PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

A low-power recording system for intracortical signal acquisition based on design specification relaxation and lower bandwidth filtering

Tesis para obtener el grado académico de Maestro en Ingeniería Biomédica que presenta:

Luighi Anthony Vitón Zorrilla

Asesor:

Julio César Saldaña Pumarica

Lima, 2023

Informe de Similitud

Yo, Julio César Saldaña Pumarica, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada "*A low power recording system for intracortical signal acquisition based on design specification relaxation and lower bandwidth filtering"*, del autor Luighi Anthony Vitón Zorrilla, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 11%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 27/10/2023.
- He revisado con detalle dicho reporte y la Tesis, y no se advierten indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:
Lima, 27 de octubre de 2023

<table>
<tr><td colspan="2">Apellidos y nombres del asesor:
Saldaña Pumarica, Julio César</td></tr>
<tr><td>DNI: 10123705</td><td>Firma:</td></tr>
<tr><td>ORCID: https://orcid.org/0000-0001-6834-6436</td><td>  </td></tr>
</table>

To my parents, María Del Socorro and Segundo Herminio.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Ph.D. Julio Cesar Saldaña Pumarica, who taught me the basic concepts and techniques demanded by the development of this thesis and also for his guidance throughout the process. I also could not have undertaken this journey without the approval of the Master program committee, primarily Ph.D. Roberto Lavarello and Ph.D. Fanny Casado, who allowed me to choose this exciting topic and focus on the application of electronics to biomedical engineering, which was my main motivation to pursue a master in this area. Additionally, this work was only possible with the support of the Microelectronics Laboratory, lead by Ph.D. Carlos Silva and Mg. Mario Raffo, who provided me with the tools required in the first stage of the development of my thesis.

In the same way, I am deeply grateful to Professor Wilhelmus Van Noije and his microelectronics research group, especially to Ph.D. Bruno Sanches, from the University of Sao Paulo, Brazil. They allowed me to use their computational infrastructure to make long simulations that, without it, I couldn't finish all the tests I needed to achieve the objectives proposed in the thesis.

Lastly, I would like to thank my parents for their endless support and belief in me. Although this process took a long time, they always trusted I could finish it and gave me their understanding and help.

Abstract

In this thesis, we designed and evaluated a circuit model at the transistor level of a low-resolution and low bandwidth ADC (analog-to-digital converter) with level-crossing architecture (LCADC), used as part of the acquisition chain of a BCI (brain-to-computer interface) device. The aim is to obtain minimal specifications that could return adequate levels of accuracy at spike detection and reduce power dissipation. In addition, we included a NEO preprocessor in the test to help in the detection accuracy. To achieve the objectives proposed, we started developing a software model for the preprocessor and the ADCs to evaluate the different variations of resolution, bandwidth, noise level, and NEO window. After finding the desired minimum values, we continued with the hardware development of the ADC. We designed the level crossing architecture and a conventional SAR to have a reference against which we compare the LCADC performance. After that, we developed a NEO circuit and applied synthesized neural recordings to evaluate power consumption with the ADC. Additionally, we designed a conventional analog frontend to estimate the power for the band of interest. Also, we estimated the dissipation for wireless transmission by calculating the approximated data stream expected in the level-crossing sampling scheme. Summing them, we obtained the power consumption of the complete acquisition chain. In conclusion, although reducing the intrinsic power of the LCADC is challenging, the scheme helps reduce the total power consumption of the acquisition chain with adequate accuracy, making it competitive against currently reported BCI devices.

Keywords: BCI, intracortical signals, LCADC, SAR, NEO, specification relaxation, lower bandwidth.

Resumen

En esta tesis se ha diseñado y evaluado un circuito a nivel de transistores de un ADC (conversor analógico-digital) de baja resolución y ancho de banda reducido con arquitectura de cruce de nivel (LCADC), utilizado como parte de la cadena de adquisición de un dispositivo BCI (interfaz cerebro-computadora). El objetivo es obtener especificaciones mínimas que puedan devolver niveles adecuados de precisión en la detección de picos y reducir la disipación de energía. Además, se incluyó un preprocesador NEO en la evaluación para ayudar en la precisión de la detección. Para lograr los objetivos propuestos, se desarrolló un modelo de software del preprocesador y los ADCs a fin de evaluar las diferentes variaciones de resolución, ancho de banda, nivel de ruido y ventana del NEO. Luego de encontrar los valores mínimos deseados, se continuó con el desarrollo de hardware del ADC. Se diseñó la arquitectura de cruce de nivel y un SAR convencional para tener una referencia con la cual comparar el rendimiento del LCADC. Después de eso, se desarrolló un circuito NEO y se aplicó las señales neuronales sintetizadas para evaluar el consumo de energía con el ADC. También, se diseñó una interfaz analógica convencional para estimar la potencia de la banda de interés. Así mismo, se estimó la disipación de la transmisión inalámbrica calculando el flujo de datos aproximado esperado en el esquema de muestreo de cruce de nivel. Finalmente, sumando los resultados se obtuvo el consumo de energía de la cadena de adquisición completa. A partir de estos, se concluye que aunque reducir la potencia intrínseca del LCADC es un desafío, el esquema ayuda a disminuir el consumo total de energía de la cadena de adquisición con una precisión adecuada; lo cual lo hace competitivo frente a otros dispositivos BCI reportados actualmente.

Palabras clave: BCI, señales intracorticales, LCADC, SAR, NEO, relajación de especificaciones, ancho de banda reducido.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC Analog-to-Digital Converter.

ALS Amyotrophic Lateral Sclerosis.

BCI Brain Computer Interface.

BER Bit Error Rate.

DAC Digital-to-Analog Converter.

DD Discrete Derivative.

DNL Differential Nonlinearity.

DSP Digital Signal Processor.

DWT Discrete Wavelet Transform.

EAP Extracellular Action Potential.

ECG Electrocardiography.

ECoG Electrocorticography.

EDA Electronic Design Automation.

EEG Electroencephalography.

ENOB Effective Number of Bits.

ERBW Effective Resolution Bandwidth.

FFT Fast Fourier Transform.

FOM Figure of Merit.

FSDE First and Second Derivative Extrema (FSDE).

IC Integrated Circuit.

INL Integral Nonlinearity.

LCADC Level Crossing Analog-to-Digital Converter.

LFP Local Field Potential.

LSB Low Significant Bit.

MEA Multielectrode array.

MSB Most Significant Bit.

MU Multi-unit activity.

NEO Nonlinear Energy Operator.

OPAMP Operational Amplifier.

OSR Oversampling ratio.

PCA Principal Component Analysis.

RMS Root-mean-square.

SAR Successive Approximation Register.

SBP Spiking Band Power.

SFDR Spurious Free Dynamic Range.

SHA Sample and Hold Amplifier.

SINAD Signal-to-noise-and-distortion Ratio.

SNDR Signal-to-(Noise plus Distortion) Ratio.

SNHR Signal-to-non-harmonic Ratio.

SNR Signal-to-Noise Ratio.

SU Single-unit activity.

TCR Threshold Crossing Rate.

TDC Time-to-Digital Converter.

TSMC Taiwan Semiconductor Manufacturing Company.

Chapter I

Introduction

1.1 Context

Communication and mobility are two of the most critical abilities for human beings, as they permit them to live and behave normally, developing their daily activities with independence. However, some diseases and injuries cause severe damage to the brain or spinal cord, affecting these abilities and dramatically changing their lifestyles.

One of the most common diseases that produce these detrimental effects is amyotrophic lateral sclerosis (ALS). It is a rare neurological disease that compromises the nerves controlling voluntary muscle movement, as referred to in the National Institute of Neurological Disorders and Stroke [1]. It is caused by a gradual deterioration and death of motor neurons which, in advanced stages, affects all voluntary muscles. Individuals with this disease lose muscle strength, affecting communication and motor abilities, and even putting their lives at risk. ALS strikes more commonly in older people between 55 and 75. Their origin could be because of genetics or environmental factors. Among the methods to treat ALS, aside from physical and occupational therapy, brain-computer interfaces (BCIs) are employed as communication support or to control equipment using brain activity.

On the other hand, spinal cord injuries affect between 250000 and 500000 people, most of the cases due to road traffic crashes, falls, or violence; according to the WHO [2]. The possibilities with this injury vary from partial loss of sensory or motor function to several problems with systems that control breathing, heart rate, and blood pressure. Unlike ALS, this injury is more prevalent in young adulthood (20-29 years) and older (+70). Its costs are higher than comparable conditions such as dementia, multiple sclerosis, and cerebral palsy. This health problem has personal and social impacts. Consequently, including appropriate assistive devices constitutes one essential measurement that could help them

to perform their daily activities. In this case, BCI also provides these devices to restore communication or help in stroke rehabilitation [3].

BCIs have enabled multiple applications to help individuals with paralysis to improve their quality of life. Slutzky [3], in his review, identifies three main clinical applications, which include restoring communication to people without articulatory or limb movement, restoring limb function to people with tetraplegia or limb amputation, and supporting stroke rehabilitation. In the case of restoring communication ability, the aim is to provide an assistive device that can interpret and reproduce a sequence of letters at a reasonable rate. Although preliminary studies only offered a communication rate of a few letters per minute, more recent ones, which employed intracortical BCI, achieved higher rates of up to 32 letters per minute, better than ALS patients said was acceptable in a survey presented by Huggins, Wren, and Gruis [4]. In the case of restoring limb function, BCIs have been used to control robot arms, prosthetic arms, and even exoskeleton training to improve patient waking. BCI for stroke rehabilitation aims to rehabilitate function by stimulating the brain, gaining plasticity, which in conjunction with physical therapy, obtains improved results compared to only physical therapy. In addition to these applications, researchers are investigating to employ BCI for restoring somatosensation or replacing cognitive functions such as memory.

However, those benefits couldn't be possible without the research on the technology behind the interfaces that interact with intracortical signals and attempt to acquire, record, and decode them to interpret and provide useful information about the neural basis of patient intentions. The advancements in this technology include moving towards wireless devices to implement viable biomedical equipment for a clinical environment. It also demands some constraints to circuits and considering other techniques to reduce the power consumption, maintaining the performance required for the application.

For example, a recent study proposed by Simeral *et al.* [5] shows the tendency to move the previous cabled intracortical interface to a wireless one. It presents the first high-resolution broadband system which records from multiple implanted microelectrode arrays by employing wireless intracortical BCI. The results demonstrate a similar performance compared with the wired counterpart, where the subjects of study with tetraplegia could interact with software applications such as Pandora, YouTube, Gmail, and the Weather app. Then, it shows an important advance as an assistive technology to help individuals with mobile and speech abilities, restoring some communication and interaction via digital means.

Despite this promising step, there are some challenges related to the electronic system that the interface relies on. The acquisition system was commercial hardware from Blackrock Microsystems for wired and wireless tests. The wired system was constituted by the array of electrodes followed by cables connected to the analog fronted, with a filter (0.3Hz to 7.5kHz) and an analog-to-digital converter (30 kS/s, 16 bits/sample). Then, it sends the information to a Neural Signal Processor, timestamped, and sent to the central suite for offline analysis. On the other hand, in the wireless system, a wireless transmitter and receiver replace the cables while including an analog frontend and converter next to the implanted electrodes, whose characteristics are: for the filter (1Hz to 7.8kHz) and for the converter (20 kS/s, 12 bits/sample). Although both systems show comparable performance, the wireless version has the challenge that it should operate powered with batteries and work for at least 24 hours. The tested wireless system had a design for high-fidelity and high electrode counts. However, the tendency is to increment the count of electrodes; therefore, it should be adapted to send the same information with lower bandwidth. This could be addressed by confining the signal acquisition to spiking events instead of considering the whole broadband data. Some studies also show that sampling at 1kHz could be sufficient. Furthermore, the design should be aware of the low power for long battery life, considering that increasing fidelity via higher resolution and sampling rate negatively impacts power consumption.

This thesis will focus on the electronic system for acquiring the extracellular action potentials from an implantable electrode. The aim will be to propose some improvements in the converter, specifically, to reduce the required resolution via an alternative architecture and the preprocessor placed before the characteristic detection, which allows applications such as those stated before. Next, the problem statement explains in detail the approach to these topics.

1.2 Problem Statement

As shown in [5], some the circuit-level characteristics impact the device's the power consumption. According to Even-Chen *et al.* [6], the acquisition chain contains several blocks that cover the sensor, the interface readout (with recording and transmitting functions), the preprocessor (sometimes included before the transmitter), and the decoder (which extracts and measures some features from the acquired signal). The interface readout contains typical components such as the amplifier, analog-to-digital converter (ADC), and

transmitter. These components have some specifications which determine their power consumption. In the case of the amplifier, it depends on the power supply voltage, bandwidth, and noise. The ADC determines it by the voltage level, resolution bits, and sampling frequency. On its side, the transmitter power depends on the number of channels (electrodes), bit error rate, and data rate. Some of these specifications could be weighed to reduce consumption; however, the introduced changes must not significantly impact system performance.

According to Slutzky [7], two main approaches would enable an adequate trade-off between power reduction and system performance. On the one hand, the bandwidth employed to record the spike signals could be reduced to a specific band (300–1000Hz) named spiking-band-power (SBP), also reducing the power requirements with a performance equivalent to the traditional method based on spike-detection threshold-crossing rate (TCR). On the other hand, the whole power system could be reduced by relaxing multiple design specifications among the components that constitute the acquisition system, maintaining a similar performance compared to the original system.

The first technique, presented by Cynthia Chestek's group [8], shows that the SBP is comparable in efficiency to the TCR approach by using simulations of recordings of neural activity. This technique relies on consuming less bandwidth than classical TCR, a common solution in the intracortical signal acquisition, but it only reduces the power after digitalization. In this case, it takes advantage of the fact the band between (300–1000Hz) contains a significant amount of power density (25% to 45%), and filtering it cuts the noise level by a factor of about two, being more robust to the noise compared with the TCR technique. Results also show that this band could reduce power, maintaining 95% of the decoding performance.

The second technique, presented by Even-Chen *et al.* [6], is centered on relaxing hardware design specifications to reduce the total power consumed. This approach is based on the fact that the number of electrodes could be more important than acquiring the whole band at high resolution (10–16 bits). High values for this parameter may interest basic neuroscience research, but it is not mandatory for movement intention decoding. Indeed, most of the specifications for the amplifier could be reduced to a lower bandwidth between 0.5–3kHz with a minimal number of bits for the ADC (7 bits). Also, in the transmission step, only transmitting binary information (obtained from threshold crossing events) could significantly reduce the bandwidth required to transmit this signal. On the other hand, although reducing these specifications could limit the ability to isolate a single neuron

recording, it offers the ability to record more signals with a higher number of electrodes. This setup could be applied to scenarios where wireless recording is essential, such as in clinical practice.

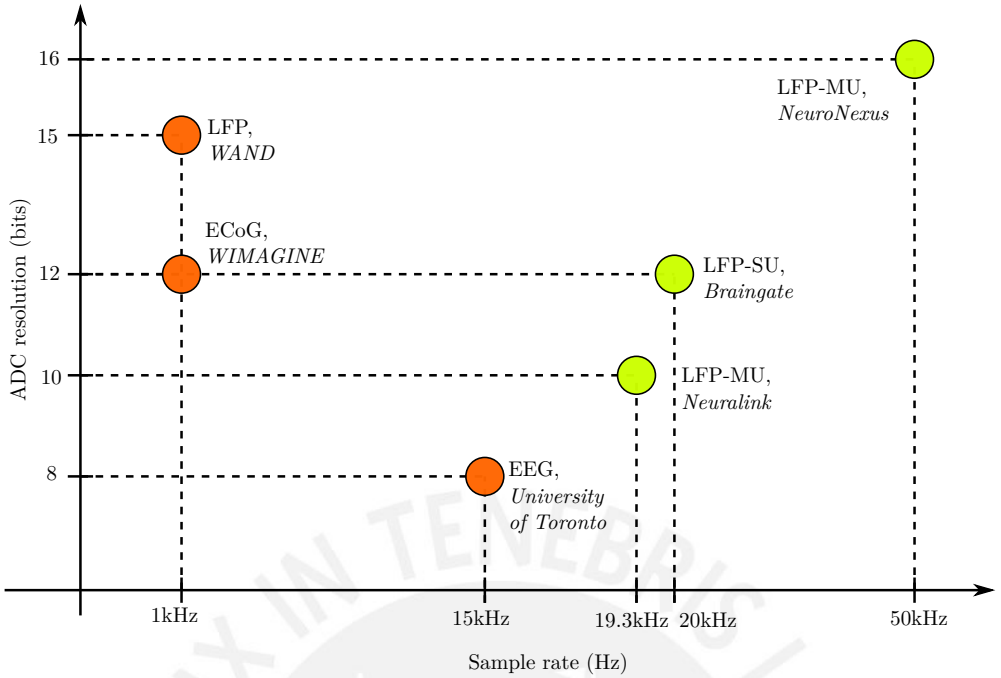
While these studies propose some design specifications and strategies to reduce the power from a theoretical point of view, Slutzky [7] suggests that those could be applied via minor modifications to current circuit designs. Moreover, combining them with other power-saving techniques could lead to greater power reductions.

In the specific case of the ADC, the circuit proposals are related to changes in the architecture. One of them is presented by De Dorigo *et al.* [9], and it introduces an ADC architecture based on a first-order Sigma-Delta ADC, designed for 11 bits of resolution and a Nyquist frequency of 10kHz. This approach minimizes the signal chain, discarding other components such as preamplifiers and filters. This circuit led to lower power consumption in the order of 40 μW. However, it employs a higher resolution than the one necessary for decoding the movement intention, according to Even-Chen *et al.* [6].

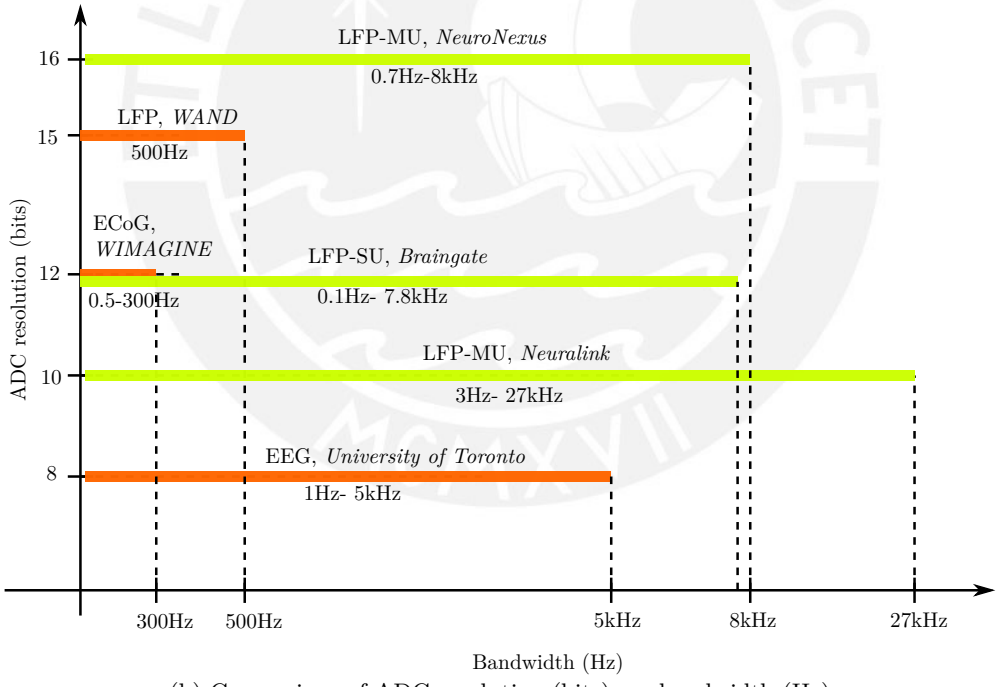
Thus, considering the proposed lower resolution requirements, other architectures have proven more energy efficient than typical ones such as successive-approximation-register (SAR) or Sigma-Delta at lower resolution. This is the case of the level crossing ADC (LCADC) architecture. Van Assche and Gielen [10] state that it converts the signal when it crosses a reference level, unlike regular ADCs that take samples at fixed rates. It is beneficial in sparse biosignals that change only sporadically, conducting a directly compressed signal and reducing power consumption. According to Van Assche and Gielen [10], this architecture is more power efficient than SAR at a cross-over point in the quantizer resolution. In extracellular action potentials (EAPs) such as the intracortical signals, this resolution is 7.6 bits. Although the study demonstrates the advantages of LCADC architecture, it doesn't validate their performance at the decoding level.

Concerning clinical trials performed with BCI, the devices had different characteristics depending on the type of neural acquisition. These characteristics were summarized by Zaer *et al.* [11]. He showed that most circuits intended to record multi-unit activity (MU) and single-unit activity (SU) had higher sampling frequency, resolution, and bandwidth specifications. However, the proposal is to reduce them to minimize power consumption. Their characteristics can be appreciated in Figure 1.1.

In addition, before the decoding stage, there is an essential step of spike detection. According to [6], this step reveals an opportunity to employ a NEO that could produce more robust protection against thermal noise, giving a margin to relax the amplifier and



(a) Comparison of ADC resolution (bits) vs. sample rate (Hz)



(b) Comparison of ADC resolution (bits) vs. bandwidth (Hz)

Figure 1.1: Comparison of different BCI devices employed in clinical trials. Here, we differentiate between the low-frequency BCI (orange) and the higher-frequency BCI (lime). The present thesis is focused on the second category. The texts in italics correspond to the study names. The acronyms used refer to: local field potential (LFP); multi-unit activity (MU); single-unit activity (SU); electrocorticography (ECoG); electroencephalography (EEG). Intracortical BCI is usually intended to measure SU and MU. Adapted from Zaer *et al.* [11].

ADC specifications. Additionally, its hardware simplicity benefits its implementation. The NEO operator is characterized as it emphasizes the amplitude-energy variation of the neural spikes, improving the signal-to-noise ratio (SNR), even in noisy environments. This preprocessing technique has been applied with an automatic threshold estimator, achieving a high level of accuracy, close to 90%, as shown by Yang, Boling, and Mason [12]. Another work, proposed by Zamani, Jiang, and Demosthenous [13], modifies the NEO to operate with non-adjacent samples improving the detection of signals with low-frequency components. Although both studies have demonstrated the use of this technique with neural signals, they didn't evaluate the NEO to samples with non-fixed frequency, low resolution, or lower bandwidth.

To sum up, the literature states that relaxed specifications for the ADC and lowered bandwidth for signal filtering demonstrate similar performance in the movement intention decoding task, as claimed by [6] and [8], respectively. Reducing the resolution is one strategy to relax its specifications. At lower resolutions, other architectures, such as LCADC, are more energy efficient, which have not been evaluated, until now, against intracortical signals. Hence, there is an opportunity to test the LCADC performance at detecting spikes in the neural signal as a previous step for decoding. In addition, in the pre-processing stage, there are techniques to improve spike detection, such as the NEO operator. However, it has not been evaluated either against the characteristic output of the LCADC or lower specifications for the input signal related to bandwidth or resolution. Therefore, there is also a chance to test the NEO operator applied to the LCADC output, determining if it could preserve its benefits of improving the SNR despite the special characteristics of the samples. Also, both components, the LCADC, and the NEO, should not increase the power consumption significantly as they will be part of a wireless device; consequently, it demands an evaluation at this level comparing them with other systems from the literature to ensure it could accomplish the power limits stated by prior developments.

1.3 Justification

BCI systems have a wide range of applications for restoring communication or motor abilities via the construction of assistive devices. This technology has been assessed in clinical trials and also via meta-analysis studies.

A recent analysis of this type was developed by Mansour *et al.* [14], which assesses BCI systems for poststroke upper-limb rehabilitation considering 12 clinical trials. As referred by Mansour *et al.*, BCI has significant effects in improving upper-limb motor functions after stroke, at the short and long-term, compared to other therapies. Although the study didn't evaluate intracortical modality, it generally exhibits the potential for BCI technology.

Indeed, from the performance perspective for restoring function, intracortical BCI has the most promising results [3]. For example, a clinical trial identified as NCT00912041 [15] applied by Pandarinath *et al.* [16] to three patients, two with ALS and one with spinal cord injury, demonstrated that the intracortical BCI has the potential as an assistive communication system. It achieved up to 39 letters per minute (7.8 words per minute), more than the satisfaction rate in a previous study (15-19 letters per minute). The setup for this experiment can be seen in Figure 1.2. In addition, in a pilot clinical trial, Simeral *et al.* [5] demonstrated an application for wireless intracortical BCI to use several mobile applications in a standard commercial tablet computer by controlling the mouse and making gestures of pointing and clicking. Recently, Willett *et al.* [17], in their study, have demonstrated that a BCI system could restore communication via decoding handwriting movements from the neural activity in the motor cortex. It has proven to achieve a 94.1% accuracy with speeds of 90 characters per minute, comparable to typing speeds in the age group (115 characters per minute). According to users, the system was easy to use and accurate enough to be useful in the real world.

On the other hand, additional studies were deployed to assess intracortical BCI stability. It is the case of the clinical trial NCT01364480 [18]. Downey *et al.* [19] evaluated an intracortical microelectrode array recording in the motor cortex BCI of two users. The results demonstrated that well-tuned units with high firing rates could remain stable for weeks and even months, enabling the possibility of using this BCI system without technical intervention in the short term. Colachis *et al.* [20] assessed an intracortical microelectrode array performance over another clinical trial in 5 years. According to this study, although it suffered from degradation that attenuated signals declining in multiple neural signal metrics, the participant could sustain high performance throughout the 5-year study.

As stated by Slutzky [3], the perspective in BCI systems is the development of completely-implantable and wireless devices with reduced power consumption and longer stability. This is in agreement with the necessity for clinical adoption and also with actual patient preferences. Although BCI developments didn't accomplish the expected performance

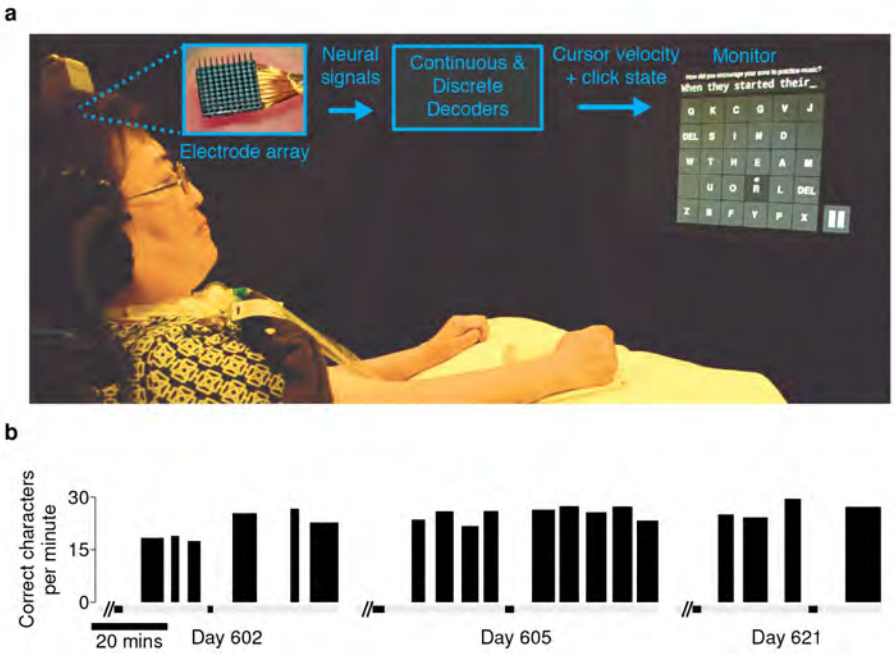


Figure 1.2: Clinical trial to perform tablet application control as a communication mean. Figure (a) presents the experimental setup for patient T6, a 51-year-old diagnosed with ALS. In this setup, the intracortical BCI has an electrode array, an acquisition system to obtain the neural signals, and decoders to interpret them as cursor velocity and click state. Figure (b) presents the performance achieved in three days for questions. The performance is measured as the correct characters rate. The average here was approximately 24 characters per minute. Obtained from [16]. CC0 1.0.

from potential users with ALS ten years ago, as showed by Huggins, Wren, and Gruis [4], the advancements in the technology have motivated the discussion about which modality could be preferred by people with paralysis. This assessment, performed by Blabe *et al.* [21], compared external and intracortical BCI technologies and their potential control capabilities. The participants' answers suggest that the desired device must be autonomous, unobtrusive, and require little to no maintenance. The conclusion is that a surgical implanted wireless device better manages these characteristics. A more recent survey, developed by Branco *et al.* [22], assessed the preferences of individuals with locked-in syndrome in communication BCI systems compared to other assistive technologies. This survey showed that patients with fewer residual movements available were more willing to use this device. However, most participants would consider them when their performance is better than current assistive technologies. In addition, contrary to the previous study, participants preferred noninvasive against the implanted device, although they had different clinical conditions and the sample size was limited.

As stated by the clinical trials and the surveys to potential users, intracortical BCIs have the potential for clinical applications. Their development should be towards an au-

tonomous system that implies wireless communication. However, they need further development to ensure long periods of use and stability, which could promote user acceptance, and the corresponding clinical trials to assess those features.

In that manner, there are advancements from the industry to the academy to achieve more reliable intracortical BCI systems. In the industry's case, tech companies such as Neuralink [23] have developed BCI systems with high channel count and single-spike resolution, with low-power electronics, which may serve as a platform for the research on the restoration of sensory and motor function. It has been tested on a pig and a macaque monkey [24]. Another company whose products are used in various research works is Blackrock Neurotech. It manufactures wired and wireless neural interfaces (including electrodes and the electronic system) applied in novel implantable clinical solutions [25]. On the academic side, an interesting project named BrainGate [26] aimed to develop BCI technologies to restore communication, mobility, and independence in persons with neurological diseases. They focus their research on assistive communication, movement restoration, neuroscience, neurotechnology, and neurotherapeutics to achieve it. It leveraged several studies referenced here, such as those performed by Simeral *et al.* [5], Willett *et al.* [17], and Even-Chen *et al.* [6].

The presented thesis aims to contribute to this field, providing an alternative implementation for the supporting circuit required by the wireless intracortical BCI. The objective is to propose some changes to the architecture inspired by the literature advancements that could reduce power consumption and increase their autonomy. We expect that, although minimal, it will constitute a step forward to their clinical adoption and the corresponding use by patients with limited mobility and communication, such as those affected by ALS or spinal cord injury.

1.4 Objectives

1.4.1 General

Propose a low-power circuit-level acquisition system for intracortical signals based on relaxed design requirements and lower band filtering employing a level-crossing ADC and a non-linear energy operator with a high accuracy at spike detection to be used in wireless medical devices.

1.4.2 Specific

1. Design and simulate a low-resolution level-crossing ADC (LCADC) architecture for intracortical signals acquisition to evaluate its conversion performance in terms of SNR.

2. Implement and simulate the non-linear energy operator (NEO) in conjunction with a lower bandwidth filtering to evaluate its performance in terms of spike detection accuracy (>90%).

3. Simulate the LCADC and NEO blocks together to evaluate the performance of the proposed design considering a power consumption estimation and spike detection accuracy compared with other commercially available and academic developments in the state-of-the-art (<30 µW).

1.5 Hypothesis

A circuit-level design combining a Level-Crossing ADC architecture with a Non-linear Energy Operator will reduce the power consumption and maintain high accuracy at the spike detection task.

1.6 Methodology overview

The research procedure to verify the presented hypothesis is in the scope of the analog integrated circuit (IC) design, which comprises several steps such as the design specifications, architecture design, circuit level design, physical design (layout), fabrication, and testing. In this case, the approach only reaches the physical design and simulation, employing for this process, the mainstream industrial-level precision software Cadence Virtuoso, provided by the *Grupo de Microelectrónica PUCP*, which has several tools to help the analysis and evaluation of the proposed design. The technology employed will be the standard stated by the Taiwan Semiconductor Manufacturing Company (TSMC) 180 nm.

According to the thesis objectives, this work is centered on two components: the ADC and the preprocessor for spike detection. The methodology to evaluate those strategies is the following:

1. *ADC design with LCADC architecture.* This design includes the study of the topology and its different variations to evaluate its performance with intracortical signals. To compare the results, it will be necessary to build a baseline with another architecture, such as the SAR, employed regularly in the state-of-the-art. The parameter of comparison will be the SNR. This step could be divided into the following tasks:

    - Build a SAR architecture employed in the state-of-the-art to establish a baseline to compare the performance of the proposed design.
    - Design the LCADC architecture of 7 bits and compare different topologies for its implementation. It includes designing internal components (comparator, digital-to-analog converter (DAC), controller, and timer) to build the ADC and employ low-power techniques to reduce block consumption.
    - Evaluate the LCADC with generated signals to test and obtain performance metrics. It includes some tests in transient time and frequency of each component. It also requires adjusting physical dimensions of transistors to obtain the required functionality.
    - Evaluate the LCADC with intracortical signals from data banks in terms of SNR. It includes running a simulation of the designed circuit against a publicly available intracortical signal data bank and measure the SNR employing tools available in Cadence.

2. *NEO design an evaluation as a preprocessor for spike detection.* It includes the implementation of different approaches for this NEO component and compares the performance of spike detection against using only a threshold crossing level approach. It also considers evaluating the characteristic variations introduced by the samples obtained from LCADC architecture. The key tasks for this step are:

    - Build a simple threshold crossing level block to serve as a baseline to compare the performance of the design NEO preprocessor.
    - Design a NEO preprocessor with two approaches: with the original equation and a proposed variation [13] to be more robust to low-frequency components.
    - Evaluate the NEO preprocessor with low-resolution samples (7 bits) and compare with high-fidelity samples (12 bits or more) regarding spike detection accuracy.

- Evaluate NEO preprocessor with lower bandwidth (300–1kHz) for both variants compared with a broadband signal regarding spike detection accuracy. Also, evaluate the performance combined with a low-resolution signal.
- Evaluate NEO with non-fixed frequency samples compared with fixed frequency samples regarding spike detection accuracy. Compare its performance by combining this non-fixed frequency with low resolution and lower bandwidth.

3. *Full chain acquisition system design, including LCADC and NEO.* This is to evaluate the performance in terms of spike detection accuracy and measure the power consumption of the whole system. It includes a comparison with other acquisition systems in terms of power consumption. The tasks considered here are:

   - Design a low-power amplifier to meet specifications for the LCADC of 7-bit resolution and the bandwidth required for the spike detection. It includes techniques to reduce this component's power to the minimum grabbed from the state-of-the-art.
   - Design the bandpass filter before the NEO preprocessor considering the spiking band power (SBP) of 300–1kHz [8].
   - Include the LCADC and NEO with the other components (amplifier and filter) and test their functionality against intracortical signals, measuring the performance at spike detection accuracy.
   - Compare the power consumption of this circuit-level design with other commercially available and academic developments, identifying the consumption of each stage and determining the contribution to reducing it thanks to the changes in ADC architecture and preprocessor.

1.7 Thesis structure

This master thesis is structured into five chapters which are the following: Introduction (current chapter), Theory, Methodology, Results, and Conclusions. The first chapter provided a general thesis overview, including the objectives, hypothesis, and methodology summary.

The second chapter is focused on the theory fundamentals to understand the concepts and processes developed in the rest of the document. It considers an overview of electronic

acquisition systems for intracortical signals, diving into the two main aspects of interest for the objectives: the analog-to-digital converter (ADC) and the preprocessor.

The third chapter describes the methodology, considering the previous steps required to evaluate the NEO preprocessor, the steps in designing the two blocks proposed in this thesis, and the strategy for power estimation.

The fourth describes the design and implementation, beginning with the software developed for the NEO evaluation and extraction of minimal specifications. Then, it follows with the design of the LCADC and SAR used as reference circuit, and finally, the NEO circuit design which is used to estimate the power dissipation with the ADC circuit.

The fifth chapter describes the results obtained from each step considered in the methodology evaluating the designed systems and circuits' performance, considering the criteria defined in the methodology. It also presents discussions of such results for the proposed objectives.

The last chapter includes the thesis's conclusions, describing if the objectives have been achieved and if the obtained results have supported the hypothesis. In addition to this, it provides some perspectives on the developed topics.

Chapter II

Theory on acquisition chain for intracortical signals

In this chapter, there is a description of the fundamentals of the acquisition chain for intracortical signals, including some concepts such as the extracellular action potentials, an overview of the acquisition chain, the state-of-the-art ADCs for neural signal acquisition, the preprocessing techniques to improve spike detection and feature extraction, and finally the current developments for extracellular signal generation, which will be employed for the evaluation of the proposed design.

2.1 Extracellular action potential

The action potentials are the mechanisms neurons use to communicate within the nervous system. These potentials are rapid and in the order of milliseconds of duration and millivolts of amplitude. These impulses are transmitted along the axon to its termination [27]. They are also referred to as spikes in the literature.

This potential originated in the neuron membrane due to an ionic variation across it, alters the resting membrane potential (around $-70\,\mathrm{mV}$). Then, when this excitation exceeds a threshold value, it generates an abrupt rise in the membrane potential reaching values near $30\,\mathrm{mV}$. This generated potential, then, is propagated along the nerve surface with a definite velocity which depends on the myelination and axon size due to the depolarization of neighbor ion channels [28]. The transmitted action potential could be measured and recorded, conducing to the intracellular action potential recording.

The intracellular recording measures the transmembrane potential using a pipette inserted into a cell and recording the differential potential relative to an extracellular

reference electrode, according to The McGill Physiology Virtual Lab [29]. It is useful for a very accurate assessment of electrical activity; however, it could risk damaging the membrane in the process.

Conversely, an extracellular recording involves measuring the potential between an electrode in the cell membrane proximity and a reference somewhere in the extracellular fluid. In this case, it measures the changes in the membrane surface instead of across the membrane.

Although this technique is less demanding than intracellular recording, it has some drawbacks. It can not obtain the exact waveform of the action potential as it will depend on the electrode position related to the membrane, and its amplitude is reduced compared to the intracellular one. This kind of recording is more useful for detecting when the action potential has occurred or for recording the activity of a group of cells rather than the waveform of the spikes [30].

When acquiring extracellular action potential (EAP), two types of signals are recorded: local field potentials (LFPs) from neural populations, which are in a frequency range from 1 Hz to 300 Hz, and the activity from individual neurons in a frequency range from 300 Hz to 5 kHz; according to Viswam *et al.* [31]. Although local field potentials contain much movement information, they cannot reach the performance of spike-based signals with higher frequency. On the other hand, there is a band at the beginning of the EAP range called spike-band power (SBP) from 300 Hz to 1 kHz. It can reduce the frequency requirements of acquisition and achieve superior performance than local potentials [8]. This thesis will use and evaluate this band from the complete range of extracellular action potentials.

2.2 Brain-computer interfaces (BCI)

As described by Shih, Krusienski, and Wolpaw [32], a brain-computer interface (BCI) is a computational system that can acquire, process, and translate brain signals to commands, received and performed by an output device. Thanks to BCI, the individual could perform actions of locomotion, movement control, environmental control, or communication. However, this will require a process of training based on the intended action and decoding similar ones by these signals, making such interaction with the user environment.

The BCI system has four main components:

- *Signal acquisition.* A set of electronic systems responsible for obtaining the signals and digitizing them to send to a processor.

- *Feature extraction.* The signal analysis process to extract the characteristics that may be correlated to the user's intent. It should minimize the physiologic artifacts to ensure the fidelity of the extracted features.

- *Feature translation.* This will convert the extracted features to a corresponding command in the output device. It also should consider the variability of those signals to ensure the control of the device by the user.

- *Device output.* It corresponds to the device executing the commands translated in the previous stage. This external device could perform some functions, such as the movement of a prosthetic arm, the letter selection or mouse movement in a computer application, or the translation of a wheelchair.

Although all BCI systems should include these components, not all will obtain the signals similarly and perform the same applications. Physiologically, BCI systems may be differentiated by the kind of signals and methodologies they use for their applications and can be classified into three categories, according to Shih, Krusienski, and Wolpaw [32]:

- *Scalp-recorded electroencephalography (EEG).* It is the least invasive with minimal risk involved. Due to the high attenuation limited to specific applications in movement control and discrete selection. It is dominated by lower frequency ($< 40\,\text{Hz}$).

- *Electrocorticography (ECoG) activity.* It records signals from the cortical surface, requiring a subdural or epidural electrode array implantation. Their recordings have higher amplitude than EEG and in a broad band, including activity up to 200 Hz.

- *Intracortical activity.* It uses an implanted array of intracortical microelectrodes. This technique provides a better recording activity resolution than less invasive technologies. However, it has some drawbacks related to their limited longevity due to their high invasiveness and continuous degradation of signal quality. The electrodes' materials and geometries are in ongoing research and development to reduce the foreign body reaction [33]. BCIs based on this methodology could be referred to as intracortical BCIs or iBICs.

In [33], Szostak offers an interesting illustration presented in Figure 2.1, which represent the different type of brain interfaces related to their position relative to the brain, corresponding to the methodologies explained above.

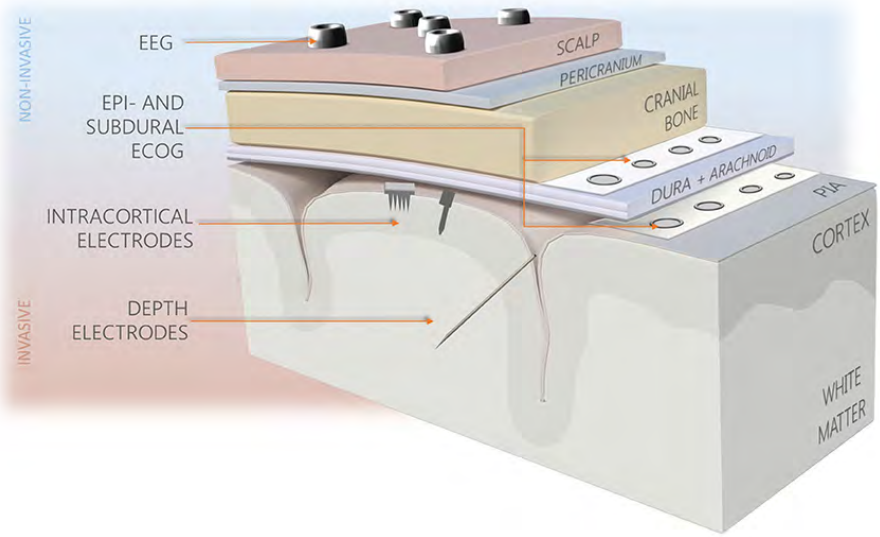


Figure 2.1: Different types of brain interfacing electrodes. Obtained from [33, Fig. 2]. Copyright © 2017 Szostak, Grand, and Constandinou.

Although the electronic system after the electrodes are similar, this thesis centers on the intracortical systems and the specific characteristics of signals and electronics they demand.

2.3 Overview of signal acquisition chain

To obtain the recording signals and extract their characteristics, it is necessary to build an electronic system that captures, transforms, and processes those signals. All of these blocks constitute the signal acquisition chain.

This acquisition chain has some defined stages responsible for a certain function in that process. Those are the electrodes, the readout circuit, the preprocessor, and the feature extractor.

