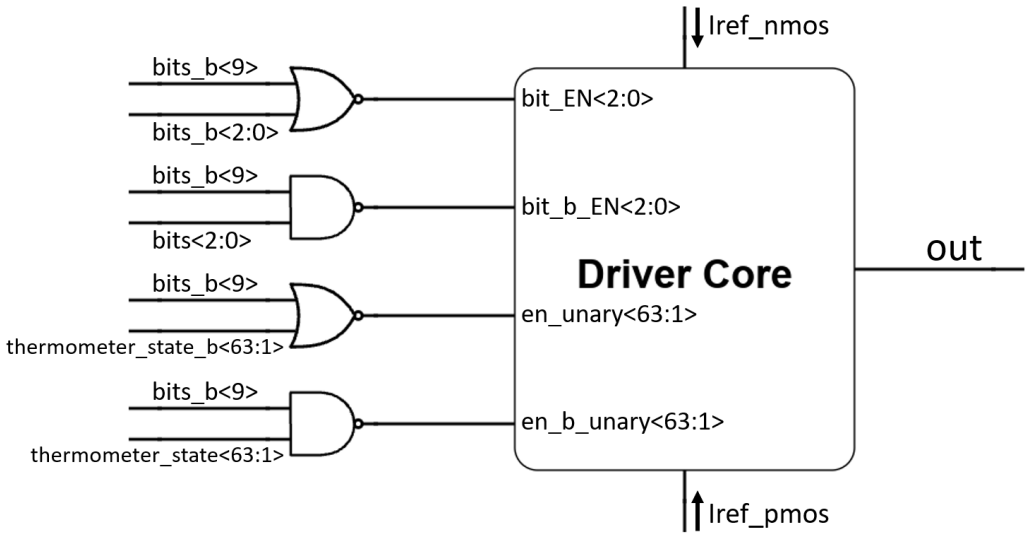

Figure 49 – Schematic circuits of the digital logic applied to the current direction bit to ensure correct activation of binary and unary branches.

This circuit is simple. All "name_b" signals in the figure correspond to the inverted versions of their respective "name" signals. The NMOS cascode transistors will only have the appropriate signal propagated on their gates if bit<9>=1 (or bit_b<9>=0). Case not, means that no current should come from NMOS, and all the switches will be turned off for the NMOS transistors, putting all NMOS cascode transistor gates in "AVSS", therefore deactivating all of them. And the same logic happens to PMOS cascode transistors when bit<9>=0 (or bit_b<9>=1).

It is important to note that one of the design attempts for this driver was to create a single type of CMOS current source (either PMOS only or NMOS only), and then just mirror its total current to a second branch in which bit<9> would choose the current direction, as shown in Figure 50.

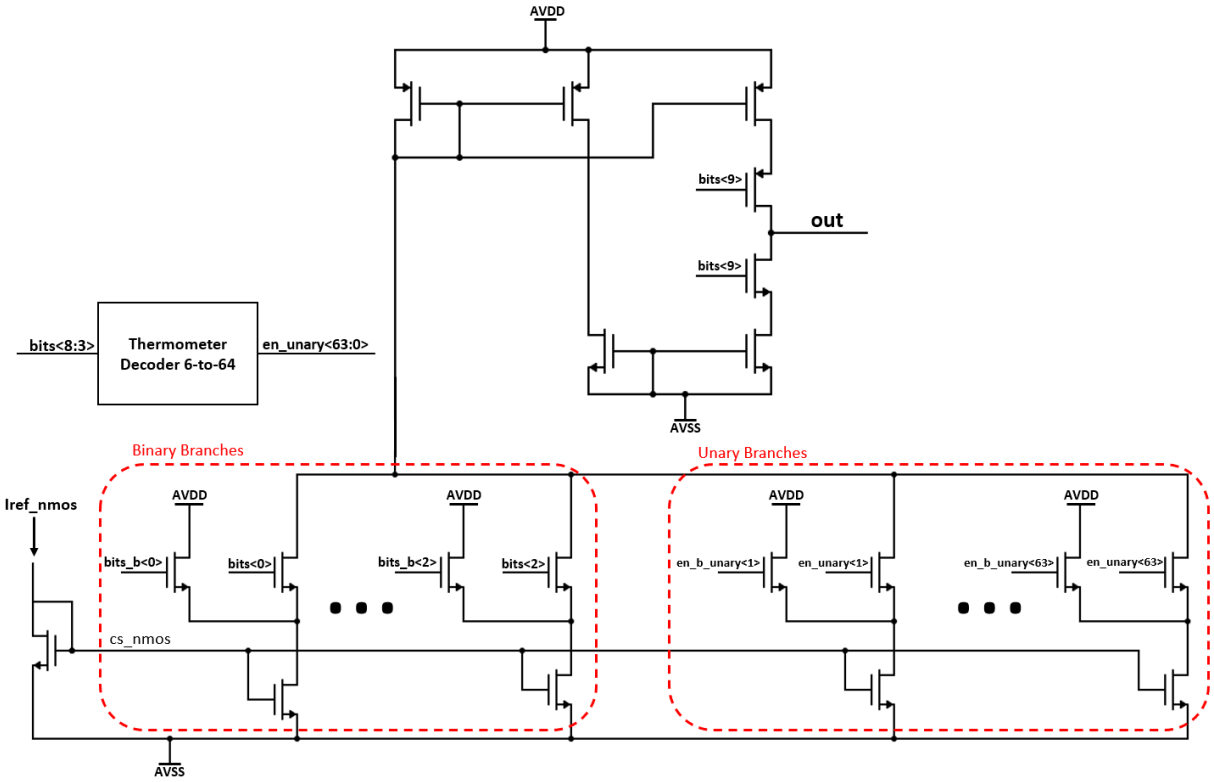

Figure 50 – Alternative segmented DAC topology tested.

This approach was attempted with both the current steering (shown in the figure) and with the switch on cascode topologies. However, the resulting INL value of this solution, for realistic area dimensions, was poor (> 2). So this solution was discarded. A solution with 4 binary bits and only 5 unary bits was also attempted, but discarded for the same reason: not enough accuracy.

6.1.5 The bank of registers

In order to reduce the glitches from the switching process between input bits, a bank of registers was implemented for all the signals that activate both the binary and the unary branches. The idea is that all the output bits that activate these instances would be sent at the same time. Considering that the unary activation bits come from a decoder and that the binary bits come directly from the SPI, it is important to have this synchronicity in signals. So the registers aim to solve these timing mismatches, helping to ensure simultaneous switching. Important to note that parasitic effects from physical layout of the instances will also cause delays that can't be solved using these registers.

The designed bank of registers is shown in figure below.

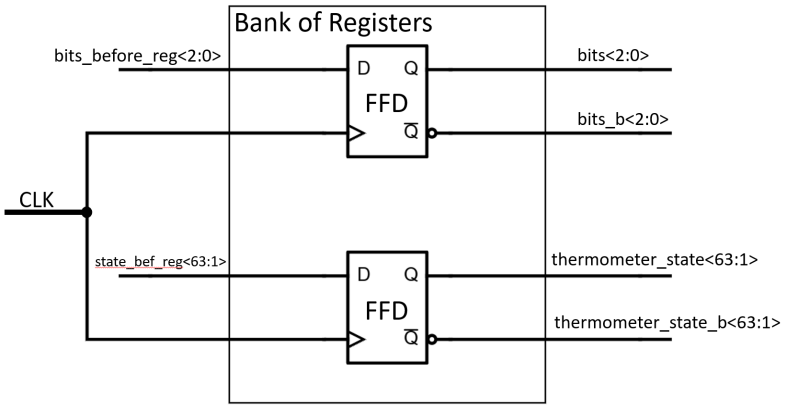

Figure 51 – 66-bit register block showing its input and output connections.

As shown in Figure 51, the 3 input bits for the binary branches and the 63 bits from the decoder for the unary branches are collected and stored, and their inverted versions are also provided at the output. These registers are implemented using simple flip-flop D (FFD) architectures called “DF1”, from AMS foundry PDK “CORELIB” library.

6.2 The chip

The top-level schematic of the full chip with 50 drivers is shown below.

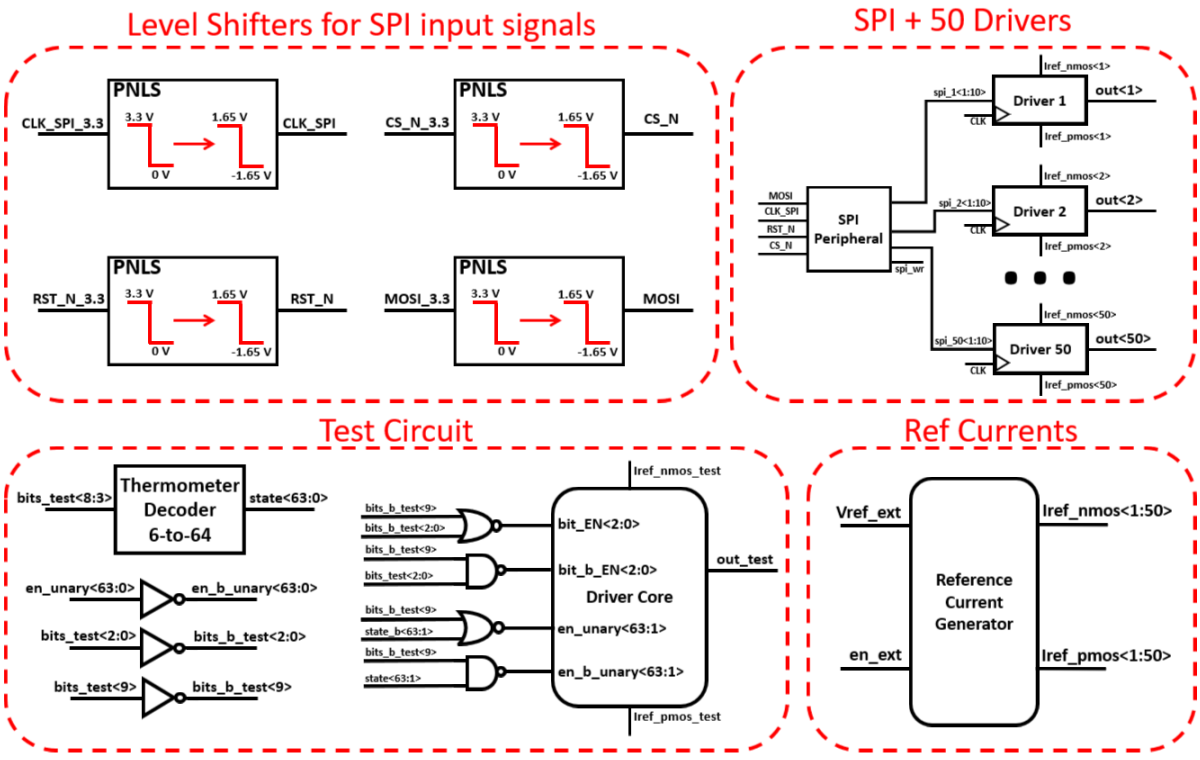

Figure 52 – Top-level schematic of the proposed chip with 50 drivers.

Its composition is explained in the following subsections.

6.2.1 Test Circuit

For lab testing, the following driver unit from Figure 53 was used.

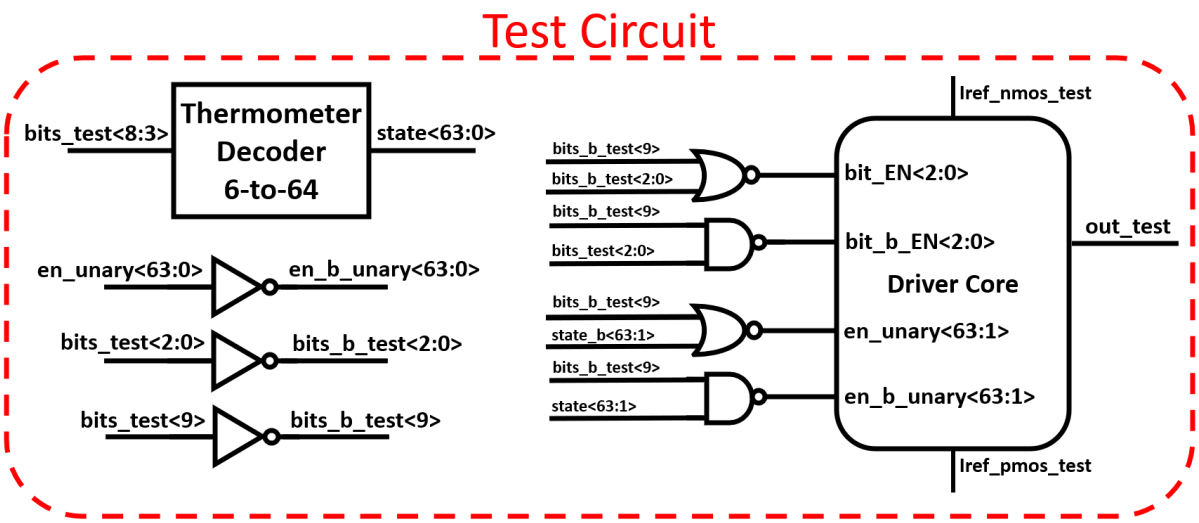

Figure 53 – Block diagram of the test circuits from the proposed chip with 50 drivers.

This test driver doesn't depend on the SPI unit and doesn't have a bank of registers for its signals, it is directly controlled by the "bits_test" signals. Therefore, it is proper for DC operation testing/calibration.

6.2.2 Level Shifters

Since the SPI input signals MOSI, CS_N, RST_N, and CLK_SPI operate at 0 / 3.3 V levels provided by the FPGA, it was necessary to implement four positive-to-negative level shifters (PNLS) to convert these signals to the required −1.65 V/1.65 V logic levels. These circuits are shown in the figure below.

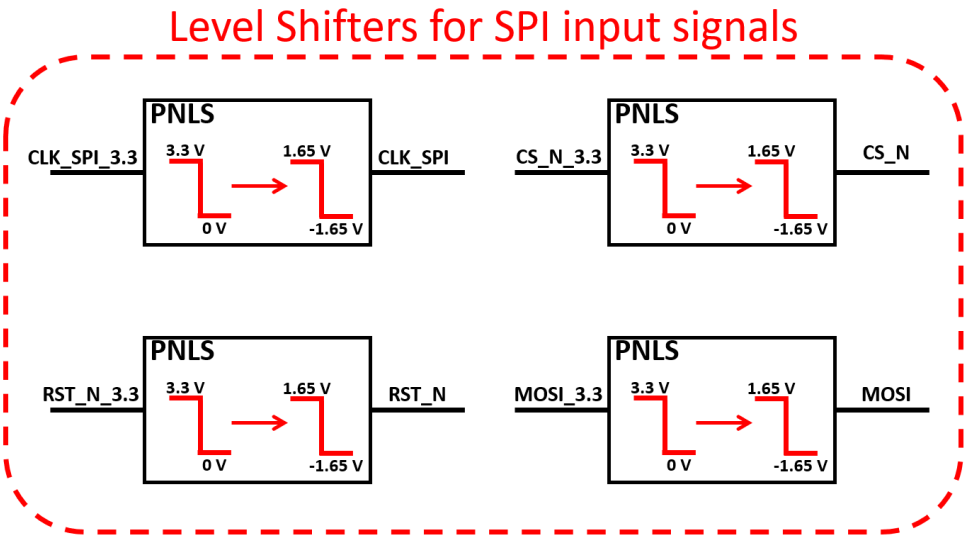

Figure 54 – Block diagram showing the implemented level shifters of the proposed chip.

The structure of each level shifter consists of a simple resistive voltage divider, as illustrated Figure 55.

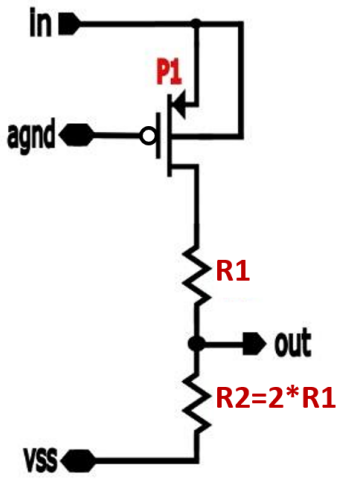

Figure 55 – Schematic circuits of the PNLS implemented in the proposed chip.

6.2.3 Reference Current Generator

The circuit responsible for generating the internal Iref current that biases all drivers is shown in the figure below.

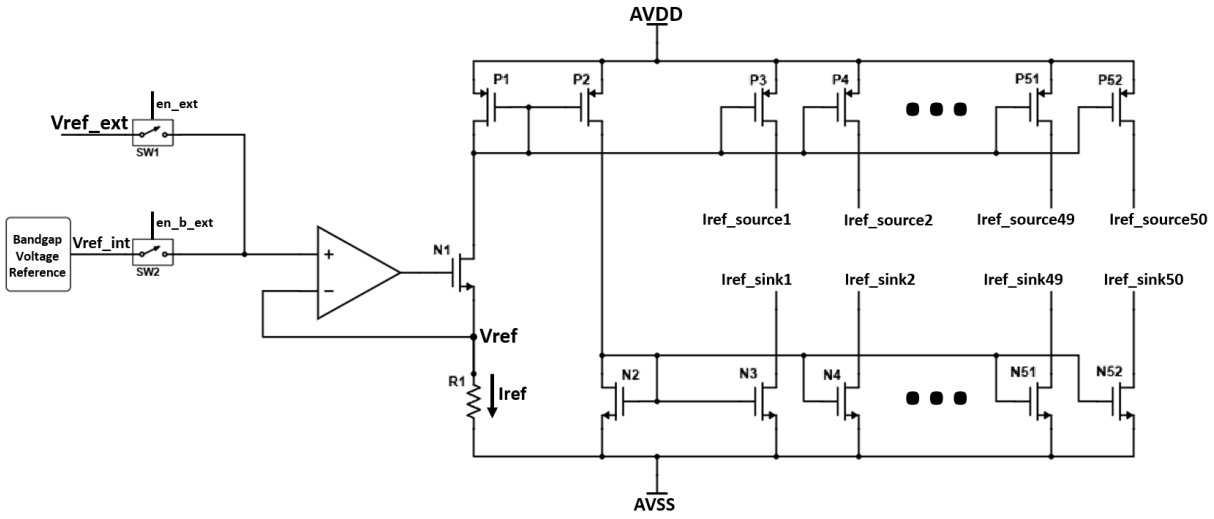

Figure 56 – Schematic circuit of the reference current generator.

This block employs a bandgap voltage reference, which produces a stable Vref value [29]. This voltage is buffered by an operational amplifier and applied across resistor R1, thereby generating the reference current Iref. The resulting current is subsequently mirrored to all source and sink branches of the driver array.

An additional input signal, en_ext, allows selecting an external reference voltage. When asserted, Vref is sourced from an external pin, providing a convenient mechanism for calibration or trimming during testing and characterization.

6.2.4 SPI + Drivers

The SPI+Drivers arrangement is shown in Figure 57.

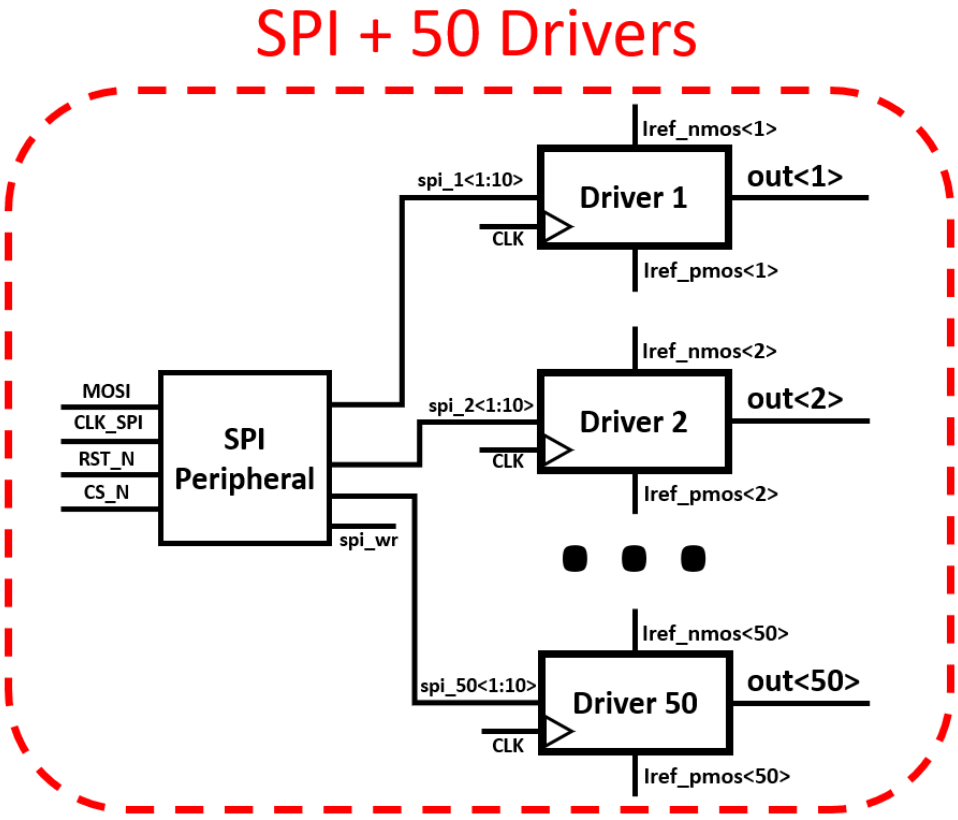

Figure 57 – Block diagram of the connections between the developed SPI unit and the 50 drivers of the proposed chip.

It can be observed that the driver array has 500 input bits, which is consistent since the chip has 50 drivers with 10 input bits on each. The chip layout is still under development. After it is done, EMIR analysis will be performed on the top level to ensure correct functioning.

6.3 Layout

The layout of the newly developed driver unit is presented in Fig. 58.

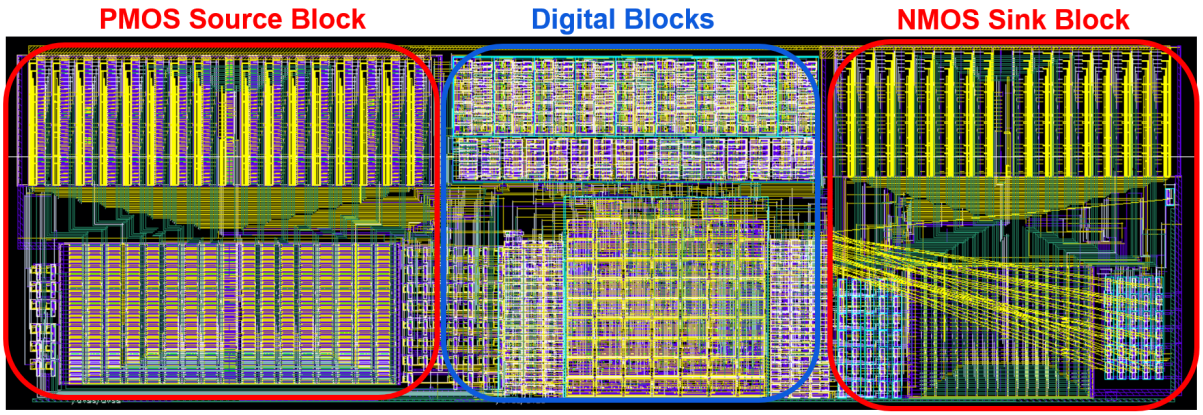

Figure 58 – Layout of the proposed driver.

The floorplan highlights the separation between the digital control circuitry and the analog current-driver stages. In this implementation, the digital blocks account for approximately 30 % of the total driver area. The complete driver occupies 0.28 mm², representing a substantial reduction compared to the first version (from chip V1) while still accommodating all required functional blocks.

6.3.1 Full chip layout area estimation

Although the layout development of the chip containing 50 drivers is currently still in progress, an area estimation was performed to verify that the driver design was carefully optimized to be able to accommodate, in the worst case scenario, at least 80% of the target number of drivers (40 drivers) within a 20 mm² chip. Considering that the driver unit represents the most area-demanding block due to the large number of instances integrated in the chip, this estimation was essential to ensure an appropriate tradeoff between linearity and the predicted area of each driver. The components considered in this area estimation were divided into 4 categories: PADs, the SPI unit, drivers, and other auxiliary circuits such as the level shifters and the current mirror transistors for the diode-connected transistors. The area analysis for these 4 categories is detailed in the following subsections.