2.3.1 Electrodes

Electrodes are those devices that are in contact with the sample or interact with the phenomena from which it requires to acquire the signals.

In BCI systems, there are several types of electrodes according to the kinds of BCI, as shown in Section 2.2. In the case of intracortical systems, the most common type is

the multielectrode array (MEA) which consists of many electrodes spatially distributed and implanted in an intracortical region to capture extracellular action potentials from the surrounding neurons.

Electrodes may differ in size, density, and material, whose characteristics will impact the recorded neural signals, mainly due to the electrode impedance [31].

Signals captured by the electrodes should be conditioned to be acquired in a proper amplitude for decoding purposes. A readout circuit performs this task.

2.3.2 Readout interface

The readout interface is the circuit responsible for adapting the signal amplitudes and converting them to the digital domain to be processed. In general, this circuit contains a low-noise amplifier for neural signals (A), an analog-to-digital converter (ADC), and a digital signal processor (DSP).

- **Amplifier.** This circuit is responsible for increasing the signal amplitude to be converted to the digital domain. It should be a differential and low-noise amplifier, which constitutes the analog frontend of the system and may be used with an intermediate block which could enhance some signal characteristics [34]. The amplifier power depends on the bandwidth, supply voltage, and noise level, so those factors may be of interest in reducing its power consumption.

- **Analog-to-digital converter (ADC).** It is the circuit block that converts the amplified signal into its corresponding representation in the digital world. These values will depend on the number of bits defining the ADC resolution, the times when the samples are taken related to the sampling frequency, and the reference voltage against the original signal that is compared and converted. All of these ADC characteristics will also impact on its power consumption. While they are more demanding, power consumption will increase.

- **Digital signal processor.** This block is optional and will be in charge of processing the digital representations of the signals samples to facilitate the following procedure to detect their characteristics or to reduce the data density to be transmitted.

- **Wireless transmitter.** Depending on the kind of BCI system, it will have a wireless or wired transmission system to the hardware/software component that process, extract and decode the movement intention. The power transmission in

wireless systems will depend highly on the bit-error-rate (BER), channel number, and data rate. The transmission frequency and data transmitted will impact how energy is dissipated in this circuit.

2.3.3 Preprocessor

This stage is commonly constituted by a bandpass filter and a threshold crossing. The first component is required to prescind from higher frequencies out of the band of interest and low frequencies, which corresponds to the local field potentials (LFP), giving a bandwidth in a range of 300Hz to 5kHz [35] or 7kHz [6]. On the other hand, threshold crossing is the common method to detect spikes in outputs a binary signal indicating the corresponding position of such spikes in the time. Some developments employ a previous block to the threshold crossing detector, which aims to enhance the signal characteristics to facilitate or improve spike detection. One of these methods is the Non-Linear Energy Operator (NEO), which emphasizes the amplitude-energy variation enhancing the spikes' amplitude.

2.3.4 Decoder

This stage estimates the user's intention from the threshold-crossing signal. It normally involves a process named spike sorting, which comprises: the spikes' alignment, responsible for aligning the spikes around a common point; the feature extraction, consisting of a dimensionality reduction; and the clustering, which groups the spikes with similar characteristics in clusters [13]. Those clusters will provide information regarding the movement intention and then apply that knowledge to a kind of assistive device.

From the described blocks, this thesis will be centered on the readout circuit, especially in the analog-to-digital converter, and in the preprocessing techniques to enhance spike detection.

2.4 Analog-to-digital converters

An analog-to-digital converter (ADC) is a type of signal converter that is responsible for translating an analog signal (continuous amplitude and continuous time signal) into a set of digital values (discrete time, quantized amplitude). Commonly, the amplitudes are relative to a voltage reference (higher than the maximum signal amplitude), while the times are related to the sampling frequency at which samples are taken, normally, at least double the signal higher frequency [36].

This circuit is constituted by four main sections, which are:

- **Antialiasing filter**, responsible for avoiding the aliasing when sampling the original signal at the sampling frequency.
- **Sampling**, which takes a sample employing switches and capacitors, which holds the voltage level during the conversion process.
- **Quantization**, responsible for converting the samples to a set of predefined fixed amplitudes, which will depend on the ADC resolution giving the maximum number of steps and the voltage reference against which the fixed levels are established.
- **Coding**, which gives each quantized sample a digital number according to the resolution and steps number. This digital value is finally the output of the ADC.

On the other hand, ADCs could have different implementations and depend on the sampling strategy [37], [10]. Then, the ADC can be classified as:

- **Fixed rate sampling**, in this case, each sample is taken at a fixed rate, so there is a correspondence between the sample number with the time. This strategy is the common implementation for ADC circuits. It can also be classified between fixed sampling at a Nyquist rate and oversampling, according to Maloberti [36]:
  - **Nyquist rate ADC**, which samples the input signal with at least twice the maximum signal frequency in agreement with Nyquist's sampling theorem. There are different types of this kind of ADCs, being the most common:
    - **Flash ADC**, known also as direct conversion. It is the fastest of the ADCs implementations and the most hardware overwhelming as it increases dramatically as the resolution number increase. A set of comparators and capacitors/resistors comprises it, directly comparing the sampled voltage against the different voltage levels generated by a voltage divider network.
    - **Interpolating ADC**. It converts through an integrator whose inverted input is connected to the input voltage (v_{in}). Then, it can ramp up for a certain predetermined run-up (t_1) time. After that time is finished, a known reference voltage (v_{ref}) of opposite polarity is applied instead of the input voltage, and then the output will ramp down until it reaches zero in a run-down time (t_2). Then, the quantized voltage could be calculated as

a relation of the two times as:

$$\frac{v_{\text{in}}}{v_{\text{ref}}} = \frac{t_2}{t_1}$$

It has the advantage of being highly accurate but extremely slow and unsuitable for audio or signal processing.

* **Pipeline ADC**, called sub-ranging ADC, has a high-resolution and better throughput. This ADC is implemented as a series of low-resolution flash converter blocks, which together achieve the full resolution for the pipeline ADC. In this case, each stage is responsible for converting the input voltage with the number of bits of its architecture. Then, the equivalent analog value to the current output is subtracted from the input voltage, which is, in turn, passed to the next stage as new input to achieve further refinement. Its advantages are the high throughput, better efficiency, and moderate sampling rate, while its drawback is the latency.
* **Successive Approximation ADC (SAR)**. It is one of the most common ADCs with multiple applications due to its low power consumption. Its architecture is simple and consists of an array of capacitors and switches, a comparator, and a register [38]. Its name comes from a "binary search" algorithm in a feedback loop used to reach the converted value [39]. It consists of a continuous comparison between the input signal and the DAC output as the count rises until the analog signal equivalent to the count is higher than the input, stopping the counter and retrieving its value as the converted output.

– **Oversampling**, these converters are used mainly for high-resolution audio bands, video bands, and currently, medium-resolution applications. This technique increases the sampling frequency further than the Nyquist value. The aim is to enclose the signal band in a small fraction of the full spectral range obtained by the sampling, enabling digital cancellation of quantization noise outside the band of interest and reducing it significantly. The most common ADC type which uses oversampling is the Sigma-Delta ADC [36].

* **Sigma-Delta ADC**, it employs a Sigma-Delta modulator that integrates differences between the signal and the converted output, providing the important characteristic of noise shaping, enhancing the oversampling ben-

efits. This converter usually has higher resolutions than SAR ADCs but with lower output frequencies.

- **Level-crossing sampling**, this technique is a kind of event-driven or event-based sampling. It is a non-uniform sampling method that only takes samples when the input changes enough to cross one of the predefined levels defining its amplitude resolution. This approach can benefit various applications characterized by sparse signals in time, such as in intelligent sensor networks or wearable, implantable, and digestible biomedical devices. The level-crossing technique compares the input signal with discrete amplitude levels and takes the samples when the signal crosses such levels. Then the maximum sampling frequency will depend on the ADC resolution (number of levels). Although there are other strategies for event-driven conversion, the level crossing is the most common and has been evaluated in different studies, as presented by Tsividis [40].

In this thesis, we will apply this type of converter to the extracellular recordings will be the level-crossing ADC (LCADC).

2.4.1 Converter specifications

Several metrics and specifications may describe the ADC operation and characteristics, and they can be of the following categories, according to Maloberti [36]:

- **General features**. These characterize the ADC properties hastily, and they are:
  - **Type of Analog Signals**. Determines if the input or output is single-ended (common ground), pseudo-differential (symmetrical to a fixed reference), or differential (not necessarily symmetrical, but operates with the difference between inputs or outputs).
  - **Resolution**, is the number of bits used in the analog output representation. It determines the minimum detectable voltage by the ADC, together with the reference voltage.
  - **Dynamic range**, is the ratio relative to the noise level of the largest signal level the converter can handle. It determines the SNR.
  - **Absolute maximum ratings**, are referred to the limits for normal operation of ADC. Such values can be classified as environmental (temperature range,

chip temperature, etc.) and electrical (voltage range, currents, impedance, etc.)

  - **Drift**, is the parameter variation (gain, offset, etc.) over a specified temperature range characterized by a drift temperature coefficient and a drift voltage coefficient.

- **Static specifications**. They measure how much the conversion for the ADC range of operation is close to or differs from the ideal static characteristic, a uniform staircase covering the full dynamic range. The most commonly used static characteristics are, according to Razavi [39]:

  - **Differential nonlinearity (DNL)**, is the maximum deviation of the step size obtained with the evaluated ADC, relative to the ideal staircase width corresponding to two consecutive digital codes.
  - **Integral nonlinearity (INL)**, is the maximum deviation of the evaluated converter transfer curve, scaled and with an adjusted offset, relative to the ideal transfer curve obtained by connecting the midpoints of each conversion step, building a straight line until reaching the endpoint.
  - **Offset**, which describes the shift for zero input. It changes the transfer characteristics so that this value shifts all quantization steps.
  - **Gain error**, which is the variation of the straight line slope that interpolates the transfer curve, relative to the expected value.

- **Dynamic specifications**. Which are related to the frequency response and the speed at which the ADC can work. The most common dynamic specifications are, according to Razavi [39]:

  - **Signal-to-noise ratio (SNR)**, which is the ratio of the input signal power (which should be known, usually using a sinusoidal signal) relative to the total noise power measured at the output.
  - **Signal-to-(noise plus distortion) ratio (SNDR)**, is the measurement of the ratio between the signal power under test and to the noise power including harmonic components at the output. It differs from the SNR, as it doesn't consider the generated harmonics in the noise count.

– **Effective number of bits (ENOB)**, is an indicator of the maximum number of bits obtained with a converter, considering the noise and dynamic range. It comes from the SNDR value. The next equation defines it:

$$\text{ENOB} = \frac{\text{SNDR}_\text{p} - 1.76}{6.02} \tag{2.1}$$

where SNDR_p is the maximum SNDR obtained with the converter, measured in decibels.

– **Dynamic range**, is defined by the ratio between the maximum and minimum values the ADC can measure. As described by Razavi [39], it can also be interpreted as the ratio of the full-scale sinusoidal power and the sinusoidal power that retrieves an SNR of 0.

These specifications will be considered in the ADC design process as part of the signal acquisition chain proposed in the thesis.

2.4.2 Architecture description

ADCs have different architectures according to their type of implementation. In the case of fixed-rate sampling, the most common and widely applied is the SAR architecture. This architecture is also employed in further analysis for power consumption in neural acquisition [6]. In this case, the thesis will focus on the level-crossing sampling method with the LCADC architecture. So, here we will discuss both SAR and LCADC architectures.

2.4.2.1 SAR architecture

The SAR ADC performs the ADC conversion by exploiting previously determined bit knowledge to determine the following one, according to Maloberti [36]. As he described, the SAR architecture comprises a sample and hold amplifier (SHA), a comparator, a controller, and a digital-to-analog converter (DAC).

The SHA block samples the input and maintains its value for successive clock cycles. The digital controller establishes the DAC output following the successive approximation algorithm. In this case, the SAR sets the most significant bit (MSB) to 1 and then compares if it matches the current value; if it is correct, it retains it; otherwise, it changes to zero. In the next clock cycle, the converter makes another prediction and follows the same criteria as the first comparison, setting the next bit. It proceeds the same way with the other bits until all have been determined.

The accuracy of this type of converter depends on the SHA, the comparator, and the DAC accuracy. So their deviations should be considered when designing this ADC and trying to minimize them.

2.4.2.2 LCADC architecture

A non-continuous sampling characterizes this kind of ADC. It only takes samples when the input changes sufficiently. Therefore, when a fast change occurs, the ADC samples quickly, and on the contrary, when there is no meaningful change, it doesn't take new samples.

As Van Assche and Gielen [10] presented, there are two LCADC topologies: an open loop topology and a closed loop. The first case comprises a comparator for each reference level, so it is unsuitable for biomedical applications. The close loop, on its side, depending on the architecture, could have one or two comparators, a DAC, a digital logic, and a time-to-digital converter (TDC). This topology has two variants:

- **Floating window LCADC**. Here, the DAC output is constituted by two consecutive quantization levels spaced one low significant bit (LSB). Then, the comparator will be triggered, and the logic adds or subtracts one LSB, if the level crossed is the high or low, respectively. If it has a DSP, samples could be processed directly; otherwise, a TDC saves the interval time between consecutive samples. For the next conversion cycle, the TDC restarts its counting after returning the digitalized time, and the DAC is updated to the new output levels, continuing with the conversion process.
- **Fixed window LCADC**. It is similar to the previous ADC; however, instead of updating the DAC at each conversion cycle, a node tracking the input is folded back to a common reference voltage. Here, the two voltage levels used by the comparators as references to trigger the level-crossing event determine the 1 LSB window.

Both architectures are equally valid for implementing the LCADC. In this thesis, both will be probed and evaluated against the neural signals to measure their performance.

2.5 Preprocessing converted signals

From the neural recordings, we aim to detect the spikes to align and sort them to extract their features. However, neural signals and their spikes are affected by noise, so detecting

them is not an easy task when noise interferes with and masks the actual signal. In this scenario, it requires a kind of preprocessing to improve spike detection. One of the most used techniques to detect the variations produced by the spikes is the Non-linear energy operator (NEO).

Proposed decades ago, the NEO calculated the instant energy for a signal. It is known as Teager's Algorithm and was published in 1990 by Kaiser [41]. It has both continuous and discrete versions. In the continuous case, it is:

$$\psi[x(t)] = [x(t)]^2 - x(t)x''(t) \tag{2.2}$$

While, in the discrete domain, it is:

$$\psi[x(n)] = x^2(n) - x(n+1)x(n-1) \tag{2.3}$$

As described by Mukhopadhyay and Ray [42], NEO operator has been used to estimate instantaneous frequency and amplitude in sinusoids and to detect discontinuities, before been applied to the detection of spikes in biomedical signals, suggested first for signals from electrocardiography (ECG).

Then, this operator was applied to neural signals to detect spikes due to its properties to enhance small variations in the signal of analysis and its extremely low computational complexity.

There are examples of its implementation in the analog domain [43], with its variations considering the current instead of voltage [44] or alternatives to it via squared-based non-linear filter [45].

In the digital domain, there are implementations for this operator [12] and with a slight variation related to the reference samples as presented by Zamani, Jiang, and Demosthenous [13]:

$$\psi(n) = x^2(n) - x(n+\omega)x(n-\omega) \tag{2.4}$$

This implementation is referred to as ωNEO or kNEO. Zamani, Jiang, and Demosthenous mention that it provides some flexibility when the spikes have low-frequency components, increasing the robustness to amplitude variations and reducing sensitivity to noise out-of-band.

The resulting signal is compared with a threshold level after the NEO application to the recordings. Then the samples over these thresholds will correspond to the spikes of the neural signal.

This thesis will consider the ωNEO to evaluate its performance under the different proposed scenarios.

2.6 Feature extraction and decoding overview

Once the spikes are detected, an alignment process occurs. It positions all the detected spikes relative to a common temporal point, taking as a reference the patterns in the spike waveform. The next step is classifying them to match the specific waveform to its corresponding neuron. These stages are enclosed in a process known as spike sorting. It aims to obtain the single-unit activity from a signal that has multi-unit activity [46].

The spike sorting process comprises some steps after spike alignment, which can be grouped into the following:

- **Feature extraction**. It commonly uses the principal component analysis (PCA). As described by Gibson, Judy, and Marković [46], the method aims to obtain the principal components by performing an eigenvalue decomposition of the covariance matrix of the data. Although its higher computational requirements, it is the most trusted and used until now. However, when the process is required to be implemented on a chip, other solutions are used, such as the discrete wavelet transform (DWT), discrete derivative (DD), or the first and second derivative extrema (FSDE), as presented by Hao *et al.* [47].

- **Dimensionality reduction**. This step has the purpose of reducing the memory and computational requirements of the clusters. In addition, it has other benefits, such as lowering the data rate of spike sorting hardware and improving the clustering accuracy. As explained by Gibson, Judy, and Marković [46], one of the best strategies to reduce the dimensionality is finding the features with multimodal distributions across spikes, as they are indicators that exist more than one spike collection in the data set.

- **Clustering**. It is the most challenging stage of the sorting process and consists of identifying the frontiers that separate a group of spikes from another. Some employed algorithms to automatically perform this process are k-means clustering, valley-seeking clustering, superparamagnetic clustering (SPC), and Osort clustering [46].

The spike sorting process outputs the spike timestamps for each identified neuron. This data will be used in the decoding, which is the following step in the BCI system.

The decoding stage is almost the final process that works directly with the spikes' information. Depending on the application, it could apply various strategies based on the identified spike trains and their corresponding times and individual neurons to match

them to the subject's intended movement. For example, in a recent development for high-performance brain-to-text communication using a BCI system, presented by Willett *et al.* [17], the decoding stage takes the results from the first three principal components, recognized after the participant attempted to write each alphabet character. Based on the trained information with a recurrent neural network (RNN), it converts the neural activity, represented by the detected spikes, to principal components and probabilities, writing a character at each moment.

Although these stages (feature extraction and decoding) are the closest to the practical application for this kind of extracellular recordings systems, this step is out of the scope of this thesis. It is only presented as a reference to highlight the required next steps before using the proposed system in a specific application.

2.7 Extracellular signal synthesis

To evaluate either the acquisition systems or the algorithms to decode neural recordings, it is mandatory to have test signals. We can obtain them directly from a clinical trial or from signals from other laboratories and research groups. Although the first strategy offers the most realistic waves, it has strict requirements to set up the trial and record the signals. Therefore, many studies prefer the second strategy, performing steps to add noise and generate the required spike trains. However, both methods have some inconveniences related to ground truth signals, as they do not know when the spikes are triggered.

Another alternative is to generate synthetic signals from neuron models and to simulate their variations due to position and noise to create realistic neural recordings. Following this methodology, there is an interesting project to generate extracellular action potential signals, which is named MEArec.

2.7.1 MEArec

As described by Buccino and Einevoll [48], MEArec is an open-source neural simulator written in Python, constituting a versatile framework to generate extracellular recordings. It has plenty of configurable options to include several phenomena that occur is in a real environment, such as bursting, drifting, noise effects, etc.

Their source code is on its author's GitHub repository `https://github.com/alejoe91/MEArec`, and its Python package is on PyPI, with extensive documentation.

This project is supported by other open source projects, which are:

- NEURON. It is a simulation environment for models of individual neurons and a network of neurons closely related to experimental data. It has been used in many scientific publications to simulate networks with thousands of neurons, as Carnevale and Hines [49] showed. It is integrated with Python and has its own package in the Python environment named neuron. Its repository on GitHub is `https://github.com/neuronsimulator/nrn`

- LFpy. It is an open-source package for Python that makes numerical simulations of extracellular action potentials. The calculation of such potentials is made in two steps. According to its author, Lindén *et al.* [50], the first one consists of the calculation of transmembrane currents of each neuron using models from NEURON. On its side, the second one performs the calculation of extracellular potential from transmembrane currents. Its repository on GitHub is `https://github.com/LFPy/LFPy`.

- SpikeInterface. It is an open-source unified framework that aims to help in the standardization and accessibility of new sorting technologies by enclosing the steps required in the spike sorting pipeline. As stated by its author Buccino *et al.* [51], it could encourage sharing datasets, results, and analysis, and supply a set of benchmarking capabilities available for future usage or development. Its repository on GitHub is `https://github.com/SpikeInterface/spikeinterface`.

In MEArec, the simulation is performed in two stages: the template generation and the recordings' generation.

- **Templates generation**. The templates are generated using realistic models of the neurons positioned around an electrode model. This stage has two steps intracellular and extracellular simulation. As described by its author Buccino *et al.* [51], the first step simulates the neuron model using the NEURON simulator. Then, the second step computes the extracellular potentials at the electrodes' position employing the LFPy software.

- **Recordings generation**. These are generated by combining templates selected with predefined conditions and simulating the spike trains. Buccino *et al.* describe that these templates and spike trains are assembled using convolution, replicating some events of spiking activity, such as bursting and drift. Finally, the software can generate different types of noise, which can be added to the resulting recordings, and

optionally includes a filter to suppress high frequencies or limit to a specific band interval.

This software has some interesting features, such as: considering a multi-electrode array for the recordings, with different available models; implementing the bursting effect (sort of rapid train of action potentials); controlling the spatio-temporal overlaps; implementing the drifting effect; modeling experimental noise by adding different types of noise (uncorrelated Gaussian, spatially correlated, and far neurons noise); and providing a testbench for spike sorting development.

It is a promising development for recording generation for testing purposes, and this thesis will use it for recording synthesis.

Chapter III

Methodology

This chapter describes the methodology followed to achieve the objectives proposed at the beginning of the document. It considers the preliminary steps to generate signals to evaluate the designs, the design of LCADC and its evaluation, the NEO combined with a lower bandwidth filter and its evaluation, and finally, a power consumption evaluation. These stages are summarized in Figure 3.1.

3.1 Simulation of extracellular action potentials recordings

For evaluation purposes, it is mandatory to obtain test signals. There are several approaches how to obtain them, via experimentation, open datasets, or simulation.

Even though experimentation is the optimal way to obtain more realistic waveforms, it has several inconveniences such as going through a clinical procedure and to have a commercial acquisition system to obtain such recordings. Therefore, this project can not follow this alternative as it doesn't have the logistics, permissions, and funding to set up those experiments. Then, the affordable method to work with those signals is via datasets released by other research studies or simulations.

In the case of the datasets, several projects have used them, which released those to the scientific community. It also leverages the development of new techniques in the processing stage based on a common data bank and generates other signals, taking them as seeds. However, they have the drawback of not providing their characteristics immediately, as they must be processed to extract them. Hence, it will depend on the accuracy of the processing task employed, and the recording's fidelity.

On the other hand, although simulations have been a matter of interest for researchers, it hadn't been used extensively due to some constraints to simulate all the effects that could

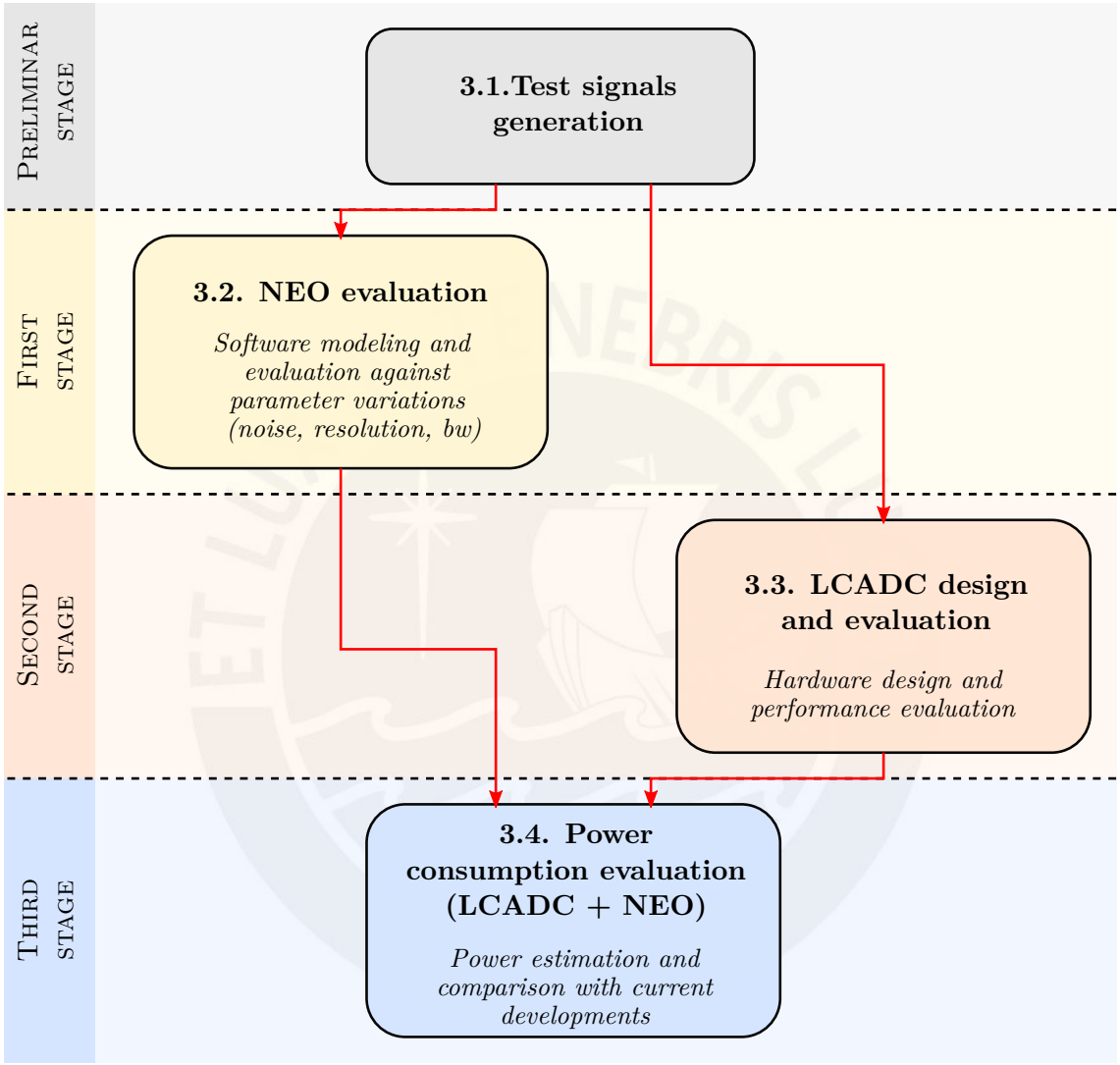


Figure 3.1: Scheme of the methodology flow process. Each subsection in this chapter corresponds to a block in the methodology flow. It starts with a preliminary stage called test signal generation, which are inputs for the NEO and LCADC evaluation in the first and second stage. Finally, results from both are inputs for the power estimation for the whole system and its comparison with current developments.

happen in a real environment and generate multiple scenarios considering the cell models and their interactions. However, recently have appeared a simulation tool which allows for establishing those characteristics and also include phenomena such as uncorrelated noise or the produced by far neurons. This software is named MEArec [48], and it will be used for this thesis.

In this stage, the following steps are required:

- *Simulate the extracellular recordings employing MEArec software.* By default, MEArec is configured to generate recordings taking as a base some cell models from the Neocortical Microcircuit Collaboration Portal (NMC), from which it generates a set of templates for the extracellular recordings and finally obtain them in the simulated electrodes combining those templates and simulating their interaction. The resulting recordings have a minimum spike amplitude of 50 µV and a maximum amplitude of 500 µV with uncorrelated noise of 10 µV level.
- *Obtain several datasets with different noise levels.* Based on the previous result, we test with other noise level values. Considering a maximum level of 500 µV, the noise levels chosen will be 5 µV, 15 µV, 20 µV, 25 µV, 50 µV, and 75 µV.

Next to having the required samples, it is possible to follow with either the LCADC design or the NEO implementation. In this case, we will start with the NEO evaluation, as it also allows testing sampling characteristics for the ADC and evaluate their performance.

3.2 NEO implementation and evaluation

The non-linear energy operator is a technique employed to maximize the differences and changes between samples, being extensively employed as a preprocessor to improve spike detection [12], [13] even at the analog stage [44]. This thesis will test this technique under other constraints, such as lower bandwidth, lower resolution, and non-continuous sampling. We propose a set of stages to evaluate its performance, comprising the task of modeling and implementation of the NEO preprocessor and its evaluation under different signal tests.

3.2.1 NEO modeling and implementation

Being an algorithm to process sample data, NEO can be modeled in a programming language to apply it to a set of values. In this case, the modeling task will be implemented

in Python in concordance with the simulation software employed to generate the signal test.

3.2.1.1 Modeling analog to digital converters

Samples obtained from simulation have microvolt values (up to 500 µV), so it is possible to model the acquisition of these values from an analog to digital converter. The modeling considers the characteristics for each converter type, one with the common continuous acquisition frequency and the other with a cross-level acquisition system. Then the required steps for this are detailed as follows:

1. *Modeling of continuous analog to digital converter.* In this case, it evaluates the maximum number of step values according to the resolution and compares the current value with the proximal step number. Here, it is considered a resolution of 12 bits (as a common value used in other acquisition systems [5]) and 8 bits as the proposal, which is considered in this thesis to reduce this value.

2. *Modeling of non-continuous (cross-level) analog to digital converter.* In this case, the values stored in the array should only be converted if the difference with respect to the previous sample is greater than the step value obtained from the converter resolution. Then, unlike the continuous converter that has a corresponding value for each sample in the array, the non-continuous only stores the non-equal values at non-continuous indexes. For this kind of converter, it is also required to store the indexes of the original array to obtain the corresponding times for each sample. To compare with the previous sampling method, it also considers the same two values of resolution 12 and 8.

Once we model the converters, we evaluate the NEO performance employing the resulting samples.

3.2.1.2 Modeling NEO

The NEO is modeled according to their equation (3.1) and applied to an array of values, which may be the extracellular recordings:

$$y(n) = x(n)^2 - x(n-1)x(n+1) \tag{3.1}$$

Although the NEO compare values in contiguous samples, here, considering the modification proposed in [13], it will introduce a width for the samples against current sample is compared, as presented in equation (3.2):

$$y(n) = x(n)^2 - x(n-w)x(n+w), \qquad w \in \mathcal{Z} \text{ and } w > 0 \tag{3.2}$$

Then, in this case, the widths will be: 2, 4, 8, 16, to evaluate the effect for distant samples and their performance in the spike detection.

3.2.1.3 Modeling Threshold Detector

An important step before the processing stage to decode movement intentions and develop the applications based on intracortical signals is spike detection, which will serve to sort the spikes and extract characteristics. The simplest approach to detect those spikes is to set a threshold level over which a spike may occur. This threshold is not necessarily a fixed value and will not be the same for original data compared to a NEO block's converted or pre-processed data. So, it is required to take an approach to set this threshold, taking into account the signal characteristics.

In this case, the reference to establishing the threshold value will be the maximum value in the data set and evaluated against fractions of it, ranging from 0.1 to 1. The threshold evaluation uses the same data for each case to obtain the number of samples considered as spikes.

3.2.2 NEO evaluation

The evaluation of NEO considers the application of threshold over modified data to detect the spikes and compare them against the original sampled data without this preprocessor. In addition, some specific characteristics may influence in the ability of NEO to improve or worsen the performance of this detection.

The following experiments have been designed to evaluate those variants:

1. *Experiment 1: Evaluation with original recordings.* It considers the original simulated values obtained from the previous stage, which were obtained at 32kHz and with different values of noise. The NEO operator will be applied to this data as well for the different values of width, and for all of them, we will apply the set of thresholds to identify the number of spikes detected and compare it to the data with no preprocessor.

2. *Experiment 2: Evaluation of lower bandwidth.* In this case, it considers sampled data with full bandwidth, which according to the simulator, ranges from 300 Hz to 6 kHz, and applies the NEO and threshold evaluation to it. Then, modify the recording with a filter to reduce the bandwidth to 300 Hz to 1 kHz, as proposed by [8]. Afterward, compare both results to the spikes detected for each threshold level.

3. *Experiment 3: Evaluation of reduced resolution.* In this case, we will reduce the sampled data resolution to 8 bits, applied to the NEO preprocessor, and then tested with the thresholds. We will compare this result with the original NEO results with 12 bits of resolution for sampled data.

4. *Experiment 4: Evaluation of non-continuous sampling.* The LCADC has the characteristic of only saving data when the difference crosses a voltage level. This fact conduces to a non-continuous sampling mode. Then samples obtained from the model of LCADC will be applied a NEO preprocessor and again evaluate the threshold results. This will be compared with the original continuous sampling data recording.

5. *Experiment 5: Evaluation of combined effects.* Combine the effects of non-continuous sampling, lower bandwidth and lower resolution, and apply a NEO preprocessor, comparing it to the original continuous sampling, full bandwidth and 12-bit resolution samples with preprocessor and without it, against the set of thresholds to determine the number of spikes detected for each case.

By this evaluation results, the thesis will afford the second objective related to the NEO effects regarding spike detection accuracy.

3.3 LCADC design and evaluation

The level crossing ADC (LCADC) architecture is chosen to implement the low-resolution converter. This architecture has promising results for lower resolutions achieving low power consumption compared with common SAR architecture, as presented in [10] due to biosignal time sparsity.

Although their advantages have been proved for general extracellular signals with no detection or encoding purposes, this thesis is going to test the feasibility of employing this architecture as part of signal acquisition for decoding movement intention. So for this purpose, the system should be able to detect accurately the spikes from which it can

decode those signals. As a figure of merit to evaluate the quality of conversion to achieve this purpose, it will be employed the SNR and SNDR compared with a common SAR architecture.

Ideally, the proposed designs could be manufactured and tested experimentally; however, integrated circuit fabrication is extremely expensive, and testing the manufactured device requires a special environment and equipment. In this case, as this project does not have that budget, the test will be limited to the simulation stage.

To evaluate it, the LCADC will be designed and simulated using a mainstream Electronic Design Automation (EDA) software, Cadence, for the technology of 180 nm TSMC, which also allows the simulation of some realistic effects due to fabrication introducing some dimension's uncertainty.

3.3.1 LCADC design

The LCADC, in contrast to SAR, requires some special blocks and logic to detect the variation of sampling against a window and proceed to convert to their digital form. It also demands a timer to store the timestamps with a corresponding digital sample.

On the other hand, both architectures share blocks such as the comparator, which is an essential part of the conversion, and DAC which will convert digital values to their analog counterpart, allowing the comparison with sampled values.

Each component design will demand its requirements, which are described as follows:

3.3.1.1 Comparator design

A critical part in the LCADC that demands a minimal offset to obtain accurate comparisons and lower power consumption. Synchronous ADCs commonly employ the Strong-Arm architecture. However, as the LCADC is asynchronous, we cannot use it as we need to know when the signal will cross the level. So then, we must optimize a static comparator to have a minimal offset and delay with reduced power consumption.

3.3.1.2 Switch design

This element is part of the digital-to-analog converter (DAC) employed for the digital sample conversion. It presents issues that must be addressed, such as current leakage, which may produce an incorrect operation. Recommended architectures are the charge pump and bootstrap.

3.3.1.3 Digital to analog converter

This component is responsible for regenerating the digital converted value to the analog domain, to be compared with the signal and evaluated if the current signal value must be sampled. This block has an array of capacitors for low-power design and the corresponding switches that may be obtained from the previous design. It must also be evaluated against the resulting analog value to determine the output accuracy of the desired values.

3.3.1.4 Time-to-digital converter design

It is timer that accurately measures the time between each sample. It should also be designed for low power and may consider an oscillator and a set of registers to store the count.

3.3.1.5 Controller design

It is the digital logic behind the conversion responsible for controlling the comparator and the time-to-digital converter. It is comprised of a set of D-flipflops that implements the required logic. It must also be evaluated concerning the speed and power consumption.

3.3.1.6 Modeling SAR architecture

This architecture won't be implemented in the simulation software but modeled employing the Verilog-A language, compatible with the Cadence software. This model will serve to compare the performance against the designed LCADC converter.

3.3.2 LCADC evaluation

ADCs have some figure-of-merit which may be quantified by measuring the signal-to-noise ratio (SNR) and the signal-to-noise-and-distortion-ration (SNDR). In addition, other characteristics such as linearity and ENOB are also important to evaluate their performance.

In this case, those evaluations can't be experimental, as all the designed circuits are intended to reach up to simulation, so the prepared evaluations are intended to use tools from the EDA software and other programming features.

For this, the following experiments are planned:

1. *Experiment 1. Evaluation of ADC performance in terms of SNR/SNDR.* This evaluation takes as a signal test a sinusoidal waveform, as it only has a unique component in the frequency domain. Then, when submitted to the converter, it should produce

a noise level due to the quantization and non-linearities (distortion). So, with these data, the SNR and SNDR could be calculated for both converters, the SAR model, and the designed LCADC.

2. *Experiment 2. Evaluation of ADC conversion accuracy and linearity.* For this evaluation, the signal test is a linear signal which may evaluate the accuracy of conversion and the response time for the conversion. It could be programmed for different slopes to test the speed of conversion. This experiment will be applied to both the SAR model and the LCADC design.

3. *Experiment 3. Evaluation effect on simulated recording signals.* Once the performance of the original design is tested, it corresponds to evaluating it with the simulated extracellular recordings, so it can assess the effects on it. This effect could be measured as the signal-to-noise ratio variation relative to the input signal. This evaluation will also be applied to both the LCADC and the SAR model to measure the impact in both architectures.

Through these experiments, this thesis aims to approach the first objective, related to the LCADC design and its performance for being used as the converter component in the signal acquisition chain.

3.4 Power consumption evaluation

An essential evaluation for the proposed system is the power consumption of its components. As all of them will be designed and simulated, this estimation will use the tools provided by the EDA software to calculate these values.

3.4.1 Estimation of power consumption in LCADC from simulation

This estimation will be calculated from the simulations executed in Cadence as the product of current demanded by the circuit from the source and the voltage of operation. It can also be helpful to identify the consumption of each component of the LCADC architecture to detect which has more impact in such value.

3.4.2 Estimation of power consumption in NEO

This estimation will consider a model for the components of NEO in a digital design, which may include the number of flip-flops for multipliers and adders as well as registers

to save their values. This estimation also considers the voltage reference and the frequency of operation.

3.4.3 Comparison with consumption of state-of-the-art systems

From the review made for the theory, we will identify the most recent research works and commercially available products regarding power consumption for their converters measured by channel.

The values obtained will be compared against the power consumption estimated from the LCADC and the power consumption for the NEO block to assess the impact of employing both components in the whole system consumption.

Once these tests and comparisons are completed, this thesis could approach the third objective, which may decide if this proposal could be energy efficient relative to other acquisition systems available in the literature or the market.

Chapter IV

Design and implementation

This chapter discusses the hardware design and software implementation, following the process depicted in the methodology. As the first objective is related to the NEO evaluation, we model and implement a test framework to perform the evaluation tasks. Then, we explore the various design criteria and processes for the second objective related to the ADC design. And finally, regarding the consumption, we detail the power estimation for both circuit blocks.

4.1 Test framework for NEO evaluation

The test framework is a set of software tools developed to deploy the different experiments established in the methodology for the NEO evaluation. This software is implemented using a programming language and integrated with the recording signals generator software to perform such evaluations.

4.1.1 Framework components

As commented in the previous chapter, synthetic extracellular recordings are used for all the test generated by the MEArec software. In addition to this, the framework should be able to emulate the ADC hardware and also the different processing and evaluation tasks required. It could be separated into components responsible for a specific job in the evaluation flow.

The software architecture is shown in Figure 4.1. The main components in this software are:

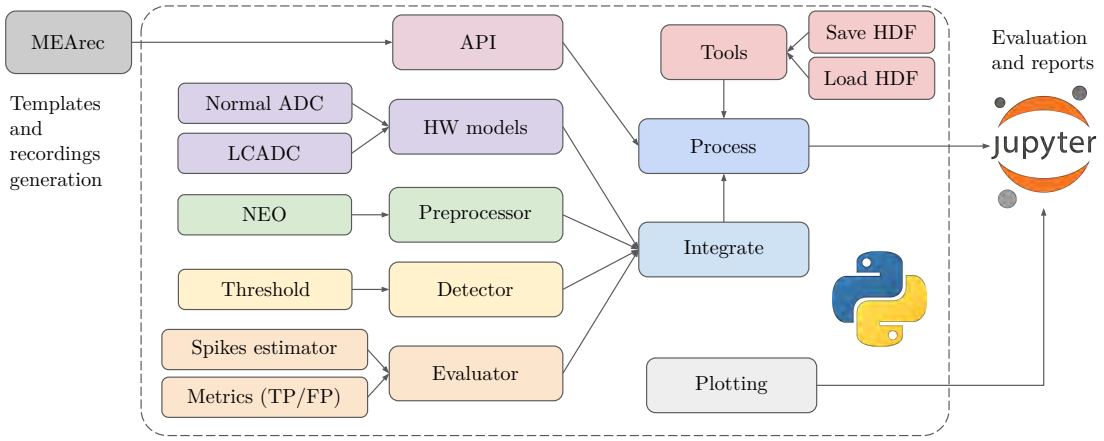


Figure 4.1: Framework components. The developed software has an interface to MEArec simulator to generate the recordings. Then, via a set of modules, the framework performs the operations required to convert the signals, apply the threshold and evaluate the results. The source code is written in Python, using Jupyter Notebooks as an interface to present the results.

- **MEArec application program interface (API)**. This is responsible for the intermediate between the framework and the MEArec software. It is prepared to receive the specific characteristics required by this evaluation and then call the MEArec methods to perform the task required.

- **Hardware models**. This component has the hardware models for the typical ADC and the LCADC. They are responsible for converting the array of samples obtained from the recording generator and convert them to digital values, emulating the behavior of each converter.

- **Preprocessor**. The NEO operator constitutes the preprocessor. This block implements and applies the NEO algorithm to both ADC architectures. It also considers the width variation for the samples taken in the NEO formula to evaluate them as a parameter.

- **Detector**. This block detects the spikes either in the recordings, the converted samples, or the NEO-modified signals. It is essentially a threshold-crossing level that obtains the samples over the threshold for predefined levels used in the evaluation tasks.

- **Evaluator**. This component has functions to estimate the spikes in the original recordings to the current timestamps and compare them against those identified by the detector block. This function also considers the kind of samples it has been

evaluating and if they come from continuous sampling or non-continuous sampling (LCADC).

- **Integrate functions**. This component is constituted by a set of blocks that aims to integrate each output from the previous block to the expected following block. For example, the incoming recording database to the hardware model conversion methods.

- **Process functions**. These functions implement the process of the tasks required for the evaluation, considering the integration functions and tools, which include the loading and saving of files.

- **Plot functions**. These functions aim to prepare the visualization plots required for the debugging and evaluation tasks. They range from other representations for the original recording signals, to evaluation representations, such as the receiver operating characteristic (ROC) curves.

- **Tools**. It contains a set of functions that helps other components, especially the process component, manage the files to load and save results from each evaluation task. The database file format is Hierarchical Data Format (HDF), the same format obtained from the recordings' generator.

All this software implementation is performed in the Python programming language. For the set of live tests, the Jupyter Notebook facilitates the tasks for reporting and plotting.