6.3.1.1 PADs area estimation

For this component, it is important to consider not only the occupied area of a single PAD, but also the space between the PADs and on the corners of the chip. All this space is not part of the “useful chip area”, where the actual circuits can be placed and routed. Therefore, a good way to analyze it is by measuring the height of the PAD and multiplying that height value with the side size of the chip. For this PDK, the PAD’s

height is about 0.35 mm. Considering the 20 mm² as a rectangle with sides of size 4 mm and 5 mm, the following image illustrates the area occupancy of the PADs.

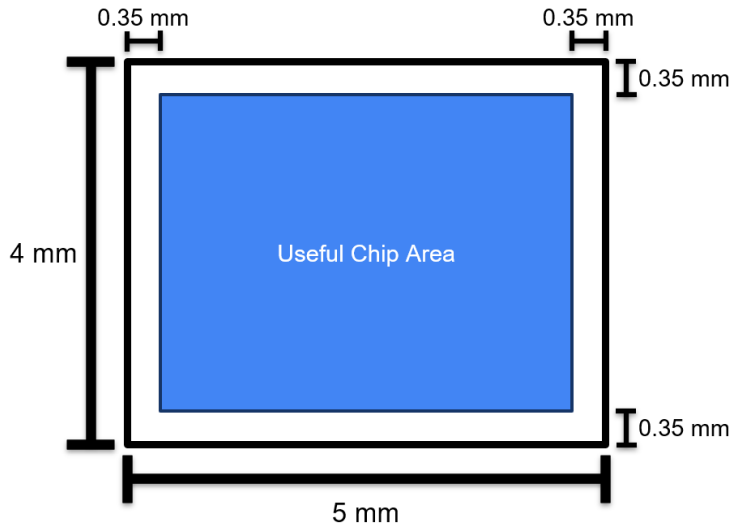

Figure 59 – Illustration of the proposed chip's usable area and the space occupied by the PAD ring.

Considering the arrangement shown in the figure, the useful chip area can be calculated as:

$$\text{Useful Chip Area} = (4 - 2 \times 0.35) \times (5 - 2 \times 0.35)\ \text{mm}^2 = 14.19\ \text{mm}^2 \tag{6.3}$$

Therefore, the area occupied by the PADs and their unused space in the surroundings is about 5.81 mm².

6.3.1.2 SPI unit area

For this one, no estimation is needed since the SPI unit was already fully created. Its area value is 0.543 mm², which can be rounded up to 0.6 mm².

6.3.1.3 Drivers area estimation

The area occupied by a single driver is 0.28 mm². Considering the target architecture of 50 drivers, the total estimated silicon area dedicated solely to the driver array amounts to approximately 14.0 mm².

6.3.1.4 Auxiliary circuits area estimation

In order to estimate the total area used by these auxiliary circuits, a tool was used in Virtuoso Layout which generates a rectangular PR Boundary with an area corresponding to the full area occupied by the instances, without any routing. And then it was considered

that the routing area would be equivalent to about 50% of the total area of the design, which it is believed to be a conservative expectation. Using this approach, the area estimated for all the level shifters and all the current mirror transistors for the diode-connected transistors is 0.2 mm^2.

Table 5 shows the predicted area for each component.

Table 5 – Estimated Silicon Area Breakdown for the Proposed Chip

Components	**Area (mm^2)**
120 PADs and surroundings	5.81
50 drivers	14.0
1 SPI unit	0.6
Other components	0.2
Testability	0.4
Total	**21.0**

Therefore, this area estimation predicts that all components could be individually placed and routed using 21 mm^2, which exceeds the 20 mm^2 available even without considering the top level connections. In this case, a reduced number of drivers need to be used in the chip, unless the driver layout is improved to occupy less area.

A preliminary view of the chip Floorplan is shown in Figure 60. With the finalized driver-unit layout, it was possible to place 40 drivers on the chip, in addition to a dedicated test driver. The array is organized into four columns: three columns contain 11 drivers each, and the third column contains 7, totaling 40 units.

The large block on the third column corresponds to the new SPI interface, and the test driver is placed directly above it. This test structure replicates the analog core of the main driver but omits the register bank, enabling direct DC evaluation by applying input codes to the circuit. The remaining central area was used to place auxiliary blocks, such as the level shifters, output buffers, and the reference generator, some of which are too small to be clearly visible in the figure.

For power distribution, the most critical rails are AVDD and AVSS, as they carry the combined current of all drivers. These supplies will be routed through the spaces between columns. The Iout signals also require careful routing, although less critical since each driver has its own output current net.

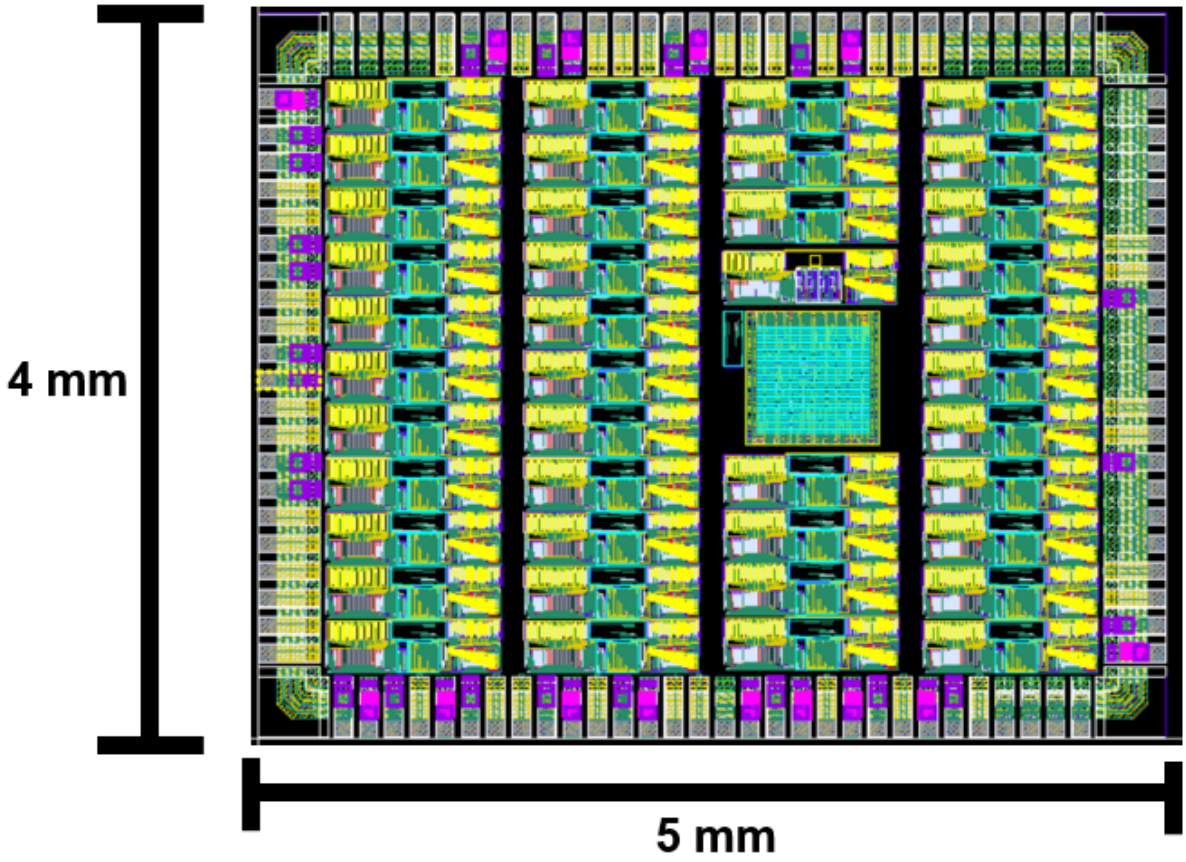

Figure 60 – Floorplan of the chip, with 40 drivers.

7 Simulations

The simulations performed to evaluate the driver's performance, as well as the chip's functioning with multiple driver units, are explained in this chapter. All simulations were performed using Cadence Virtuoso Analog Design Environment (ADE), Cadence Spectre (for analog-only simulations) and Cadence Xcelium (for mixed-signal simulations).

7.1 Optimizations

In order to find the best solution through the design space that could achieve good INL and DNL values while having compact area and being able to maintain all transistors in the saturation region, in which they would be more robust for corner variations, an optimization process was implemented. Using the Advanced Optimization (AOP) method in ADE, which employs a Machine Learning algorithm called "Neuron" to search through the design space, and by defining expressions that can evaluate all important metrics from the design, it is possible to achieve good results, which will be described in the next sections.

For this to be done, expressions for DNL and INL calculation, transistors' area occupancy and transistors' saturation were defined in ADE, with specifications about their acceptable values, so that the optimization algorithm could act on meeting all the defined requirements. Specifically the area expression was the hardest to conceive, requiring thorough analysis of the area occupied by the PMOS source and NMOS sink blocks when having different numbers of gates defined, based mainly on these parameters: length, width, number of gates, drain area, source area and additional fixed spacing defined.

7.2 Driver DC Simulations

A full DC sweep simulation with all 1024 possible input bit values (from 0 to 1023) was performed, in order to assess INL and DNL metrics for this driver, not only for nominal approach, but also for different corners. The results are presented in the next subsections.

7.2.1 INL and DNL performance

One of the main concerns of this project was to achieve acceptable INL and DNL values while keeping reasonable area occupancy. The table below shows the reported worst INL and DNL values from a full DC sweep under nominal conditions.

Table 6 – Worst INL and DNL Metrics from Nominal DC Simulations

Metric	**Value [LSB]**
Worst INL	0.22
Worst INL from PMOS	0.15
Worst INL from NMOS	0.19
Worst DNL from PMOS	0.003

It can be observed that, for the nominal operation, very good results were achieved for both INL and DNL values, both easily reaching the desired accuracy goal of 1% (would be 200 μA, considering the full range of 20 mA, and 1 LSB is only about 19.6 μA).

7.2.2 PVT Corner analyses

This work was validated in a vast list of PVT (Process, Voltage and Temperature) corner combinations, which are listed here:

- Temperature values of 0 ºC, 27 ºC and 100 ºC.
- Process corners TM, WP, WZ, WO and WS.
- Supply voltage variations of ±5%.

Since we have 3 temperature corners, 5 process corners and 3 voltage supply values (including nominal of 0%), when running the combination of all corner types we have a total of 45 corners checked. The results of these runs are shown below.

Table 7 – Worst-Case INL, DNL and Output Current Across PVT Corners

Metric	**Value**	**Which Corner**
Worst INL	1.93	Worst-One, 0°C, $\|V_{DD}\|$ - 5%
Worst INL from PMOS	1.34	Worst-Speed, 27°C, $\|V_{DD}\|$ - 5%
Worst INL from NMOS	1.00	Worst-Speed, 100°C, $\|V_{DD}\|$ - 5%
Worst DNL	0.02	Worst-One, 0°C, $\|V_{DD}\|$ + 5%
Max Current (mA)	10.1	Worst-Power, 100°C, $\|V_{DD}\|$ + 5%
Min Current (mA)	-10.1	Worst-Power, 100°C, $\|V_{DD}\|$ + 5%

Also worth noting that, when setting the circuit to have the maximum nominal current, the delta current among the corners is only 0.10 mA for the PMOS side and 0.06 mA for NMOS side. Therefore, there is a worst case scenario of 1% of current variation. It is important to note that an ideal current source was used for this test, considering that the current that feeds the current mirror is an input pin for this driver design.

7.2.3 MC analysis

A Monte Carlo (MC) run is a type of simulation that aims to analyze how statistical variations in device parameters can affect the circuit's behavior, being a good method to analyze circuit robustness. Monte Carlo runs apply statistical distributions on both process and mismatch parameters. "Mismatch" refers to the local random variations which can happen between "identical" devices, such as the many current mirrors employed in this work. Since process corner variations were already assessed in the previous section, the mismatch is the more critical effect that needs to be assessed with the MC run.

Therefore, in order to check for the device's mismatch influence over the output results, and also to account for the combination of these mismatch issues and the process variations, a 1,000 points Monte Carlo analysis was performed for the driver unit. Results from this run are summarized in table below.

Table 8 – Monte Carlo simulation results based on 1,000 points.