4.1.2 Evaluation process flow

The framework developed should be used following a set of continuous steps for each experiment which involves generating and evaluating the results. Figure 4.2 shows the set of steps required for the evaluation, and these are:

- **Recordings generation**. This step uses the API component to interact with MEArec and generate the recordings. It performs the following:

  - Generate the recording templates from the cell models obtained from the MEArec software and save them to the templates file.

  - Generate the recording signals from the generated templates taking different noise values and saving them to the corresponding files, one per each noise level/sampling frequency.

- **ADC conversion**. In this step, the recordings saved into files are passed to the converter model and, depending on the conversion type (normal ADC and LCADC), returns the corresponding converted samples. It uses the ADC component and performs the following:

  - Load the recordings saved previously and convert them according to the type of converter.
  - Normalize the signals (required for the NEO preprocessing) converted.
  - Save converted and normalized values to the corresponding files, one per each original file and resolution.

- **NEO preprocessing**. In this step, the preprocessing algorithm is applied to the normalized samples considering the variation for each width of adjacent samples for the NEO evaluation. It uses the preprocessor component and includes the following.

  - Load the converted values containing the normalized ones that should be the source for the preprocessor.
  - Apply the NEO preprocessor to the normalized values with predefined widths.
  - Save the NEO values in a new file with the information of the converted values and the recordings attached to it.

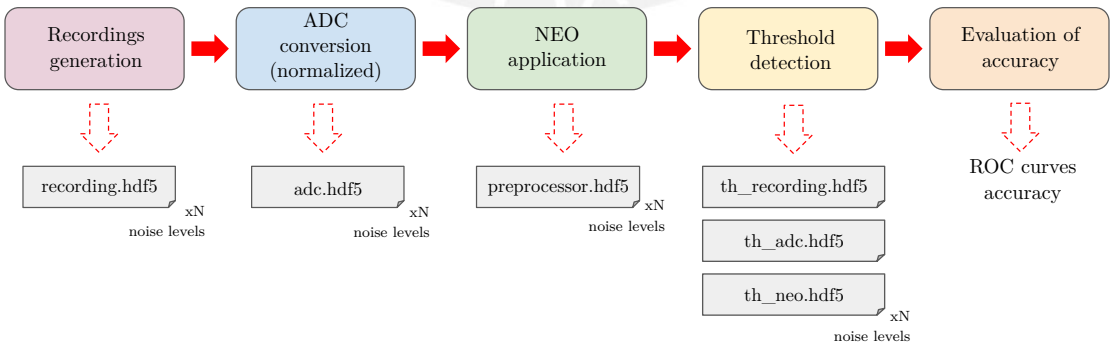


Figure 4.2: Evaluation process flow. The process is divided into five main steps, which cover from the signal generation to the evaluation stage. The first four stages have as output HDF files, whose number will depend on the parameter variation, such as the noise level.

- **Threshold detection**. This step applies the threshold level to the results obtained in recording, normalized converted values, and NEO values. It uses the detector component and includes the following steps:
  - Load the NEO files with the recordings' information, the converted and the NEO values.
  - Apply the thresholds to the recordings taking a fixed number of levels relative to the maximum amplitude in the signal.
  - Apply the thresholds to the normalized values taking the same fixed number of levels relative to its maximum amplitude.
  - Apply the thresholds to each NEO evaluated signal relative to each maximum amplitude.
  - Save the results in separate files as not all the operations could be performed in RAM due to the large data size.
- **Results evaluation**. In this case, the evaluation consist of taking the measurements for each case and comparing them with the quality of classification using the simple threshold. This step uses the evaluator component and performs the following steps:
  - Estimate the positions in the timestamps array of the original spike trains retrieved from the recordings files.
  - Take the spike trains selected in the detection stage and compare them against the results for each threshold level.
  - From the comparison, determine the number of true positives, false negatives, true negatives, and false negatives. Then, calculate the true positive rate (TPR) and the false positive rate (FPR) for each threshold level.
  - With the results of TPR and FPR, construct the ROC curve where each point represents the evaluation against each threshold level. Repeat the procedure for each signal to be compared among recordings and NEO.
  - Plot the results to identify the behavior of each type of signal and determine the variations according to the criteria established in the experiments.

4.2 SAR ADC design

The idea behind SAR ADC architecture is to approximate the sampled value via a series of successive approximations made by a DAC and digital logic. For this, the system has to perform several comparisons to obtain the closest representation for an analog input based on the knowledge of previously determined bits [36].

4.2.1 Building blocks

There is a general approach to implementing SAR ADC systems, consisting in several blocks, including a sampler and hold circuit, a comparator, an n-bit DAC, and its corresponding logic [36].

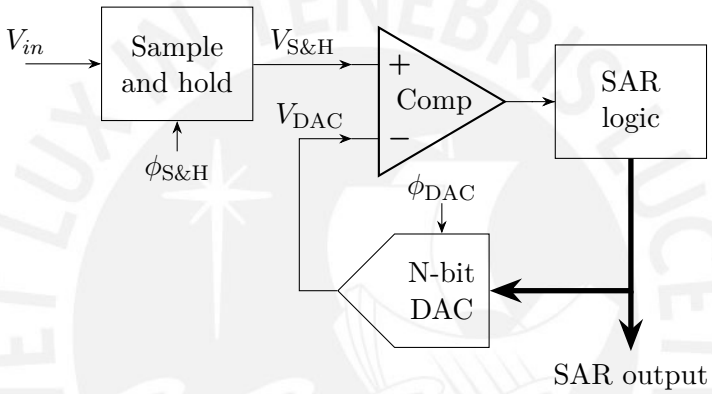


Figure 4.3: Basic architecture for SAR ADC. Although the hardware implementation could present some variations, it contains the same conceptual components which enclose the main idea behind this ADC architecture, where a sampled signal is compared multiple times to approximate its representative digital value. Adapted from [36].

Figure 4.3 shows a general scheme for the SAR architecture. Here, the input signal feeds a sample and hold block (S&H), controlled by a signal clock ($\phi_{S\&H}$). During the sample stage, it captures an instant voltage, while in the hold stage, it maintains this voltage to compare it with the voltage generated by the DAC. In this stage, successive comparisons occur where the sampled voltage is compared with different threshold levels generated by the SAR logic and the N-bit DAC. For each cycle of the signal clock that controls the DAC component (ϕ_{DAC}), the ADC obtains a bit of conversion, beginning from the MSB until it reaches the LSB.

The logic behind the SAR is to generate the threshold voltage V_{DAC} that a '1' state could produce for each bit. For example, if it is the first bit, the threshold voltage will be $V_{REF}/2$. Then, if the actual input is lower than this threshold voltage, the output is changed to '0'; otherwise will remain at '1'. For the next bit, the DAC adds to the

previous value a voltage equivalent to that bit being '1', which will be $V_{REF}/4$. This new voltage is compared to the input again and follows the same logic: if it is lower, it changes to '0,' and if it is greater, it remains to '1'. The process continues several times equal to the number of resolution bits considered in the design.

Although there are two clock signals in this scheme, one for the sample and hold and another for the DAC, generally, designs share the same clock signal. Here, we use the first half clock for the sample stage in the sample and hold block. In addition, while both blocks are depicted as separate entities, some designs include the sample and hold together with the DAC, which will be discussed later in the DAC topologies.

4.2.2 Comparator

It is one of the most relevant influencing in the ADC's precision. There are various alternatives, but we can differentiate between dynamic and static comparators.

- **Static comparator**. It is characterized because it is always active, constantly comparing the values in its inputs. That means that it has static power dissipation. It also constitutes the most prominent percentage of consumption in ADCs that use them.

- **Dynamic comparator**. It only performs the comparisons in determined time instants, and the most relevant power consumption is in the transitions. This architecture is preferred against the static comparator when the input comparison is required only in discrete times, as it happens in synchronous analog-to-digital conversion. The SAR has this type of conversion, and designs usually use this architecture for their comparators.

The most used architecture that implements dynamic comparators is the Strong Arm, proposed initially by Kobayashi *et al.* [52] and used for almost three decades until now. The functioning of this block is well explained by Razavi [53]. The three main reasons for its popularity are its consuming zero static power, its rail-to-rail output (other comparators need a final inverter stage to achieve this), and its offset comes, principally, from only one differential pair [53].

Figure 4.4 presents a model for the Strong Arm architecture. We can differentiate four phases in the functioning of this circuit [53]: the pre-charge when the clock is in a low state; the amplification, when the clock changes from low to high, activating the

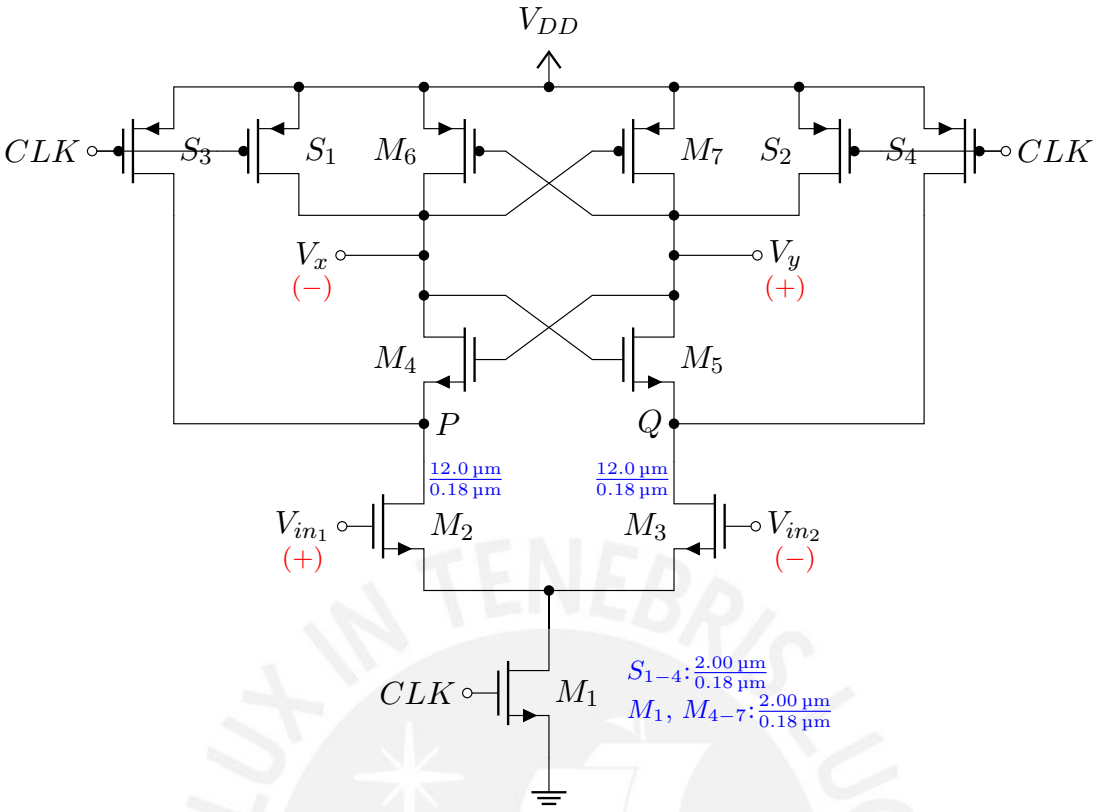


Figure 4.4: Strong Arm architecture for the dynamic comparator. It uses a clocked-based signal to evaluate the signals only in the clock transition in its positive edge. Based on [53].

differential pair; the regeneration phase, when after producing a difference in currents in the differential pair, produces a voltage difference in the outputs, as well; and the fourth phase, which speeds up the difference to the rail-to-rail levels.

Considering all the transistors of the same dimensions, we can have a starting point for the circuit design with the minimum value for the length $L = 180\,\text{nm}$ and width $W = 2\,\mu\text{m}$. After that we can adjust the values to make faster the amplification stage.

Simulating this circuit will return a plot like Figure 4.5. It shows the response for the variation of the differential voltage of 100 mV while the common voltage is 400 mV. The first advised characteristic is that the changes are only produced when there is a rising edge of the clock signal. As expected, when the difference $V_{in_1} > V_{in_2}$ the positive output (in Figure 4.5 out_p and Figure 4.4, V_x) results in '0' logical, while in the opposite case, this output results '1' logical. Defining the final output, we can see the tendencies of voltages in P and Q. If P is lower than Q, the output is '0' while, on the contrary, it is '1'. The delay times measured for both cases when the input $V_{in_1} > V_{in_2}$ and $V_{in_1} < V_{in_2}$ are around 15 ns.

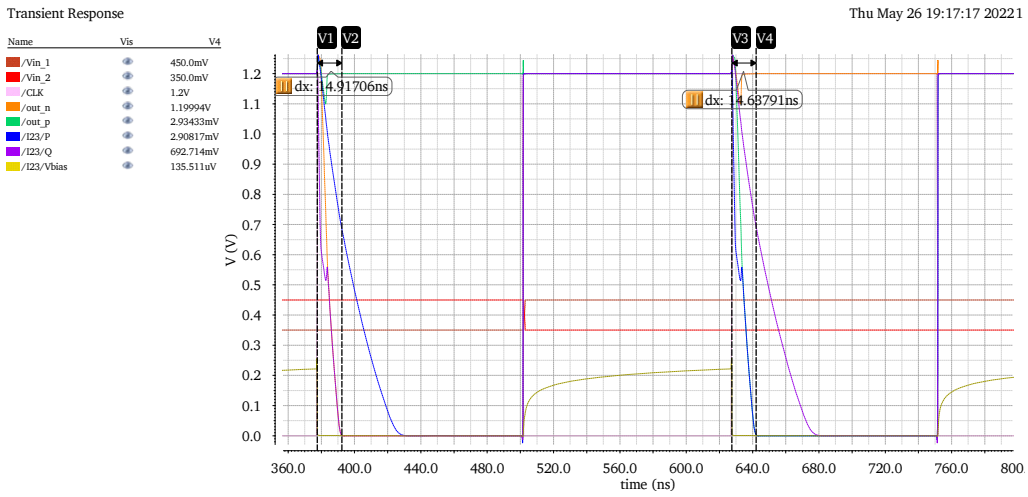


Figure 4.5: Transient simulation for the dynamic comparator testbench circuit. The common mode voltage is 400 mV, while the differential voltage is 100 mV. The figure presents the behavior for all the inner signals in the comparator.

One important thing to notice about the delay time is that they change depending on the differential voltage. By simulation, we obtain that the delay is more significant when the difference is lower; for example, it is approximately 45 ns for 1 mV of difference. This could be a bit higher than other designs [54]; however, it is enough for the required frequency we will use, whose period is around several microseconds.

Likewise, we can appreciate that if the clock signal is high ('1'), both outputs keep their values, while when the clock is low ('0'), both are V_{DD} or '1' logical. This is problematic as any comparator design will require that the output keeps its value until the next clock-rising edge. That implies the addition of a block that accomplishes this function. This block is a latch. First, we add an inverter to each output in the comparator, acting as a buffer. The latch is activated only when V_x or V_y falls to '0', as when they change from they default value in an idle state. It sets that value ('0') to the corresponding output, while the inverted value is set to the complementary one, ignoring the value in another terminal in the comparator. Consequently, when $V_{in_1} > V_{in_2}$ returns 1 and $V_{in_1} < V_{in_2}$ returns 0, with V_{in_1} acting as the non-inverting input and V_{in_2} as the inverting output. The schematic for this circuit is presented in Figure 4.6. The dimensions for the transistors are similar to the comparator.

Simulating this circuit, we obtain Figure 4.7. The purpose is to measure the offset. Here, we use a CLK period of 100 ns and a differential input that ranges from -1 mV to 1 mV. The rising time for the differential input (which acts as a ramp) is set to 10000

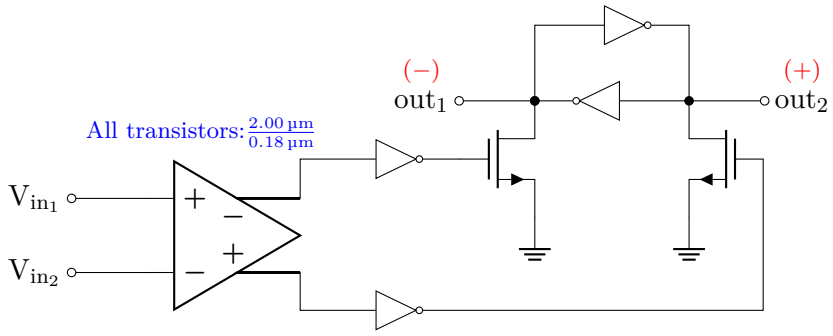


Figure 4.6: Dynamic comparator connected to latch circuit. This complete block assures the comparison output holds its value until the next rising edge clock.

times the CLK period, giving a resolution of 2/10000mV equal to 200 nV. The result shows an offset of 410 nV.

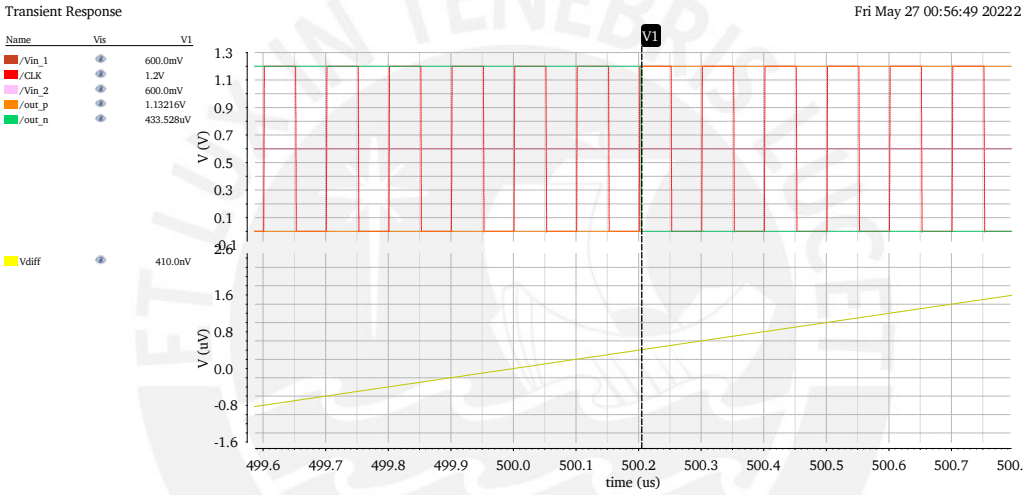


Figure 4.7: Transient simulation for dynamic comparator testbench circuit and parametric value of differential voltage. The upper plot shows signal result in a time range around the output transition. The lower plot shows the difference between the inputs for VDIFF. The vertical marker is positioned in the time where the output is 0.6 V, and the VDIFF value corresponds to the offset.

However, the mismatch in the comparator affects the offset considerably due to the variations in the differential pair M_2 and M_3. To obtain a more realistic value for this quantity, we consider this mismatch via a Monte Carlo simulation, configuring its corners properly. Figure 4.8 shows this simulation. The setup is similar to the previous simulation, with the difference that the offset value is calculated multiple times considering variations in the physical dimensions of the transistors. To speed up the simulation, we use a VDIFF value of 20 mV with a resolution of 40/100mV (rising time equal to 100 times the CLK period), which results in 400 µV. According to the results, the mean value is 211.3 µV, while the standard deviation is 6.28 mV. This demonstrates that, although the

ideal value when no mismatch is present, could be as low as 410 nV, it is higher due to the dimensions' uncertainty. In the present case, we could predict that voltages below the standard deviation may present inconsistencies that affect the conversion's precision, specially in the last bit.

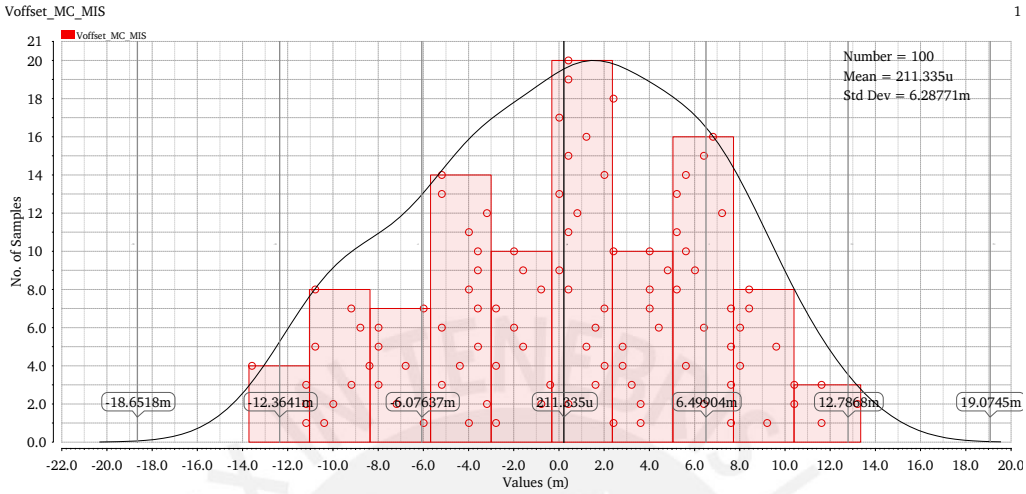


Figure 4.8: Histogram for offset with Monte Carlo simulation. This figure shows offsets distribution for variations in transistors' physical dimensions. The vertical lines mark the standard deviation σ and multiples around a mean value (211.3 µV).

4.2.3 Sample and hold

This circuit keeps stable the signal input while the reference voltage generated in the DAC approximates the analog sample.

The simplest circuit for this block is a switch and a capacitor that samples the analog input in one stage and then keeps the voltage as charge in the capacitor while the rest of the circuit performs other tasks. The scheme is presented in Figure 4.9.

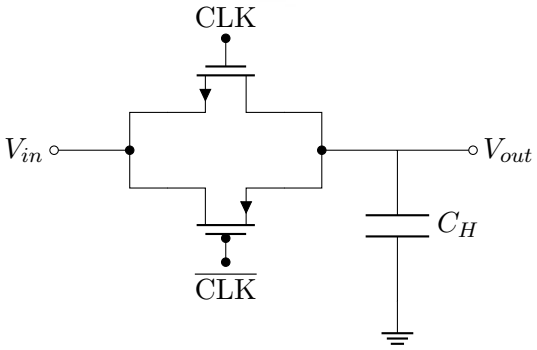


Figure 4.9: Sample and hold circuit implemented with a complementary MOS switch followed by a capacitor. The control signal is represented as CLK, usually a square periodical signal. $\overline{\text{CLK}}$ represents the negative value of CLK. C_H is the capacitance of hold, which stores the value from V_{in} as V_{out}. Based on [55].

To obtain the dimensions for the switch-capacitor circuit, we need to consider that the settling time should be lower than the clock period, and the final voltage error should be lower than an LSB. We can consider the circuit as an RC block, for which the output voltage is determined by:

$$V_{\text{out}} = V_{\text{step}}(1 - e^{-\frac{t}{\tau}}) \tag{4.1}$$

where V_{out} is the output voltage after the switch control is enabled, V_{step} is the difference between the input and the previous stored voltage in capacitor, τ is the constant time, that is equal to the product of the switch resistance (R_{sw}) and the hold capacitor, resulting in $\tau = R_{\text{sw}}C_H$.

We can rewrite the equation (4.1) to obtain the time in function of the voltage error:

$$t = \tau \ln\left(\frac{V_{\text{step}}}{V_{\text{step}} - V_{\text{out}}}\right) = \tau \ln\left(\frac{V_{\text{step}}}{V_{\text{error}}}\right) \tag{4.2}$$

In a worst case, we can consider a voltage change equal to the reference 1.2 V and an error equal to 0.5LSB. As LSB is the reference divided by the 2^8, then the required time is $t \approx 6.238\tau$ which should be lower than half of the clock period (sampling pulse) $T_{\text{samp}} = T_{\text{CLK}}/2 = 1/(16 \times 20\,\text{kHz}) = 3.125\,\text{us}$:

$$t \approx 6.238\tau < 3.125\,\text{us} \tag{4.3}$$

We use the minimal length $L = 180\,\text{nm}$ and a minimal width $W = 300\,\text{nm}$ for the transistor dimension, as it minimizes the charge in the channel and reduces the charge injection effect. We chose a small width because the switch drives a low current. Simulating this switch, we found that has a variable resistance between 2.8 kΩ and 66.46 kΩ, as shown in Figure 4.10.

As the load resistance is too high, this switch resistance variation has almost no effect on the final steady voltage value. However, it will limit the hold capacitance we can use. According to the previous factor calculated for the settling time in (4.3), we can use the worst resistance value to obtain the maximum capacitor, which results in:

$$C_{\text{max}} = \frac{t_{\text{max}}}{6.238 \times R_{\text{sw}}} = \frac{3.125\,\text{us}}{6.238 \times 66.46\,\text{k}\Omega} \approx 7.5\,\text{pF} \tag{4.4}$$

Although we can use up to this value, increasing excessively would impact the power consumption. For that reason, we ended up using a much lower value, 100 fF, as we intend a design with low power dissipation.

After simulating with a ramp input, we obtain the result shown in Figure 4.11. While the S_H is in a high state, the output follows the input, with the switch acting as a short

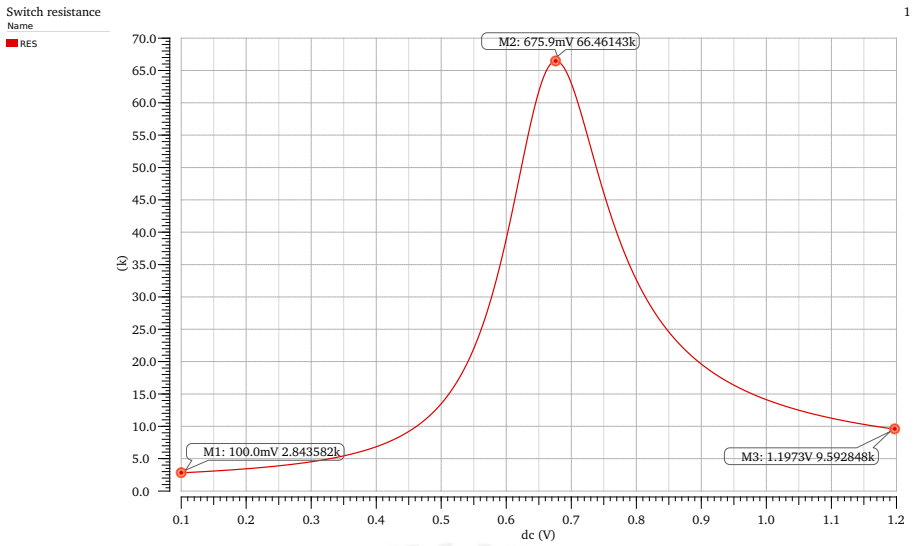


Figure 4.10: Switch resistance variation for different input values when control voltage is 1.2 V. Simulation using a test resistance of 1 GΩ, required to measure the current flowing through the switch and the voltage difference between input and output, from which calculate the switch resistance.

circuit. When the S_H falls, the voltage in the capacitor remains equal to the last value just before the control signal's transition. Something interesting to notice is the delay once the change occurs to track the input in the sample stage. This delay is minimal for almost all samples; however, there is a sample in which the delay is visibly higher than the others and occurs at an input voltage of 720 mV. Also, if we check the currents that go through the switch, this particular sample is significantly lower than in others. We infer that the resistance is higher around these voltages, producing a lower current and, consequently, a higher delay. Although it is minimal and does not impact significantly when samples are taken at fixed times, in other cases, such as the level-crossing scheme, these different delays could impact the precision of sampling.

4.2.4 DAC logic

This circuit controls the DAC via a pulse sequence generator, enabling or disabling the branches in the switched capacitors. It also takes the result from the comparator, determining the conversion codes for each bit in the output.

The architecture for this circuit was proposed initially by Anderson [56], and it is frequently used now as a reference for SAR logic implementation [57], [10], [58]. Figure 4.12 shows its scheme with a slight variation from the original.

The presented circuit has two rows of flip-flops. The first row D_{n-1} to D_e generates a pulse that is propagated in each rising clock, while the second row outputs the binary

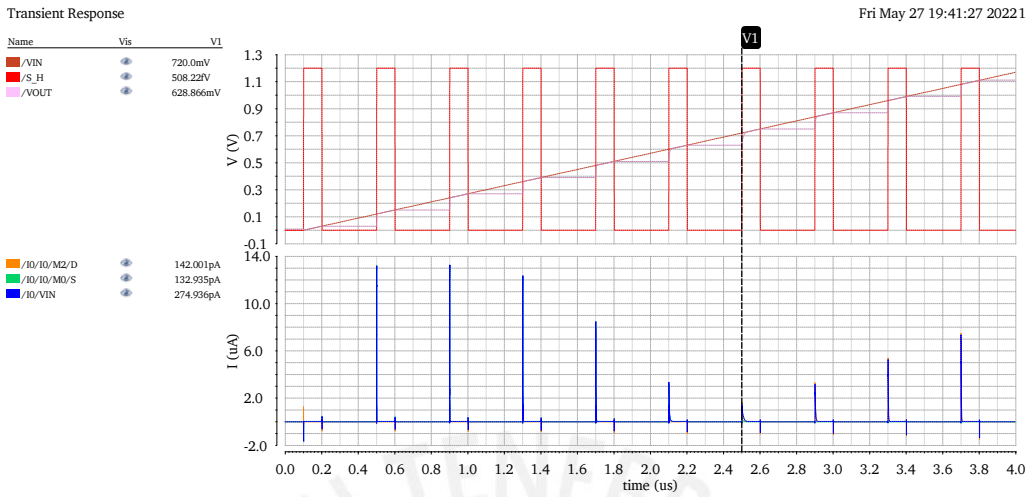


Figure 4.11: Transient simulation for the sample and hold circuit. The upper plot shows the voltage signals for all the terminals in the sample and hold component. The lower plot presents the switch's current flow during each transistor's activation.

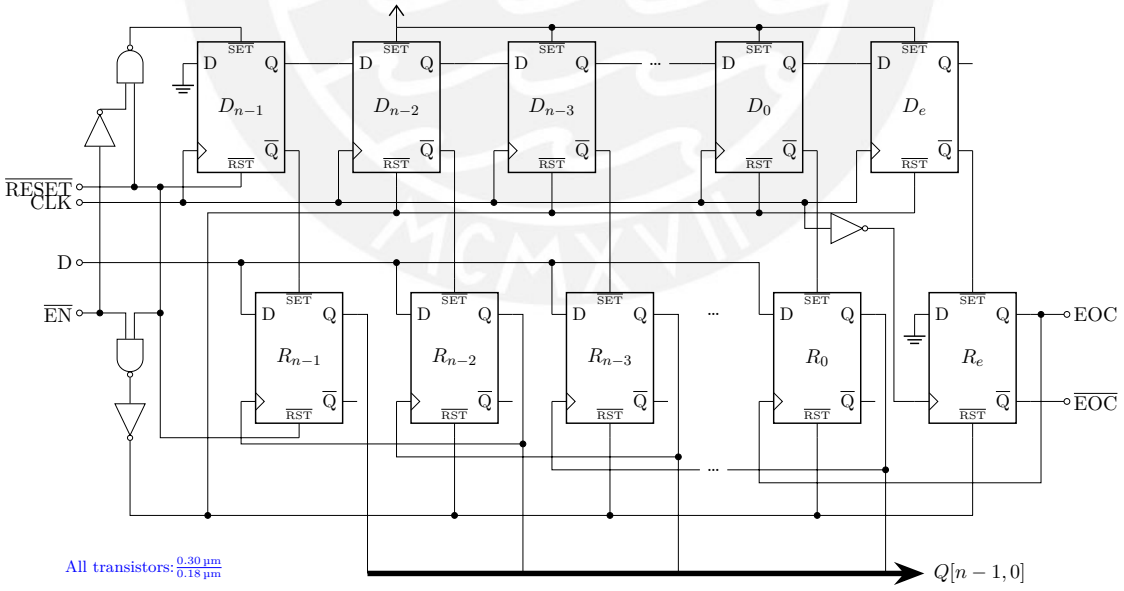


Figure 4.12: Scheme for DAC logic of SAR architecture. Implementation with D Flip-flops and control logic for the reset ($\overline{\text{RESET}}$) and enable ($\overline{\text{EN}}$) inputs. The n sub-index represents the number of output bits.

input for the DAC. The cycle starts when the SET input for the first flip-flop is activated. This occurs when the reset is OFF, and the enable input is ON. That sets a high value ('1') in the first flip-flop of the second row R_{n-1}. In the DAC, it corresponds to the most significant bit, while others remain at a low value ('0'). Afterward, the sequence starts synchronized by the clock signal. In the first rising clock after the initialization, the high value generated in D_{n-1} changes to low ('0') while the high pulse ('1') is propagated to the next flip-flip D_{n-2}. It activates the next flip-flop in the second row R_{n-2}, setting its value at '1'. This activation also produces a rising clock that triggers the previous flip-flop R_{n-1}. It will set the value from the D input (connected to the comparator). Therefore, it will store the result from the comparison while this flip-flop output is high ('1'). The cycle continues for each bit in the output until reaching the flip-flop D_0 and generating the output Q_n. The purpose of the last flip-flop R_e is to generate a pulse that indicates the end of conversion EOC. This pulse could be set indefinitely until the reset is activated. However, we added logic to generating a pulse with a finite duration, which will trigger the next cycle in a continuous conversion. Its duration is half of a clock cycle. That is the purpose of using the inverted CLK signal for the last flip-flop. We choose this duration to use part of the first comparison window; otherwise, it will increase one clock cycle and the conversion will take one more cycle than the number of output bits. Concerning the reset and enable signals when active, all the flip-flops except the first of each row output '0'. This is used to reset the output and, when the enable input is activated, to prepare for the next conversion cycle.

We expect to work with sampling frequencies around 20 kHz in the SAR architecture. Thus, depending on the resolution, the clock frequency will equal the sampling frequency times the number of bits $f_s \times N$. With a resolution of 8 bits, the clock frequency will be 160 kHz.

For simulation purposes, we chose a slightly higher frequency of 200 kHz. The result is presented in Figure 4.13. Here, the RESET_n ($\overline{\text{RESET}}$) is set to start at a high level (deactivated) and then go to a low level (activated). During this stage, independent of the state of other control signals, all the outputs are zero (except EOC_n whose deactivated state is 1). After this state is set, the EN_n ($\overline{\text{EN}}$) is activated to start the conversion stage. If the comparator input is zero along the whole conversion cycle, the final states for each output are zero, and we can see the propagation of the first pulse in each output bit. Then, if the comparator input is one during a certain pulse in one output bit, this value remains in that output bit, until the next conversion cycle. At the end of the conversion,

a final pulse is generated in the EOC signal, which means the cycle is ending. It will serve as an ”enable” pulse that will start the conversion again.

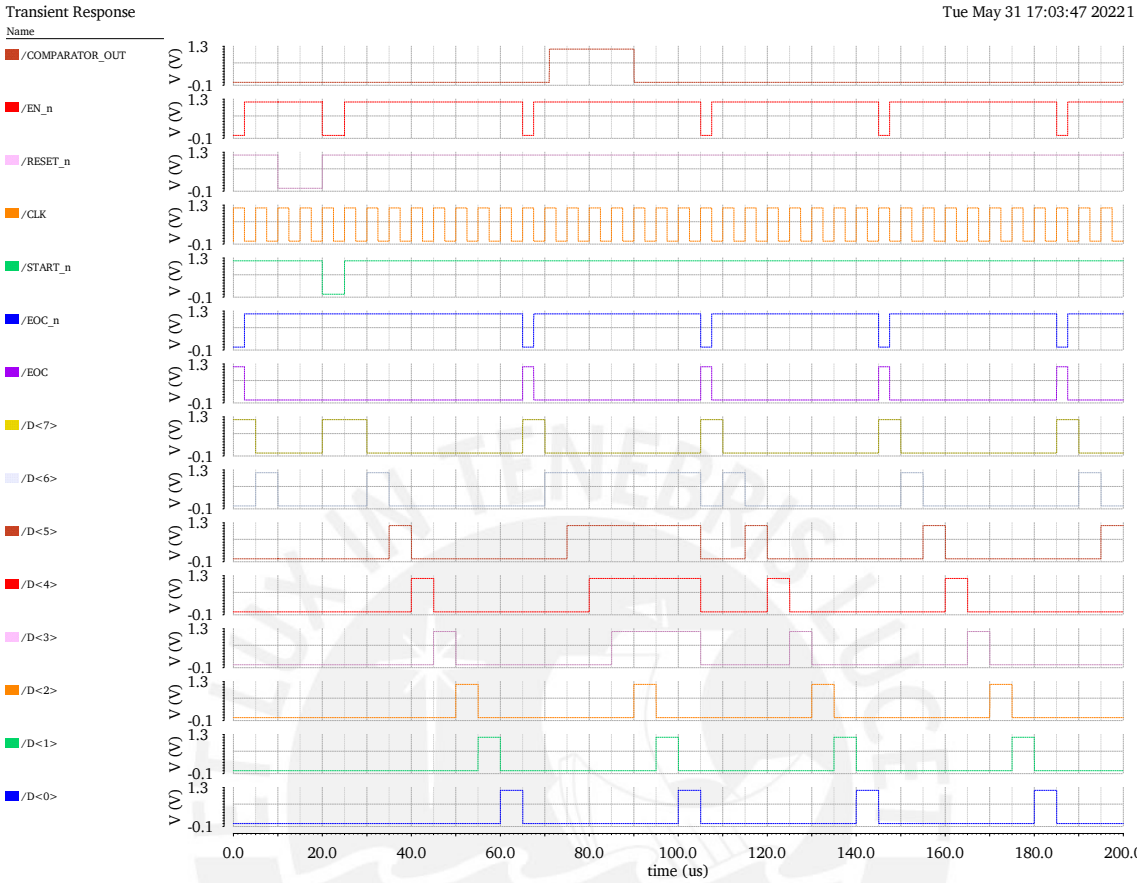


Figure 4.13: Transient simulation for testbench of SAR logic. Here, the signals show the output bits' variation along the time, synchronized by the clock signal and controlled by the START_n and RESET_n. The clock frequency is 200 kHz, and its duty cycle is 50%.

4.2.5 DAC

A DAC can be implemented by three different approaches: by using resistors, capacitors, and current sources. The most used ones we found for low-power ADCs are capacitive DACs. They are based on the switched principle, where the charge is distributed in each step according to the branches' signal levels connected to the reference voltage or the ground. Ahuja *et al.* [59] compare two common architectures for capacitive DACs: the binary-weighted and the split capacitors. According to its results, the binary DAC is more area and power-efficient.

The idea behind this architecture relies on an array of capacitors whose values are proportional to their corresponding weight in the analog output. As each branch is con-

nected to one bit in the digital input, their values are a scaled factor of a power of two multiplied by a base capacitor value corresponding to the LSB.

On the other hand, depending on how the array converts the signal: referred to the ground (single-ended) or taking the difference between two input signals (differential architecture), the topology has slight variations. Both architectures are presented by Bonetti [60], while Li *et al.* [61] emphasize differential architectures.

4.2.5.1 Single-ended DAC

We can differentiate two approaches for the capacitive DAC: one which acts just as a DAC that generates the voltage compared against a sampled signal from another circuit; or a DAC that also includes the function of sample and hold, receiving the input signal and comparing the result against a fixed reference voltage. The last one is the preferred implementation; however, the first is the simplest to explain. We will start with the basic approach, and then we will continue to the implementable design.

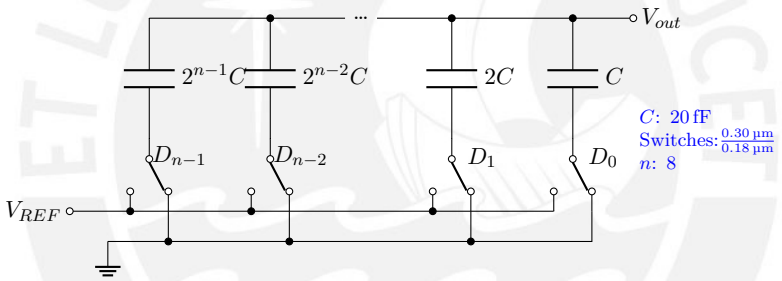


Figure 4.14: Conceptual circuit for binary-weighted DAC capacitive architecture for n bits. The n value represents the number of bits for the DAC, and C is the unit capacitor from which the others are multiples. The D inputs represent the digital value that controls the switches in the DAC.

The conceptual DAC is presented in Figure 4.14. Its operation relies on the charge distribution when the branches are connected to V_{REF} or ground acting as a capacitor divider. In a general capacitor divider with two series capacitors (C_D and C_U) connected to V_{REF} and ground, where C_U is the capacitor connected to V_{REF} and C_D is the capacitor connected to ground, and the output is the node that connects both capacitors, the output voltage can be expressed as:

$$V_{out} = V_{REF}\frac{C_U}{C_D + C_U} \tag{4.5}$$

Supposing we have a binary number $X = \overline{X_{n-1}X_{n-2}\ldots X_1X_0}$ whose bits are connected to the control switches. When the bit is '0', it is connected to ground (GND); when it is '1', it is connected to V_{REF}. We can group the capacitors connected to V_{REF} into an

equivalent capacitance, while we can sum all the capacitors for the denominator in (4.5). Finally, the output will be:

$$V_{out} = V_{REF}\frac{X_{n-1}2^{n-1} + X_{n-2}2^{n-2} + \ldots + X_0}{2^{n-1} + 2^{n-2} + \ldots + 1} = V_{REF}\frac{X}{2^n - 1} \tag{4.6}$$

Here, we can see that the V_{REF} voltage is scaled in a proportion equal to the digital number divided by the maximum representable number with n bits. Other DAC designs include an additional capacitor C_e with the same value as the base capacitor C, to scale the voltage against 2^n instead of $2^n - 1$ as presented in the equation (4.6).

The circuit uses a comparator that compares the DAC output against the sampled V_{in}. Then, the comparator takes the difference:

$$V_{comp} = V_{in} - V_{REF}\frac{X}{2^n - 1} \tag{4.7}$$

This result will define if the corresponding bit will be '1' or '0' in the final digital output.

In the design, we use a width of 300 nm for NMOS and PMOS in the switches, with a minimum channel length (180 nm). In this case, we followed the same analysis as in the sample and hold switch to minimize the effect of charge injection and clock feed through, with the benefit of lower area.

On the other hand, for the capacitor, we have to consider that in the charge stage, the DAC will use a sum of several capacitors, which, according to (4.5) and (4.6), will be equal to $255C$ (and if we use an additional C_e capacitor, it will be $256C$). Then the previous value of 7.5 pF, using the equation (4.4), should be equivalent to the total capacitance. Considering that, the maximum base capacitance will be:

$$C_{\max} = \frac{7.5\,\text{pF}}{255} \approx 29\,\text{fF} \tag{4.8}$$

In this case, we use a lower value of 20 fF to favor a low power dissipation, but not too low because of the capacitance mismatch. According to the Monte Carlo - mismatch simulation, the current value returns a standard deviation of %0.54. This deviation is propagated to the DAC output as it depends on the scaled value of the unit capacitor, as shown in (4.5). According to our calculus, the maximum variation occurs when DAC targets $V_{REF}/2$, obtaining a deviation of 0.043LSB at DAC output for this capacitor mismatch, lower than %10 the limit of 0.5LSB. For more details on this, please, refer to Appendix C and check our procedure to get this value.