Name	**Min**	**Max**	**Mean**	**Std Dev**
Maximum Power (mW)	17.12	18.29	17.64	0.16
Lowest Current (mA)	-10.46	-9.59	-10.00	0.13
Maximum Current (mA)	9.68	10.33	10.00	0.10
Worst DNL (LSBs)	0.07	**0.60**	**0.27**	0.08
Worst INL (LSBs)	0.42	6.54	2.08	1.21
Worst INL from NMOS block (LSBs)	0.27	**1.18**	**0.57**	0.15
Worst INL from PMOS block (LSBs)	0.19	1.17	0.46	0.13

Regarding differential non-linearity (DNL), the results were encouraging: the mean DNL across all samples was only 0.27 LSB, and the worst-case value observed was 0.60 LSB. These values indicate that the converter maintains monotonic behavior with comfortable margin and shows limited susceptibility to mismatch-induced step-size distortion.

For the integral non-linearity (INL), it is common in the literature to separate the analysis of the positive and negative output current regions. This is because device mismatch affects the NMOS and PMOS current blocks differently, leading to distinct non-linearity profiles for sourcing and sinking currents. Following this methodology, the analysis shows that the NMOS-driven region exhibited a mean INL magnitude of 0.57 LSB and a maximum deviation of 1.18 LSB across the 1,000 Monte Carlo trials. These results demonstrate that, even under combined process and mismatch variations, the overall static accuracy remains within acceptable bounds for the intended application. Histograms from figures below show the distribution of DNL and INL values from the different points simulated.

Figure 61 presents the histograms of the INL values obtained separately for the PMOS and NMOS current blocks. Both distributions exhibit very similar statistical behavior, indicating that mismatch affects the sourcing and sinking paths in a comparable

manner. The vast majority of samples fall below approximately 0.75 LSB, demonstrating that most realizations maintain low non-linearity. Only a small number of Monte Carlo points exhibit INL values exceeding 1 LSB, which reinforces that large deviations are rare under combined process and mismatch variations.

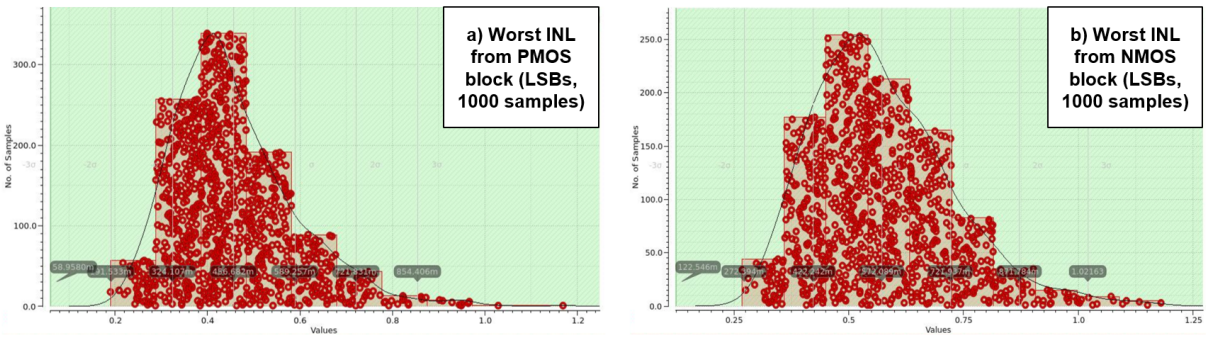

Figure 61 – Histograms showing the variation of the worst-case INL from a) PMOS blocks and b) NMOS blocks across 1,000 Monte Carlo simulation points. The horizontal axis is expressed in LSB.

7.3 Driver transient simulations

In order to assess the transient performance of the designed driver, a transient simulation was performed sweeping all the 1024 possible bit code values. Mainly the glitches during clock value changes and during transistor switching were observed.

7.3.1 Level Shifter test

The 4 implemented level shifters that aimed to convert the voltage levels of 0/3.3 V coming from the FPGA into -1.65/1.65 V were tested for the necessary high frequency operation of 2.5 MHz. All 4 behaved as expected. Figure 62 shows the results from conversion for the SPI clock signal.

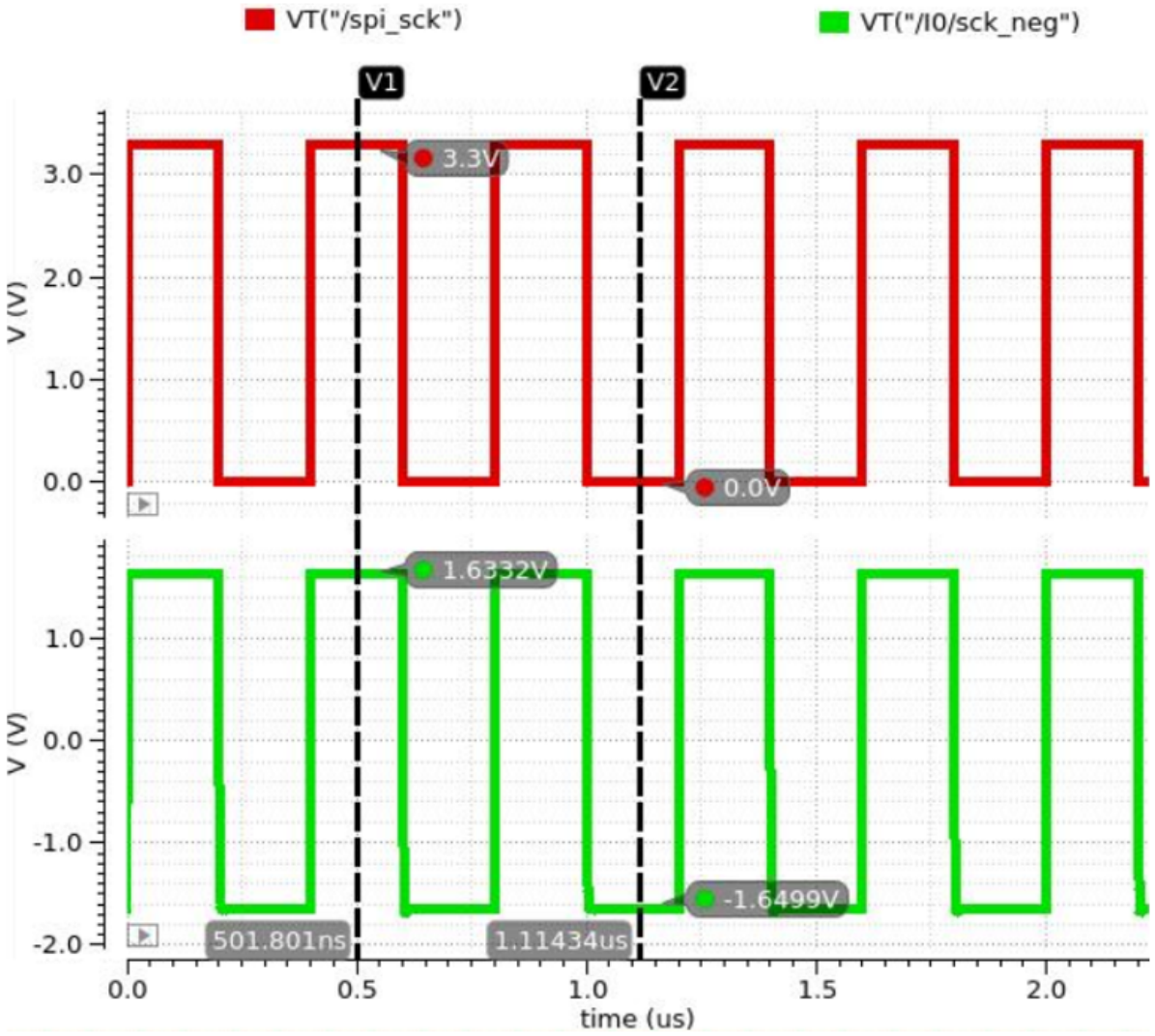

Figure 62 – SPI clock signal before and after the PNLS.

Signal "/spi_sck" corresponds to the clock signal received from the FPGA, while "/I0/sck_neg" corresponds to the converted signal to proper voltage levels. We can see that it worked fine for the proposed frequency of operation.

7.3.2 Single Driver Transient Test

A transient simulation was performed to check the time taken for the output signal settling, the SFDR value and the glitches during unary and binary transitions.

A counter circuit was used to provide the full sweep of digital input values to the driver. This counter was set to count in reverse order, starting from 1111111111, until it reaches 0000000000. The output current result from this test is shown in Figure 63 below.

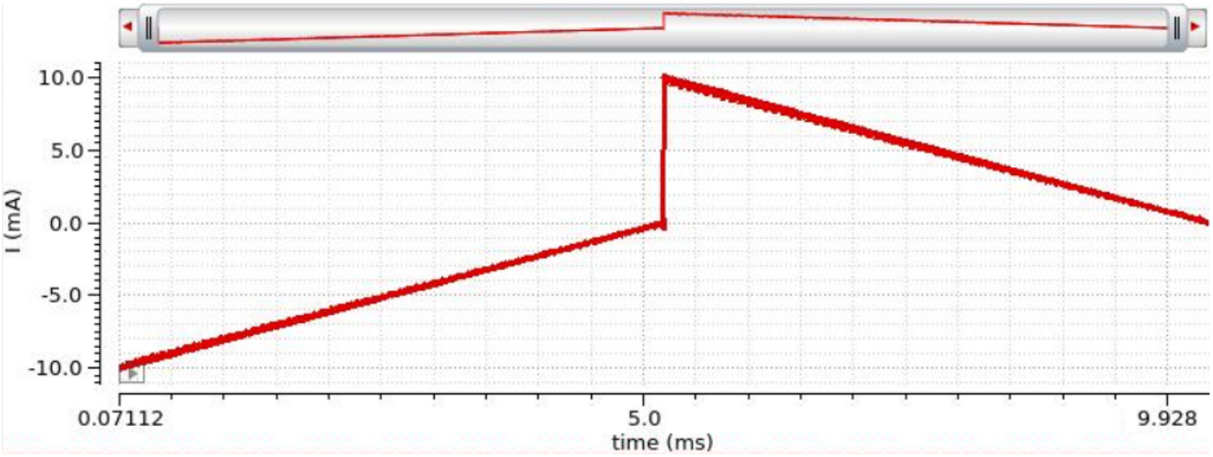

Figure 63 – Output current from transient simulation of the proposed driver with a full input sweep from 0 to 1023 in descending order.

Since this simulation goes from 1111111111, this corresponds to the maximum negative current of -10 mA. Then, the current magnitude reduces until it reaches 1000000000, which corresponds to 0 mA. The next code is 0111111111, which corresponds to the maximum value of 10 mA, and this magnitude is reduced until code 0000000000, which also corresponds to 0 mA. Glitches can be observed better when zooming into the output current waveform, as shown in Figure 64.

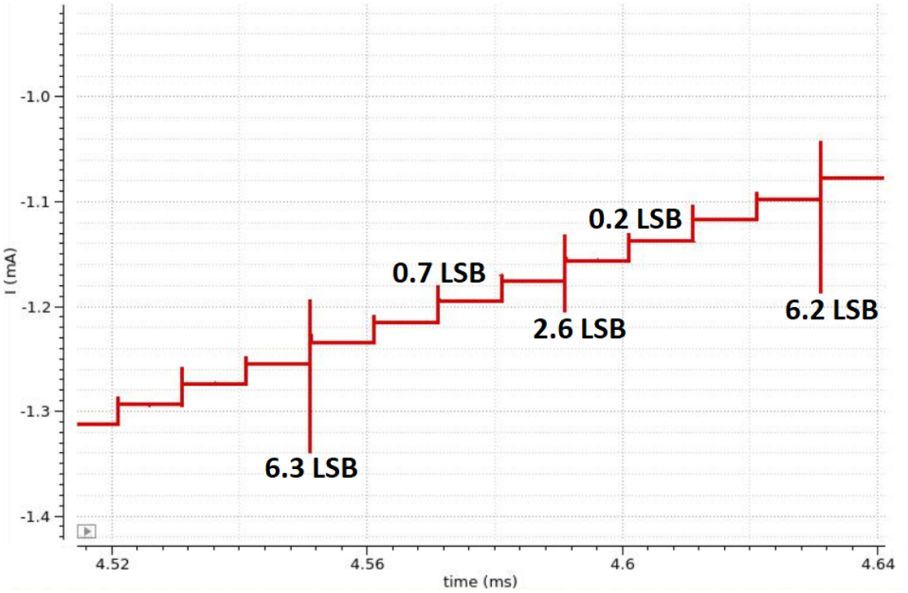

Figure 64 – Zoomed-in view of the transient output current highlighting the binary and unary glitches, along with their respective glitch size values.