Figure 4.15 shows the result of the transient simulation. The analog output shows the expected rising ramp as the digital input grows. There are glitches, but they can be

explained due to the transitions between the bits in the digital input. On the other hand, although we expect a 1.2 V for the last digital value, the results show a voltage of 1.197 V. This difference is caused by the leakage current through a switch connected to the V_{out} used as RESET for the output value. The effect is appreciable due to the sweep along 256 cycles in the present test; however, in the SAR circuit, it is only 8, and this effect will be negligible.

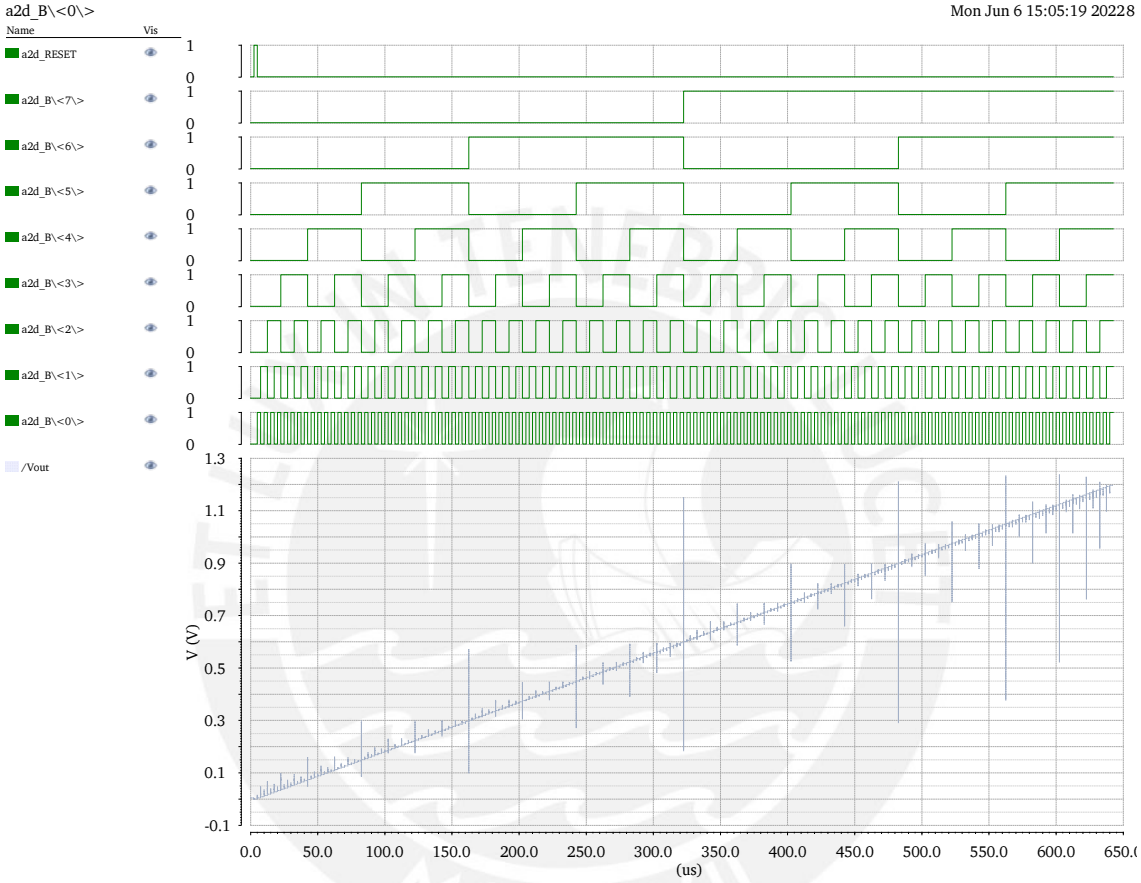


Figure 4.15: Transient simulation for DAC testbench. The minimum period is set to 5 µs (D0). All others signals are multiple of this minimum period to generate the rising ramp.

In this architecture, the previously stored charge or the activation sequence is irrelevant, as the voltage starts in '0' and progressively approaches the signal level, sampled by another circuit. Its disadvantage is that the comparison level is variable (comes from the input) and may reach regions where the comparator doesn't work correctly (the transistors are not saturated). That's why we prefer a fixed reference.

To use a fixed reference, the DAC acts as a sample and hold block and also as a DAC. Frequently, it employs a common mode voltage V_{CM} against which we will compare the DAC output [58]. The operation can be divided into two stages: the sampling stage and the conversion stage. The circuit for this kind of DAC is presented in Figure 4.16.

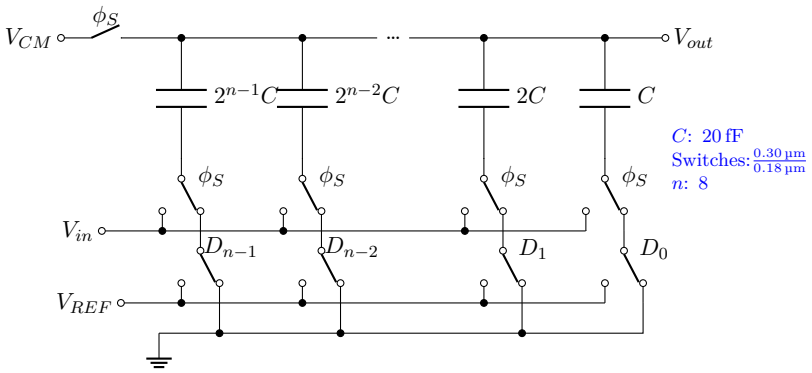


Figure 4.16: DAC circuit with sample and hold function included. Each branch is connected to two levels of switches that will be used in the phase of sampling and conversion, respectively.

In the sampling phase, the switch ϕ_S is activated and connects the output to the V_{CM} while all the capacitive branches are connected to V_{in}. This produces an accumulation of charge that is equal to:

$$Q = (V_{CM} - V_{in}) \times (2^n - 1)C \tag{4.9}$$

During this stage, $V_{out} = V_{CM}$. When the sampling switches open, the new voltage is $V_{out} = V_{CM} - V_{in}$ and starts the conversion stage. Following the same logic as conventional DAC, the first cycle connects the first branch ($2^{n-1}C$) to V_{REF}. It will distribute the stored charge among all capacitors obtaining the following equation:

$$2^{n-1}C(V_{out} - V_{REF}) + V_{out}\overbrace{(2^{n-2} + 2^{n-3} + \ldots + 1)}^{2^{n-1}-1}C = (V_{CM} - V_{in}) \times (2^n - 1)C \tag{4.10}$$

Grouping and simplifying this expression results in:

$$\begin{aligned}(2^n - 1)V_{out} &= (V_{CM} - V_{in}) \times (2^n - 1) + V_{REF}2^{n-1} \\ V_{out} &= V_{CM} - V_{in} + V_{REF}\frac{2^{n-1}}{2^n - 1}\end{aligned} \tag{4.11}$$

In the SAR architecture, the other terminal of the comparator is connected to V_{CM}. Then, the comparison voltage will be:

$$\begin{aligned}V_{comp} &= V_{CM} - \left[V_{CM} - V_{in} + V_{REF}\frac{2^{n-1}}{2^n - 1}\right] \\ V_{comp} &= V_{in} - V_{REF}\frac{2^{n-1}}{2^n - 1}\end{aligned} \tag{4.12}$$

Here, if V_{in} is greater than the scaled value of V_{REF} (approximately a half of V_{REF}), the comparison will output '1', otherwise will return '0'. This is the same behavior as the DAC without the sampling feature.

The scaled factor for the V_{REF} in equation (4.12) could generate some variations in the conversion. Then to fix it, we can add a capacitor equal to the base value C. The architecture with the additional capacitor is presented in Figure 4.17.

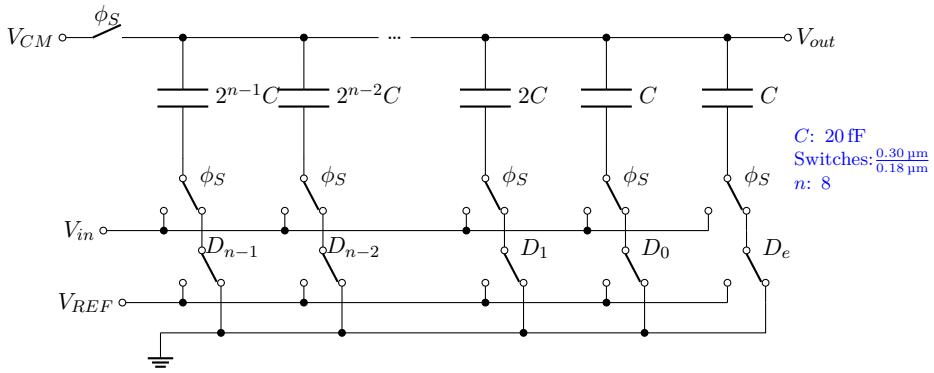


Figure 4.17: DAC circuit with sample and hold function included with an additional capacitor to fix the scale factor. In the SAR logic, D_e is always 0, and the terminal is connected to the ground in all cases.

With this addition in the circuit, the charge stored in the sampling stage equals:

$$Q = (V_{CM} - V_{in}) \times 2^n C \tag{4.13}$$

When the sampling switches are open, the voltage stored equals $V_{CM} - V_{in}$. Then, following the logic sequence of the SAR, in the first cycle, where D_{n-1} is '1', the capacitor $2^{n-1}C$ is connected to V_{REF}, and the others are connected to ground, the resulting voltage output is:

$$\begin{aligned} 2^{n-1}C(V_{out} - V_{REF}) + V_{out}2^{n-1}C &= 2^n C(V_{CM} - V_{in}) \\ 2^n C V_{out} &= 2^n C(V_{CM} - V_{in}) + V_{REF}2^{n-1} \\ V_{out} &= V_{CM} - V_{in} + \frac{1}{2}V_{REF} \end{aligned} \tag{4.14}$$

The final comparison value will be:

$$V_{comp} = V_{in} - \frac{1}{2}V_{REF} \tag{4.15}$$

To simulate this DAC, we use a fixed analog value. We evaluate the behavior similarly as it was controlled for the SAR logic block. We use a test voltage of 400 mV and a reset pulse active from the beginning until the first 30 µs. This is important to reset the capacitors, as the remaining charge can alter the voltages in the evaluation sequence.

The result of the transient simulation is shown in Figure 4.18. Here, we can appreciate how the DAC voltage starts at 800 mV. It is equal to the result from the relationship in (4.14). In the next iterations, it decreases the comparison voltage in $V_{REF}/4$, $V_{REF}/8$ until

$V_{REF}/256$. Finally, when all digital values are '0', the output voltage reaches $V_{CM} - V_{in} =$ 200 mV, as expected from the theory.

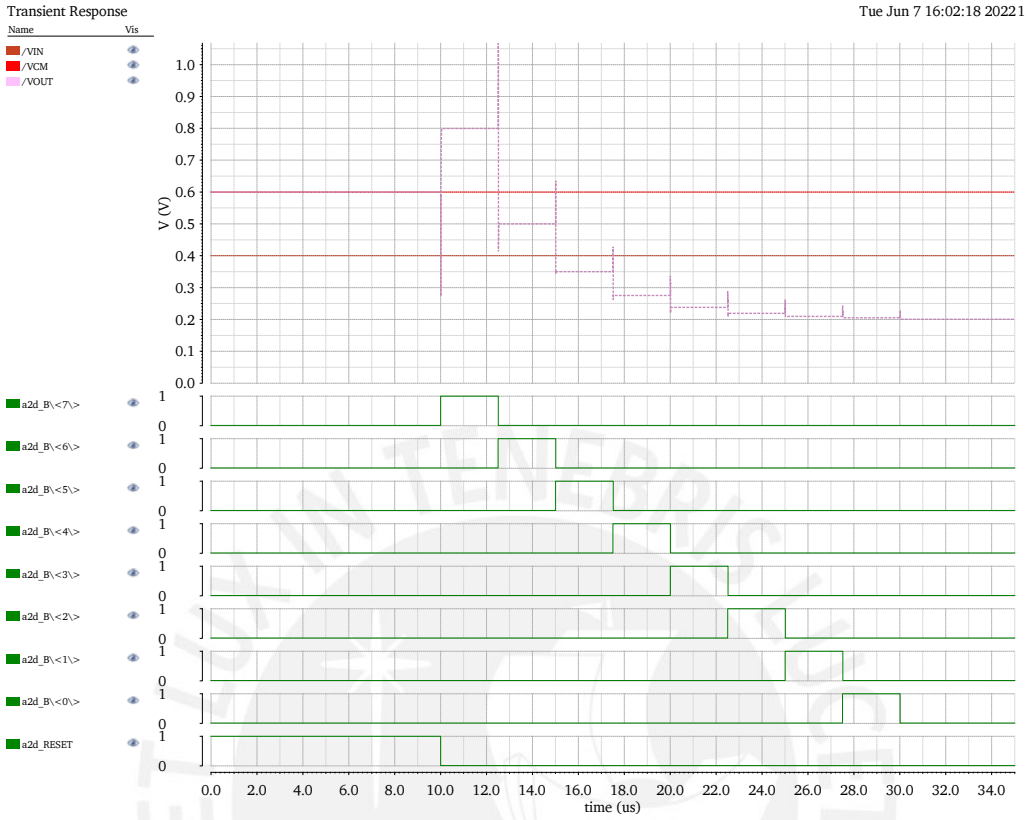


Figure 4.18: Transient simulation for DAC testbench. The pulse width is set to 2.5 µs. The simulation time covers the complete sequence of pulses in the digital inputs.

4.2.5.2 Differential DAC architecture

The differential DAC with sampling and hold functionality can be considered two single-ended DACs, one for each input V_{in_p} and V_{in_n}. However, if we analyze the voltage result from the previous design in the conversion stage, we identify that both inputs are affected by the same scaled V_{REF} voltage and with the same sign. Subtracting them in the comparator will cancel the voltage we want to compare with the analog input. This means that, independently of the combination of the digital input, the result will always be the same, which does not help obtain its digital representation. To fix it, we can invert the digital input or swap the connection between V_{REF} and the ground. It produces a generated comparison voltage, positive in one input and negative in the other, obtaining a difference of twice its value instead of zero.

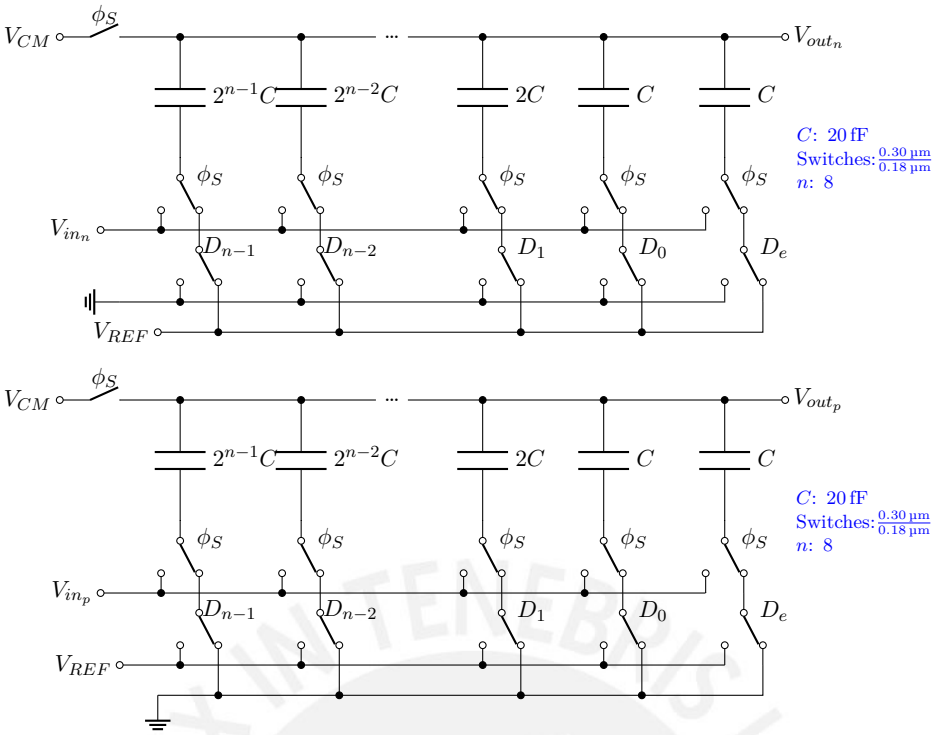


Figure 4.19: Differential DAC circuit with sample and hold function. The inputs are V_{in_p} and V_{in_n} which stands for the positive and negative inputs, while V_{out_p} and V_{out_n} are the corresponding outputs.

From Figure 4.19 we can see two DAC structures, one per each comparator input. The voltage for the positive input will be equal to the previous single-ended architecture:

$$\begin{aligned} 2^{n-1}C(V_{out_p} - V_{REF}) + V_{out_p}2^{n-1}C &= 2^nC(V_{CM} - V_{in_p}) \\ 2^nCV_{out_p} &= 2^nC(V_{CM} - V_{in}) + V_{REF}2^{n-1} \\ V_{out_p} &= V_{CM} - V_{in_p} + \frac{1}{2}V_{REF} \end{aligned} \tag{4.16}$$

In contrast, the negative input has a slight variation in the output voltage:

$$\begin{aligned} 2^{n-1}CV_{out_n} + (V_{out_n} - V_{REF})2^{n-1}C &= 2^nC(V_{CM} - V_{in_p}) \\ 2^nCV_{out_n} &= 2^nC(V_{CM} - V_{in}) + V_{REF}2^{n-1} \\ V_{out_n} &= V_{CM} - V_{in_n} + \frac{1}{2}V_{REF} \end{aligned} \tag{4.17}$$

The difference voltage will be:

$$V_{out_p} - V_{out_n} = V_{in_n} - V_{in_p} \tag{4.18}$$

Although this result gives just the sign of the output, we note a difference in the next iteration. If we follow the conventional logic for the single-ended SAR, and supposing the

signal is positive, the D_{n-1} is '0'. Then, enabling the following bit D_{n-2} returns a voltage in the positive input equal to:

$$2^{n-2}C(V_{out_p} - V_{REF}) + V_{out_p}(2^{n-1} + 2^{n-2})C = 2^nC(V_{CM} - V_{in_p})$$
$$2^nCV_{out_p} = 2^nC(V_{CM} - V_{in_p}) + V_{REF}2^{n-2}$$
$$V_{out_p} = V_{CM} - V_{in_p} + \frac{1}{4}V_{REF} \tag{4.19}$$

In the negative input, it is:

$$2^{n-2}CV_{out_n} + (V_{out_n} - V_{REF})(2^{n-1} + 2^{n-2})C = 2^nC(V_{CM} - V_{in_n})$$
$$2^nCV_{out_n} = 2^nC(V_{CM} - V_{in_n}) + 3V_{REF}2^{n-2}$$
$$V_{out_n} = V_{CM} - V_{in_n} + \frac{3}{4}V_{REF} \tag{4.20}$$

Then, the comparison voltage is equal to:

$$V_{out_p} - V_{out_n} = V_{in_n} - V_{in_p} - \frac{1}{2}V_{REF} \tag{4.21}$$

On the other case, when the result in the equation (4.18) is negative, the comparator output is '1' and the corresponding value in the positive input is:

$$(2^{n-1} + 2^{n-2})C(V_{out_p} - V_{REF}) + V_{out_p}2^{n-2}C = 2^nC(V_{CM} - V_{in_p})$$
$$2^nCV_{out_p} = 2^nC(V_{CM} - V_{in_p}) + 3V_{REF}2^{n-2}$$
$$V_{out_p} = V_{CM} - V_{in_p} + \frac{3}{4}V_{REF} \tag{4.22}$$

While, in the negative input, it is:

$$(2^{n-1} + 2^{n-2})CV_{out_n} + (V_{out_n} - V_{REF})2^{n-2}C = 2^nC(V_{CM} - V_{in_n})$$
$$2^nCV_{out_n} = 2^nC(V_{CM} - V_{in_n}) + 3V_{REF}2^{n-2}$$
$$V_{out_n} = V_{CM} - V_{in_n} + \frac{1}{4}V_{REF} \tag{4.23}$$

The comparison voltage is equal to:

$$V_{out_p} - V_{out_n} = V_{in_n} - V_{in_p} + \frac{1}{2}V_{REF} \tag{4.24}$$

We can observe from equations (4.21) and (4.24) that considering the voltage sign of $(V_{in_n} - V_{in_p})$, the term we compare against it has the same sign in the next iteration (following the same logic as in the single-ended), which doesn't perform an effective comparison to obtain the magnitude bit. Then, it requires a special logic or an adaptation

from the single-ended to fix it. That will be discussed in the differential SAR architecture section.

Something important to notice is the fact that, while in the single-ended, we can compare against $\frac{1}{2}V_{REF}$ with the first bit D_{n-1}, in the differential case, we need to use the next bit D_{n-2} to obtain the same term. This means that, while in the final bit, we compare with a voltage of $\frac{1}{2^n}V_{REF}$, in the differential circuit, the comparison voltage is $\frac{1}{2^{n-1}}V_{REF}$. Consequently, the resolution with the same number of capacitors is lower in the differential than in the single-ended case. The solution we considered is an additional capacitor of value 2^nC.

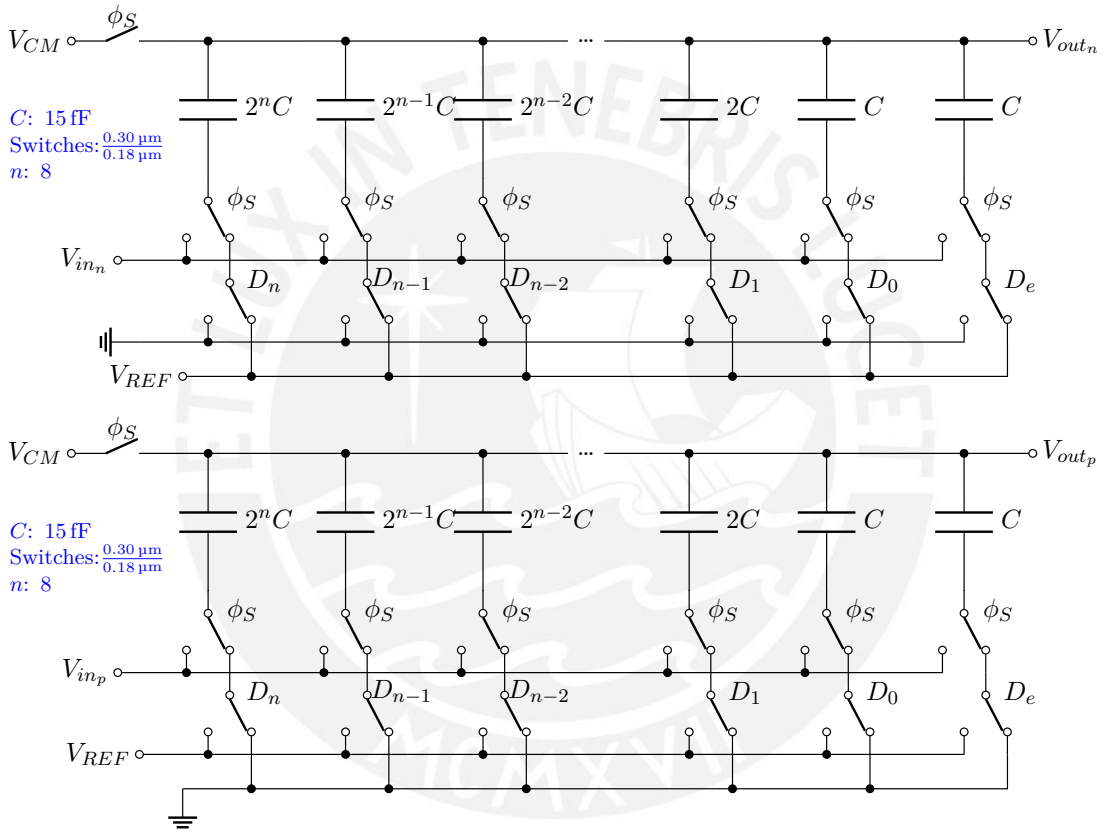


Figure 4.20: Differential DAC circuit with sample and hold function with an additional capacitor to achieve the required n-bits of resolution. Here, D_n can be interpreted as the sign bit. It doesn't provide information on the voltage magnitude.

Figure 4.20 presents the circuit with that capacitor to achieve the n-bits of resolution. The first bit gives information about the sign, while the others correspond to the magnitude. It means that the conversion could require an additional clock cycle. To fix it, we can adapt the digital input knowing the maximum amplitude of the analog signal. For example, if V_{CM} is $V_{REF}/2$ as usual, the first bit of magnitude D_{n-1} is always '0', as it cannot be higher than $V_{REF}/2$. We can assume this bit value and continue converting the

remaining bits. Finally, we can obtain the complete digital value in the same number of cycles as in the single-ended case.

In relation to the dimensions, for the switches, we use the same dimensions as all previous cases. However, for the capacitor, we reduced the size because in the previous analysis (4.8), when we obtained a maximum capacitance of 29 fF, the whole capacitance was equal to $256C$; however, in this case, as we added a capacitor with the same value, the new whole capacitance is $512C$. Therefore, the maximum capacitance will be:

$$C_{\max} = \frac{7.5\,\text{pF}}{512} \approx 14.5\,\text{fF} \tag{4.25}$$

As the estimation employed for the calculus was pessimistic, considering extreme cases for voltage change across the capacitor and maximum resistance for the switch, we used a value slightly higher than the limit, in this case, 15 fF.

In this case, we would ask why don't we reduce further the capacitance to values of 10 fF or even 1 fF if they could be faster and use less area? There are several reasons for that. One of them is the possibility of implementing those values. We are using the TSMC 180nm technology, and with hard constraints, we only achieve a minimum unit capacitance of 7.4 fF. Another reason is the mismatch, as the lower the dimension, the more the variability of the size and consequently on the capacitance. And finally, because of the thermal noise. According to Razavi [62], the capacitor has a root-mean-square (RMS) noise equal to:

$$v_n = \sqrt{\frac{kT}{C}} \tag{4.26}$$

where $k = 1.38 \times 10^{-23}\,\text{JK}^{-1}$ is the Boltzmann constant, T is the absolute temperature, and C is the capacitance. For ambient temperature ($T = 300\,\text{K}$), we would find values of noise equal to 2 mV if the capacitance is $C = 1\,\text{fF}$, close to 0.5 LSB, and 640 uV if it is $C = 10\,\text{fF}$. For the chosen value of 15 fF, it is slightly lower: 525 uV, which is almost 10% of the LSB, that we consider adequate.

Regarding the circuit simulation, it will return the same result as the single-ended version, with the difference that it has two inputs. Then, we consider it more interesting to test its behavior as part of the entire SAR circuit, developed in the next section.

4.2.6 Complete SAR architecture

The complete SAR architecture considers the blocks discussed in previous sections: sample and hold, DAC, logic, comparator, and register. Depending on the DAC used, the ADC can be single-ended or differential.

We can use the ENOB as the first evaluation metric to test the ADC performance. Here, we design an ADC of 8 bits; therefore, we expect an ENOB, decimals below 8, as the noise introduced by the circuits is added to the quantization noise reducing the SNDR.

4.2.6.1 Single-ended architecture

The first proposal is a single-ended architecture that uses independent blocks for the sample and hold and the DAC. The scheme is presented in Figure 4.21. In this architecture, the voltage input V_{in} is sampled by the sample and hold block, controlled by an enabled signal EN, which is activated when the $\overline{\text{EOC}}$ or $\overline{\text{START}}$ signals are active. The signal $\overline{\text{RESET}}$ resets all the digital blocks, which includes the logic and register (all digital bits to zero), and corresponds to the first pulse in the initialization process. The next pulse is the $\overline{\text{START}}$, which initiates the logic sequence while sampling the input voltage. Following the steps reviewed in the SAR logic design, the DAC_{in} has its MSB in '1', and the others in '0'. Then, the output DAC is $V_{REF}2^{n-1}/(2^n-1)$. Depending on the sampled value, when the falling edge of CLK occurs (rising edge of $\overline{\text{CLK}}$), the comparator output is '1' or '0' if the DAC_{out} is higher than the sampled V_{in} or vice versa. The comparator maintains the value until the rising clock sets it to the MSB and continues to the $n-2$ bit. If it is '1', the next DAC output is $V_{REF}3\times 2^{n-2}/(2^n-1)$; otherwise, it is $V_{REF}2^{n-2}/(2^n-1)$. This term is compared with the sampled voltage following the same logic as the previous bit. The sequence continues until the last bit arrives. After that, the logic clock generates the $\overline{\text{EOC}}$ flag, which initiates the next conversion cycle.

In this architecture, the result of the last bit is set simultaneously as it triggers the loading of the digital value to the register. However, it doesn't provide enough time to save the last bit in DAC_{in}. Thus, we use the comparator output directly with the other bits in DAC_{in}, to set the register output. On the other hand, as the DAC saves the charge in each iteration, we need to reset the voltage for the next conversion cycle to avoid any residual charge that could be added. For it, we employ a pulse with a fixed duration (monostable), lower than a clock cycle, triggered by the rising edge of EN. We generate that pulse by combining a delayed EN with the non-delayed signal via a NAND gate. That creates a pulse width equal to the delay time of the signal.

For the circuit implementation, we use transistors with a minimal width $W = 300\,\text{nm}$ to reduce the power dissipation, while in the DAC, we use 20 fF for the base capacitor. We consider it an adequate value that balances the delay time without significant thermal noise contribution. Another important dimension to take into account is the capacitance

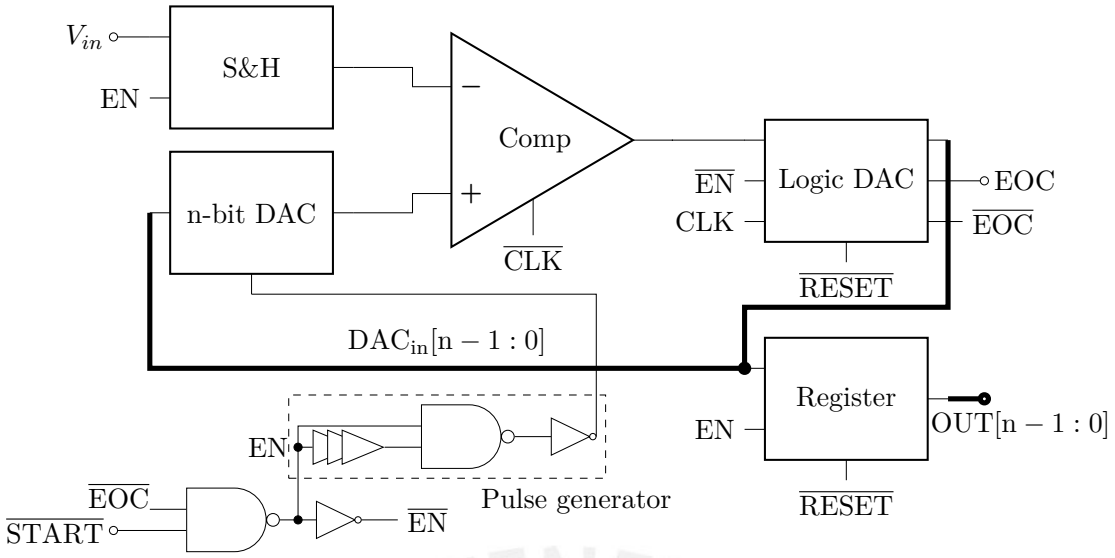


Figure 4.21: Scheme for single-ended SAR ADC based on independent DAC and sample and hold blocks. The inputs for this circuit are V_{in} which is the analog input signal; the $\overline{\text{RESET}}$, which stands for the inverse of the reset function; the $\overline{\text{START}}$, which initiates the sequence of conversion and CLK, which is the clock that synchronizes the sampling and conversion functions. The outputs are EOC, which stands for the end of conversion, and the OUT[n − 1 : 0], which is the digital output.

in the sample and hold block. For it, we choose 1 pF, because, simulating with lower values, we identify a voltage increment in the capacitor during the comparison phase of the dynamic comparator. This effect can be explained due to the pre-charged capacitances during the comparator's OFF phase, whose current flows to the hold capacitor, increasing its voltage.

To test the behavior, we use two transient experiments. One for a constant voltage and another for a sinusoidal input voltage. The first aims to analyze the behavior of the DAC and other blocks during the conversion process, while the other aims to measure the ENOB.

In the first scenario, we use a square signal in two stable voltage values 0 V and 800 mV. By design requirement, we expect that the SAR must operate with a frequency of 20 kHz ($T_s = 1/20\,\text{kHz} = 50\,\mu\text{s}$), though, we use the corresponding period equal to the eighth part of this cycle ($50/8 = 6.25\,\mu\text{s}$).

The result of the transient simulation is presented in Figure 4.22. As expected, the voltage in the DAC tries to follow the sampled input value. First, it is set to a half of 1.2 V, then, in the next iteration, the value is $\frac{3}{4}V_{REF}$. As this value is higher than V_{in}, the next comparison voltage in the DAC is $\frac{1}{2} + \frac{1}{8} = \frac{5}{8}V_{REF}$. That continues until the last

bit. The final value obtained in decimal representation is 169. However, the expected is:

$$ADC_{out} = 2^n - 1 \times \frac{V_{in}}{V_{REF}} = 255 \times \frac{0.8}{1.2} = 170 \tag{4.27}$$

Hence, there is a slight variation in the representation. It can be explained due to the variations detected previously in the DAC simulation.

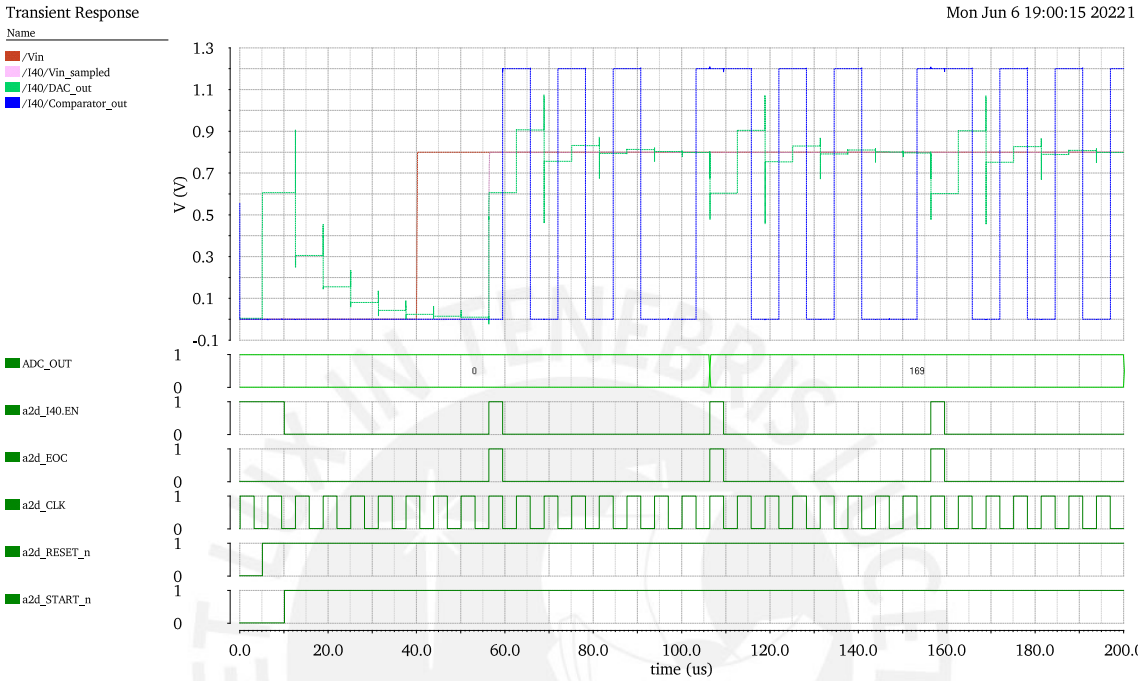


Figure 4.22: Transient simulation for single-ended 8-bit SAR architecture. The simulation runs until 200 µs and permits showing at least the conversion for two values 0V and 800 mV, and in the last case, at least two successive conversion cycles. The digital results were 0 and 169, respectively.

The second test applies a sinusoidal wave to SAR ADC input, and from the results, we measure the ENOB. We use the coherent sampling technique to improve the spectrum representation. In this case, we choose several complete cycles $M_C = 7$ and samples $M = 128$, then the frequency input is[1]:

$$f_{in} = f_s \frac{M_C}{M} = 20\,\text{kHz} \frac{7}{128} = 1.093\,75\,\text{kHz} \tag{4.28}$$

To calculate the ENOB, we use a zero-order hold in ADC output. We create this block by describing its behavior via Verilog-A. It emulates an ideal DAC that generates the corresponding analog output for each digital code obtained from the ADC. It includes an offset value to subtract the DC component, which is set to 600 mV.

The response is presented in Figure 4.23. Although the sampled voltages follow the input, the ADC output doesn't. As the input is from 100 mV to 1.1 V, we expect the

[1] In Cadence, we let the software calculate this value while we enter the expression as a function of the sampling frequency.

signal with no DC component to range from −500 mV to 500 mV. That range occurs just in the first cycle; however, the signal rises above the expected limit in the next ones. On the other hand, we appreciate that, the comparator is not working properly for input voltages lower than 300 mV, obtaining a '0' value in any case. These errors in the output significantly impact to the SNDR, as they introduce harmonics reducing the ENOB to a detrimental level.

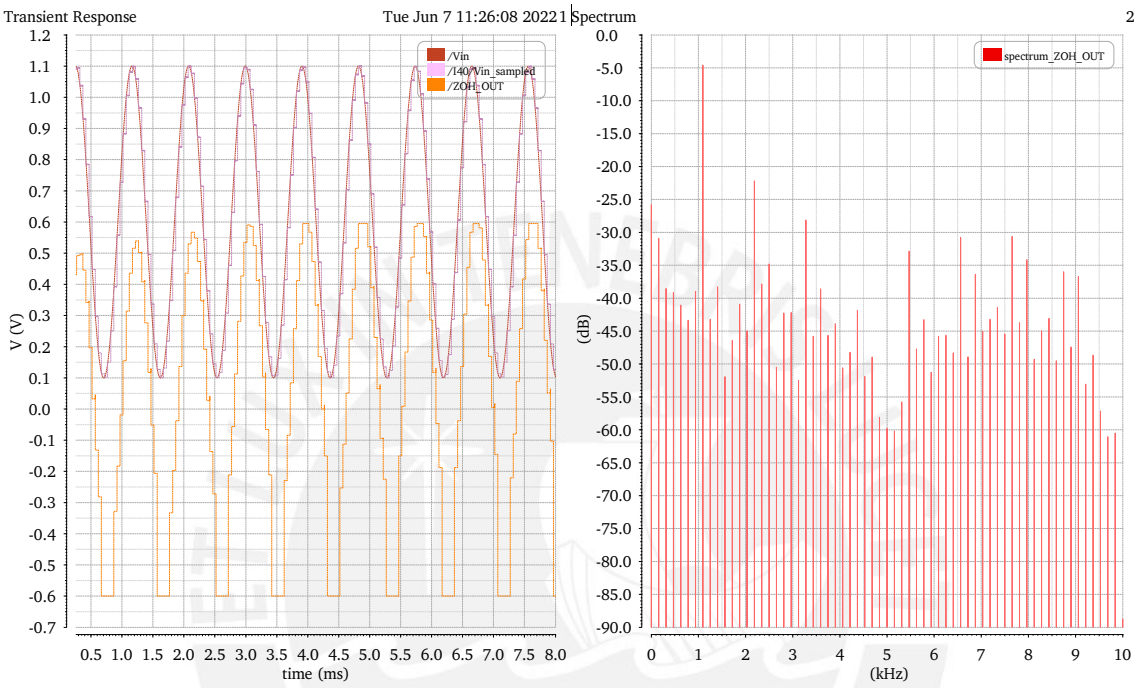


Figure 4.23: Transient simulation for single-ended 8-bit SAR architecture. The simulation runs until 8 ms to obtain the required seven complete cycles from which the spectrum is calculated. On the right is the transient response while on the left is the spectrum obtained from the ZOH output. According to the measurements in Cadence, the SNDR is 13.8 dB giving an ENOB of 2.01.

The increased output voltage can be explained due to a non-adequate reset time that keeps the charge from previous conversion cycles and adds it to the next, altering the output value. It could be solved using another time window for the reset signal. However, it isn't the most critical issue. The non-functioning of the comparator is due to the non-polarization of its differential pair transistors, as the common voltage falls below its threshold voltage. This problem cannot be addressed easily, at least with this configuration, as the comparison voltages always change depending on the input. Therefore, we opt to change to an architecture that uses a fixed reference voltage for the comparison. That alternative employs a DAC that shares the sample and hold function, also discussed in the DAC section.

Figure 4.24 presents the SAR architecture with fixed reference. It employs a DAC that receives the input voltage and returns as output the difference between the common voltage and the sampled input combined with a scaled reference created from its digital input. The sequence of functionality is similar to the previous SAR architecture. However, it introduces two improvements to solve the issues identified in the floating reference architecture.

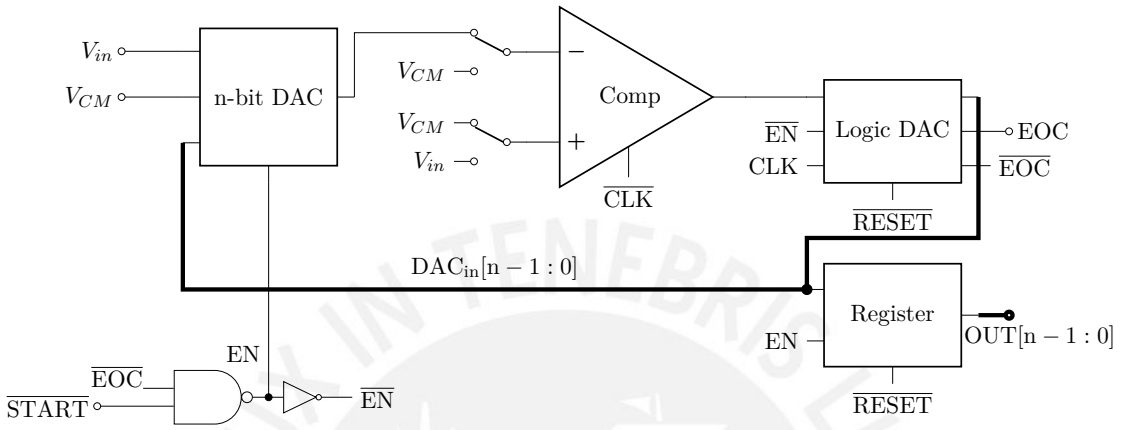


Figure 4.24: Scheme for single-ended SAR ADC based on a DAC with sampling function. The inputs for this circuit are V_{in} which is the analog input signal; the $\overline{\text{RESET}}$, which stands for the inverse of the reset function; the $\overline{\text{START}}$, which initiates the sequence of conversion; the CLK, which is the clock that synchronizes the sampling and conversion functions and the V_{CM} which is the common voltage used in the DAC. The outputs are EOC the end of conversion, and the OUT[n − 1 : 0], the digital output.

The first improvement is the RESET time in the DAC, essentially the sampling window equal to the full EOC or $\overline{\text{START}}$ instead of a pulse generated after the rising edge. It ensures enough time to discharge the capacitor plates for the next conversion cycle. During this time, the DAC output is V_{CM}, and it is compared with V_{CM} in the other terminal, which doesn't give any meaningful information for the output bit. In addition, as we are taking the first half cycle for this process and immediately comes the comparison for the first bit, we need another strategy to obtain the MSB, as the DAC output is still not charged completely at that time. Analyzing the inputs, we notice that the V_{CM} is $V_{DD}/2$, and $V_{REF} = V_{DD}$. Hence, the comparison voltage in the first iteration is equivalent to:

$$V_{comp} = V_{in} - \frac{V_{REF}}{2} = V_{in} - V_{CM} \tag{4.29}$$

Therefore, we can obtain the MSB while resetting the DAC by just comparing V_{in} and V_{CM}. We employ a double input switch for the comparator, also controlled by the EN signal. For the other bits, it follows the DAC sequence obtained in (4.14), equal to:

$$DAC_{out} = V_{CM} - V_{in} + V_{REF}\frac{\overline{X}}{2^n} \tag{4.30}$$

where $\overline{X}$ is the binary number in the DAC_{in} signal.

The second improvement is related to the kind of DAC employed. As we can see in the scheme, the positive input of the comparator has a fixed reference during the whole conversion stage. Although the input can be lower than the threshold voltage of comparator transistors during the first bit, we ensure that the other input (connected to the common voltage), is over the transistor threshold voltage. That ensures that at least one will be polarized and work at a higher range than the previous SAR architecture, where both inputs are below the threshold voltage, reducing the comparator's dynamic range.

For the circuit implementation, we use the exact dimensions in the transistors as in the previous architecture, which is $W = 300\,\text{nm}$, and for the base capacitor in the DAC, we employ 20 fF, as well.

To test the behavior, similar to the previous single-ended architecture, we use two transient experiments, one with a constant value input and the other with a sinusoidal input. In the first case, we use a square signal in two stable voltage values, 0 V and 800 mV.

The result of the transient simulation is presented in Figure 4.22. Different from the SAR based on an independent sample and hold function, in this case, the voltage in the output for the DAC follows the V_{CM}. That is desirable as it ensures the comparator will operate correctly throughout the conversion process. On the other hand, we can see that, in the first half cycle, the output voltage in the DAC is 600 mV. After that, the voltage is the difference of $V_{CM} - V_{in}$, which gives $-200\,\text{mV}$ plus a half of the reference voltage resulting in 400 mV. As V_{CM} is higher than this difference, the comparator result is '1'. The next cycle compares the difference ($-200\,\text{mV}$) plus $(\frac{1}{2} + \frac{1}{4})V_{REF} = 900\,\text{mV}$ resulting in 700 mV. As this value is higher than V_{CM}, the output is '0'. The sequence continues until it arrives the last bit.

Concerning the code obtained from the conversion of 800 mV, we expect the value:

$$ADC_{out} = 2^n \frac{V_{in}}{V_{REF}} = 256 \times \frac{0.8}{1.2} = 170.7 \tag{4.31}$$

The comparison procedure truncates the value; then, the digital code for this input voltage is 170 equal to the obtained one.

Although the digital codes are coincident, we can only ensure the conversion is done correctly by calculating its ENOB. We also employ the relationship for a coherent sampling with seven complete cycles and 128 samples.

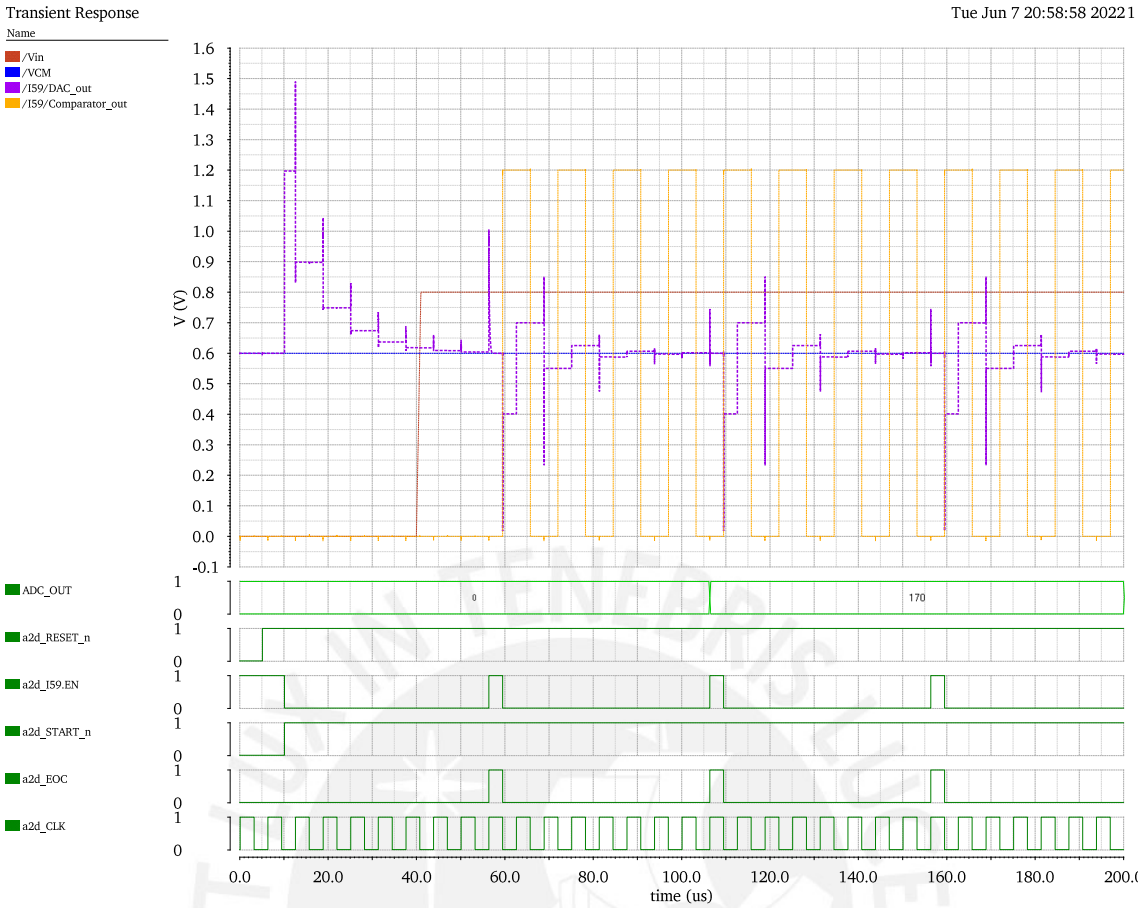


Figure 4.25: Transient simulation for single-ended 8-bit SAR architecture. The simulation runs until 200 µs as well. The digital results were 0 and 170 for 0 V and 800 mV, respectively.

Figure 4.26 presents the result for the sinusoidal input. As we can see on the left side of the image, the output range coincides with the expected values, from −500 mV to 500 mV. On the other hand, the output waveform is similar to the input and doesn't present errors for low voltages. For the spectrum calculus, we clip the waveform from 1.04 mV to 7.44 mV, which contains exactly seven complete cycles. On the right side, we can see the result with significantly lower values for other components (below 65 dB), apart from the fundamental located at approximately 1.09 kHz. This gives a more margin for the SNDR, which is 48.21 dB, according to the calculations from Cadence. For an ideal ADC with rail-to-rail input, the expected SNDR is:

$$\text{SNDR} = 6.02 \times 8 + 1.76 = 49.92\,\text{dB} \tag{4.32}$$

However, as the voltage input, in this case, is scaled, we should use:

$$\text{SNDR} = 6.02 \times 8 + 1.76 + 20\log_{10}\frac{1}{1.2} = 48.33\,\text{dB} \tag{4.33}$$

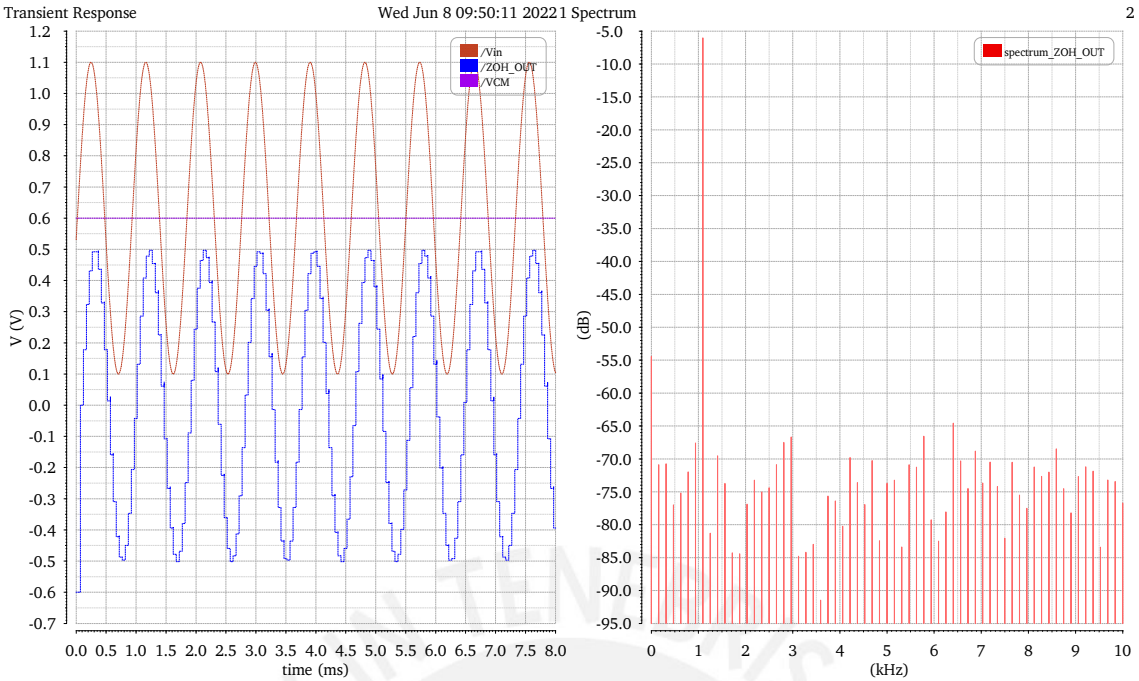


Figure 4.26: Transient simulation for single-ended 8-bit SAR architecture. The simulation runs until 8 ms to obtain the required seven complete cycles from which the spectrum is calculated. On the right is the transient response, while on the left is the spectrum obtained from the ZOH output. Unlike the previous architecture, the SNDR is 48.21 dB, and the ENOB is 7.716.

This value represents the maximum achievable SNDR with the input amplitude employed for 8 bits. If we compare it with our results, they are almost identical.

Cadence uses the classical ENOB formula for a rail-to-rail in its calculations, and applying it to the present values will return:

$$\text{ENOB} = \frac{48.21 - 1.76}{6.02} = 7.716 \tag{4.34}$$

However, we know that our input is not rail to rail, thus we need to adjust the formula to:

$$\text{ENOB} = \frac{48.21 - 1.76 - 20\log_{10} 1/1.2}{6.02} = 7.979 \tag{4.35}$$

With this result, we consider an appropriate design for reference. However, as the standard industry is differential architecture, we adapt this single-ended design to the required differential one discussed in the next section.

4.2.6.2 Differential architecture

The differential architecture for the SAR ADC shares various components of the single-ended architecture. It calculates the conversion for the difference between two input analog signals instead of a signal referred to ground. One of the main differences is the differential

DAC, which receives two inputs instead of one and outputs two signals. These signals are compared, and according to the result, the logic establishes the output value. A referential scheme for this configuration is presented in Figure 4.27. Here, we can notice that the inputs for the comparator are populated either by the DAC outputs or directly the analog inputs V_{in_p} and V_{in_n} selected from switches controlled by the EN signal.

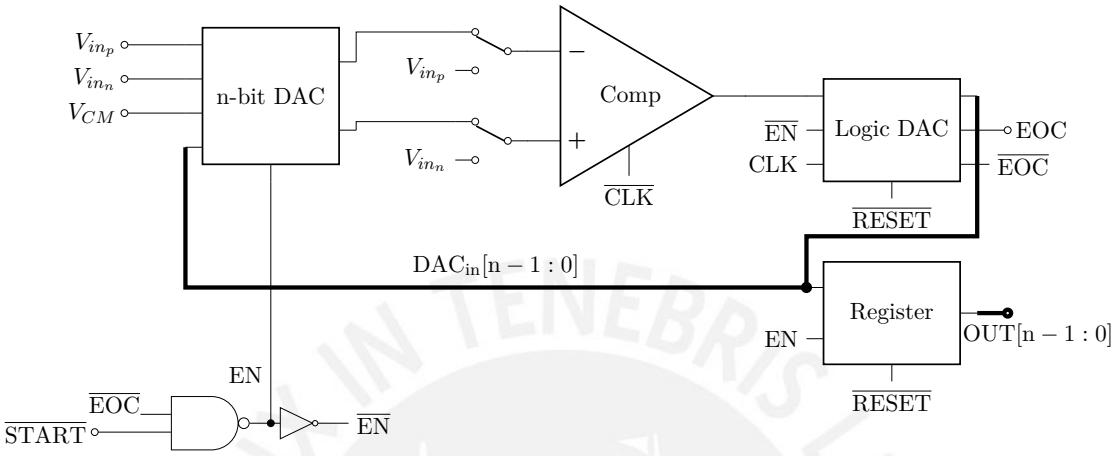


Figure 4.27: Scheme for differential SAR ADC based on a differential DAC with sampling function. The differential inputs for this circuit are V_{in_p} and V_{in_n}, while the control inputs are the $\overline{\text{RESET}}$, which stands for the inverse of the reset function, the $\overline{\text{START}}$, which initiates the conversion sequence; and the CLK, which is the clock that synchronizes the operation. It also uses a common voltage V_{CM} employed internally in the DAC. The outputs are EOC, which stands for the end of conversion, and the OUT[n − 1 : 0], which is the digital output.

Although the scheme is similar to the single-ended version, some details deserve special attention. Equal to the single-ended, the initialization sequence starts with a RESET signal, followed by the START pulse which also activates the EN signal. During this step, the V_{in_n} is compared against V_{in_p}. We choose an inverse polarity for this step to obtain '1' if V_{in_n} is greater than V_{in_p}, contrary to the usual scheme, to obtain directly a signed digital output (different from the single-ended whose output is unsigned). Therefore, if it is negative, the output will start with '1'; conversely, if the difference is positive, it will start with '0'. As we analyzed in the differential DAC section, we used an architecture with an additional capacitor to obtain the complete resolution output. However, we also identified that it would add a conversion cycle. To solve this, we limit the range of the differential input. We assume that, although the differential input can be negative or positive, it will only range from $-V_{REF}/2$ to $+V_{REF}/2$. Then, the first magnitude bit is known. If the signal is positive, the expected next bit is '0'; if it is negative, the next bit

is '1'. On the other hand, while the logic can preserve this sequence, we need to identify the required inputs for the differential DAC.

As adverted in the DAC section (where we assumed an equal behavior to the single-ended case), we need special logic to obtain a functional differential ADC. We recall that the SAR algorithm subtracts the original value against a reference generated from the bits in the DAC. In the differential case, we are working with signed signals that can be positive or negative; therefore, to make valuable comparisons with the reference, it could have the appropriate sign.

From (4.18) we observe that the output for the sign bit $V_{out_p} - V_{out_n}$ is equal to $V_{in_n} - V_{in_p}$ which returns directly the correct value ('1' for negative and '0' for positive). However, for the first magnitude bit, recalling the equations (4.21) and (4.24), we identified that the comparison is meaningless if we maintain its value in the DAC. Then, we need to invert it, which also inverts the comparison reference ($V_{REF}/2$) sign and obtain a worthwhile comparison term. If we make this change, the equations (4.21) and (4.24) transform to the following:

- If the difference $V_{in_p} - V_{in_n}$ is positive, the sign bit is '0' and the first DAC bit is '1':

$$V_{out_p} - V_{out_n} = V_{in_n} - V_{in_p} + \frac{1}{2}V_{REF} \tag{4.36}$$

- If the difference $V_{in_p} - V_{in_n}$ is negative, the sign bit is '1' and the first DAC bit is '0':

$$V_{out_p} - V_{out_n} = V_{in_n} - V_{in_p} - \frac{1}{2}V_{REF} \tag{4.37}$$

We can notice from the equations that a '1' in the DAC branch adds the scaled V_{REF} to the difference while a '0', subtracts it. From the equation (4.36), if the difference was greater than $V_{REF}/2$, it should be compared in the next iteration against $3V_{REF}/4$, requiring a '1' value in the next bit. However, the comparator output in (4.36) returns '0'. In the negative case, if the value was greater than $V_{REF}/2$, in the next iteration, the subtraction should be $3V_{REF}/4$, requiring a '0' value. However, the comparator returns a '1'. This suggests a pattern where the final input for the DAC is the opposite of the comparator. This can be solved by using the negative output of the dynamic comparator.

Once the magnitude and sign bits are solved by inverting the input for the DAC through the logic block, we need to develop a strategy to obtain the first magnitude bit to complete the conversion in exactly eight clock cycles. Considering that scenario, if the

difference is positive, the first bit will be 1, with the corresponding first magnitude bit 0, and conversely for the negative case. Therefore, we use the inverted sign bit value for the magnitude bit, set simultaneously in the DAC, just after the first comparison.

Implementing this circuit we also set all digital blocks' transistors to a width value of 300 nm, including the switches. On the other hand, as stated in the design stage, we reduced the base capacitance to 15 fF as we increased the total DAC capacitance, and also we obtained better results in the simulations with this value than with the previous employed.

We also employ two transient experiments to test the architecture's performance. One to verify the behavior when converting a fixed voltage input and the other to obtain the ENOB measurement. In the first experiment, we use a square wave in the positive, which alternates into two values to obtain the conversion for both. Those values are relative to the fixed constant voltage at the negative input. The test time is 250 µs to give enough for at least two consecutive converted values for each test voltage. We recall that our operating frequency is eight times the sampling frequency, which is 20 kHz, equal to previous tests. Then, the samples are taken every 50 µs while the clock period is 6.25 µs.

Figure 4.28 presents the result for the transient simulation with constant input voltages. The differential value for the input is −400 mV and 400 mV. The expected value for them are:

$$ADC_{out} = 2^n \frac{V_{in}}{V_{REF}} = 256 \times \frac{400}{1.2} = 85.33 \tag{4.38}$$

Then, as the digital output is truncated, for the positive voltage, it will be 85, while for the negative, it will be -86. Those are the same numbers we obtain in the experiment.

Apart from the digital values, which confirm that the differential architecture is working at least for constant values, we can observe some details that are characteristic of this proposal. First, we can see that both DAC outputs vary along the conversion cycle, different from single-ended, where the unique DAC changes while the other input for the comparator is fixed. On the other hand, we are using the negative output of the comparator as input for the logic, which is set consequently to the DAC input. In addition, in both input cases (positive and negative difference), we can observe that both DAC outputs converge to a voltage that coincides with the V_{CM} less or plus half of the input difference. This can be explained from the equations (4.16) and (4.17). Considering the

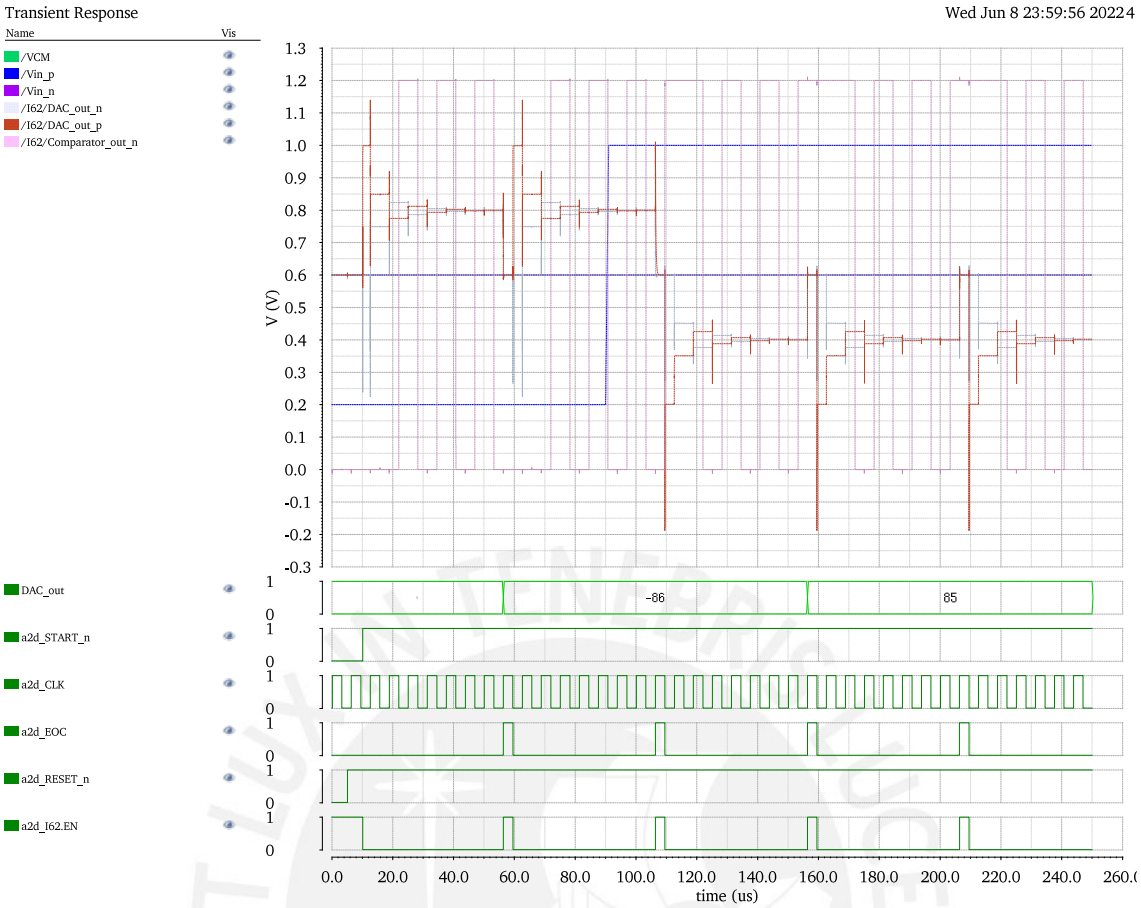


Figure 4.28: Transient simulation for differential 8-bit SAR architecture. The simulation runs until 250 µs. The digital results were -86 and 85, for −400 V and 400 mV as differential voltages input, respectively.

positive difference, $V_{in_p} = 1\,\text{V}$ while $V_{in_n} = 600\,\text{mV}$:

$$V_{out_p} = 600 - 1000 + 600 = 200\,\text{mV} \tag{4.39}$$

$$V_{out_n} = 600 - 600 + 600 = 600\,\text{mV} \tag{4.40}$$

Those values can be observed in the simulation output.

Then, the convergence voltage is located in the middle of those outputs which is 400 mV. Analytically, it can be calculated from:

$$V_C = \frac{V_{out_p} + V_{out_n}}{2} = V_{CM} + \frac{V_{REF}}{2} - \frac{V_{in_p} + V_{in_n}}{2} \tag{4.41}$$

In our case, the negative input V_{in_n} equals half of V_{REF}; therefore, for this setup:

$$V_C = V_{CM} - \frac{V_{in_p} - V_{REF}/2}{2} \tag{4.42}$$

The same applies to the negative value where $V_{in_p} = 200\,\text{mV}$ while $V_{in_n} = 600\,\text{mV}$. Then, the output voltages are $V_{out_p} = 1\,\text{V}$ and $V_{out_n} = 600\,\text{mV}$. Finally, the convergence voltage is 800 mV which is exactly the value obtained in the simulation.

This value is important as we have identified the problem with the comparator when both inputs vary in the first single-ended architecture with independent sampling and hold circuit. Their oscillations produce that the comparator goes off the correct operation region and outputs wrong comparison values. The differential architecture doesn't suffer for it, at least in the range we choose for the test. However, we should consider it if the voltages approach the limits of the full range, especially to the upper limit, as the convergence voltage will be low, and we don't want this convergence level to go beyond 300 mV.

Once we have verified the proper functioning for constant values, we will analyze its behavior with a sinusoidal input. For that, we choose a value for the input frequency that will help to obtain a coherent sampling, with seven complete cycles, 128 samples, and 20 kHz of sampling frequency. Something important to notice here is that the digital output is connected to a bipolar ideal DAC. It is modeled with Verilog-A, and it is similar to the one used in the single-ended case, with the difference that this accepts signed digital codes, as it is the type returned by the differential ADC. Likewise, the sign of the output is obtained directly from the digital code, and we don't need to subtract the DC component because the converted signal has only AC components.

The result of this simulation is presented in Figure 4.29. The image shows the transient response for a sinusoidal input in the positive terminal of the ADC while the other is fixed at $V_{REF}/2$ (600 mV). The output waveform is almost identical to the one obtained with the single-ended version.

To evaluate the precision of the conversion, we need to get the SNDR. According to the calculations in Cadence, this value is 48.119 dB. From the results in equations (4.32) and (4.33), we can observe that the maximum achievable SNDR for 8 bits, in the case of rail-to-rail input, is 49.92 dB, while for the input range we employ, it is 48.33 dB. If we calculate the ENOB using the conventional formula, it will be:

$$\text{ENOB} = \frac{48.119 - 1.76}{6.02} = 7.70 \tag{4.43}$$

While if we use the adapted formula that considers the input amplitude, the result is:

$$\text{ENOB} = \frac{48.119 - 1.76 - 20\log_{10} 1/1.2}{6.02} = 7.963 \tag{4.44}$$

Although not high as the result in the single-ended with a fixed reference, the value is acceptable to take as a reference against which we will compare the level crossing architecture performance. In addition, we must consider that this simulation includes transient noise until 100 MHz, which makes it more realistic, as well as the ENOB measurement.

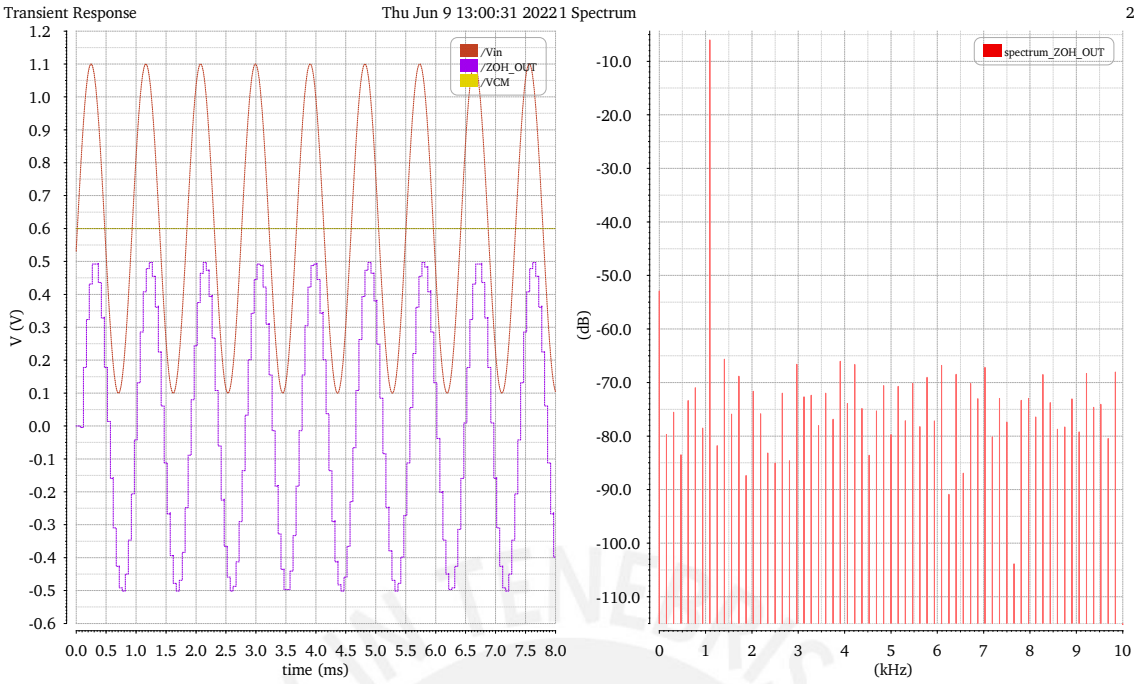


Figure 4.29: Transient simulation for differential 8-bit SAR ADC. On the left, the image shows the transient response while the negative input is fixed at 600 mV, and the positive input oscillates between 1.1 V and 100 mV. On the right is the resulting spectrum for seven complete cycles. According to Cadence, the resulting SNDR is 48.119 dB with an ENOB, equal to 7.70.

4.3 Level crossing ADC design

We will use the SAR ADC as a reference for the precision and power dissipation in the acquisition process. However, this thesis's objective is to use the LCADC architecture, which we expect has lower power dissipation under certain conditions characteristic of the intracortical signals.

From the designs for low-power applications such as in biomedical interfaces, the LCADC can be of two main types: with a floating-window comparison (uses N-bit DAC) or with a fixed-window comparison (uses 1-bit DAC) [10]. On the other hand, according to the results obtained by Van Assche and Gielen [10], when comparing the architectures against the SAR ADC, they found that the architecture based on fixed window (1-bit DAC) is more competitive in terms of power dissipation than the floating window; thus we will employ this architecture as reference for the LCADC design in this section.

The base architecture we use is the one proposed by Li, Zhao, and Serdijn [63], which includes some improvements from their previous works in [64], such as the architecture of the 1-bit DAC. It is extended in [65] as part of a whole circuit for an ECG acquisition system.

4.3.1 Building blocks

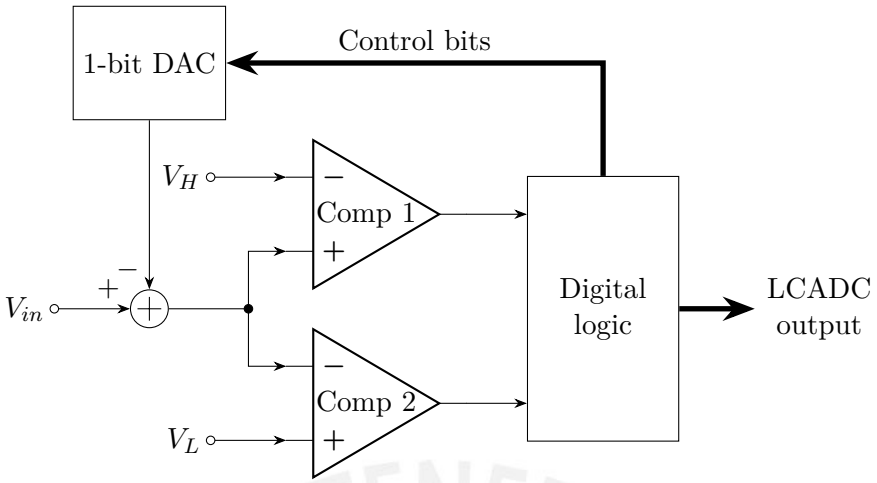


Figure 4.30: General architecture for an LCADC with a fixed window of comparison. The input signal is V_{in} which is subtracted by the 1-bit DAC output and compared against two comparators to define if the DAC should subtract the next upper or lower reference level. Based on [65].

The basic idea behind the level-crossing ADC is obtaining the corresponding sample of the analog input almost exactly at the time when it crosses a certain reference level. In the case of the fixed window variant, we only have two fixed reference levels (upper and lower level) defining the comparison window. Then, we must adapt the sampled voltage to fit into that window. As long as this new sampled voltage is within the window, the corresponding code in the digital output doesn't change. When it goes above or below the reference voltages, it adds or subtracts a bit to the digital output and performs an analog operation to reinsert the signal into the comparison window.

Figure 4.30 shows a conceptual model for the fixed-window LCADC. Here, the 1-bit DAC tracks the input voltage based on the comparison result against the reference voltages. If the track voltage is under the lower reference V_L, the DAC adds an offset to return to the comparison window, while if it is upper than the higher reference V_H, it subtracts that offset. This requires a special logic that controls the DAC depending on the values obtained from the comparators. Likewise, it uses a counter that stores the digital value for the output, whose count changes each time the tracking voltage goes off the limits of the comparison window. All these blocks used in this architecture will be developed in detail in the following sections.

4.3.2 One-bit DAC

This is one of the main components of the LCADC. It is responsible for keeping the comparison signal in a fixed window while comparing against the fixed reference levels.

The operation's principle for this circuit is the charge sharing in switching capacitors. The first proposed circuit, presented by Li *et al.* [64], only had two capacitive branches: one acts as the tracking branch while the other performs the offset injection. The advantages of this circuit against others from the literature are: an input not limited by the reference, a lower power consumption, and the tolerance to charge injection and clock feed-through. Later, the author in [63] improved the original architecture and introduced a third branch for offset injection. This additional branch reduces the settling time to reach the respective references before it is connected to the tracking branch.

A similar structure to the one proposed in [63] is presented in Figure 4.31. The variations are the additional switches controlled by $\phi_{\overline{\text{RST}}}$. The operation for this circuit resides in the charge distribution from a previously charged capacitor. The procedure starts with the reset signal ($\overline{\text{RST}}$). When it is active ('0') and connects the branches to the medium voltage V_M, which is:

$$V_M = \frac{V_H + V_L}{2} \tag{4.45}$$

where V_H is the upper reference voltage and V_L is the lower reference voltage.

During this time, both branches are connected, and the output voltage is $V_{out} = V_M$. In this stage, we can connect the upper node to V_{in}, and we will start the comparison and tracking exactly in the value of V_{in}. At this moment, we disconnect from V_M. The other switches are grouped in sets "1" and "2", which can be identified by their numeral sub-index. We can start with any set of switches. For example, the switch ϕ_1. It also connects the switches ϕ_{1L} or ϕ_{1H}. Initially can be any; then, when the comparison with V_M is significant, it changes to the correct reference. When one set of switches is connected, the other is disconnected, and when they alternate, there is a non-overlapping time, which avoids that both switches ϕ_1 and ϕ_2 are enabled simultaneously.

Supposing that the initial $V_{in} = V_{in_0}$, if the voltage changes to V_{in_1} while only the ϕ_1 and ϕ_{1L} are connected, the change in V_{out}, initially in V_M, can be obtained considering the previously stored charge as a voltage source which will be equal to $V_{in_0} - V_M$ for both C_U capacitors and V_M for both C_D capacitors. Then the output voltage for the new V_{in_1} will be:

$$V_{out_1} = V_M + (V_{in_1} - V_{in_0})\frac{C_U}{C_U + C_D} \tag{4.46}$$

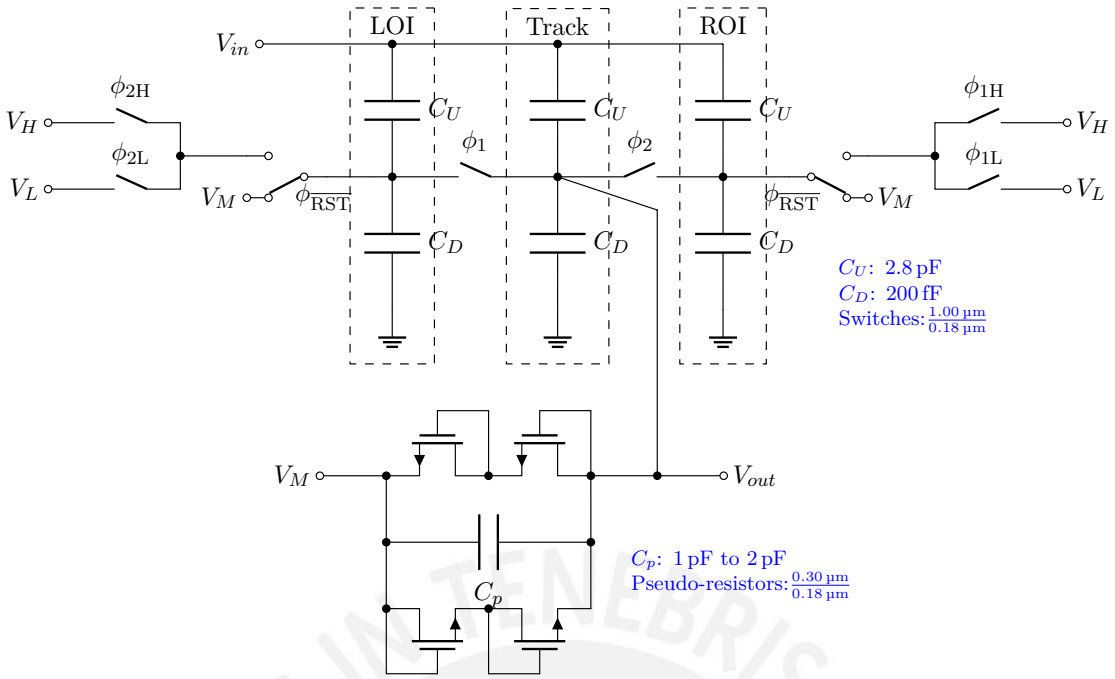


Figure 4.31: Circuit scheme for the 1-bit DAC employed in the LCADC architecture. Based on [63]. It adds reset switches to the original circuit and a capacitor in parallel with the pseudo between the output and the mean voltage V_M.

As long as C_U is higher than C_D the added voltage to the output will be closer to the increasing voltage in the input.

After the V_{out} has reached a reference level, for example, V_H, the DAC needs to inject an offset to keep the output into the comparison window margins $[V_L; V_H]$. This injection occurs due to the change in the set of switches, in this case, to "2". This modifies the current voltage V_{out} when connecting to the pre-charged capacitors that store V_L in the right offset branch (ROI) by ϕ_2. The new voltage after the ϕ_2 connection results from the charge sharing between capacitors, considering the total charge of the plates in the output node before and after the connection. After ϕ_1 is disconnected, and before ϕ_2 is connected, the total charge in the tracking branch (Q_{track}) and the offset branch (Q_{ROI}) are:

$$Q_{track} = (V_{out} - V_{in})C_U + V_{out}C_D$$

$$Q_{ROI} = (V_L - V_{in})C_U + V_L C_D$$

Next to the ϕ_2 connection, the total charge will be equal to:

$$Q_{total} = 2(V_x - V_{in})C_U + 2V_x C_D$$

We can equal the total charges in both phases:

$$Q_{track} + Q_{ROI} = Q_{total}$$
$$(V_{out} - V_{in})C_U + V_{out}C_D + (V_L - V_{in})C_U + V_L C_D = 2(V_x - V_i n)C_U + 2V_x C_D$$
$$V_{out}(C_U + C_D) - 2V_{in}C_U + V_L(C_U + C_D) = 2V_x(C_D + C_U) - 2V_{in}C_U$$
$$V_x = \frac{V_{out} + V_L}{2}$$

This result suggests that the output voltage after the injection equals the mean between the previous output voltage and the reference voltage pre-charged in the offset injection branch. Consequently, if the last voltage $V_{out} = V_H$, the new voltage will be V_M. This is the desired behavior to keep the tracking voltage always in the window margins of V_H and V_L. Conversely, when the input voltage is descending, we use a pre-charged voltage V_H activating the switch ϕ_{1H} instead of ϕ_{1L}. Therefore, when the voltage reaches V_L, it changes, ideally, to V_M with the transition.

This behavior is ideal, as it supposes immediate switch transitions. However, comparators have delays, and the input signal is not steady in these transitions, as we assumed. It introduces some variations in the output, returning wrong values in comparators. Then, we need also to smooth the output signal employing a capacitor C_p connected between V_{out} and V_M, as introduced in Figure 4.31, in parallel with the NMOS transistors, which act as a pseudo resistor.

However, the addition of this capacitor, introduces a variation in the charge distribution, altering the slope in (4.46):

$$V_{out_1} = V_M + (V_{in_1} - V_{in_0})\frac{2C_U}{2(C_U + C_D) + C_p} \tag{4.47}$$

It also alters the charge injection modifying the stored charge when both switches are off:

$$Q_{track} = (V_{out} - V_{in})C_U + V_{out}C_D + (V_{out} - V_M)C_p$$
$$Q_{ROI} = (V_L - V_{in})C_U + V_L C_D$$

And after the offset injection:

$$Q_{total} = 2(V_x - V_{in})C_U + 2V_x C_D + (V_x - V_M)C_p$$

Calculating the new voltage in V_x it will be equal to:

$$\begin{aligned} Q_{track} + Q_{ROI} &= Q_{total} \\ V_{out}(C_D + C_U + C_p) - 2V_{in}C_U - V_M Cp + V_L(C_D + C_U) &= \\ V_x(2C_u + 2C_D + C_p) - 2V_{in}C_U - V_M C_p & \\ V_x = V_{out}\frac{C_D + C_U + C_p}{2C_u + 2C_D + C_p} + V_L\frac{C_D + C_U}{2C_u + 2C_D + C_p} & \end{aligned}$$

Recalling that $V_L = V_M - \text{LSB}$ and $V_H = V_M + \text{LSB}$, and taking into account that in the moment of offset injection $V_{out} = V_H$, then, we can simplify the equation to:

$$V_x = V_M + \text{LSB}\frac{C_p}{2C_u + 2C_D + C_p} \tag{4.48}$$

It indicates that the value after the injection won't be exactly V_M, but a slightly different, depending on the value of C_p and its proportion relative to the other capacitors.

In the circuit implementation, we use a width of 1 µm for all switches and a capacitance of 2 pF for the output. We choose them adjusting from simulations of the whole circuit. That was because we identified that offset injection produces a glitch at the output that overpasses the V_M reference and produces an activation of recurrent code logic which is undesired. The adjustment of the capacitor aimed to reduce that glitch until it was not able to trigger the recurrent logic.

On the other hand, the capacitors C_U and C_D are parametrized, and set in the higher level block, when we instantiate it in the complete LCADC architecture. The criteria to choose them was to use a $C_U \gg C_D$ to minimize the attenuation factor of the output signal, according to (4.47). Taking as a starting point the values used by Li, Zhao, and Serdijn [63] and after verifying with simulations we ended up with $C_U = 2.8\,\text{pF}$ and $C_D = 200\,\text{fF}$, which showed an acceptable level of performance.

As the control for this DAC has several signals, which are not necessarily periodical, we opt to test the block in the complete circuit. However, we can evaluate the functioning for a case of use, for example, when the signal is rising, and it needs to inject offset with V_L to keep it in the comparison window.