The figure shows examples of measured glitch sizes for different code transitions. The glitch size here is defined by the following equation:

$$\text{Glitch Size} = \frac{\text{Max}_{\text{Current glitch}} - \text{Min}_{\text{Current glitch}}}{\text{Ideal LSB}} - 1 \tag{7.1}$$

Therefore, the glitch size measurement represents any value above the expected ideal 1 LSB step during the signal transition.

An explicit pattern can be observed from the presented results: we have regions with small glitches (less than 1 LSB) and regions with high glitches (> 1 LSB). In this figure, the 2 regions with glitch sizes above 6 LSBs are unary transitions, in which there is a higher change in the cascode voltages, since an unary current cell branch is being added to the output current and 3 binary branches are being deactivated at the same time, causing higher current transition. The 2 regions with glitch sizes below 1 LSB represent binary transitions of bit<0> and bit<1>, which cause less change and therefore have less impact during transition. The other transition, which has a glitch size of 2.6 LSB, represents the binary branch of bit<2> being turned on while bits 0 and 1 are being turned off at the same time, therefore causing higher glitch value.

Figure below shows the zoomed-in region of worst case mentioned before, in which the glitch size was 6.3 LSB.

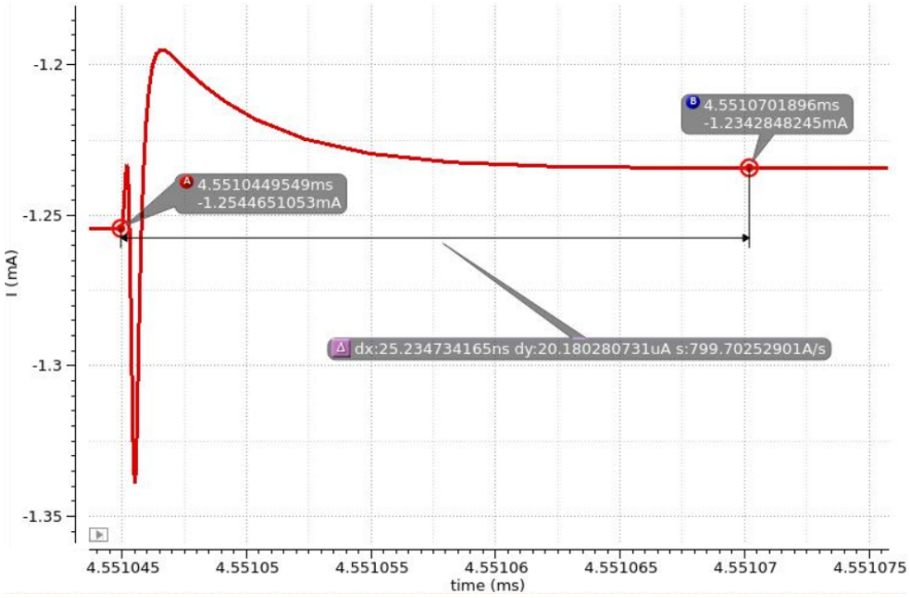

Figure 65 – Figure: Zoomed-in transient waveform of the output current during a unary bit transition, illustrating the worst-case driver's settling time.

From that image, we can observe that the time for the current signal to stabilize after it started the transition was about 25 ns. This is a considerable amount of time which can be relevant if dealing with high speed applications, since they need to do fast switching. A 25 ns time translates to a frequency value of 40 MHz, so any frequency above that would not be applicable for this design. However, this was expected when the

topology of using a switch in the cascode to control the activation of the binary and unary branches was applied. Since the application for the proposed driver is low frequency (about 2 kHz, which corresponds to a period of 500 μs), this settling time is quite satisfactory. It represents only 0.005% of the total period. Moreover, due to the inherent mechanical characteristics of MEMS actuators and their inherently slow response time, these glitches are not considered problematic.

The other important transient metric is the Spurious-Free Dynamic Range (SFDR), as discussed in Chapter 2. To measure this value properly, a Verilog-A script was developed to generate a 10-bit binary sinusoidal signal, which was then used as input stimulus for the driver. The resulting driver output current is presented in Figure 66.

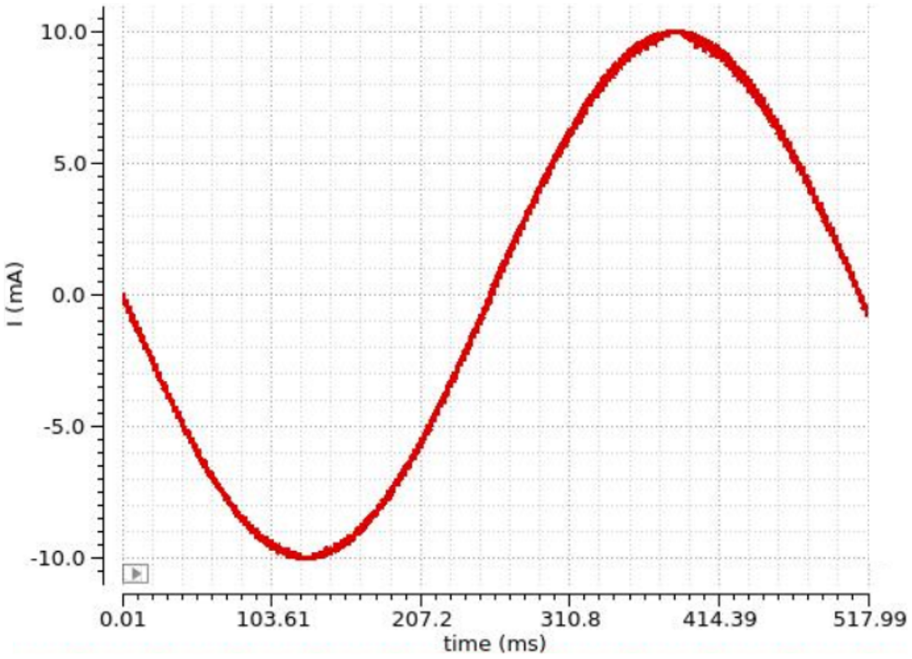

Figure 66 – Output current waveform of the proposed driver under sinusoidal excitation for SFDR analysis.

Given that the operational frequency for each bit update is 2 kHz, an appropriate sine-wave frequency can be obtained by taking this value and dividing it by 1024, resulting in a desired frequency of 1.95 Hz. To have a more comprehensive analysis of the output current behavior for the frequency analysis, reducing spectral leakage and having cleaner spectrum, a total of 49 cycles of the same sinusoidal signal was simulated. The corresponding waveform is shown in figure below.

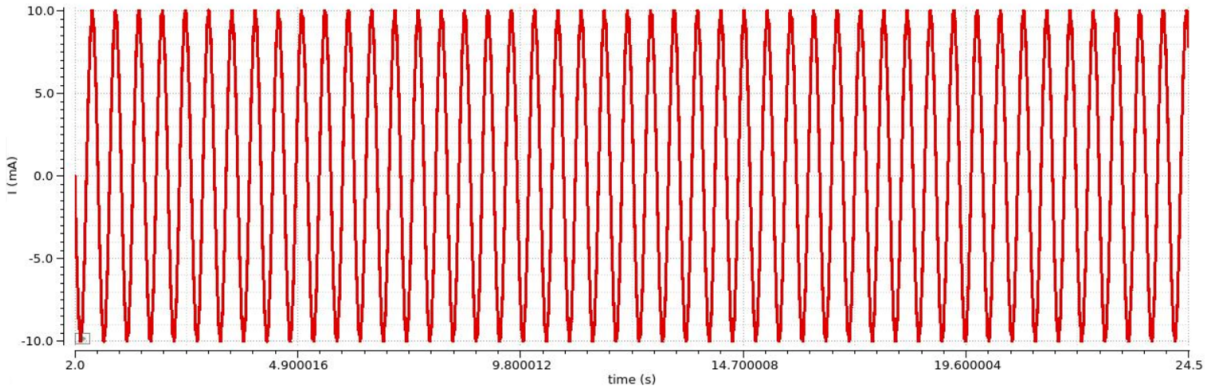

Figure 67 – 49-cycle output current waveform of the proposed driver under sinusoidal excitation used for SFDR analysis.

A Discrete Fourier Transform (DFT) was then applied to this waveform to obtain its frequency-domain representation, which can be seen in Figure 68. The DFT expression used had specifications of: 49 full cycles of 1.95 Hz each, rectangular window type, ADC span value of 10 mA (sine wave amplitude), 16384 total samples.

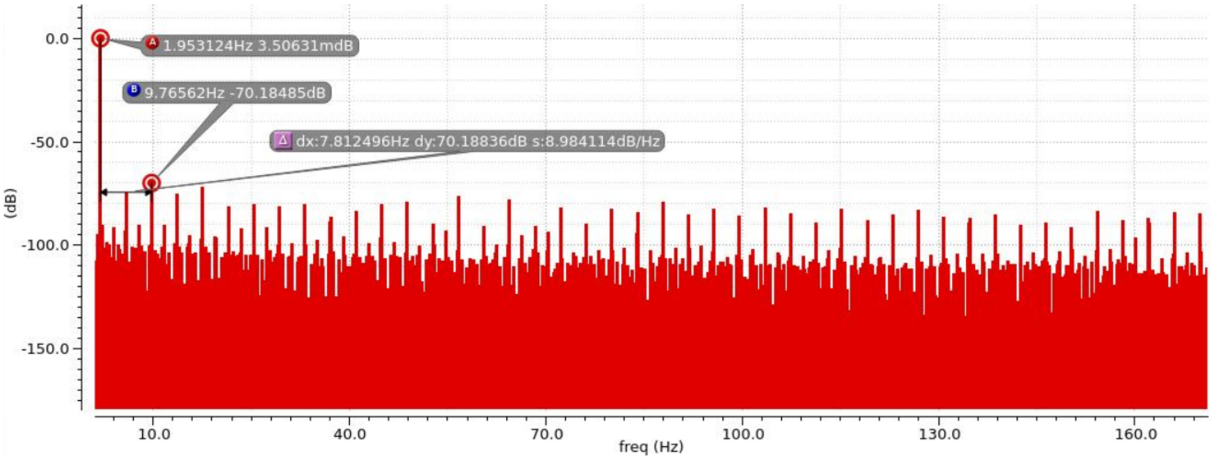

Figure 68 – Discrete Fourier Transform (DFT) of the 49-cycle output current waveform of the proposed driver used for SFDR measurement.

As expected, the dominant frequency component corresponds to the sinusoidal input frequency of 1.95 Hz, with a magnitude close to 0 dB. The second highest frequency bin appears to occur at 9.76 Hz, with a magnitude of -70.18 dB. From this analysis, the calculated SFDR for the designed driver is 70.18 dB. However, it must be noted that this value corresponds to the simulated pre-layout nominal condition, and a lower value is expected during experimental measurements due to the parasitic components, mismatch, noise and other non-ideal effects.

7.3.3 Full Chip Transient Test: SPI + Drivers

Considering that the full chip transient simulations take a long time, in order to test the proper functioning of the full chip with the SPI controlling the drivers, it was decided to do a full sweep testing in 3 of the 50 drivers implemented, using the SPI to write the digital codes for these 3 units. Therefore, in this testbench, a series of 'mosi' signals coming from different bit sources had to be used, associated in series by using OR logic gates, in order to make sure the SPI unit received all the desired input signals for all 3 drivers, to make a full sweep of the digital input codes (from 0 to 1023). The SPI clock was tested with the intended nominal frequency value of 2.5 MHz, which corresponds to a period value of 400 ns.

The resulting current signals from this test are reported in figures below.