Figure 4.32 shows the result for the transient simulation of the 1-bit DAC. As expected from the theory, the voltage output equals V_M during the reset phase. Then, immediately after it is disconnected from the reference voltage V_M, the output starts tracking the input in a relationship that comes from (4.47). Replacing the values of C_U and C_D, it gives:

$$V_{out} = V_M + (V_{in} - V_{in_0})\frac{2 \times 2.8\,\text{pF}}{2(2.8\,\text{pF} + 200\,\text{fF}) + 2\,\text{pF}} = V_M + (V_{in} - V_{in_0})\frac{7}{10} \tag{4.49}$$

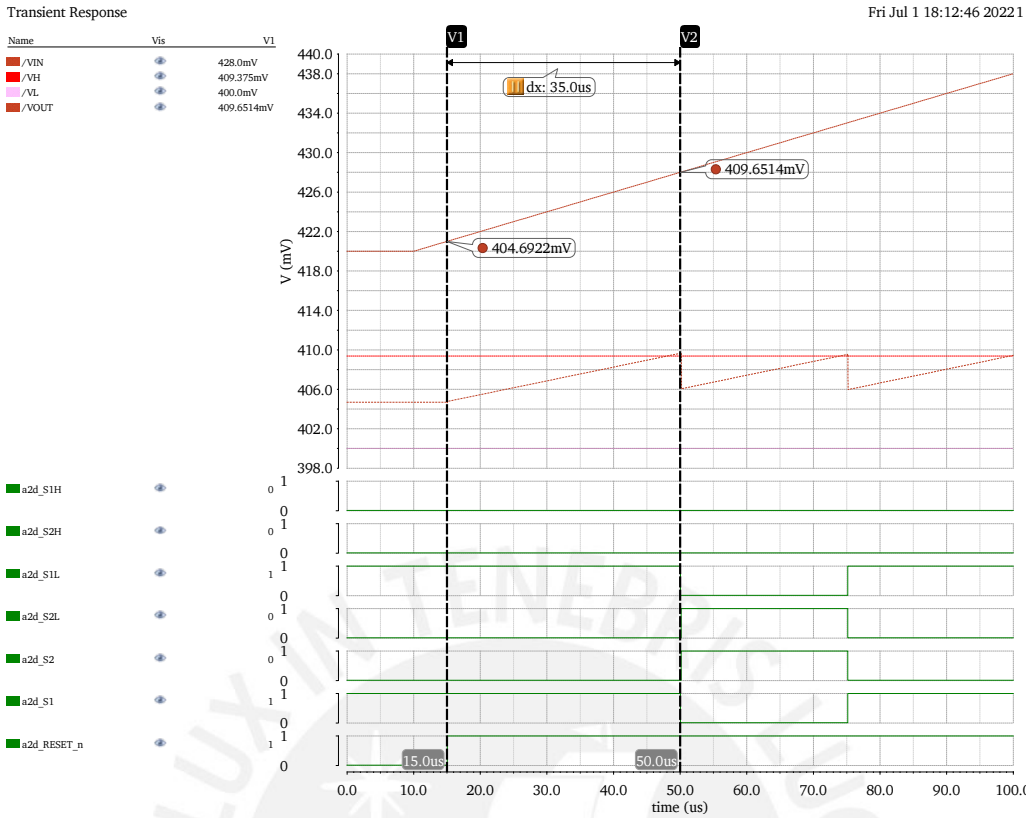


Figure 4.32: Transient result for simulation of 1-bit DAC. It evaluates a ramp signal and the offset injection when the output reaches the higher voltage reference level VH.

The input rate is 200 V/s (increase 20 mV in 100 µs). Then the expected rate will be $200 \times \frac{7}{10} \approx 140\,\text{V/s}$. The measured rate from the figure is:

$$\text{Output rate} = \frac{409.65 - 404.69}{0.035} = 141.7\,\text{V/s} \tag{4.50}$$

We can confirm that the output rate has minimal difference against the updated formula, although it is lower than the original value without the additional capacitor C_p.

The other effect of this capacitor is that the final output doesn't go to V_M after the offset injection, reaching a slightly higher voltage. This is explained by the equation (4.48). Therefore, using the output capacitor introduces some errors and undesirable effects; however, not introducing it produces glitches that also present errors but in the final digital code, significantly altering the output.

4.3.3 Logic blocks

To have a functional 1-bit DAC, the logic blocks are responsible for generating the control signals, which, in turn, perform the tracking, change the voltage reference, and inject the offsets, when necessary.

The origin of these control signals comes from the comparators' output. From them, we obtain two different types of information. The first is to know if the signal has exceeded the limits of the comparison window, either if it goes above the higher limit or below the lower limit. This means a code change requiring an output logic to generate the changed code signal. The second type of information comes while the signal is in the window, below or above the middle reference voltage V_M. From it, we can obtain the signal tendency, if it is rising or falling, and consequently, if it may exceed the lower or upper limit. Depending on that, we can use the higher reference V_H or lower reference V_L for the voltage injection. While the signal is in this range, the output has a recurrent code, and we use a special logic to generate the recurrent code control signal each time it crosses the V_M. Finally, the recurrent and the changed code signals are inputs for the DAC control logic, responsible for activating or deactivating its switches.

4.3.3.1 Recurrent code logic

This block generates the control signal that indicates that the input has crossed the V_M reference level and is either in the upper or the lower region of the fixed comparison window.

Figure 4.33 shows a scheme for the logic inside. When the input signal has crossed the crossover point and activates the inverter gate, it will set the logic value to the SR latch output. Then, if the logic value is '1', the UD will be '1'; conversely, when it is '0', UD will be '0', as well. When the transition occurs it generates also a pulse that indicates the state of UD has changed either to '1' or to '0'. The generated pulse is '0' when there is no change, and it is '1' when the input has changed, maintaining the value for a certain time, returning it to '0' again. The inverted pulse ($\overline{\text{C}}$) has the opposite behavior. We can obtain such behavior with a logical gate that outputs '0' when the inputs are equal and '1' when they are different (XOR gate). Therefore, we can use a delayed signal and a non-delayed as the inputs. When there are no changes, both are the same, returning '0'. But, when the non-delayed signal changes, the delayed one keeps the previous value, generating a difference for the inputs in the XOR gate and, consequently, returning '1'. It keeps this '1' level until the delay finishes, and it changes, returning the output to '0'. This operation can be expressed as $C = A \oplus A_d$, where A is the non-delayed input, and A_d is the delayed input. In the case of the inverted signal, we can also express the previous formula as its equivalent:

$$C = A \oplus A_d = \overline{A \oplus \overline{A_d}} \tag{4.51}$$

This is the expression that we used in Figure 4.33, for the pulse generator.

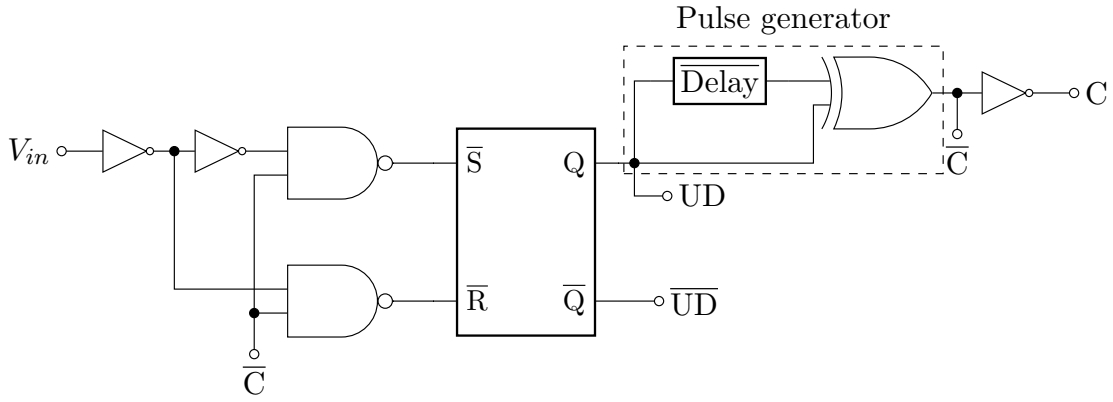


Figure 4.33: Scheme for recurrent code logic, based on [63]. This circuit receives as input the output from the comparator (V_{in}) and outputs the pulse for recurrent code (C) and the state of the signal if it is high (UD=1) or low (UD=0). Here, $\overline{\text{Delay}}$ returns the inverse value of the delayed input.

On the other hand, while the pulse is active, the SR latch is disabled to prevent the transition from influencing the final value of the UD signal. Therefore, we employ NAND gates as enablers in both inputs of the latch which are disabled when $\overline{\text{C}}$ is '0', corresponding to the time period when the pulse is active.

In the final implementation, we added a delay block to the outputs of UD to ensure that the change in other blocks, that depends on UD will occur after the C pulse is generated. However, this delay is minimal compared to the most significant delay that generates the C pulse. The capacitors' values in this component were obtained via simulations with the whole LCADC to avoid undesirable code changes during the UD transitions. For the widths of the transistors, we use the value of 300 nm, as employed in other digital circuits before.

The result of the transient simulation is presented in Figure 4.34. Here, we observe that when the input changes to '1', the UD signal also changes to '1', and when it changes to '0', the UD signal changes accordingly. On the other hand, we can appreciate the generation of the C pulse in each transition, either in the falling or the rising edge of UD. Also, we notice that the pulse widths are slightly different for the UD falling and rising delays, being approximately 150 ns and 120 ns, respectively. Although, we don't consider that it could significantly affect the output for the ADC.

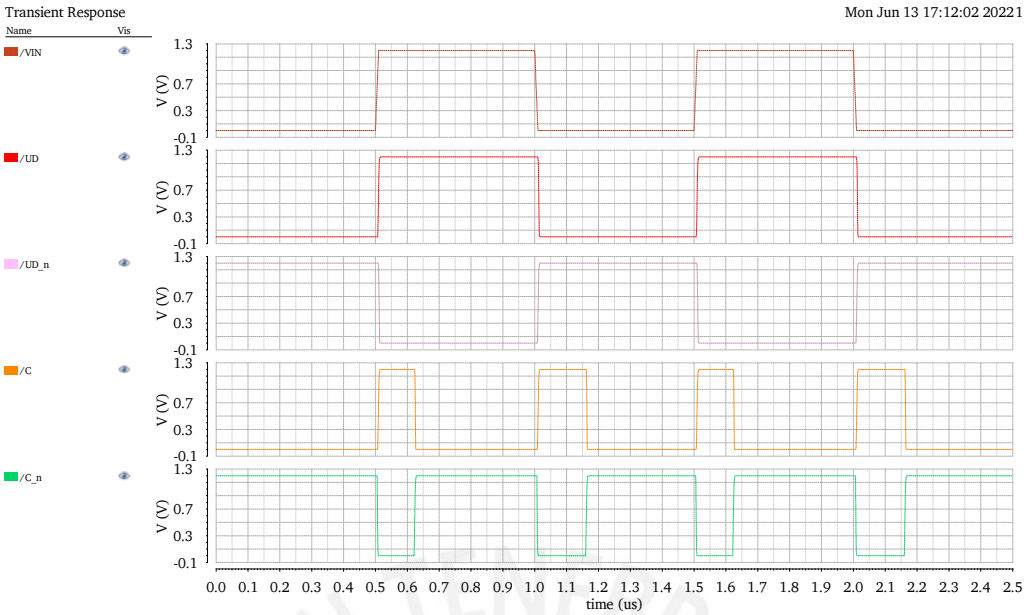


Figure 4.34: Transient result for simulation of the recurrent logic circuit. It evaluates the control signals when the input alternates between 1 and 0.

4.3.3.2 Changed code logic

This logic is pretty simple and can be directly obtained from the comparator output with a simple buffer. We implement it in the whole circuit and we don't design it as an individual block as the recurrent code logic.

4.3.3.3 DAC control logic

This block generates all the control signals for all the switches in the DAC circuit. It will alternate between the set of switches "1" and "2", and decide if the pre-charged voltage in the offset branch will be V_H or V_L, activating the corresponding switches.

Figure 4.35 shows a scheme for this circuit. It is based on a flip-flop whose clock signal comes from the changed code logic (C). Each time this input signal has a rising edge; it activates the flip-flop, alternating its value between '0' and '1'. Then, the flip-flop outputs control the switches "1" and "2". In addition, the state signal UD determines which offset switches are enabled. If the UD signal is low (UD = 0), the input signal is in the lower region of the comparison window, requiring an offset that could increase the voltage value. Then, the control connects to a V_H reference by switching S_{1H} or S_{2H}. Conversely, when $UD = 1$, the signal requires an offset that could reduce the voltage using the reference V_L. It corresponds to the switches S_{1L} or S_{2L}.

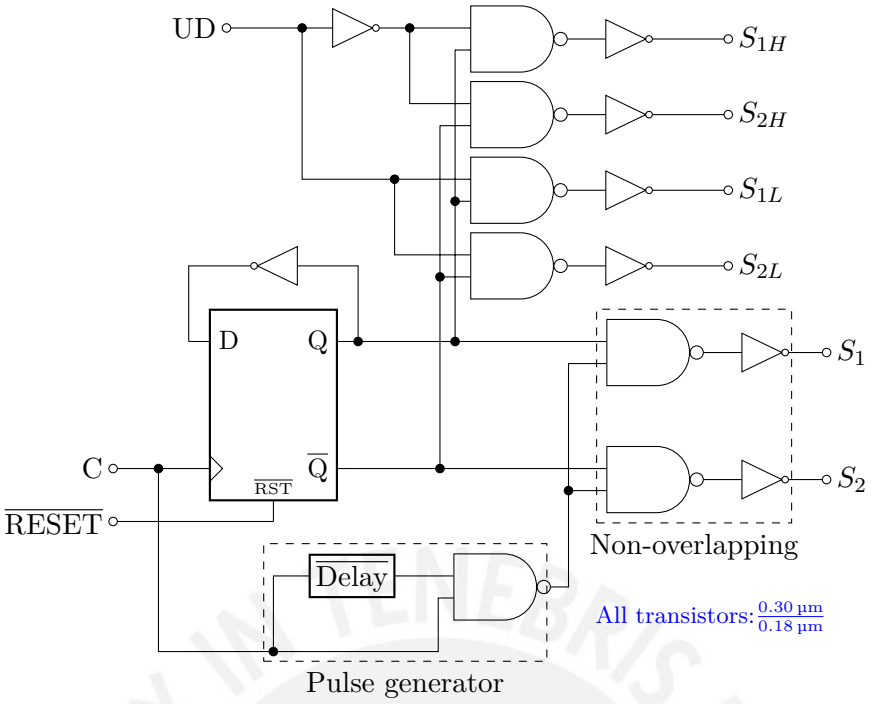


Figure 4.35: Scheme for DAC control logic. This circuit receives the change code from the recurrent code and changed code logic (C) and the state of the signal in the comparison window from UD. Then it outputs all the signals for DAC switches.

In addition to this, the output for switches S_1 and S_2 must be non-overlapping. To obtain that behavior, we generate a pulse when a rising edge occurs on the C signal input, and during this pulse, we set the switch control outputs to '0'. The pulse can be obtained with a NAND gate and a delay block. The NAND gate will ensure that it will generate a pulse only on the rising edge. We can analyze the result first considering an unchanged signal. The result of the original signal A and the delay signal A_d will be '1' for the NAND, whether A is '0' or '1'. Then, if A changes from '1' to '0' (falling edge), the inverted delay signal is '0', and the result is still '1'. However, if A changes from '0' to '1' (rising edge), the inverted delayed signal is '1', and the result is '0'. It keeps the state until the inverted delayed signal changes to '0' and the NAND output returns to '0'. The generated pulse temporarily disables the switches S_1 and S_2, setting them to '0', preventing that in, the transition, both activates simultaneously.

In the circuit implementation, we also added an enable signal for the switches. On the other hand, we set a capacitor value of 200 fF for the delay block, similar to the short delays we use on other circuits which just provide tens of nanoseconds of delay used for the overlapping circuit. For the transistor widths, we use 300 nm, as in other digital circuits.

To test the block's behavior, we use a periodic waveform with a frequency of 5 M, as in previous circuits for the LCADC architecture.

The result of the transient simulation is shown in Figure 4.36, which shows the result for each rising edge of the C input. It alternates the switches "1" and "2" while the UD keeps the same value. This alternation from "1" to "2" changes the set of switches but keeps the reference voltage associated with, i.e., from S_{1L} to S_{2L} or S_{1H} to S_{2H}. However, when the UD changes, the associate reference also changes, maintaining the set of switches. For example, if the active set of switches is "2" and the UD changes from '0' to '1', the switch alternation is from S_{2H} to S_{2L}.

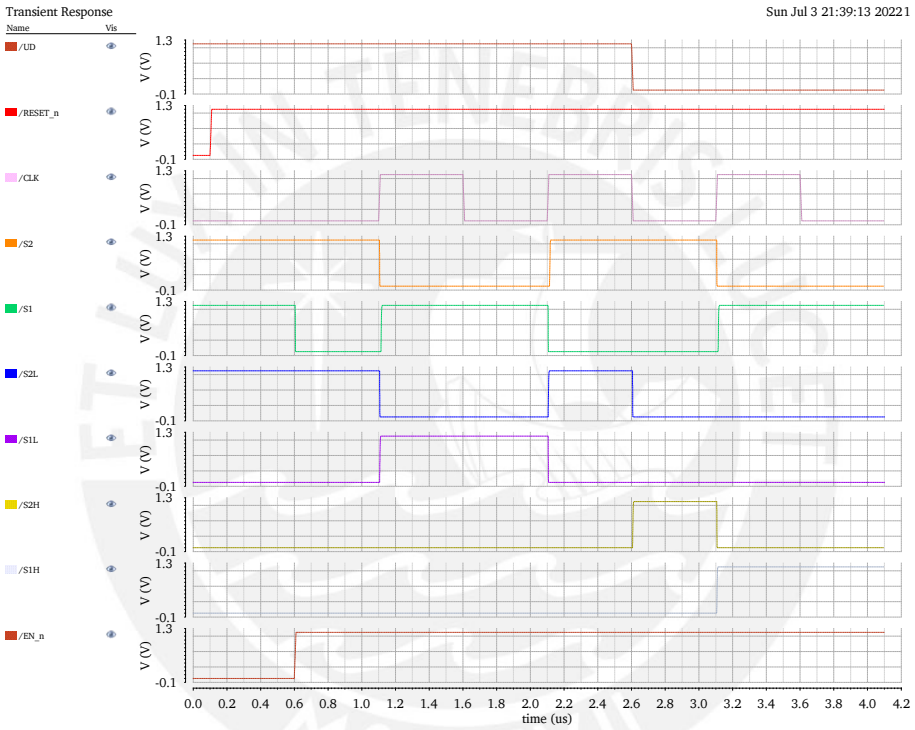


Figure 4.36: Transient result for simulation of DAC control logic circuit. It shows the control signals when the signal state UD alternates between 1 and 0, and when the change code input is triggered.

4.3.4 Counter

This circuit stores the output value for the LCADC architecture. This counter can increase or decrease the value depending on a state signal UD, changing it at each clock cycle.

The increase can be implemented by the addition of '1' to the previous count value and can be expressed as:

$$S = A + 1 \tag{4.52}$$

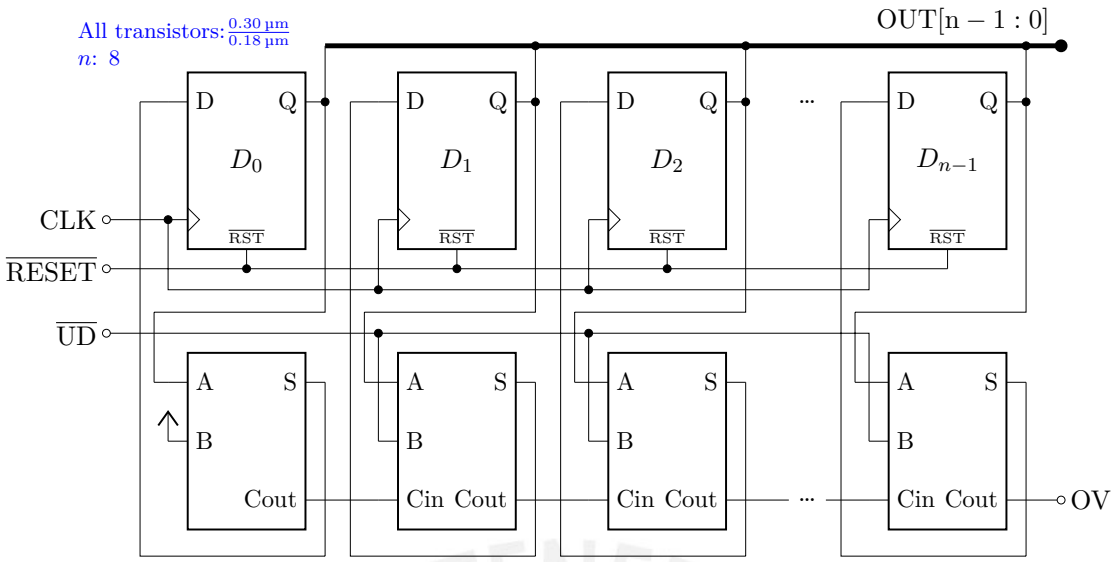


Figure 4.37: Up/down n-bit counter scheme. Here, $\overline{\text{UD}}$ is the control signal defining whether the count goes up or down. CLK is the clock signal that executes the count and OV is the overfull flag, meaning that the count has exceeded the maximum value.

On the other hand, the decrease or subtraction can be expressed as the addition by the two's complement of '1', which is $\overline{111\ldots1}$. As we can see, the expression for the counter output can be written as:

$$S = A + \overline{XXX\ldots X1}, \quad \text{where: } X = 0 \text{ (up)}, \ X = 1 \text{ (down)} \tag{4.53}$$

Figure 4.37 presents the circuit that translates this operation. This circuit shows a counter with UP/DOWN option. We implement the counter output with flip-flops while for the next state, which is the input for the flip-flops, is implemented with the expression in (4.53) through an adder that sums the previous count plus the number $\overline{XXX\ldots X1}$. For the first bit, we employ a half adder with a constant input in VDD representing the '1' in the expression. For the other bits, we use full adders, which receive the carrier outputs from the previous bit plus a value that depends on the signal $\overline{\text{UD}}$, where $\overline{\text{UD}} = 0$ executes the up count and, $\overline{\text{UD}} = 1$ performs the down count.

For the circuit implementation, we use a minimal width $W = 300\,\text{nm}$ and length $L = 180\,\text{nm}$ for the transistors in each digital block considered for this design.

Figure 4.38 shows the transient simulation. Here, we can see how the value in the output increases, reflected in the scaled pulse widths of individual bits (the pulse of one bit is twice the pulse width of the previous one). Something to notice is the activation of the overflow flag, enabled in the last value. It is not a registered output; however, if required, it can be stored in an additional flip-flop.

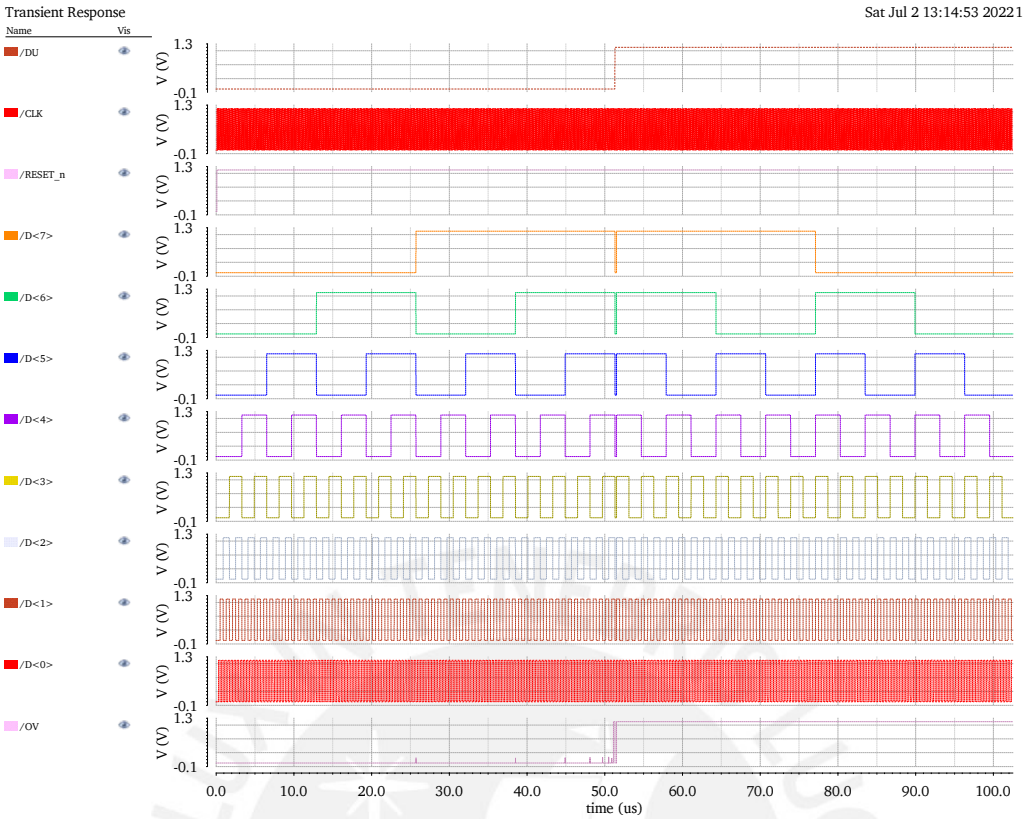


Figure 4.38: Transient result for the up/down counter circuit simulation. It evaluates the circuit through each value the counter could have ascending or descending the count.

4.3.5 Comparator for the LCADC

Differently from the SAR ADC, where the comparisons occur only in the positive edge of the clock, in the LCADC, the comparator is always active, as the next code could occur at any time. The latch-based architecture employed on the SAR is not useful in this case. Therefore, an operational amplifier (OPAMP) based architecture is the most appropriate. The disadvantage of keeping the comparator always active is that it drains static current with a fixed power dissipation during the whole circuit operation.

Considering this, we will reduce the current bias in all the comparator stages to reduce the whole comparator power dissipation. In addition to this, we will take as main design parameters the delay and offset, as according to simulations that we have done, the architecture is more sensitive to their variations (long delays and large offsets contributes to reduced precision in the sampled time for the converter).

We have explored several architectures here, for which we calculated the delays and offset, and they will serve as a reference when probing the whole circuit architecture.

4.3.5.1 Three-stage comparator

This is the original architecture presented in the article we are basing our design in [63]. It has an initial differential stage, followed by a second-stage differential amplifier, with a current mirror as a load that outputs to a single-stage amplifier. The original article has different values for the transistors as the operating voltage (0.8 V), distinct from the voltage in our design (1.2 V). Initially, we scaled up to work with higher voltage; however, it increased the total current, which is undesirable due to power consumption. Then, we needed to tune the dimensions to achieve a reduced value.

Figure 4.39 presents the topology for this architecture. Here, the voltage bias, identified as V_{bias}, is set to 600 mV to achieve an overdrive voltage of approximately 150 mV.

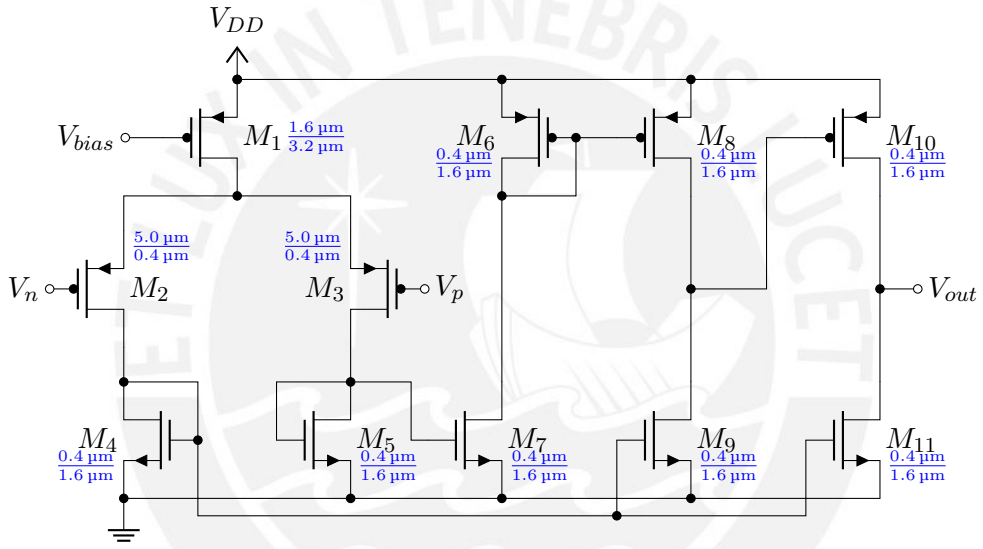


Figure 4.39: Three-stage comparator. Based on [63]. The topology follows the same as in the original paper; however, the dimensions are adapted to the current operation voltage 1.2 V.

In the circuit implementation, the relationship $W/L = 0.25$ for the loads and the single branches while the differential has a relationship of $W/L = 0.5$ to drive the double of current and has equivalent current densities. On the other hand, the differential pair has a proportion of $W/L = 12.5$ ($gm/I_d = 25$) to increase the transconductance and have a better response against noise.

The result of the simulation is presented in the Figure 4.40. It uses a square input for the transient simulation in the positive input, with 1 ns of rising time and a period of 1 µs. As the graphic shows, the delay for the rising edge is around 34 ns; however, for the falling edge, it takes almost 100 ns to achieve the % 50 of the output. That would be a problem when the output must go from '1' to '0'. We can explain this response due to

the different delay sources for the output stage. While the rising edge (PMOS) uses the three-stage amplification, the falling edge (NMOS) only uses a two-stage amplification. Although it is the original proposal in the reference article [63], we consider changing it to obtain a more symmetric response.

In the case of the offset, it is in a fair value of around 1 µV, which may increase to the millivolts order due to the mismatch effect.

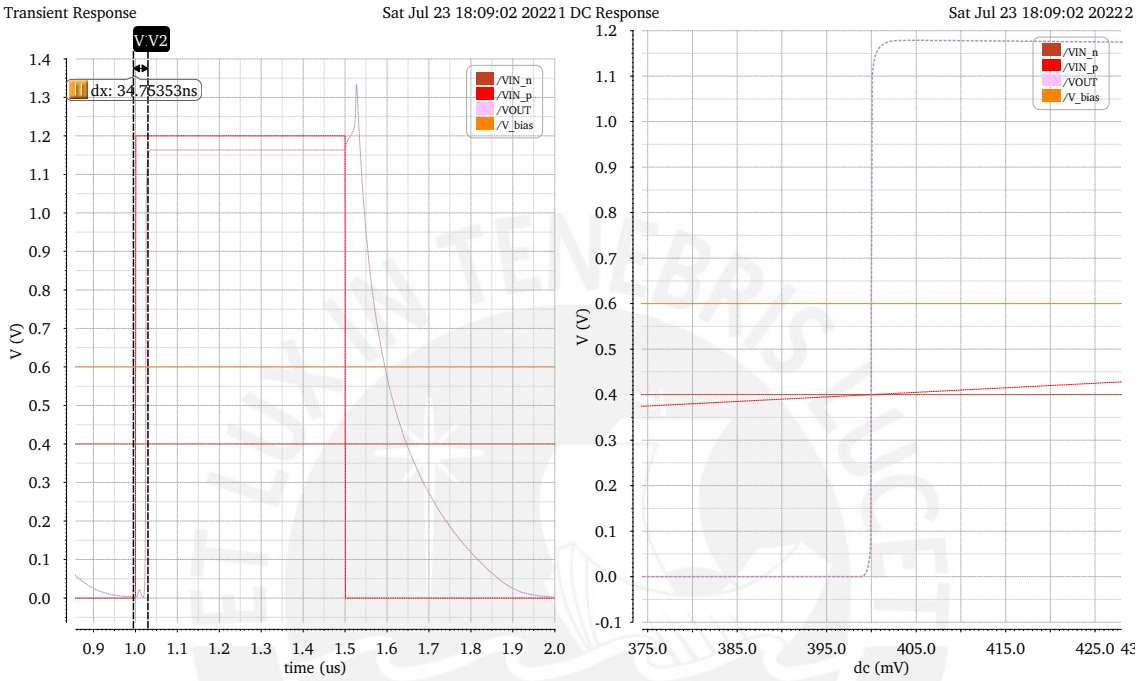


Figure 4.40: Result for simulation of the three-stage comparator. On the left, it shows the transient response where we can calculate the delay, around 34 ns. On the right, it shows the DC response, which calculates the offset. As it is tiny, it cannot be visualized, but according to the calculator is around −1.6 µV.

Apart from the issue identified, we also notice that this architecture consumes around 1.25 µA for the whole circuit, and consequently, we consider changing the architecture to reduce this value.

4.3.5.2 Differential stage and A-class amplifier

The previous circuit has the drawback of an important power dissipation level due to the three active stages. A proposed alternative consists of a simplified version that only includes the first differential stage and a single-stage common-source that acts as an A-class amplifier.

In this design, we scaled the dimensions from the previous architecture after a noise analysis to minimize the effect of thermal and flicker noise on them. We chose adequate

sizes for the output stage to reduce the systematic offset. Likewise, we added two inverter gates with minimum dimensions to respond quickly and improve the output slew rate. The scheme for this architecture is presented in Figure 4.41.

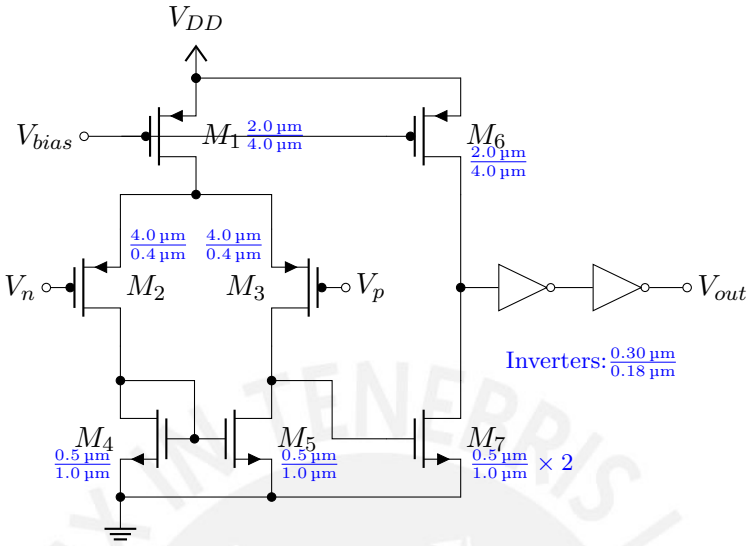


Figure 4.41: Comparator based on differential and simple A-class amplifier. The output is connected to two inverter gates to improve the slew rate. Those have minimum dimensions to increase the comparator speed.

In the circuit implementation, the expected tail current for the differential branch is 500 nA, which will rise up to 1 µA considering the output branch. On the other hand, the relationship for the current mirrors is $W/L = 0.5$. We also follow this relationship in the loads in the differential stage and in the common-source amplifier. The dimensions for the output inverter gates are set to L minimum and $W = 300\,\text{nm}$, as used in other digital gates, to have faster responses and short rising times.

The simulation result with this design is shown in Figure 4.42. The transient input is also a square wave with a rising time of 1 ns, while the period, in this case, is 200 ns. As depicted in the figure, it doesn't suffer from a huge delay for one of the edges. The delays for falling and rising are 12 ns and 19 ns, respectively. On the other hand, the offset is −5 µV which we also consider adequate for this architecture.

Although this circuit achieves lower response times and minimum offset, we test other variants, either to pursue lower power consumption, lower delay or to fix glitches that are present in the whole simulation scheme that cannot be corrected just with the logic blocks.

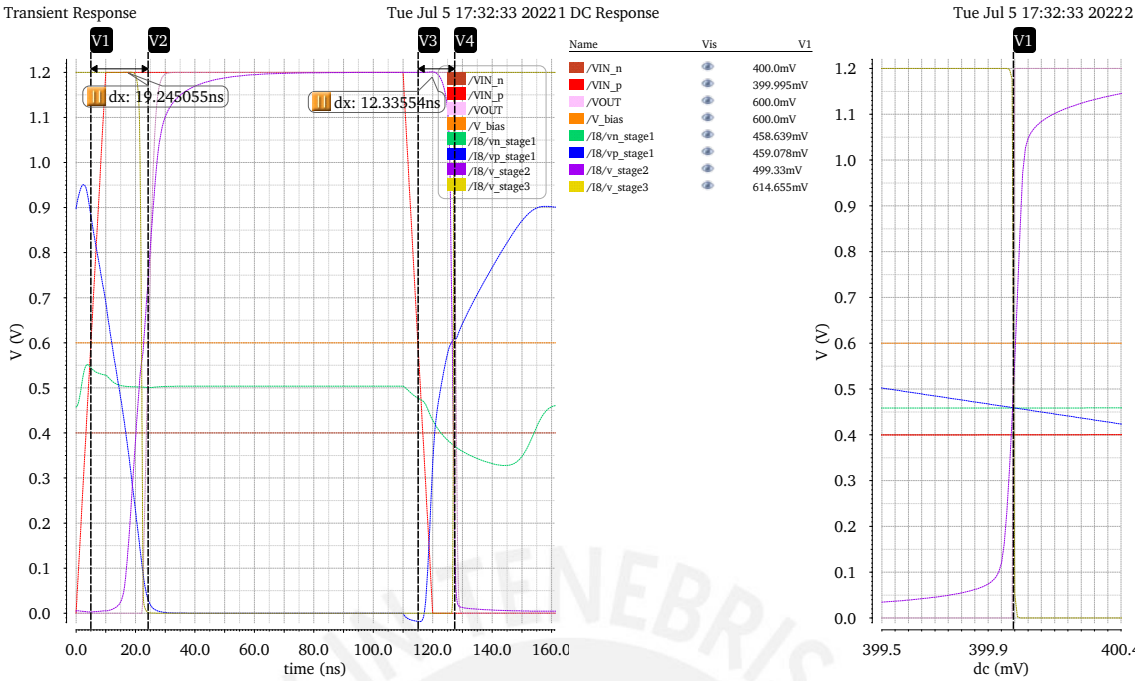


Figure 4.42: Result for simulation of the comparator with an A-class amplifier. On the left, it shows the transient response where we can calculate the delay, which is around 12 ns for the falling edge and 19 ns for the rising edge. On the right, it shows the DC response, which calculates the offset. According to the calculator is around $-5\,\mu V$.

4.3.5.3 Differential stage with a level shifter

It uses a similar topology to the previous architecture; however, the next stage to the differential pair doesn't amplify and just sets the output level to reduce the offset and minimize the delay. For this, it uses minimum dimensions for the transistors in this stage.

The scheme is presented in Figure 4.43. To obtain the dimension of the level shifter, we perform simulations to minimize the offset.

We use the same values for the differential stage in the circuit implementation. We first obtain the differential stage output voltage for the level shifter when both inputs equal 400 mV. Here, we obtain 458 mV. We want the inverter output to be 600 mV with this input level. After some tweaks, we obtain the values using a minimum value for L and small values for the widths. This procedure gives us the starting point for the dimensions. After that, we adjust them in the complete circuit to minimize the offset output.

The simulation for the final component is presented in Figure 4.44. Here, we can see that while the offset keeps its absolute value, the delay is reduced in approximately 5 ns, better than the previous architecture with an A-class amplifier. This can be explained due to the short dimensions in the common-source stage amplifier.

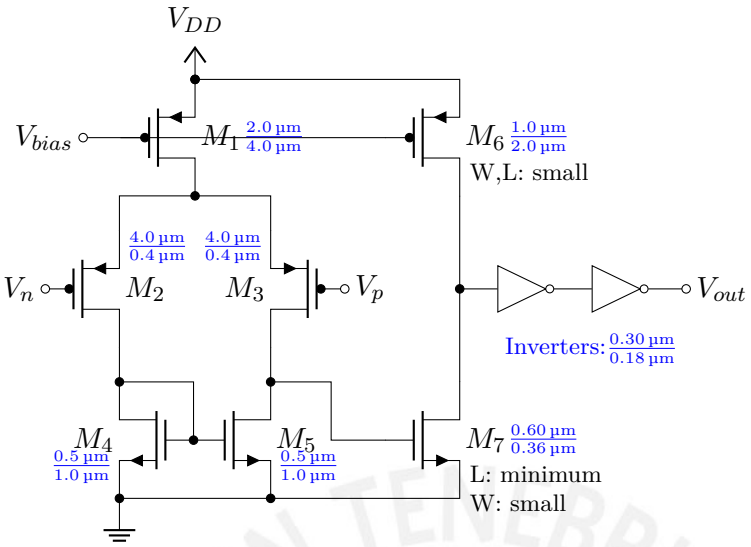


Figure 4.43: Comparator based on differential and a level shifter. It is similar to the previous architecture with the difference in dimensions of the stage next to the differential input, which only shifts the voltage level output without amplifying.

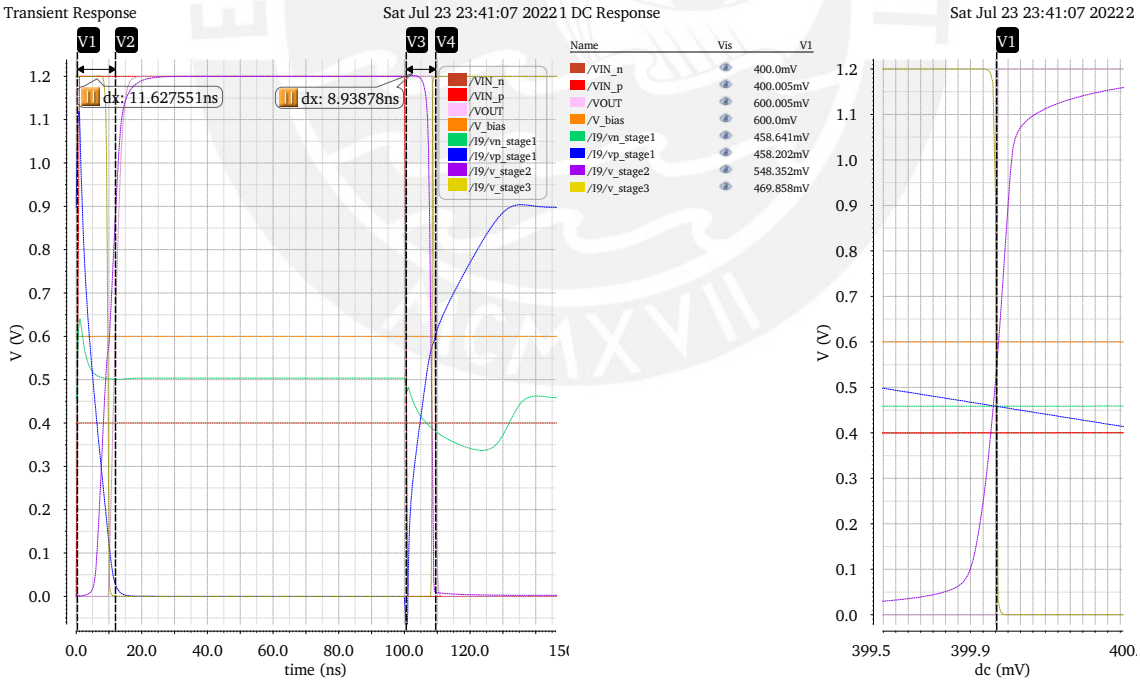


Figure 4.44: Result for simulation of the comparator with a level shifter. On the left, it shows the transient response where we can calculate the delay, which is 9 ns for the falling edge and 11.6 ns for the rising edge. On the right, it shows the DC response, which is used to calculate the offset, around 5 µV.