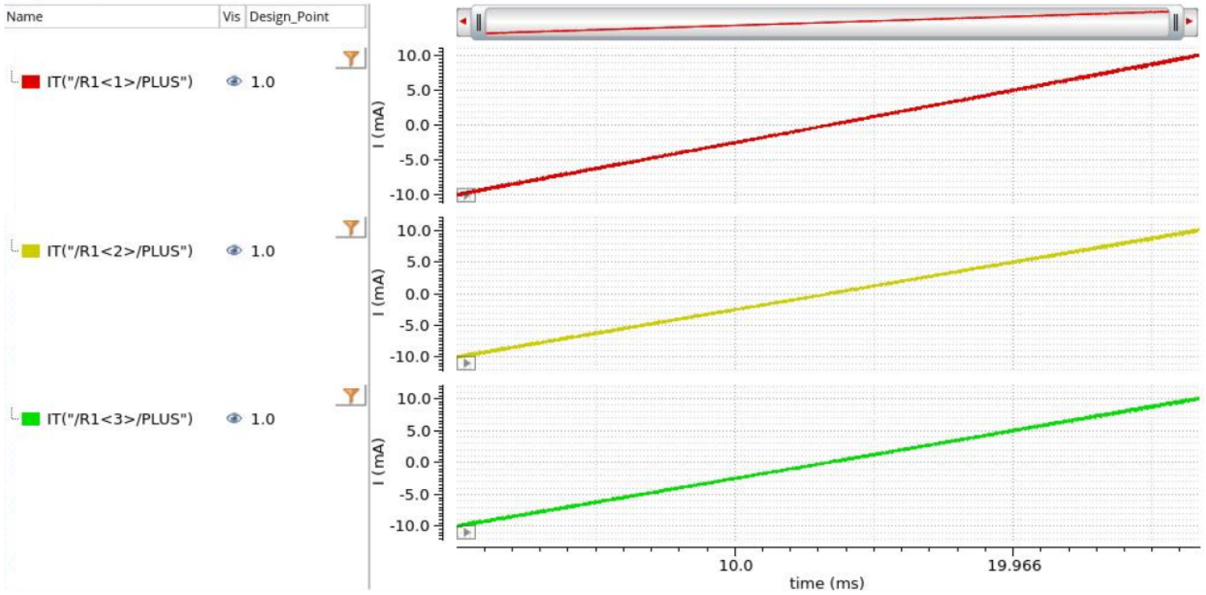

Figure 69 – Figure: Transient waveforms of the output current from three drivers after a full input sweep applied in specific order that results in ascending output currents from -10 mA to 10 mA.

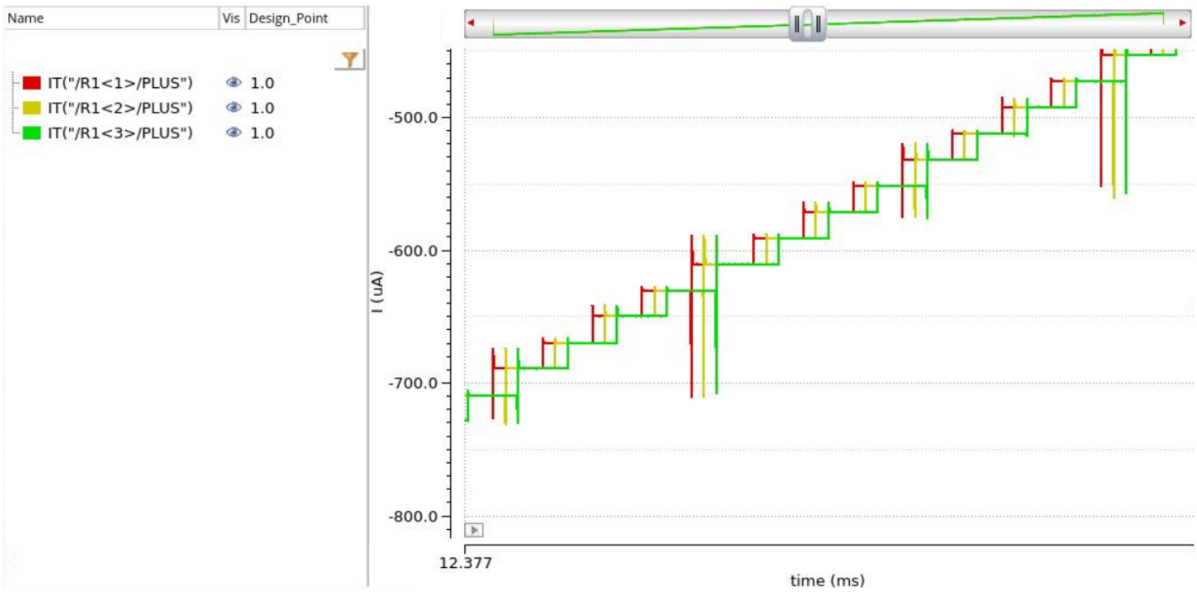

Figure 70 – Zoomed-in transient waveforms of the output current from three drivers, illustrating their transition behavior.

Figures show that a full current sweep, from -10 mA to 10 mA, was successfully done for all 3 drivers, with their current matching well and their codes changing with a 6.4 μs of difference, which corresponds to 16 times the SPI clock period (400 ns). This is expected, since the SPI takes 16 bits at a time, and writes the 10 LSBs bits on its output for each driver. Therefore, the full chip was working as expected.

7.4 Power consumption analysis

The total power consumption from the driver unit was also checked from simulation. From DC simulations, therefore without considering the power used by the registers, the highest power consumption happens for nominal output currents of -10 mA and 10 mA. Both show the same power value of 17.5 mW.

The SPI unit had a DC power consumption of 5.1 mW. Therefore, considering the worst-case scenario in which all the 50 drivers are operating with highest nominal current, a total of 880 mW of power consumption can be estimated.

7.5 EMIR Analyses

After the new driver layout was finished, it was important to run Electromigration and IR Drop (EMIR) Analysis in the design to make sure the currents provided, specially the ones from supply rails, were not causing any issues in the wires.

In order to assess the main scenarios, these simulations were performed:

- DC EMIR run with nominal output of 10 mA for the driver.
- DC EMIR run with nominal output of -10 mA for the driver.
- Transient EMIR run with full current sweep.

During the DC electromagnetic (EM) analysis of the layout, two relevant violations were identified in the metal routes supplying the AVDD and AVSS guard rings, shown in figures 71 and 72. This observation reinforced an important design principle: although one might assume that certain narrow or auxiliary connections would carry negligible current, the actual current distribution can differ significantly from intuition. The EM simulation revealed that these paths were indeed subject to higher current densities than expected, highlighting the necessity of performing EM verification rather than relying only on manual reasoning.

Additionally, the transient EM analysis showed that the DVDD and DVSS networks experienced relatively high peak currents during major code transitions, as shown in Figure 73. These effects were not captured in the DC EM analysis, which, by nature, does not account for dynamic switching activity. The transient-domain results therefore proved essential for identifying these short-duration current surges.

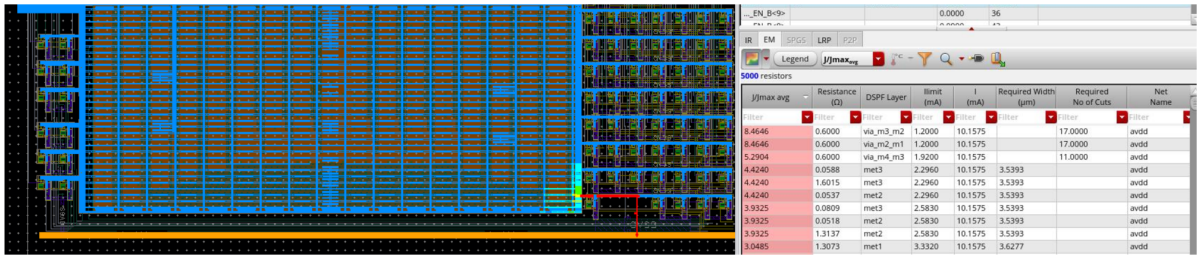

Figure 71 – Initial DC EM results from proposed driver, with nominal output current of 10 mA.

Figure 72 – Initial DC EM results from proposed driver, with nominal output current of -10 mA.

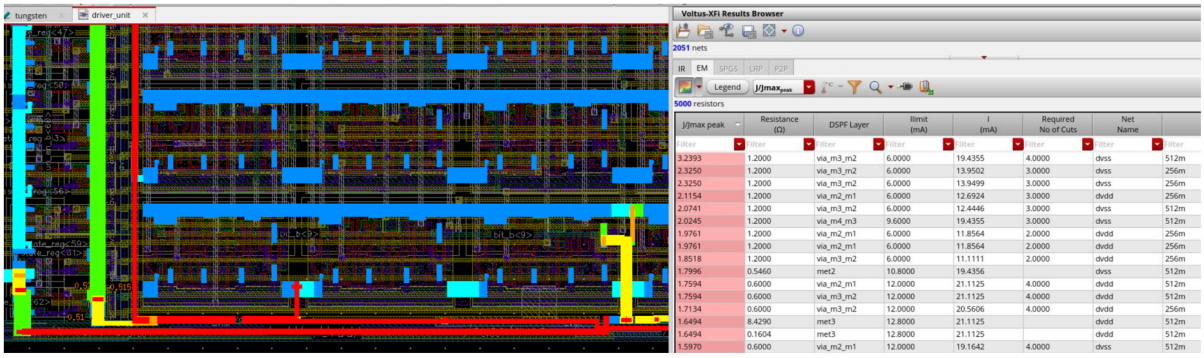

Figure 73 – Initial Transient EM results from proposed driver, with full output current sweep.

All EM issues were corrected by increasing metal widths and adding additional via arrays in the affected regions. After these adjustments, the updated EM simulations demonstrated that the current densities were safely within the allowable limits, as shown in figures 74 and 75.

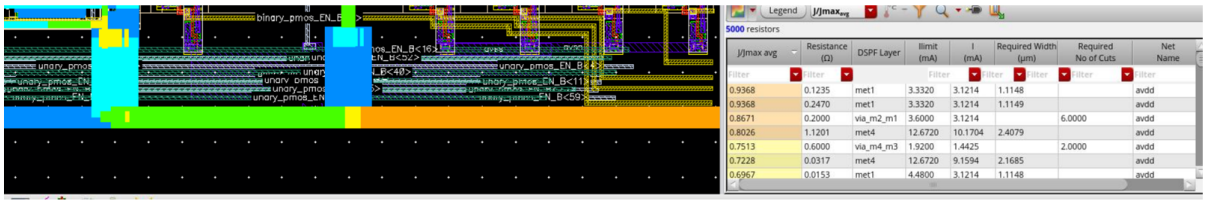

Figure 74 – Final DC EM results from proposed driver, with nominal output current of 10 mA.

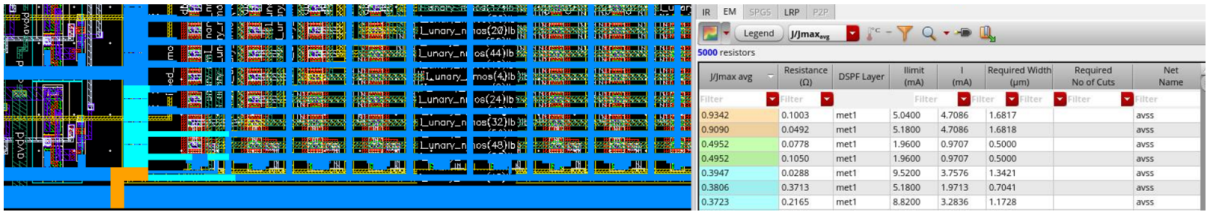

Figure 75 – Final DC EM results from proposed driver, with nominal output current of -10 mA.

Regarding the IR-drop analysis, the initial results were already satisfactory and required no further layout modifications. In the DC domain—where the register network is not included—the peak IR drop was only about 20 mV, as shown in Figure 76. For the transient analysis, higher values were observed, reaching approximately 450 mV, as seen in

Figure 77. However, these last occurred exclusively on digital nets during switching events, which is expected due to instantaneous current peaks. Since these variations do not impact the critical analog nodes, they do not represent a functional concern for the system.