This variant is focused on reducing the delay. Instead of that, the next two architectures are focused on fixing the glitches in the output generated by the recurrent code logic block. To reduce them, we introduce hysteresis to the output. For that, we test two proposals, one with a Schmitt trigger inverter and another different architecture with hysteresis.

4.3.5.4 Differential stage, amplifier, and Schmitt trigger

This proposal takes the same architecture from the comparator with an A-class amplifier, with a difference in the output inverter that uses a Schmitt trigger instead and introduces hysteresis in the order of a few millivolts.

The scheme for the Schmitt trigger component is presented in Figure 4.45. It requires that the transistors that follow the inverter I_1 have small dimensions of width than the input stage. In addition to this, it requires the L minimum for all the transistors.

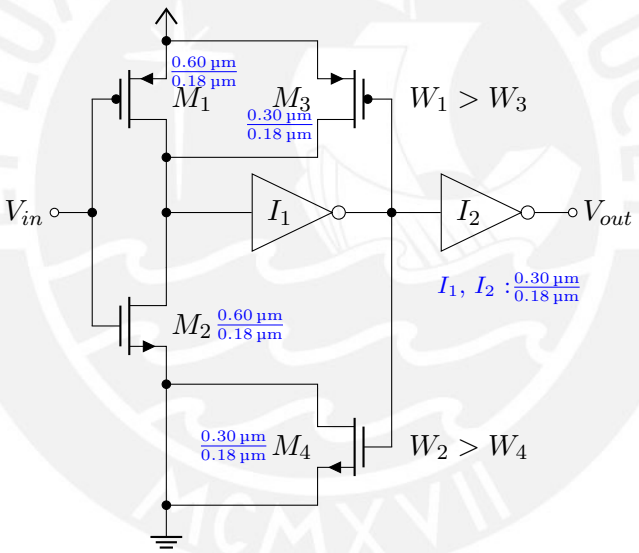


Figure 4.45: Schmitt trigger scheme. Here, the L for all transistors must be minimum, and the widths of the NMOS (M_1) and PMOS (M_2) of the input branch must be higher than the widths of the opposite ones (M_3 and M_4, respectively.)

In the circuit implementation, apart from using minimum values for the channel length, we use the previously used value of 300 nm for the widths, while for the input transistors, we use twice this value, which is 600 nm.

The result of the circuit simulation is presented in Figure 4.46. It shows a hysteresis of approximately 247 mV. It is a higher value, but we must consider that it follows the amplifier stage. Then, the input-referred hysteresis is significantly lower than this value, as it is divided by the amplification factor.

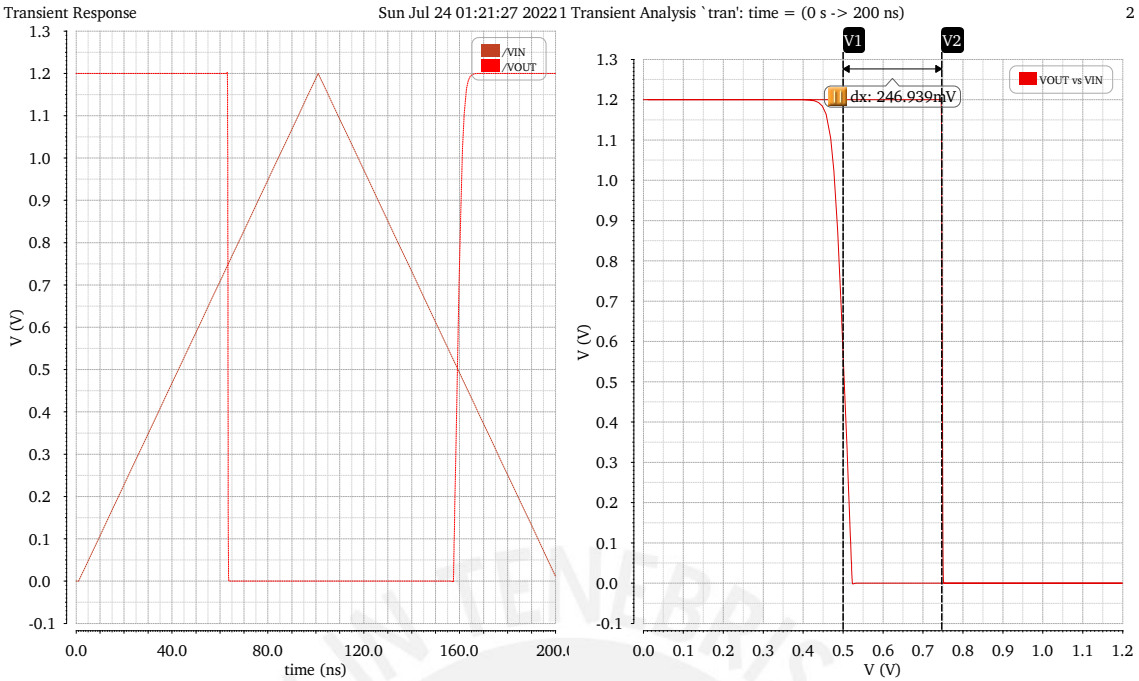


Figure 4.46: Result for simulation of the Schmitt trigger. On the left, it shows the transient response. On the right, it shows the voltage vs. V_{in} to obtain the hysteresis value of 247 mV.

Finally, this circuit can replace the output inverter in Figure 4.41, and obtain a comparator with a hysteresis value. Figure 4.47 presents the scheme for the modified circuit that includes the Schmitt trigger component. All the other components are identical to the original.

The simulation result for this circuit is presented in Figure 4.48. Here, the measured hysteresis is equivalent to 4.6 mV. As expected from the previous analysis, it is considerably lower than only the Schmitt trigger component, as it is referred to the input of the two-stage amplifier.

4.3.5.5 Comparator with hysteresis

As the previous architecture doesn't provide enough level of hysteresis (tested with the whole architecture), we move on to another one. That changes the topology but guarantees hysteresis levels of around several millivolts.

The scheme for this circuit is presented in Figure 4.49. It shows two relationships between the widths of the load transistors in each differential branch, whose difference produces the unbalanced current that is translated to a hysteresis effect. The difference between a and b also provides asymmetry in the hysteresis curve.

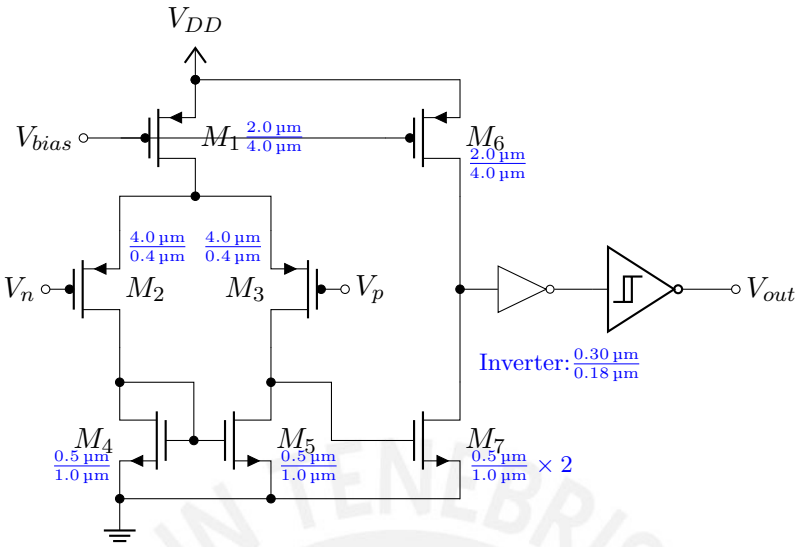


Figure 4.47: Comparator based on a differential amplifier and Schmitt trigger.

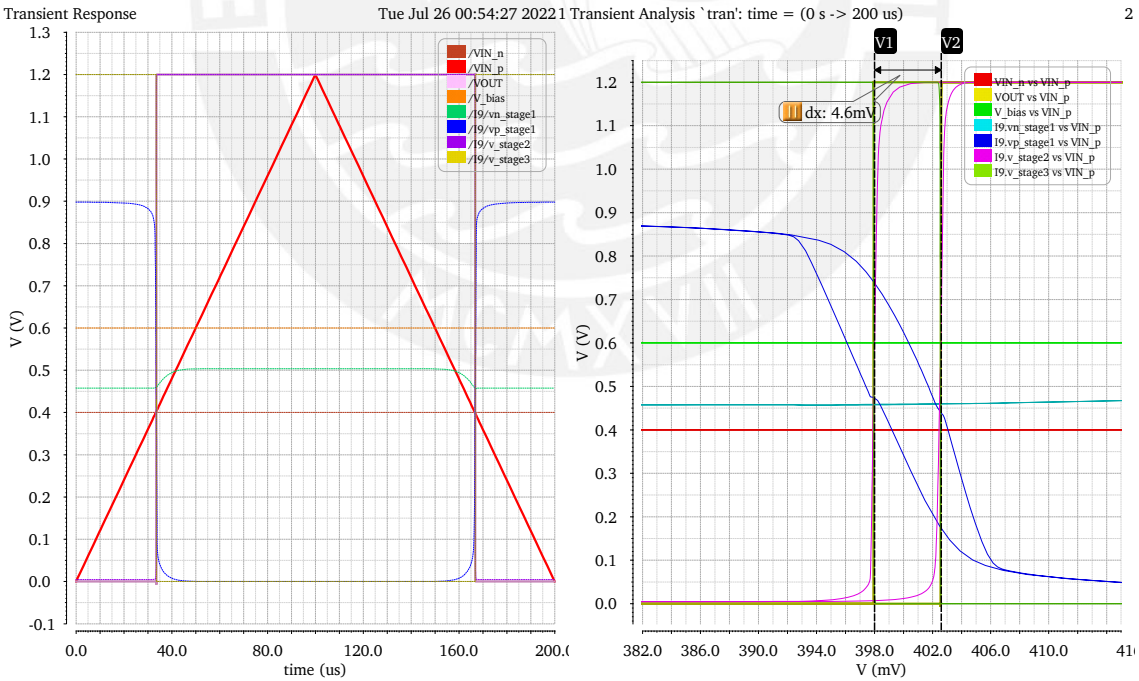


Figure 4.48: Result for simulation of the comparator with Schmitt trigger. On the left, it shows the transient response. On the right, it shows the voltage vs. V_{in} to obtain the final hysteresis value of 4.6 mV.

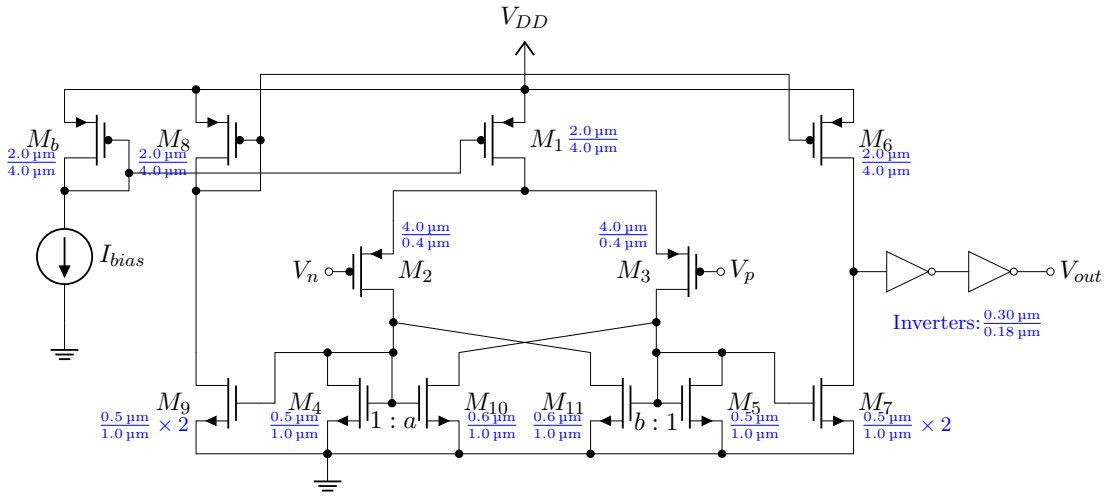


Figure 4.49: Comparator with hysteresis scheme. Based on [66]. While a and b differ from 1, it will generate a hysteresis. If $a \neq b$, then it will achieve a hysteretic asymmetry.

The dimensions of the transistors are the same as those obtained for the A-class amplifier version. The only difference is in the transistors that have the relationship $1 : a$ and $b : 1$ in their widths. In this case, while the unity represents a width of 500 nm, for both a and b factors, the widths are equal to 600 nm. It could produce a lower hysteresis value for this architecture; however, according to the next simulations, it is enough for our purposes.

The simulation result depicted in Figure 4.50 shows a hysteresis value of 5.9 mV. Although the relationship of $1 : a$ and $b : 1$ are equal and have a minimum value of 1 : 1.2, the hysteresis is higher than using just the trigger component. We consider this value enough according to simulations with the whole architecture, and, therefore, we keep the relationship level.

4.3.6 Complete LCADC architecture

Once all the required building blocks are designed, we can follow up with the whole LCADC architecture. As presented in the simplified scheme in Figure 4.30, it requires a 1-bit DAC, two static comparators, and a set of logic blocks responsible for returning the digital codes.

Figure 4.51 shows the detailed scheme with the final blocks constituting the ADC. This scheme includes an analog multiplexer, a logic for the change code pulse C_c and a logic for the recurrent code C_r pulse. These pulses are inputs for the DAC control logic and the counter that outputs the ADC result.

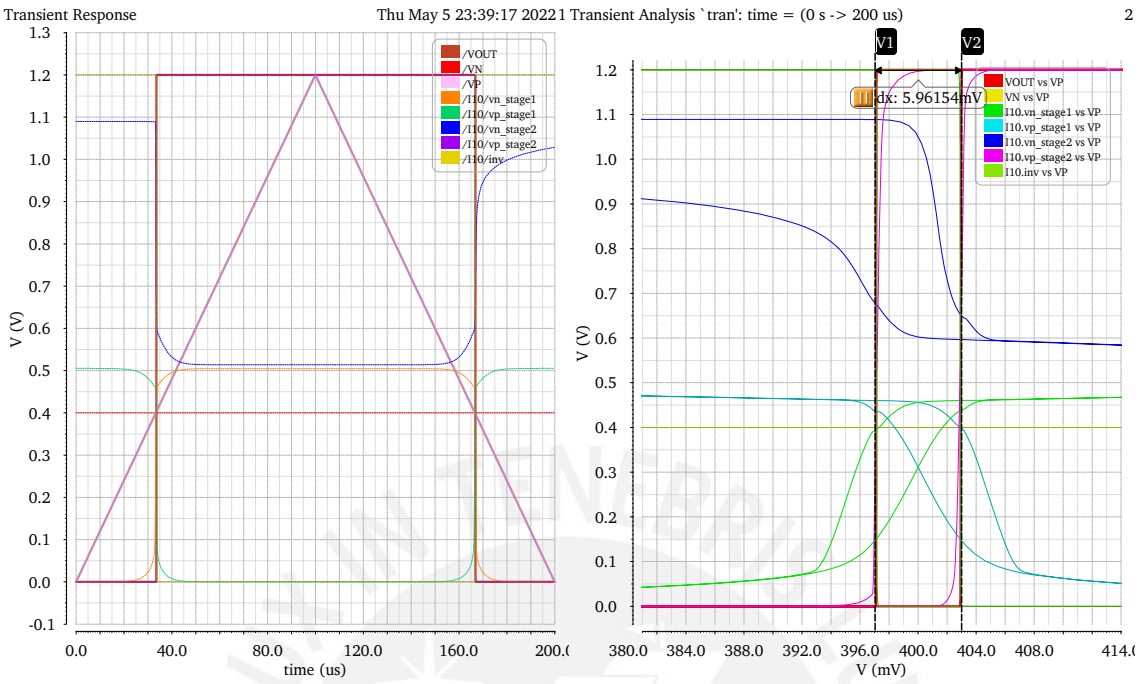


Figure 4.50: Result for simulation of the comparator with hysteresis. On the left, it shows the transient response. On the right, it shows the voltage vs. V_{in} to obtain the final hysteresis value of 5.9 mV.

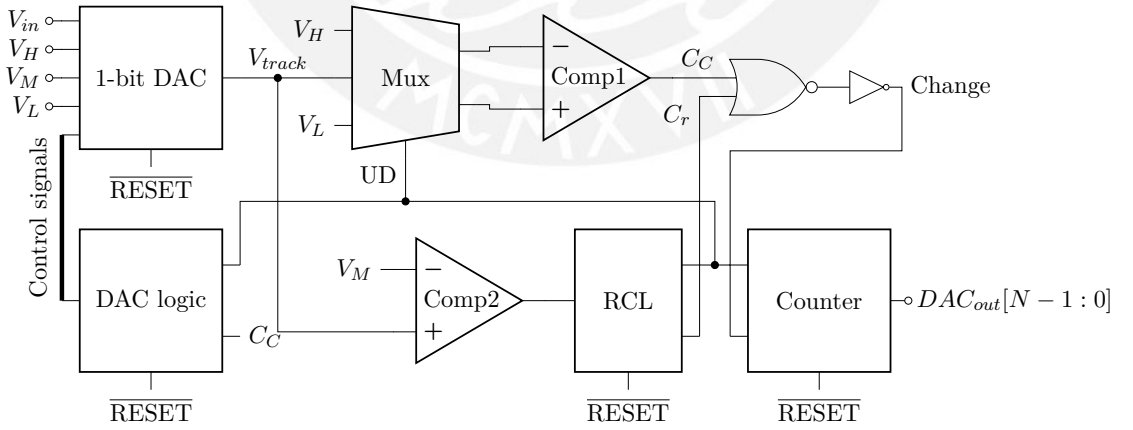


Figure 4.51: Complete scheme for LCADC architecture of 1-bit DAC. This circuit uses fixed references for the comparisons, represented as V_H, V_L, and V_M. In addition, it has a reset signal, $\overline{\text{RESET}}$, to start the conversion . The output is represented as $DAC_{out}[N-1:0]$.

The functioning starts with a low-value $\overline{\text{RESET}}$ signal, which sets all outputs to 0. Then, when this state is released, the V_{track} starts following the input into the comparison window between the high reference voltage V_H and the low reference voltage V_L, and it starts at V_M, which is the mean value of both. If the tracked voltage in the 1-bit DAC is in the upper window region ($[V_M : V_H]$), the second comparator (Comp2) returns '1', and the UD signal, which comes from the recurrent code logic (RCL) is '1'. In the other case, if the tracking voltage V_{track} is in the lower window region ($[V_L : V_M]$), the comparator returns '0', and the UD signal is '0', as well. This value also conditions the multiplexer outputs: V_H and V_{in} if '1' or V_{in} and V_L if its value is '0'.

In the first case, (UD = 1), if the V_{track} is below V_H, then the first comparator (Comp1) outputs '0', and nothing happens to the DAC. However, when the V_{in} exceeds V_H, it will trigger a pulse that activates the DAC logic to inject a negative offset in the voltage input to return the V_{track} to V_M. This also produces the counter increment in '1'.

In the second case, (UD = 0), if V_{track} is above V_L, the first comparator (Comp1) also outputs '0'. When it crosses the reference V_L, it also triggers a pulse activating the DAC logic, but, in this case, it injects a positive offset to return the V_{track} to V_M. It will also change the count value, decreasing it in '1'.

A special event occurs when the V_{track} transits from the lower window region to the upper window region and vice versa. That produces a pulse in C_r which also increases or decreases the count value (depending on the final value of UD).

Some special features required for the circuit include alternating offset injection for the 1-bit DAC (two branches of offset), which can inject a voltage to the V_{track} with minimum delay, as the offset branch is pre-charged with the required voltage. In addition to this, the switches that control the execution must be non-overlapping to avoid an active state in both offset branches at the same time. On the other hand, the logic of the recurrent and changed codes generates a pulse with a fixed time (monostable) being the C_C pulse of a shorter duration than the C_r pulse. That will prevent glitches in the transition of UD.

Concerning the comparators, we have designed several alternatives in the previous section. The aim for them is to achieve a minimum delay and offset. They also must be compatible with lower power dissipation. We will test each comparator version in the results to evaluate which achieves the best performance metrics.

In the circuit implementation, we added a delay block after the Change signal. The purpose is to ensure that the UD has a stable value before we change the count, which may affect its behavior by increasing or decreasing the value.

For the simulation, we use a sinusoidal waveform as the input signal with the same characteristics as in the SAR case (with the frequency $F = 7 \times \frac{20000}{128} = 1.093\,75\,\text{kHz}$ and with the amplitude 500 mV over a DC of 600 mV). It also has the same number of cycles (7 complete cycles), determining the simulation duration. We extend the simulation a bit further to clip the output signal approximately at phase 0.

We also connect the circuit output to an ideal DAC modeled with Verilog-A, similar to the SAR case. It converts the digital codes to analog amplitudes, which may be used to obtain the metrics and compare them to the original signal.

Figure 4.52 shows the simulation obtained with the circuit. It covers more than 7 complete cycles which provides us the flexibility to choose the most appropriate range to obtain the performance metrics. In the image, we can appreciate that the output signal has a DC value, because of the initialization process. On the other hand, the amplitude is lower than the original, as the relationship obtained in the 1-DAC outputs a V_{track} that has a proportion of $\frac{7}{10}$ with respect to the input. Despite those variations, which are intrinsic to the architecture, the shape of the output is almost identical to the original. However, we must extract the metrics to evaluate if this architecture has adequate precision and is comparable to the SAR, our reference. This process offers the challenge of calculating those metrics from a non-uniform sampled signal.

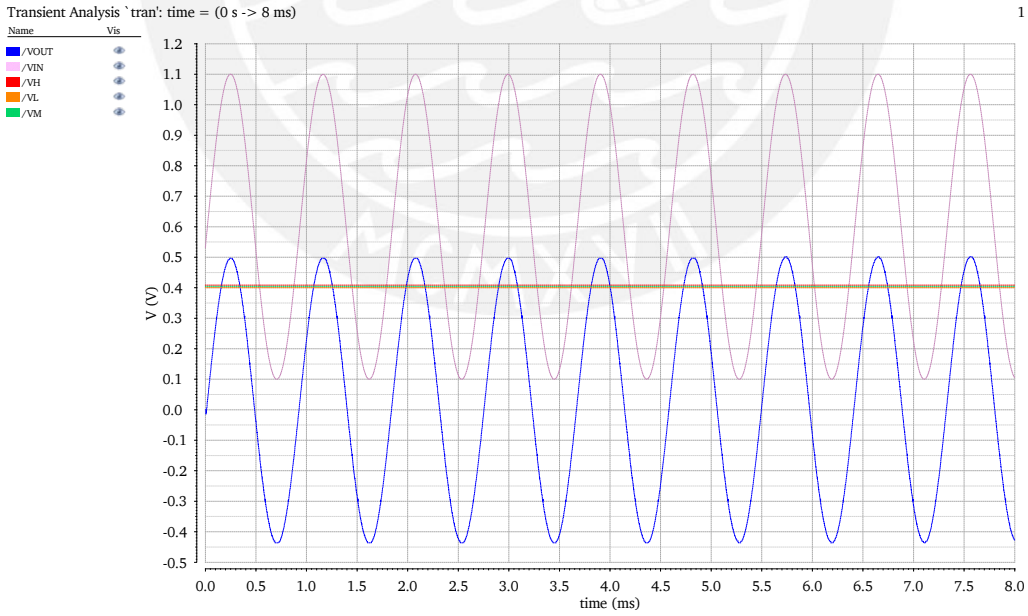


Figure 4.52: Result for simulation of the whole LCADC architecture. The simulation runs until it completes 7 cycles.

4.3.6.1 How to calculate metrics for non-uniform sampled signals?

The metrics we used to evaluate the SAR were the SNDR and the ENOB. However, to obtain such values, we need the spectral density of the output, which requires a uniform sampled signal. In the special case of the LCADC, we don't have such a condition as the sample only occurs when the amplitude level is reached, in non-quantized time intervals.

Therefore, we must process the output from the LCADC to obtain a uniform sampled signal and then apply the techniques we know to calculate the metrics. To achieve this, we employ an interpolation technique. This stage aims to approximate the original samples to uniform distributed intervals of times, along the same range as the original signal. We can use polynomials of order 1 (lines) or 3. According to [67] up to 3rd order is enough to recover the signal with an adequate SNDR level.

At this point, we could ask, how the interpolated signal from level-crossing samples differs from the SAR output, and which effects have in the performance comparison? First, we should recall that SAR output takes samples at specific times and for each sample, there is a quantization stage that introduces errors, which depending on the amplitude resolution will determine the maximum SNDR and consequently the maximum effective number of bits. In the case of LCADC, ideally, we don't have a quantization error as such as the samples are taken almost at the time it crosses the level, and only have the time uncertainty. This characteristic of LCADC helps to enhance the SNDR as the output values are closer to the real value. However, as we cannot process them directly, we need to interpolate the samples. The interpolation process will add values that were not taken from the original signal. This could be an advantage or drawback. It is advantageous if the consecutive samples have short time intervals, approximating better the input signal, and consequently grabbing more accurate information that helps reconstruct the signal. However, it could be a drawback if they are distant samples whose behavior has more variation that cannot reproduce accurately introducing errors. Both effects to some extent could increase or decrease the SNDR and will depend on the input signal characteristics. In general, simpler interpolators introduce less error values. For more information regarding the differences, we added Appendix A which explains both sampling schemes.

Once the signal is interpolated, we can apply the Fast Fourier Transform (FFT) and obtain the spectral density. Then, we can remove the DC component and identify the maximum component in the spectrum corresponding to the sinusoidal spectrum. We can

calculate the energy for the sinusoidal component and all the other components (noise and distortion). The relationship between them (subtraction if they are converted to decibels), will return the SNDR value. Finally, we can apply the ENOB relationship to calculate the value for that SNDR.

Applying this procedure to the different architecture versions that include different alternatives for the comparators Comp1 and Comp2, we obtain the result presented in Table 4.1. In this case, the "Three-stage" is the comparator in Figure 4.39, the "A-class amplifier" is the comparator in Figure 4.41, the "Level shifter" is the comparator in Figure 4.43 and the "Hysteresis" is the comparator in Figure 4.49. As the library we use for the interpolation doesn't allow even order values, we employ only 1 and 3 orders. On the other hand, for the frequency, we use a similar frequency as in the SAR case (20 kHz), and to test if increasing the sampling could increase the ENOB, we also use higher sampling rates (40 kHz and 100 kHz).

Table 4.1: Results of SNDR and ENOB for different variants of LCADC architecture. We utilize different types of comparators for which we test the interpolation and sampling frequency to obtain the metrics for the architecture.

Comp1	**Comp2**	F_s	**Order**	**SNDR**	**ENOB**
Three stage	Three stage	20000	1	46.520169	7.435244
			3	46.468178	7.426608
		40000	1	46.544109	7.439221
			3	46.492331	7.430620
		100000	1	46.527168	7.436407
			3	46.476187	7.427938
A-class amplifier	A-class amplifier	20000	1	47.237217	7.554355
			3	47.254551	7.557234
		40000	1	47.177163	7.544379
			3	47.262942	7.558628
		100000	1	47.171922	7.543509
			3	47.265298	7.559020
Level shifter	Level shifter	20000	1	45.564743	7.276535
			3	45.941958	7.339196
		40000	1	45.602634	7.282830
			3	45.854533	7.324673
		100000	1	45.585374	7.279962
			3	45.802977	7.316109
Hysteresis	Hysteresis	20000	1	38.604771	6.120394
			3	38.797036	6.152332
		40000	1	38.704752	6.137002
			3	38.775304	6.148722
		100000	1	38.700017	6.136215
			3	38.762364	6.146572
A-class amplifier	Hysteresis	20000	1	48.105117	7.698524
			3	46.612710	7.450616
		40000	1	48.150400	7.706047
			3	46.576856	7.444660
		100000	1	48.085552	7.695274
			3	46.601431	7.448743

The results show that the higher ENOB and SNDR are obtained in the architecture that mixes the variants of comparators, using the A-class amplifier type for the comparator that outputs the changed code pulse, and the hysteresis type for the comparator that outputs the recurrent code pulse. The maximum achieved ENOB is 7.7, equivalent to an SNDR of 48.1 dB, with a linear interpolation and a sampling frequency of 40 kHz. On the other hand, the architecture that shows more regular values, independent of the interpolation order or the frequency, is the one that uses A-class amplifiers for both comparators, obtaining a mean ENOB of 7.55 and an SNDR of around 47 dB.

Something we highlight here is that, although the optimizations we applied to the architecture, including changing the comparators type, the ENOB is slightly lower than the SAR architecture, which we use as the reference. Then, we must compare them with neural recordings to verify if the LCADC can still be a good replacement for the SAR with that kind of signal.

4.3.6.2 A reset block for accumulative offset

Testing with long neural signals, we identified a problem that we were not aware of just with the sinusoidal input. This problem is about an accumulative offset that progressively increases the output, making it mount over a kind of linear signal dependent on the time. This effect significantly affects the expected result, being worst while the conversion continues.

We identified that the origin of this offset was due to different slopes in the 1-DAC tracking voltage. The rising slop was higher than the falling slope. To fix it, we implemented two modifications to the circuit. First, in the 1-DAC, we changed the V_M to the $V_{DD}/2$, which is 600 mV, where, by simulation, we found that the slopes are almost the same. We also check that the pseudo resistors are properly connected to avoid the influence of ground or VDD in their resistances.

Another strategy is using a reset block that ensures the return to a known reference for the signal. In this case, we reset the 1-bit DAC and the counter, each time the input reaches the '0' value. Any offset accumulated during the conversion process will be suppressed in the DAC capacitors and the output, returning the tracking signal to V_M and the counter to '0'. We employ an additional comparator to implement this block that triggers the reset event after the ADC input signal crosses the V_M level. We also added the same logic employed for the C_r signal, to generate a pulse each time the crossing event occurs, which is used to reset the other blocks.

Although it cannot make any difference if the signal stays a long time in positive or negative values, in the specific case of biomedical signals, it is expected that most of the time, they will oscillate around zero, making the method effective in this scenario.

After those updates in the circuit, it is ready to test with neural recordings and to obtain the metrics for its comparison against the SAR architecture.

4.4 NEO design

The NEO is a preprocessor block that receives the output from the ADC and alters it to enhance the spikes and facilitate their detection. In this section, we present the circuit design for this digital component, constituted essentially of multipliers and registers, for which we explain their implementation.

4.4.1 Signed multiplier

The multiplication is the sum of one of the factors multiplied by each bit of the other and shifted according to the bit position. As the input can be positive or negative, we need to consider the sign of the factors for the result.

There are two ways we can deal with that. One extends the sign to cover the expected number of bits in the result: for n-bit factors to 2n-bit. Or to consider the current value of the negative representation in a full-length word extended by zeros and add a correction term to the multiplication. Following the last approach and considering a negative factor, we can express the multiplication bit a bit as:

$$A \times (2^n - B) = 2^n A + A \times (-B) \tag{4.54}$$

We observe an additional factor $2^n A$ which we need to subtract from the result obtained. Then, the expression will be:

$$A \times (-B) = A \times (2^n - B) + 2^n(-A) \tag{4.55}$$

In the case both factors are negative, the expression is:

$$(2^n - A) \times (2^n - B) = 2^{2n} - (A + B)2^n + (-A) \times (-B) \tag{4.56}$$

As the new representation is $2n$ in length, the first term $2^2 n$ is meaningless. Then, the expression may be expressed as:

$$(-A) \times (-B) = (2^n - A) \times (2^n - B) + (A + B)2^n \tag{4.57}$$

From both equations, we can multiply the n-bit representation directly, using n-bit adders, adding a correction term. Looking at the equations (4.55) and (4.57), this term is the n-bit complement to two of the other factor if one is negative and of both factors if both are negative. This can be easily implemented by logic gates that will control the addition of the correction term, depending on the sign bit of the factors.

Figure 4.53 shows the implementation of the digital multiplier. In this case, the product of the factor B[n − 1 : 0] with each bit in A[n − 1 : 0] is implemented by AND gates. The output for the first bit OUT[0] is directly obtained from B[0], and the successive bits equally come from it but shifted according to the A bit position that affects the B factor. Once, it covers all A bits, for the next position n, it implements the additional term to fix the result, as we are using n bit complement to 2 in each bit a bit product, instead of the 2n bit complement. The adders in the base of the figure return the n-bit complement to 2 for each factor, while the AND gates connected to their outputs decide whether to pass this complement depending on the sign bit of the factors. Finally, it sums this result to the accumulative sum obtained from the product bit a bit, obtaining the last bits for the output OUT[2n − 1 : n].

4.4.2 NEO architecture

The original equation for the NEO was presented (2.4). It uses delayed samples, including factors forward to the evaluation sample $(x + \omega)$. However, this is not possible in real life as we cannot predict future samples. Therefore, we need to adapt the equation to use previously stored samples:

$$f[n] = x[n - \omega]^2 + x[n]x[n - 2\omega] \tag{4.58}$$

The delay in hardware are implemented by shift registers which store a sample each time receives a positive edge clock and shifts the current value to the output register connected next to it. Using this strategy and employing the previously designed multiplier, we can implement the full circuit for the digital NEO.

Figure 4.54 shows the architecture for the NEO block. In this case, we are using a delayed clock to save the NEO result in the output register before it changes the value of the chained registers. The term $x[n - \omega]^2$ is generated by the output of the ω position in the shift register, while the multiplication of the input and output of the 2ω shift register generates the term $x[n]x[n - 2\omega]$. The multipliers have $2n - 1$ bit outputs. Their results input the output adder after we obtain the 2n complement to 2 of the negative term,

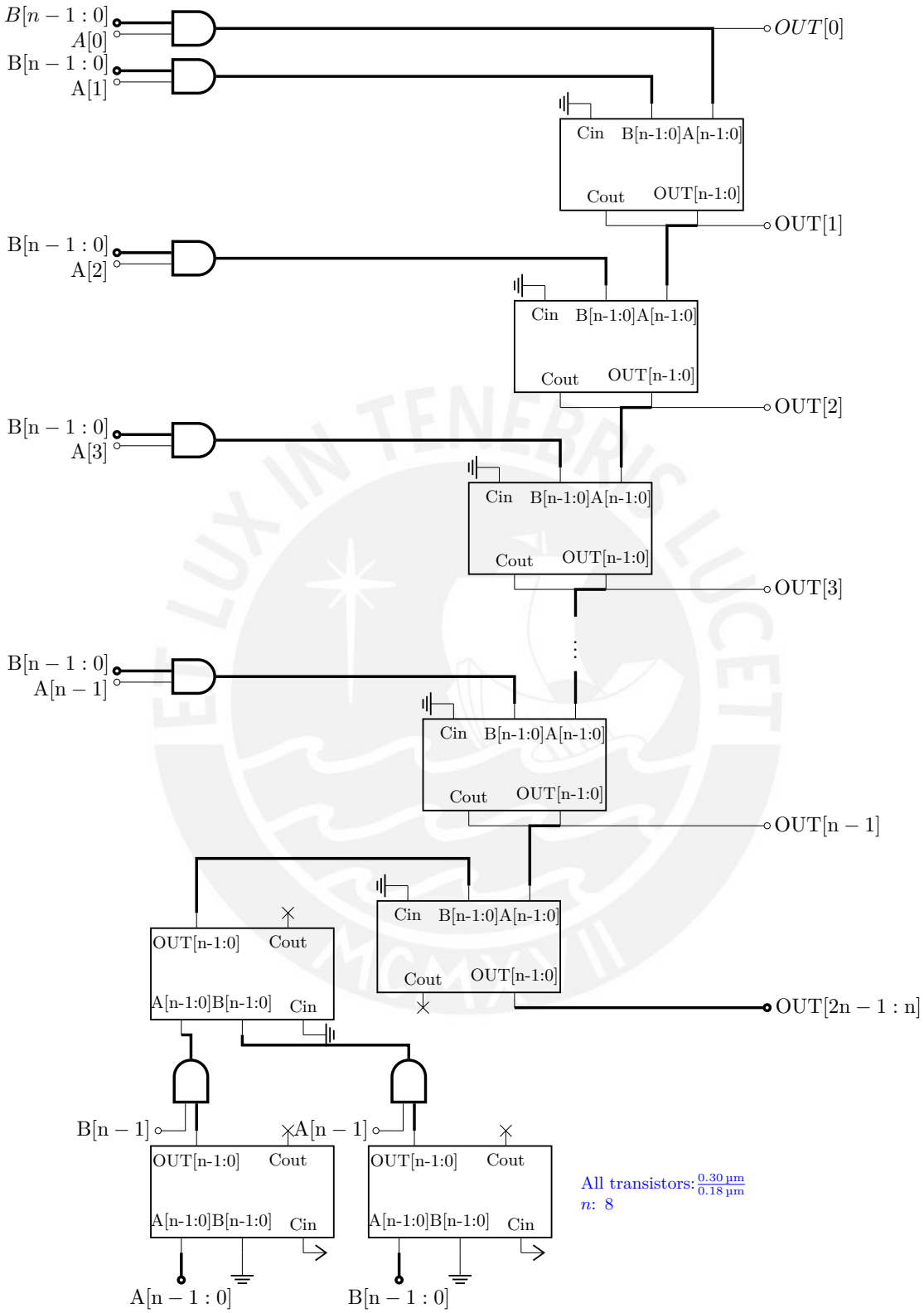


Figure 4.53: N-bit signed multiplier. A[n − 1 : 0] and B[n − 1 : 0] are the two n-bit signed factors that inputs the multiplier, returning the 2n-bit OUT[2n − 1 : 0] output.

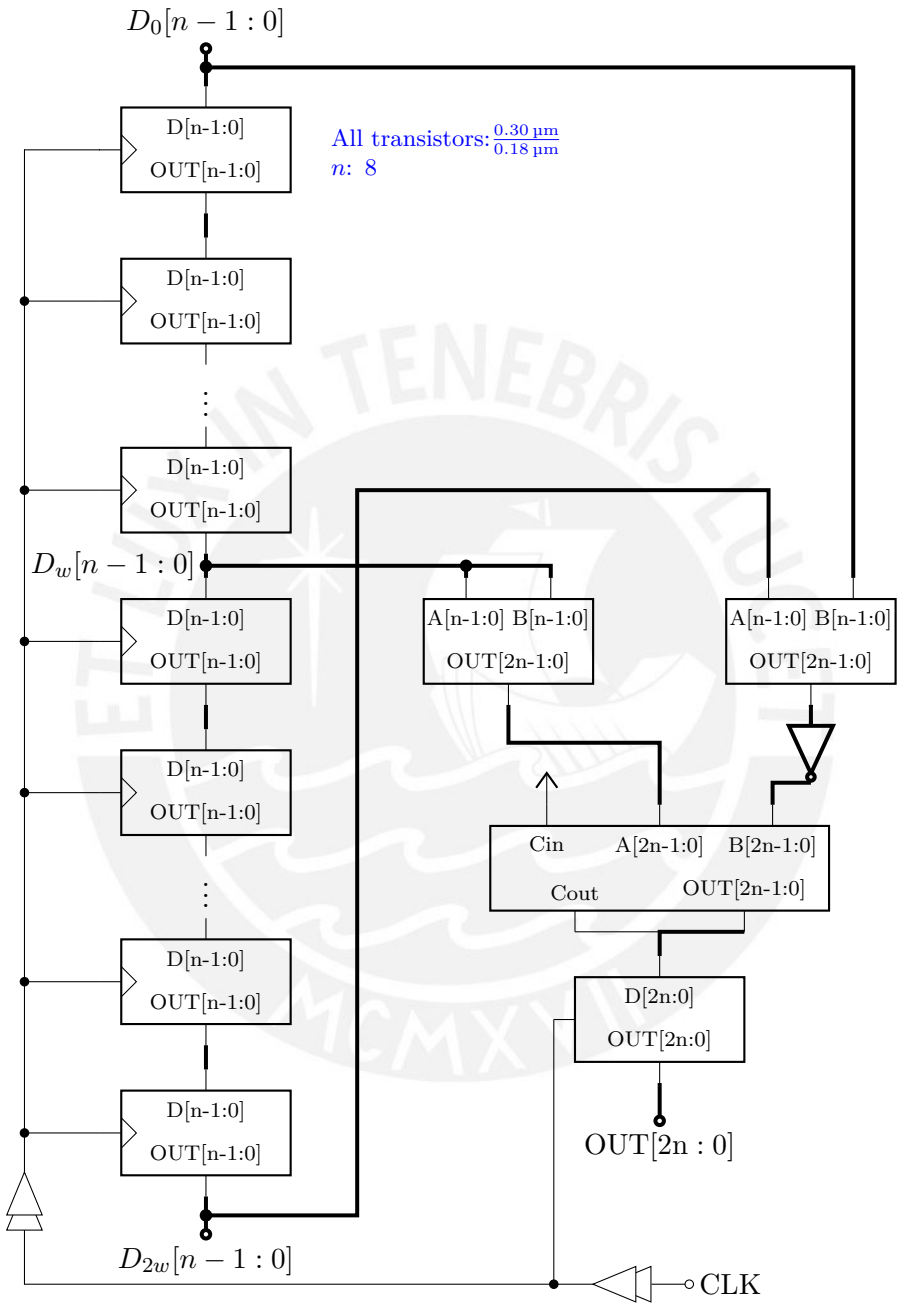


Figure 4.54: N-bit NEO architecture. $D_0[n-1:0]$ is the n-bit input, and CLK is the clock that controls the shift and the output registers. $OUT[2n:0]$ is the 2n-bit output.

which consists of adding its 1's complement and an additional one to the sum. The final output is $2n$ size as it can be higher than the $2n-1$ order. This sum also includes the sign for the result.

We use minimum transistor sizes for the circuit implementation in the current technology, with $L = 180\,\text{nm}$ and $W = 300\,\text{nm}$. In addition, as the NEO width is 4, then the number of registers is 8, using the 4th register output for the term $x[n-4]^2$ and the first register input with the 8th register output for the other term in NEO equation $x[n]x[n-8]$.

We used a pseudo random-sequence that inputs the NEO block to test this circuit. With a Verilog-A block, we verify each result relative to the ideal NEO functionality. Figure 4.55 shows the simulation for this testbench.