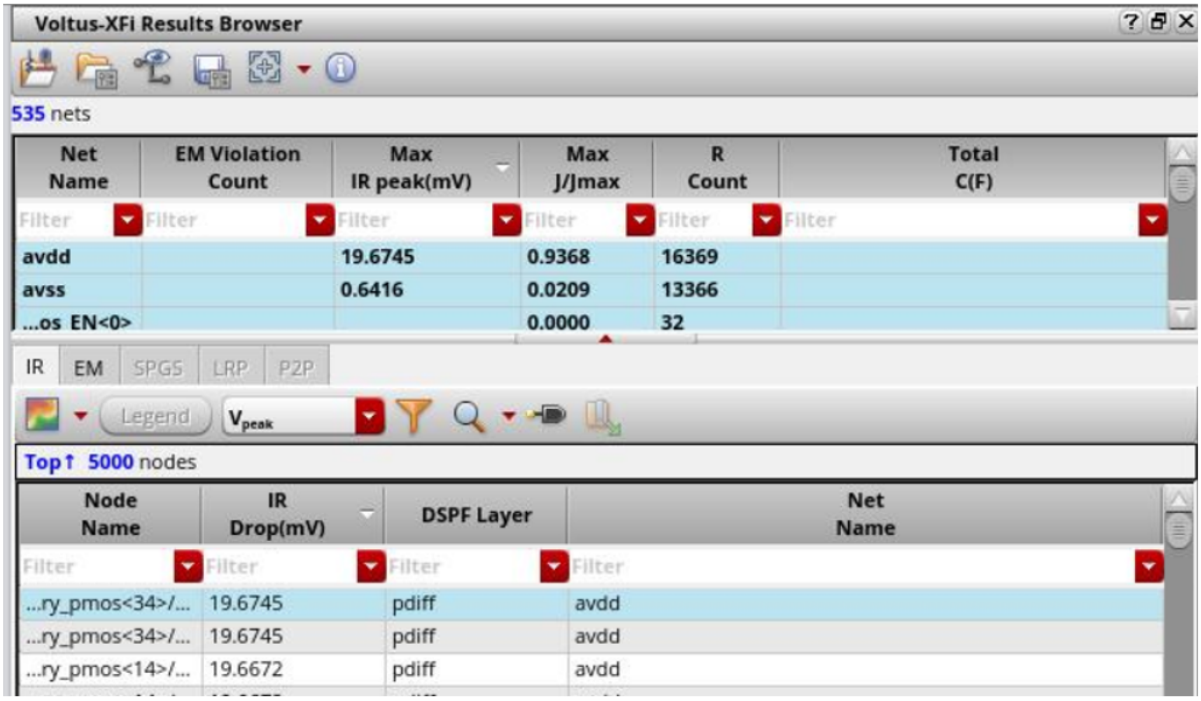

Figure 76 – DC IR Drop results from proposed driver, with nominal output current of 10 mA.

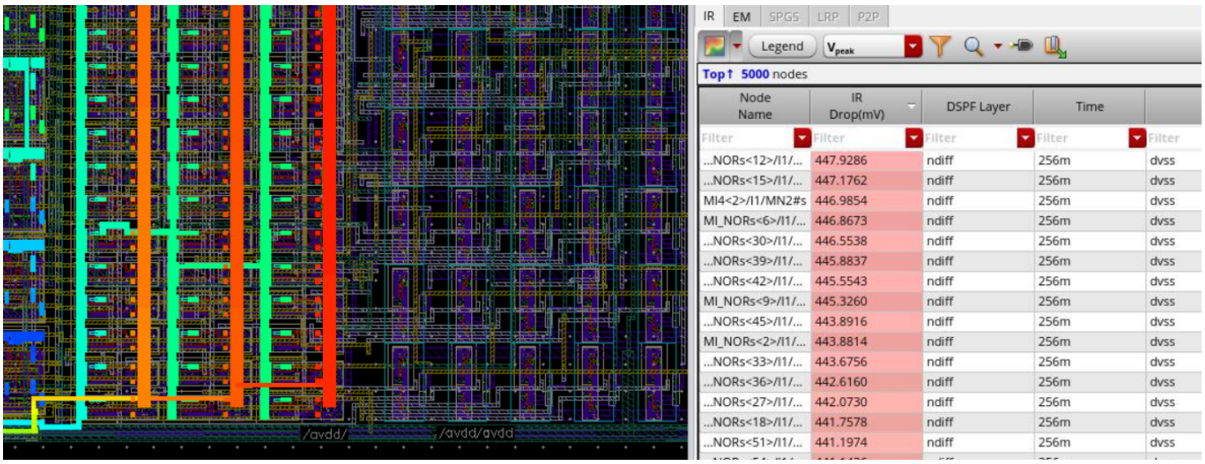

Figure 77 – Transient IR Drop results from proposed driver, with full output current sweep.

7.6 Comparison Table

To better assess how the proposed work compares with other high-current driver implementations, a comparison table was created and is presented below.

Table 9 – Performance summary and comparison with other high-current driver implementations.

Metric	**This work***	**[8]**	**[30]**	**[31]**	**[32]**	**[2]**	**[1]**
Technology	350 nm	130 nm	90 nm	140 nm	65 nm	350 nm	350 nm
Voltage Supply (V)	−1.65 to 1.65	1.2 / 2.5	1.2 / 2.5	1.8	1.0 / 2.5	3 / 3.6	±15
INL (LSBs)	0.57	4.4	1.2	1.8	2.5	0.25 (Peak measured)	–
DNL (LSBs)	0.27	2.2	0.51	2	1.6	0.1 (Measured)	–
Power per driver (mW)	17.6	226	128	270	82	120	150
Resolution (bits)	10	14	12	14	14	8	8
Current Range	20 mA (Full-scale)	16 mA (Full-scale)	16 mA	20 mA	16 mA	5 mA (Full-scale)	10 mA (Full-scale)
Silicon Area per driver (mm^2)	0.28	1.58	0.825	1.27	1.56	–	0.24

Overall, the proposed design demonstrates favorable results in terms of both linearity and silicon area when compared to the other works summarized in the table. It is important to highlight, however, that the referenced designs report measured INL/DNL performance, whereas the results presented in this dissertation are derived from Monte Carlo (MC) simulations that include process and mismatch variations, since the chip was not fabricated yet.

Among the surveyed implementations, the work in [5] is particularly relevant because it targets the same application—current drivers for MEMS actuators, and is fabricated in the same 350 nm technology node. Unfortunately, no per-driver silicon area is reported in that publication. Only the total test-chip area is available (approximately 12 mm^2), making a direct comparison difficult.

The work in [6], corresponding to the doctoral thesis of Colin Ross, is also closely related to the present application. Although it achieves a slightly smaller per-driver area compared to this work, several important distinctions must be noted. First, its power

consumption is approximately five times higher, likely due to the significantly larger supply voltage range used. Second, the maximum output current is limited to ±5 mA, which is only half of the full-scale current supported by the proposed driver. As higher output currents generally require wider or more numerous transistors, which directly increases area, matching this work's current range would likely push the design in [6] well beyond the reported silicon footprint. Additionally, the thesis does not report measured values of INL or DNL, nor does it provide a quantitative accuracy analysis, which is a key metric for high-performance current drivers. Also, the solution implemented in [6] adopts a voltage-mode actuation approach, which is intrinsically more sensitive to load variations, especially in the context of MEMS devices whose impedance may change substantially depending on operating conditions. In contrast, the current-mode architecture presented in this dissertation inherently provides better robustness to load fluctuations, ensuring more predictable actuation forces and improved overall system performance.

In summary, when considering current range, linearity, power consumption, area efficiency, and robustness to load variations, the proposed design achieves a well-balanced set of performance metrics and compares favorably with the closest related works in the literature.

8 Conclusion and future work

The biggest challenge in the driver design was to balance area and linearity. It a point where, for the same design topology, the driver could achieve an INL below 0.1 and a DNL below 0.01, estimated from nominal simulations, but with an area of approximately 0.5 mm^2, which is very large for a single driver and unfeasible for an array of 50 drivers within a total area of 20 mm^2, which is the target of this project. This process became a valuable learning experience in finding the balance between area efficiency and linearity while maintaining good performance across PVT corners, therefore ensuring robustness. The challenge was even greater due to the high total current range of 20 mA, associated with the requirement of having 2 different current directions, making it necessary to implement both sink and source current paths. Most of the literature on DACs and current drivers don't address both directions simultaneously, and often evaluate the accuracy and the linearity values under simplified conditions.

In the first stage of this master's research, the focus was on doing the layout design of the first chip prototype, allowing subsequent important analyses, such as EMIR, to be conducted giving important insight into the effects of high current densities and power distribution on the reliability of the circuits. During the course of this work, several topologies were examined, compared and studied in the area of current driver design, leading to the identification of a topology that provides a good tradeoff between linearity and area efficiency. Thanks to the optimization process performed using a machine learning algorithm, the optimized design was able to minimize layout area while still meeting high linearity specifications and ensuring saturation of the transistors, maintaining good robustness. The layout of the final version of the proposed chip is currently under development, aiming to fabricate it for future experimentation and validation of its performance. During this master's program, two articles directly related to the work developed here were published: [23] and [27].

Only a few works in literature address DACs that are capable of both sinking as well as sourcing current. The general works are usually concerned with only one of the current directions, when their accuracy and linearity are evaluated. This work is, on the contrary, devoted to the design of a completely bidirectional current driver array with good linearity metrics over the full current range. Consequently, extra work has been put into the design process, but the application range has also increased. The final chosen topology combined the segmented DAC architecture with the cascode switching technique applied to it, being specially well suited for a low-speed, high-linearity and low power application. This configuration, which integrates source/sink capabilities, unary and binary structures, and cascode switches, has not been previously reported in the literature and represents

the main contribution of this master's thesis.

The next steps of the project will encompass finalizing the layout for the driver array, as well as integrating the complete top level design with the PAD rings. Finalization of this design will follow with layout-extracted simulations, and verification of the power distribution and reliability via a complete IR-drop analysis of the entire chip.

A complete component datasheet will also be compiled, containing the chip pinouts and appropriate electrical specification, supporting future measurements on it. Following the fabrication of the chip, the work will proceed on the design of the printed circuit board (PCB) for testing and experimental validation. Finally, the results obtained from the fabricated prototype will be consolidated and submitted for publication in a peer-reviewed journal.

Bibliography

[1] C. G. A. Ross, "Next generation adaptive optics," PhD dissertation, Dalhousie University, Halifax, Nova Scotia, Canada, July 2020. Cited 9 times on pages 9, 12, 13, 16, 17, 28, 29, 30, and 97.

[2] J. Lee, J. Weiner, H.-H. Chen, Y. Baeyens, V. Aksyuk, and Y.-K. Chen, "Cmos-based mems mirror driver for maskless lithography systems," in *2007 IEEE Custom Integrated Circuits Conference*, 2007, pp. 401–404. Cited 4 times on pages 20, 21, 22, and 97.

[3] A. Riboullet, F. Nabki, Y. Blaquière, and G. Cowan, "Configurable and intelligent switched cmos current driver powering arrays of electrothermal actuators for mems switches," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2024, pp. 1–5. Cited 3 times on pages 20, 22, and 68.

[4] D. Osipov, S. Paul, S. Strokov, A. K. Kreiter, A. Schander, T. Tessmann, and W. Lang, "Current driver with read-out hv protection for neural stimulation," in *2016 IEEE Nordic Circuits and Systems Conference (NORCAS)*, 2016, pp. 1–4. Cited on page 20.

[5] Y. Wu, P. Langlois, R. Bayford, and A. Demosthenous, "Design of a cmos active electrode ic for wearable electrical impedance tomography systems," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 846–849. Cited 2 times on pages 20 and 21.

[6] J. Li, Y. Wu, R. Bayford, D. Jiang, and A. Demosthenous, "A low power, low thd current driver with discrete common-mode feedback for eit applications," in *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2022, pp. 1–4. Cited on page 20.

[7] T. Moody, S. Ren, and R. Ewing, "10 bit current steering dac in 90 nm technology," in *NAECON 2014 - IEEE National Aerospace and Electronics Conference*, 2014, pp. 337–341. Cited 3 times on pages 20, 65, and 71.

[8] X. Li, H. Fan, Q. Wei, Z. Xu, J. Liu, and H. Yang, "A 14-bit 250-ms/s current-steering cmos digital-to-analog converter," *Journal of Semiconductors*, vol. 34, no. 8, p. 085013, aug 2013. [Online]. Available: https://doi.org/10.1088/1674-4926/34/8/085013 Cited 3 times on pages 20, 27, and 97.

[9] J. Li, D. Jiang, Y. Wu, N. Neshatvar, R. Bayford, and A. Demosthenous, "An 89.3 Cited on page 20.

[10] S. Choi, Y. Chon, and S.-M. Park, "A cmos current-mode vertical-cavity-semiconductor-emitting-laser diode driver for short-range lidar sensors," *Sensors*, vol. 24, p. 3513, 05 2024. Cited 3 times on pages 20, 21, and 65.

[11] U. Bihr, T. Ungru, H. Xu, J. Anders, J. Becker, and M. Ortmanns, "A bidirectional neural interface with a hv stimulator and a lv neural amplifier," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, pp. 401–404. Cited on page 21.

[12] V. Valente, A. Demosthenous, and R. Bayford, "Output stage of a current-steering multipolar and multisite deep brain stimulator," in *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2013, pp. 85–88. Cited on page 21.

[13] N. Dumas, C. Trigona, P. Pons, L. Latorre, and P. Nouet, "Design of smart drivers for electrostatic mems switches," *Sensors and Actuators A-physical - SENSOR ACTUATOR A-PHYS*, vol. 167, pp. 422–432, 06 2011. Cited on page 21.

[14] S. Li, L. Zhang, X. Qin, and Q. Han, "Design of a 12-bit segmented current-steering dac," in *2023 International Conference on Applied Physics and Computing (ICAPC)*, 2023, pp. 460–463. Cited on page 23.

[15] S. J. Tilden, T. E. Linnenbrink, and P. J. Green, "Standard for terminology and test methods for analog-to-digital converters: a case study of utilization of ieee-std-1241," *Computer Standards Interfaces*, vol. 22, no. 2, pp. 103–112, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548900000398 Cited on page 24.

[16] V. Kommangunta, K. Shehzad, D. Verma, and K.-Y. Lee, "Low power 10-bit 100 msps segmented current steering dac with $>$ 78 db sfdr," in *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, 2021, pp. 1–3. Cited 2 times on pages 25 and 26.

[17] A. N. Serov, N. A. Serov, and E. A. Dolgatcheva, "Influence of nonlinearity of adc conversion function on rms measurement error," in *2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2018, pp. 1–6. Cited on page 26.

[18] G. Manganaro. (2020) High speed digital-to-analog converters – a tutorial. YouTube video. Video available at https://www.youtube.com/watch?v=0hPApkpShmE (accessed December 9, 2025). [Online]. Available: https://www.youtube.com/watch?v=0hPApkpShmE Cited 7 times on pages 27, 49, 50, 51, 52, 53, and 54.

[19] W. Kester, "Understand sinad, enob, snr, thd, thd + n, and sfdr so you don't get lost in the noise floor," Analog Devices, Inc., Tutorial MT-003, 2008, rev. A, October 2008.

[Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf Cited 2 times on pages 27 and 28.

[20] L. Constantinou, I. F. Triantis, R. Bayford, and A. Demosthenous, "High-power cmos current driver with accurate transconductance for electrical impedance tomography," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 575–583, 2014. Cited on page 31.

[21] D. Colombo, G. Wirth, S. Bampi, and P. Srinivasan, "Voltage reference design using 1 v power supply in 0.13 µm cmos technology," in *2013 IEEE 4th Latin American Symposium on Circuits and Systems (LASCAS)*, Cusco, Peru, 2013, pp. 1–4. Cited on page 34.

[22] C. W. V. Casañas, T. H. P. de Castro, G. A. F. de Souza, R. L. Moreno, and D. M. Colombo, "A review of cmos current references," *Journal of Integrated Circuits and Systems*, 2022. Cited on page 35.

[23] D. M. Colombo, D. C. Lott, P. S. Locatelli, K. El-Sankary, and G. Burley, "±10 ma cmos current driver," in *IBERCHIP Workshop (IWS 2025)*, Manaus, Brazil, 2025, pp. 1–4. Cited 2 times on pages 45 and 99.

[24] B. Razavi, "The current-steering dac [a circuit for all seasons]," *IEEE Solid-State Circuits Magazine*, vol. 10, no. 1, pp. 11–15, 2018. Cited 4 times on pages 49, 50, 51, and 54.

[25] G. Manganaro, "An introduction to high data rate current-steering nyquist dacs: Fasten your seat belts," *IEEE Solid-State Circuits Magazine*, vol. 14, no. 3, pp. 24–40, 2022. Cited on page 52.

[26] V. Kaipu, "Design and layout of a high-speed high-resolution current steering dac based on an optimized switching sequence," Ph.D. dissertation, 07 2017. Cited on page 52.

[27] B. Guimarães, B. R. Raimundo, D. C. Lott, and D. M. Colombo, "High-fanout spi design: From rtl to mixed-signal physical implementation," in *Workshop on Circuits and System Design (WCAS 2025) – Chip in the Jungle*, Manaus, Brazil, 2025. Cited 4 times on pages 56, 57, 61, and 99.

[28] B. M. R. Raimundo, "Síntese física de um spi em tecnologia cmos 0,35 µm," Belo Horizonte, Brazil, July 2025. Cited on page 56.

[29] D. Martini Colombo, G. Wirth, and S. Bampi, "Sub-1 v band-gap based and mos threshold-voltage based voltage references in 0.13 a mu m cmos," *Analog Integrated Circuits and Signal Processing*, vol. 82, pp. 25–37, 01 2014. Cited on page 77.

[30] W.-H. Tseng, C.-W. Fan, and J.-T. Wu, "A 12-bit 1.25-gs/s dac in 90 nm cmos with >70 db sfdr up to 500 mhz," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2845–2856, 2011. Cited on page 97.

[31] Y. Tang, J. Briaire, K. Doris, R. van Veldhoven, P. van Beek, H. Hegt, and A. van Roermund, "A 14b 200ms/s dac with sfdr>78dbc, im3<83dbc and nsd<163dbm/hz across the whole nyquist band enabled by dynamic-mismatch mapping," in *2010 Symposium on VLSI Circuits*, 2010, pp. 151–152. Cited on page 97.

[32] Z. Qi, L. Ran, Q. Dong, Y. Ting, L. B. Yang, and H. Zhiliang, "A 14-bit 1-gs/s dac with a programmable interpolation filter in 65 nm cmos," *Journal of Semiconductors*, vol. 34, no. 2, p. 025004, feb 2013. [Online]. Available: https://doi.org/10.1088/1674-4926/34/2/025004 Cited on page 97.

Annex

ANNEX A – SPI Verilog Code

```
1  `timescale 1ns/1ps
2  module spi_lott_50drivers_10bits #(parameter number_bits=10,
       number_drivers=50)(
3       rst_n,    // reset ativo em n?vel baixo
4       sck,      // clock do SPI (fornecido pelo master)
5       csn,      // chip select (ativo em n?vel baixo)
6       mosi,     // linha de dados do master para o slave
7       dat_ol,   // vetor de sa?da (dados capturados organizados por
            driver)
8       spi_wr    // sinal de escrita habilitada
9  );
10
11 //============================
12 // Entradas / Sa?das
13 //============================
14 input sck;
15 input csn;
16 input mosi;
17 input rst_n;
18 output [number_bits*number_drivers-1:0] dat_ol;  // sa?da com todos
       os bits organizados
19 output spi_wr;
20
21 //============================
22 // Registradores internos
23 //============================
24 reg spi_wr;
25 reg [15:0] reg_bdat_o;   // registrador de 16 bits para capturar um
       frame inteiro recebido no SPI
26 reg [number_bits-1:0] reg_dat_o;   // registrador com os 10 bits
       ?teis extra?dos
27 reg [number_bits*number_drivers-1:0] reg_dat_ol; // registrador que
       acumula a sa?da final (todos drivers)
28 reg [3:0] cont_pos;      // contador de bits (dentro do byte
       recebido)
29 reg [3:0] cont_neg;      // contador de bits (em borda negativa
       para controle de sa?da)
30 reg [9:0] cont_byte_i;   // contador de "bytes" recebidos (endere?a
       cada driver na entrada)
31 reg [9:0] cont_byte_o;   // contador de "bytes" enviados (endere?a
       cada driver na sa?da)
32
33 wire csn_or_sck;
```

```
34 wire rst_csn_n;
35
36 //==========================
37 // Atribui??es cont?nuas
38 //==========================
39 assign dat_ol = reg_dat_ol;
40 assign csn_or_sck = csn || sck;      // usado para detectar borda
      de csn
41 assign rst_csn_n = rst_n && (!csn);  // reset ativo quando CSN sobe
42
43 //==========================
44 // Contador de bits recebidos (entrada MOSI)
45 // Avan?a a cada borda negativa do clock SPI
46 //==========================
47 always@(negedge sck or negedge rst_csn_n)
48 begin
49     if(!rst_csn_n) begin
50         cont_pos <= 0;
51         cont_byte_i <= 0;
52     end
53     else if(cont_pos == 4'b1111) begin // terminou de receber 16
          bits
54         cont_pos <= 0;
55         if(cont_byte_i >= number_drivers)
56             cont_byte_i <= 1;           // reinicia contador de
                drivers
57         else
58             cont_byte_i <= cont_byte_i + 1;  // avan?a para pr?ximo
                driver
59     end
60     else
61         cont_pos <= cont_pos + 1; // incrementa contador de bits
            dentro do byte
62 end
63
64 //==========================
65 // Contador de bits transmitidos (sa?da)
66 // Avan?a a cada borda negativa do clock SPI
67 //==========================
68 always@(negedge sck or negedge rst_csn_n)
69 begin
70     if(!rst_csn_n) begin
71         cont_byte_o <= 0;
72         cont_neg <= 4'b0001;
73     end
74     else if(cont_neg == 4'b1111) begin
75         cont_neg <= 0;
```

```
76          if(cont_byte_o >= number_drivers)
77              cont_byte_o <= 1;  // reinicia contador de drivers
78          else
79              cont_byte_o <= cont_byte_o + 1; // avan?a para pr?ximo
                    driver
80      end
81      else
82          cont_neg <= cont_neg+1;
83  end
84
85
86  //==========================
87  // Recep??o dos dados SPI
88  // Captura 16 bits, guarda apenas 10 (LSBs) e escreve no vetor de
        sa?da
89  //==========================
90  always@(negedge sck or negedge rst_n)
91  begin
92      if(!rst_n) begin
93          reg_bdat_o <= 0;
94          reg_dat_o <= 0;
95          reg_dat_ol <= 0;
96          spi_wr <= 0;
97      end
98      else if(csn) // se chip select n?o ativo, desabilita escrita
99          spi_wr <= 0;
100     else begin
101         if (cont_byte_i >= 0) begin
102                 case (cont_pos)
103                     // carrega bits do frame de 16 bits em
                            reg_bdat_o
104                     4'b0000, 4'b0001, 4'b0010, 4'b0011,
105 4'b0100, 4'b0101, 4'b0110, 4'b0111: begin
106                         reg_bdat_o[15 - cont_pos] <= mosi;
107                         spi_wr <= 0;
108                     end
109                     4'b1000, 4'b1001, 4'b1010, 4'b1011, 4'b1100,
                            4'b1101, 4'b1110: reg_bdat_o[15 - cont_pos]
                            <= mosi;
110
111                     // ?ltimo bit recebido (16?)
112                     4'b1111: begin
113                         // monta palavra com apenas os 10 LSBs
114                         reg_dat_o <= {reg_bdat_o[9:1], mosi};
115                         // armazena no vetor de sa?da a palavra
                                formada na sequencia anterior
```

```
116                         reg_dat_ol[(number_bits*cont_byte_i)-1 -:
                               number_bits] <= reg_dat_o;
117                         spi_wr <= 1;
118                     end
119
120                     default: begin
121                         reg_bdat_o <= 0;
122                         reg_dat_o <= 0;
123                     end
124                 endcase
125
126         end
127         else begin
128             reg_bdat_o <= 0;
129             reg_dat_o <= 0;
130             spi_wr <= 0;
131         end
132     end
133 end
134
135 endmodule
```

ANNEX B – SPI Genus Reports

Area Report

```
====================================================================
Generated by:        Genus(TM) Synthesis Solution 21.19-s055_1
Generated on:        Sep 27 2025  01:13:53 pm
Module:             spi_lott_50drivers_10bits
Technology libraries: c35_CORELIB_TYP 3.02
                   physical_cells
Operating conditions:_nominal_
Interconnect mode:    global
Area mode:         physical library
====================================================================

Instance     Module       Cell Count     Cell Area    Net Area    Total Area
-----------------------------------
spi_lott_50drivers_10bits  669      191191.000    52326.060    243517.060
```

Power Report

```
Instance: /spi_lott_50drivers_10bits
Power Unit: W
PDB Frames: /stim#0/frame#0
-------------------------------------
Category Leakage Internal Switching Total Row%
-------------------------------------
memory 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00%
register 1.54012e-09 2.14726e-04 6.25383e-06 2.20963e-04 75.72%
latch 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00%
logic 6.08620e-11 3.61482e-06 7.00668e-06 1.06216e-05 3.64%
bbox 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00%
clock 8.84000e-13 1.05337e-06 5.91817e-05 6.02351e-05 20.64%
pad 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00%
pm 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00%
-------------------------------------
Subtotal 1.60186e-09 2.19394e-04 7.24242e-05 2.91820e-04 100.00%
Percentage 0.00% 75.18% 24.82% 100.00% 100.00%
-------------------------------------
```