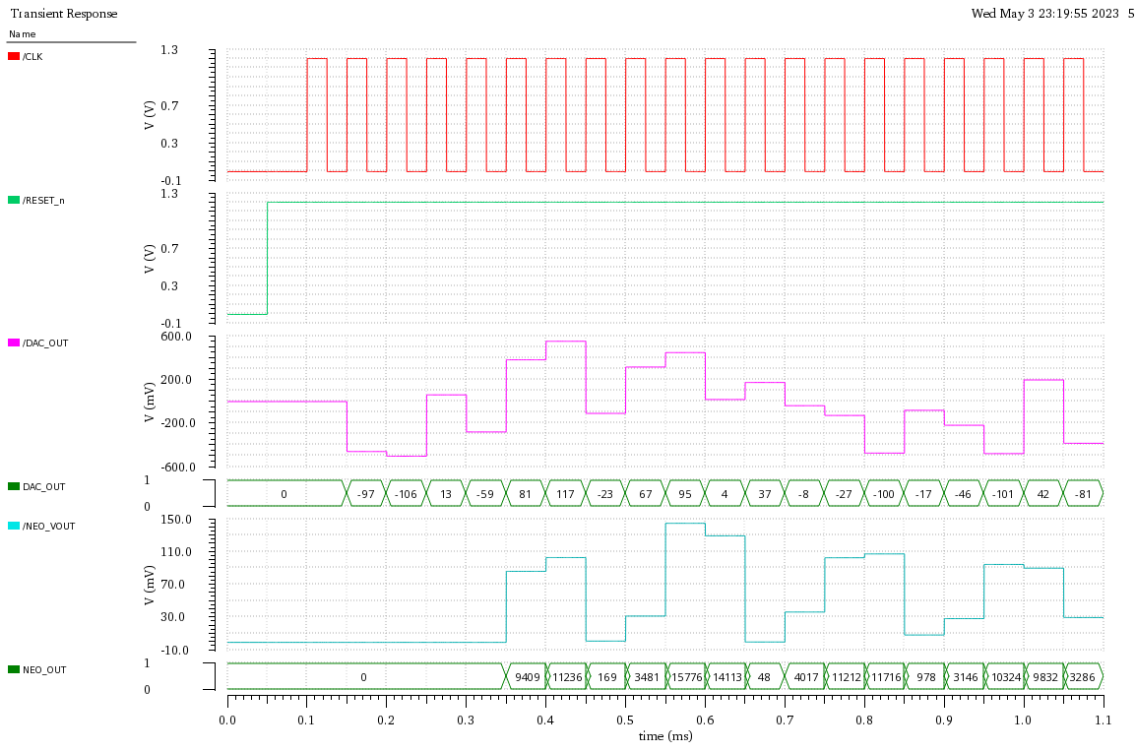


Figure 4.55: Results for simulation of the NEO architecture. The simulation runs for several random samples for which the NEO digital block calculates the result according to the NEO equation. In this case, DAC_OUT represents the input for the system, and NEO_OUT represents the output.

The simulation shows that the circuit starts with all the stored values in the shift register as zeros. Then, while the generator feeds a new value, it replaces the previous stored, while this one is shifted to the following register. For the first four samples, the result is 0 due to the two terms in the equation (4.58) are zero. For the next sample, a value is stored for the $x[n-4]$; consequently, the result differs to zero. For the next four

samples, the value in NEO is $x[n-4]^2$ as the other term is zero because it doesn't still store enough samples. Finally, when the 9 sample arrives, both NEO terms differ to zero, and the result will apply its equation.

Then, we make a validation for the values obtained in Table 4.2. We can confirm that all the values obtained are the same as the expected, applying the NEO equation. Then, the module implemented is working correctly.

Table 4.2: Verification of simulation values in Figure 4.55

x	$x[n-4]^2$	$x[n-8]x[n]$	$x[n-4]^2 - x[n-8]x[n]$	**Obtained**
-97	0	0	0	0
-106	0	0	0	0
13	0	0	0	0
-59	0	0	0	0
81	9409	0	9409	9409
117	11236	0	11236	11236
-23	169	0	169	169
67	3481	0	3481	3481
95	6561	-9215	15776	15776
4	13689	-424	14113	14113
37	529	481	48	48
-8	4489	472	4017	4017
-27	9025	-2187	11212	11212
-100	16	-11700	11716	11716
-17	1369	391	978	978
-46	64	-3082	3146	3146
-101	729	-9595	10324	10324
42	10000	168	9832	9832
-81	289	-2997	3286	3286

Finally, we have this block tested and prepared to use with the ADC and measure power consumption which is the third thesis objective.

Chapter V

Results and discussion

This section is divided into the stages described in the methodology section. It will start with the recording simulation, followed by the NEO and ADC evaluation. In the end, it finishes with the power consumption evaluation.

5.1 Extracellular recordings simulations

These simulations were produced via a slight variation of the original MEArec software, registered in this repository: `https://github.com/LuighiV/MEArec`. Then with these modifications, another repository was created `https://github.com/LuighiV/eapprocessor`, which interfaces the modified one to vary the noise level by a Python module[1].

The simulation software allows users to vary the configuration parameters of the templates and recordings generation. In this aspect, for the templates' generation, it used the default ones as suggested in the software repository, changing only the parameters for the recording stage.

With respect to the recordings, most of the parameters remain unchanged, only modifying those related to the noise level, sampling frequency, and band filter, the last one was introduced as a dictionary for the interface method in the framework developed.

From the parameters which preserve their default values, the most important to consider are:

- **Spike trains (neurons)**. It employs 10 spike trains, 7 of the type excitatory and 3 inhibitory. Their average firing rate is 5 Hz and 15 Hz for the excitatory and inhibitory cells, respectively.

[1] This repository is private until the thesis is defended.

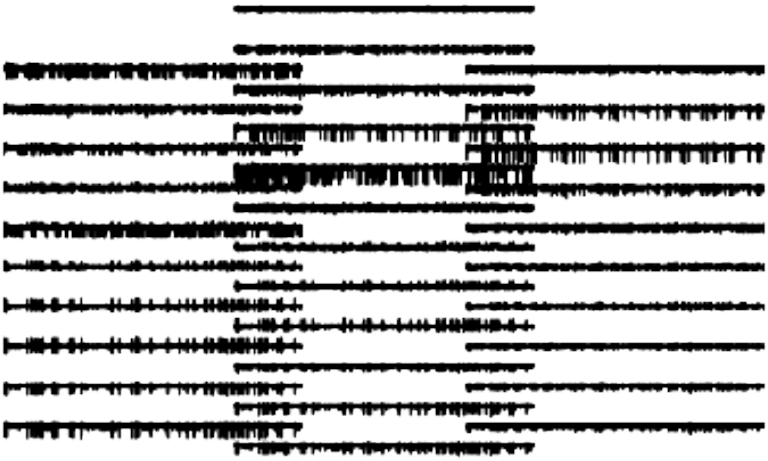
- **Seeds**. These are the reference numbers employed for random functions. It should be a constant value to reproduce the same patterns in the generated recordings. Here, we use a number of 100 for all the seeds.

- **Templates**. It refers to the templates' characteristics for their selection and use in the recording generation. The most important parameters are the minimal distance between neurons, established to 25 µm, the minimum amplitude, equal to 50 µV and the maximum amplitude, set to 300 µm.

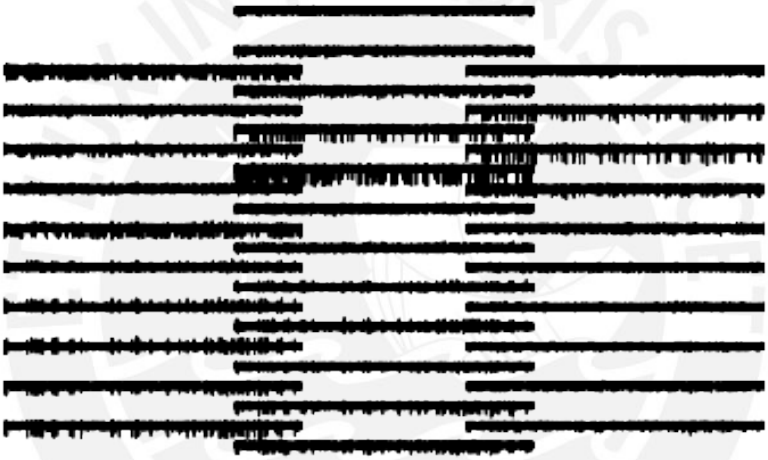
- **Recordings.** Here, we set the parameters referred to neuron signal characteristics such as the sampling frequency, the bursting and the drifting phenomena, the level and type of noise, and the output filter. In this case, both bursting and drifting are not considered in the evaluations performed, and they are set to false. With respect to the type of noise, the tests will contemplate an uncorrelated noise (Gaussian noise).

One variation introduced by the interface developed is the noise level. The type of noise evaluated here is an uncorrelated noise, and the amplitude represents the value of the deviation standard, while the mean value remains zero. Through all the process of experimentation, the tested values for noise have changed, as not all electrodes and spike trains have the same amplitude, and one level of noise could be low for one electrode, but high for another. Then, the values will depend on the electrode that will be evaluated. In this case, Figure 5.1 shows the recordings for 32 electrodes with three different noise levels: 5 µV, 25 µV, and 50 µV. These results shows how the amplitude affects the visibility of spike trains. While in the first subfigure with 5 µV, the spikes could be distinguishable, with higher values of noise levels, the spikes are hidden by the amplitude of the noise. This figure also depicts that the noise level affects differently to each electrode signal. For some of them, with 25 µV, it only shows a uniform signal along the time, other electrodes could preserve, although minimally, the position of the spikes with higher amplitude.

Related to the spike trains, they are shown in Figure 5.2. This set of images depicts the spike trains for different noise levels. It is important to notice that, independent of the noise levels the positions for the spikes remain intact. This is a fundamental condition as it allows us to compare the results from one noise level to another. In addition, the figures show the 10 spike trains set in the configuration for the recording generation. Here, it also exhibits the two different types (0-6 are excitatory, while 7-9 are inhibitory), and their corresponding average frequency. According to the configuration, the inhibitory rate



(a) Recordings with 5 µV noise level



(b) Recordings with 25 µV noise level

(c) Recordings width 50 µV noise level

Figure 5.1: Recording simulation of 32 electrodes with different noise levels. It shows the variation in amplitude for the noise, hiding the spike trains in the resulting signal.

is higher than the excitatory rate, and this is demonstrated in the figure, as the first set of spike trains (0-6) are more spaced than the second set of them (7-9).

5.2 Evaluation of NEO

As described in the methodology, there are 5 experiments designed to evaluate this component: evaluate the original recording, evaluate with lower bandwidth, with lower resolution, with non-continuous sampling, and with their combined effects.

5.2.1 Experiment 1: Evaluation of the original sample

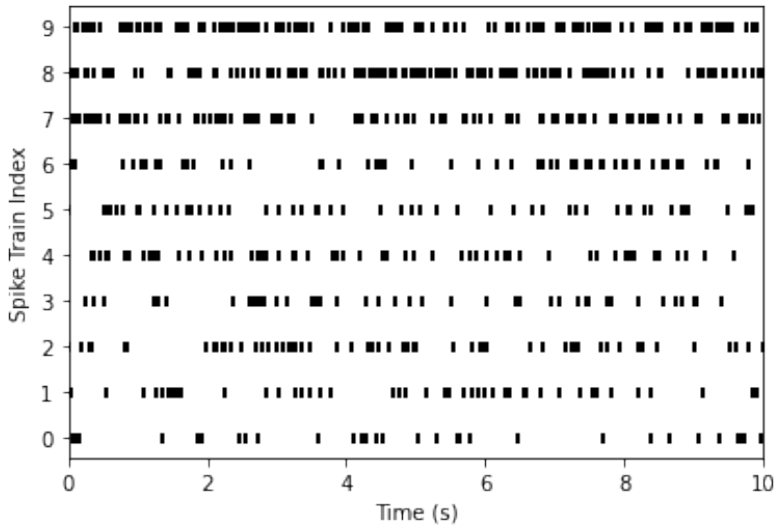
After we generate the recordings with different values of noise level, sampling frequency, and bandwidth, we process those signals by applying the different stages of evaluation. Those include the conversion with a software model of ADC, the NEO preprocessing, and the detection via the threshold level.

These stages are:

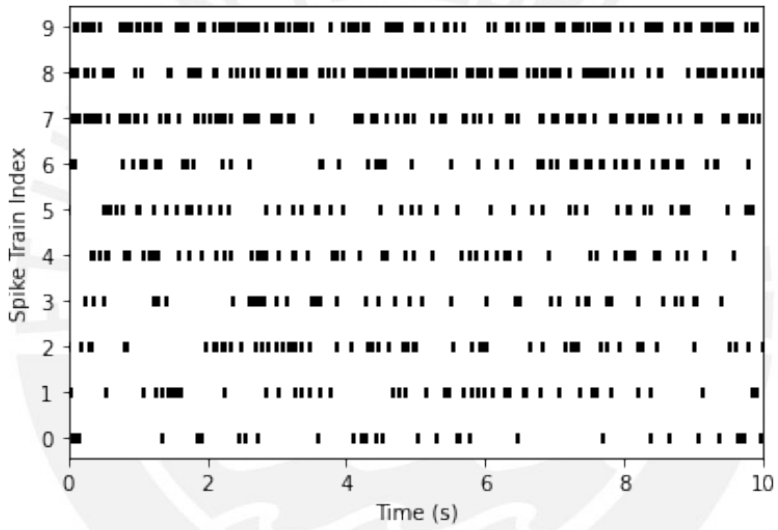
- **Sample conversion via an ADC model.** This conversion considers an ADC model implemented in software, which converts the signals into encoded integer values. This model emulates the behavior of a quantizer, comparing the samples with a set of levels depending on the resolution value. In addition, it considers a reference voltage and the type of signal to be converted, between bipolar (positive and negative samples) or of one-polarity samples. In this case, the type of signal is bipolar, and the established reference level is 500 µV, as the maximum level that spikes could achieve according to the value defined in the model (300 µV). As the ADC model is symmetric, then the range of values that it could convert properly is from -500 to 500 µV. With respect to the resolution, for these tests, it will remain at 12 bits, although this value will vary in the next experiments.

- **NEO preprocessing.** The model for NEO is taken from the equations specified in the theory and methodology (3.2). In this case, it will consider various values of window widths for the spaced samples in the NEO formula and not only the classical contiguous samples for the calculations. During the experiments, this parameter was changing, but in the end, it will consider 4 values: 1, 4, 16, and 32.

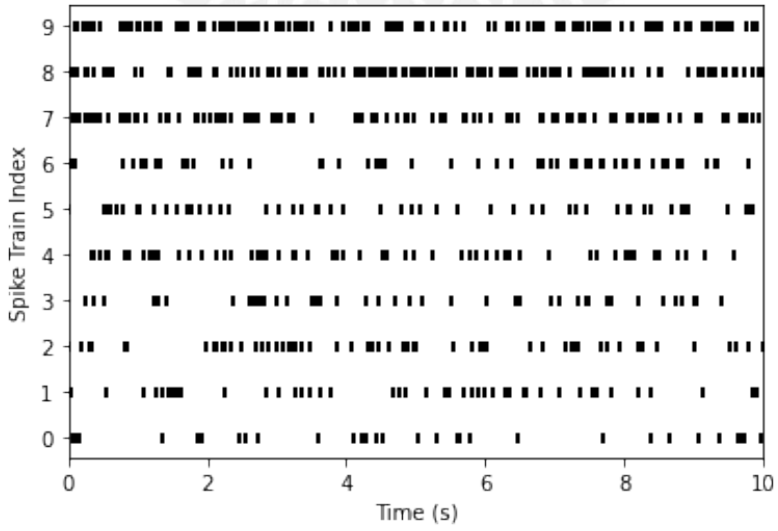
- **Threshold crossing level** This step applies a set of crossing levels to the whole signals (recordings, ADC values and normalized values, and NEO signals obtained



(a) Spike trains with 5 µV noise level



(b) Spike trains with 25 µV noise level



(c) Spike trains with 50 µV noise level

Figure 5.2: Spike trains with different noise levels. This figure demonstrates that they are equal to each other, as required for the next experiments.

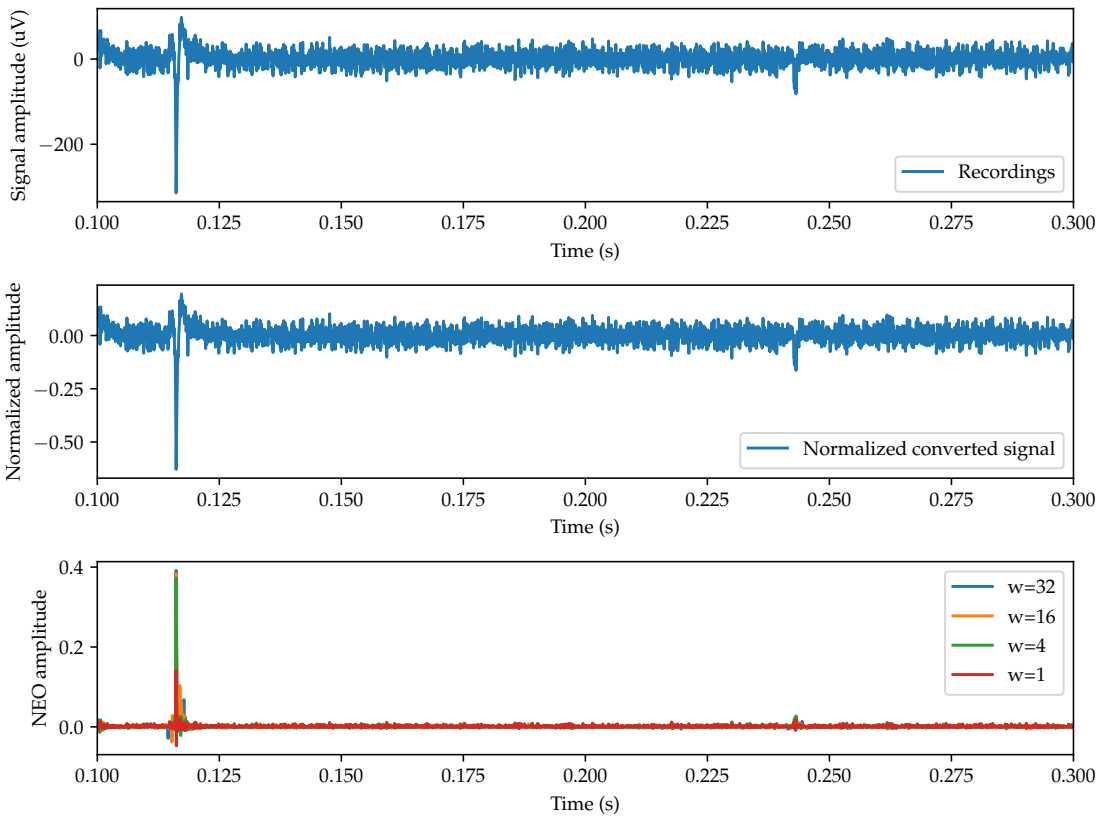
for different values of window width. The number of steps is a parameter configurable in the software developed. However, after a set of tests, we achieve an adequate value of 50 levels. The levels are relative to the corresponding signals. This means that each level represents a fraction of the maximum value. This is also an important condition for comparing the results between signals from different sources.

In this stage, the output signal is a set of vectors composed of zeros and ones, each vector with the same size as the input. Here, a zero value (0) represents a sample below the threshold level, and one (1) represents a sample above the threshold. In this case, as the converter model is for continuous sampling, the converted signal has the same length as the original recording signal. This condition will remain for all the tests that use this model of the converter, and only changes when it evaluates the non-continuous sampling model.

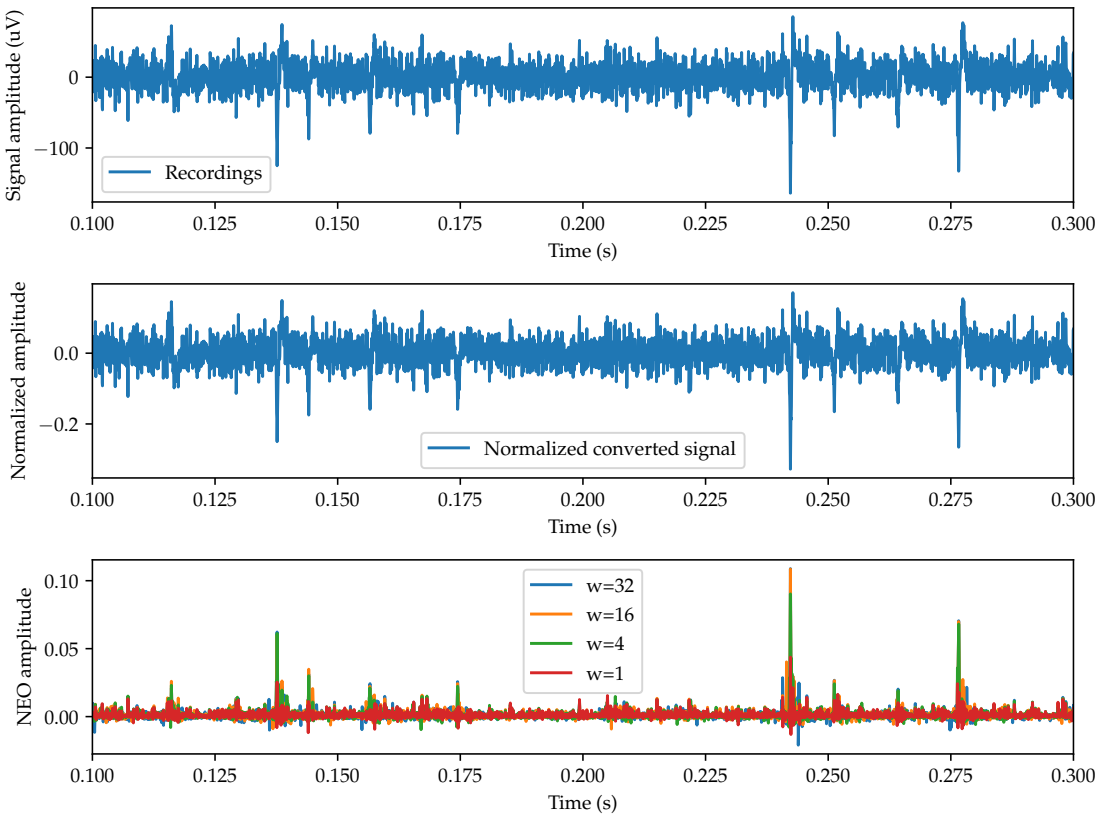
Figure 5.3 compares NEO applied to two channels for 10 µV of the noise level. First, we can appreciate that, although the time window is the same for both cases, the signals are notably different in the position of the spike trains and their amplitudes. With respect to the spike trains, depending on the position, some have more intensity than others for a specific electrode so the relative amplitude will be different. This causes some spike trains are better captured by some electrodes than others. On the other hand, the normalized signal has the same reference for both channels however, the amplitude is different. While in electrode 0 it is near 0.75; in electrode 27, its maximum is around 0.4. This is important to notice as it indicates that the same noise level, won't affect in the same manner to all electrodes or spike trains. This should be considered when evaluating a specific spike train with different noise levels and establishing the range of these levels.

On the other hand, the figure depicts that NEO emphasizes the picks in signal, keeping the rest of the signal at a lower level, independent of the window width chosen for its formula. In addition, reduces the relative amplitude of noise with respect to the detected spikes, favoring its detection with threshold levels.

Although for all the NEO values, the signal shows an enhancement with respect to the original converted signal, we also evaluate the width for the samples used in the NEO formula (3.2). Figure 5.4 shows the differences for distinct width values. In this case, the most significant difference is in the amplitude if width $w = 1$ is compared to $w = 4$. While in the first case, the amplitude only rises to 0.05, in the second case is close to 0.1. Additionally, negative values are reduced for widths greater than 4. Finally, the



(a) NEO application for channel 0.



(b) NEO application for channel 27.

Figure 5.3: Results comparison of NEO applied to two channels with 10 µV of noise, a bandwidth from 0.3–6 kHz, and a sampling frequency of 20 kHz, both with a time window from 0.1 to 0.3 seconds.

differences between the results for widths 16 and 32 are almost indistinguishable. However, it could change when there are more different levels of spikes where minimum differences in amplitude could influence their detection or not.

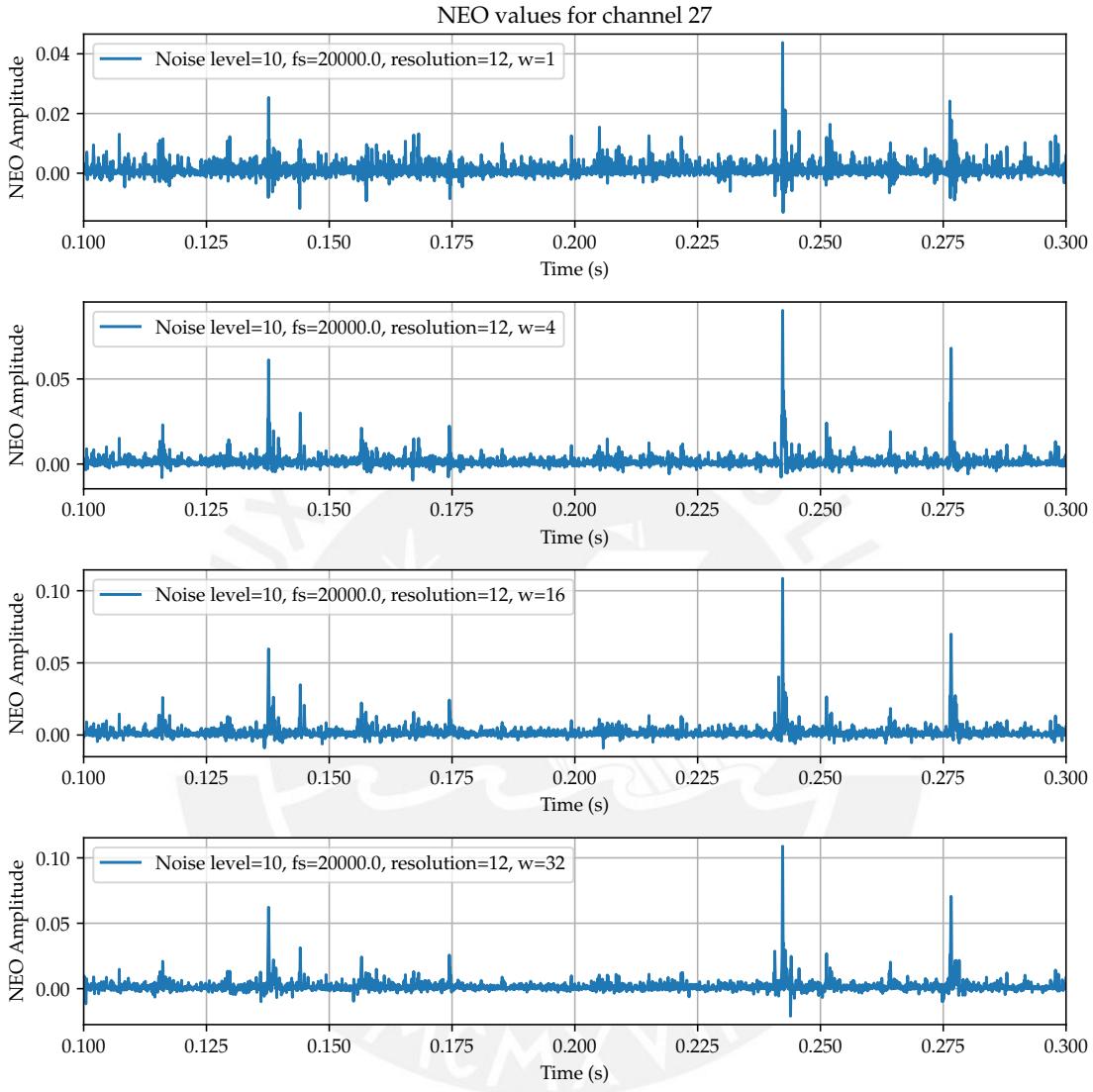


Figure 5.4: Results comparison of NEO for different widths applied to signal with 10 µV of noise and 20 kHz of the sampling frequency, captured in channel 27.

One of the benefits introduced by the NEO operator is that, against the same relative threshold level, the number of samples that crosses the threshold level is reduced. It implies that the quality of false positives could be decreased. Figure 5.5 shows that effect. In the case of the original recording signal and the normalized signal, the threshold is for the absolute values. In the case of NEO, the threshold only applies to positive values. The number of samples (orange dots) that crossed the threshold is significantly higher in

both the recording and normalized signal with respect to the NEO. It suggests that the NEO is more selective than the original signals against a relative threshold level.

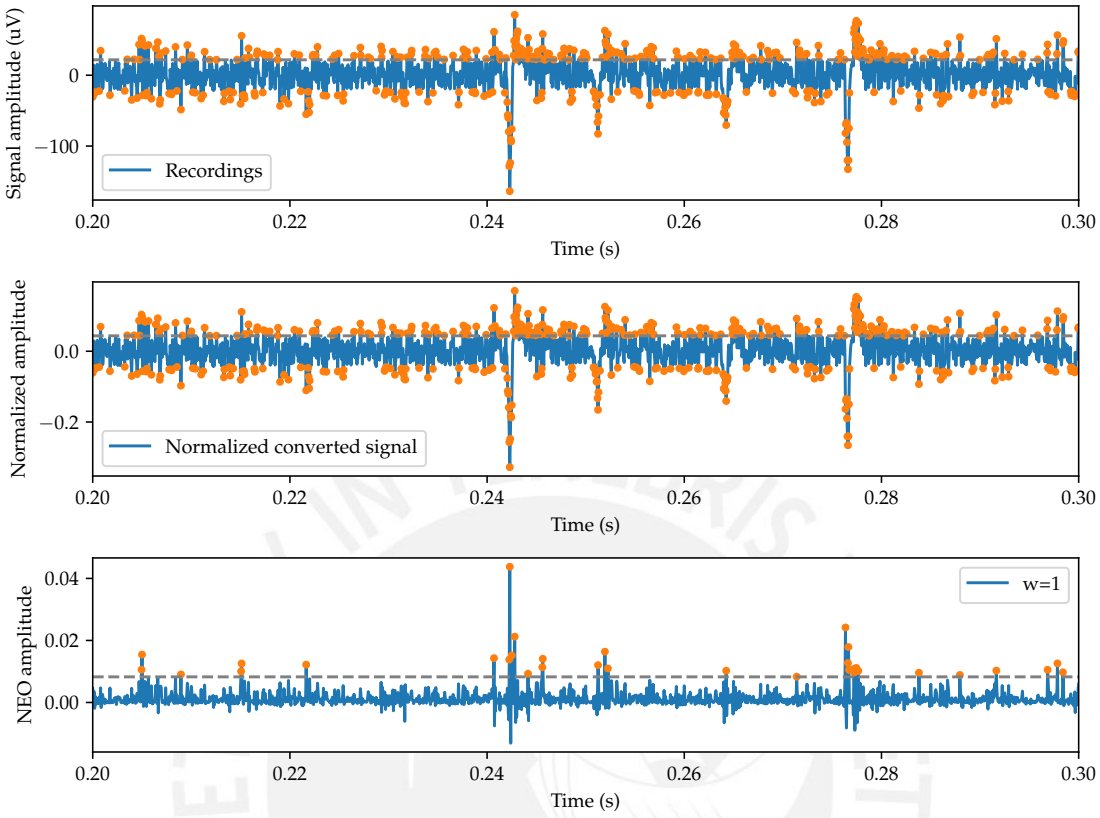


Figure 5.5: Results for the crossing level detection with a signal of 10 µV of noise and 20 kHz of the sampling frequency, captured in channel 27. Here the orange dots represent the samples that crossed the level, considered ones in the output signal. The dashed gray line represents the relative level against which they are evaluated.

However, it is not enough visual inspection to evaluate the NEO detection capacity. It must consider other metrics to measure the improvement. For example, Figure 5.6 shows the number of values over each threshold level and compares the curves against the recordings, the normalized signal, and the different widths for the NEO operator. According to this figure, for lower values of thresholds, the number of samples that crosses them is higher, and consequently, those levels don't constitute adequate detectors for spikes. Then, as the threshold level increases, the number of samples that crosses the threshold is minor for all the signals. Nevertheless, for the NEO signal, the approaching to reduced values are faster than the recording case. It indicates that for the same relative threshold, NEO selects the spikes better than the normal signal. In addition to this, a variation in the width doesn't have too much relative influence when its value is higher than 4.

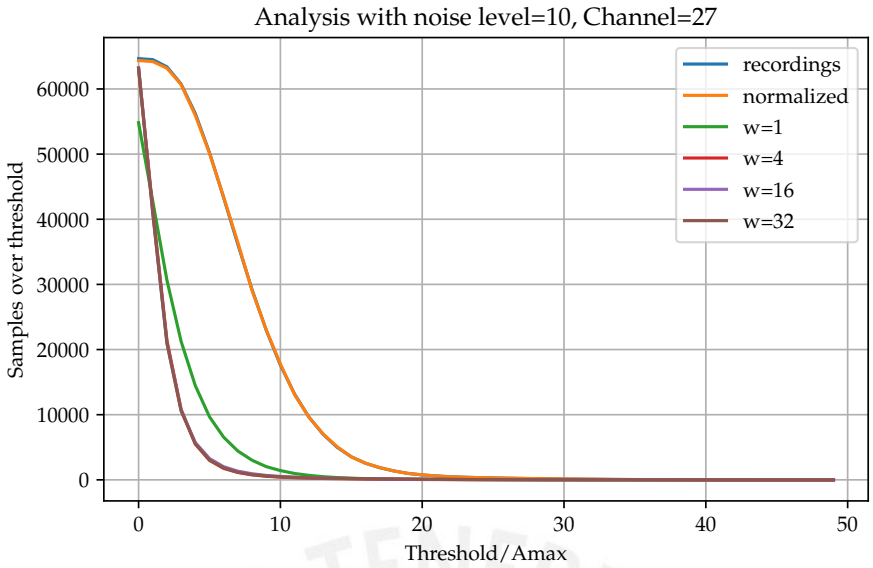


Figure 5.6: Results comparison of the number of spikes for original signal, sampled signal and NEO for different widths applied to signal with 10 µV of noise

5.2.1.1 Selection of parameters for evaluation

Due to the realistic variations implemented by the MEArec simulator, they produce variations in the spikes amplitude and electrode signals as shown in Figure 5.3. Then, to uniform the next evaluations, it is required to set up some fixed parameters.

Firstly, to understand the spatial distribution of the electrodes and signal origin in the neuron, we plot them in Figure 5.7. The spike trains' amplitude captured in the electrodes is influenced by the original amplitude in the neuron soma, its SNR, and the distance from the soma to the electrode position. Those factors will produce those electrodes to capture better some spike trains than others. In addition, depending on the density of neurons around a specific electrode, it could capture a determined number of spike trains.

Observing the different recording signals in the electrodes, we identify that electrode 27 has minor variations between spike amplitudes, and we choose this electrode number for further evaluations of resolution, bandwidth, and sampling mode.

Then, another parameter to choose is the sampling frequency. In this case, the default frequency is 32 kHz for both templates and recordings. However, to use a more realistic sampling frequency, we changed it to 20 kHz. This frequency is set from the generation stage in the MEArec software, and all the posterior results will be executed against these new recording signals with the modified sampling frequency. It is important to highlight that the templates remain with the same high frequency and, inside the generation process, the software resamples those templates to generate the recording with a new sampling

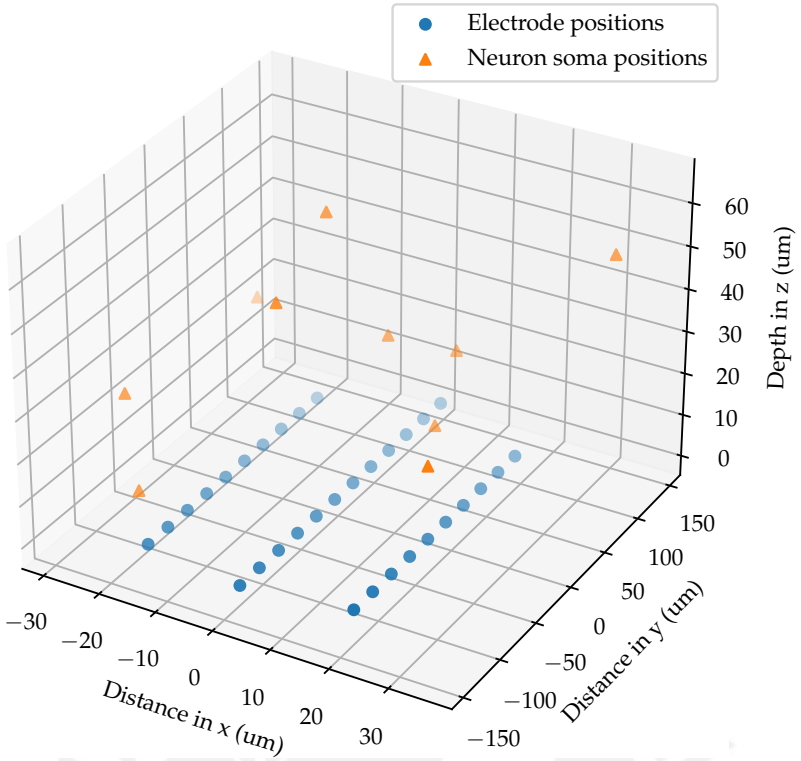


Figure 5.7: Positions for the electrodes and neuron somas as the origin for the different spike trains. The x-y plane represents the base for the electrode array, while the positive z-axis represents the depth inside the cortical tissue.

frequency. There is another alternative which consists of resampling the high-frequency recording directly. However, it has inconvenient that samples that correspond to the maximum values in spikes could be missed out in the process. Although it could happen in a realistic recording, the intention is that all spikes could be present to have a proper reference to evaluate changes in other conversion or preprocessing parameters.

Finally, the noise levels must be selected. In the beginning, we thought of a wide range of values from 5 µV to 200 µV; however, taking into account the selected channel to analyze (27), we chose low values for the noise levels, which are 5 µV, 15 µV, and 25 µV. Although the last one could appear as a low noise value, it has almost the same amplitude as the acquired spikes in the electrode. It is adequate to evaluate the performance when the noise is comparable to the signal amplitude.

5.2.1.2 Metrics calculation

To properly evaluate the detection performance for each scenario, it is required to uniform the calculated metrics. The classical approach for this is to generate curves of observation ratio (ROC) or accuracy curves, both based on the numbers of true positives (TP), false

positives (FP), true negatives (TN), and false negatives (FN). However, to obtain these values, it is required to properly define the references and decide which are spikes and which are not.

Which are spikes? The first question to decide how to evaluate the results is determining what constitutes a spike. The spike in the physiological basis is the complete action potential signal. Although, it could be distorted in the extracellular acquisition due to the distance and noise, it will preserve a value with a high amplitude and a resting time when no more action potentials could be triggered.

The spike trains are a mix of different neurons' signal amplitudes and their noise levels in the generated signal. Then, there is no clear reference to determine which are spikes and which are not, apart from the maximum local values that may be confused with noise. In addition, the information obtained from the simulation software only provides the position for the maximum value for the spike. It could be later used for the spike alignment process, but it is not enough to determine the length of a specific spike.

Therefore, planning a process to determine the samples that may be considered spikes is required. Then, those will be compared to the original maximum positions for the known spike trains, and the desired values of true positive, false positive and others will be calculated.

The first approach considers only the sample of maximum value as one spike and the other as not spikes. Then, when comparing to the original maximum values, if the corresponding sample matches the maximum value position, it is considered as a true positive, while others that cross the level and do not match as false positive. This procedure, although simple, has some inconveniences in the obtained metrics, such as overweighting the number of false positives. This is because it considers several contiguous sampling as different spike trains, although only one spike could be triggered at that time. In addition, it considers contiguous samples to the maximum value as no spikes, although we know that for a specific neuron, it can not trigger too close spikes. Then this approach for the calculation should be staked out.

The second approach considers the refractory time to determine whether a spike is detected or not. Although this strategy could be addressed by different means, the proposal is to split the whole time reference into potential spike periods. Here, we use the reference position for the maximum value of the spike as the center for each period. Then, the complete period represents one spike, and it is enough that one sample crosses the

level in this period to be considered as only one detected spike. On the other hand, we split the remaining time into periods with the same window considered for the true spikes. This could lead to some periods in proximity to the true spike windows with less or more duration than the others. However, they are inferior in quantity to the equally spaced period. These periods that don't contain a spike are considered true negative spikes.

Computation of TP, FP, TN, and FN. To compute these values, we consider the second approach described above, and they are:

- **True positives (TP)**, For each true positive window centered on the reference position for its maximum value, we compare to the vector of zeros and ones obtained from the threshold detection level. Here, if one sample of this signal crosses the level (value equal to 1), we consider the whole window and samples as one true positive.
- **True negatives (TN)**, For each window where we do not expect spikes if no sample crosses the threshold (all values are 0), we consider the window as one true negative.
- **False positives (FP)**, Similar to the previous one. In this case, we use the potential true negative spike windows, and if one sample crosses the level (value equal to 1), we consider the entire window as one false positive.
- **False negatives (FN)**. In this case, if in the window, we expect there is a spike, all the samples are below the cross-level (they are equal to zero), then we consider this window as one false negative. This value could also be interpreted as the number of missed spikes.

Based on these values, we can calculate the True Positive Rate (TPR) as the total of TP divided by the number of expected positives in the signal, which will correspond to the total of spike trains:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.1}$$

In the case of False Positive Rate (FPR), it is calculated as the total of FP divided by the number of expected negatives (no spikes) in the signal, which results from the total of windows established minus the total of spikes. Then:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{5.2}$$

These sets of values build the ROC curve, which shows how good the classification of a spike or not is for a determined signal. Here, each point in the ROC curve corresponds

to one threshold level in the following interval:

$$\text{Threshold level} \in \left[\frac{A_{\max}}{\text{Number of thresholds}}; A_{\max}\right] \tag{5.3}$$

In addition to these, another metric used is the accuracy. Here, we employ a modified version of the general formula, as it does not consider the TN in the calculation. Then, it results in:

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{5.4}$$

5.2.1.3 Selection of spike trains

As we commented before, not all the spike trains will have the same amplitude in a specific electrode, and consequently, only specific spike trains could be detected with a threshold level.

The following evaluations require selecting which spike trains will be considered. The method followed is to choose an intermediate threshold level value and apply it to the recordings. Then, we obtain the values for TP and FN for each spike train and evaluate them against the windows where we expect spikes. Finally, we plot those results for each spike train as bars. Here, the number of detected spikes is represented as filled bars, while the number of not detected spikes is represented as not filled regions.

This visual graphic help to decide which spike trains could be considered and would provide more valuable information about the detection capacity for the different signals. Figure 5.8 shows this plot, which is obtained from a recording signal with a lower noise level (to have preferably only spikes), and with a relative threshold level of 15/50 with respect to the maximum amplitude in the recording. We use this level as it shows an intermediate quantity of fully detected spike trains (lower values will result in all spike trains detected but could be confused with noise and high values only show the spike train of maximum amplitude). According to this figure, at least 5 spike trains are almost fully detected. From them, 3 correspond to excitatory cells (lower frequency) while 2 correspond to inhibitory cells (high frequency). In the next calculations, we can consider any of them, preferably from 5 to 8 as they have a higher number of spike trains and could provide fine resolution in the metrics calculation (TPR, FPR).

Then, we can appreciate how the selected spikes trains match the signal characteristics. Figure 5.9 shows the recording, converter, and NEO signals for channel 27, with 5 µV of noise level, and in the top region of each subfigure, the selected spike trains positions are marked as red crosses. Here, this plot reinforces the selection of the spike trains as

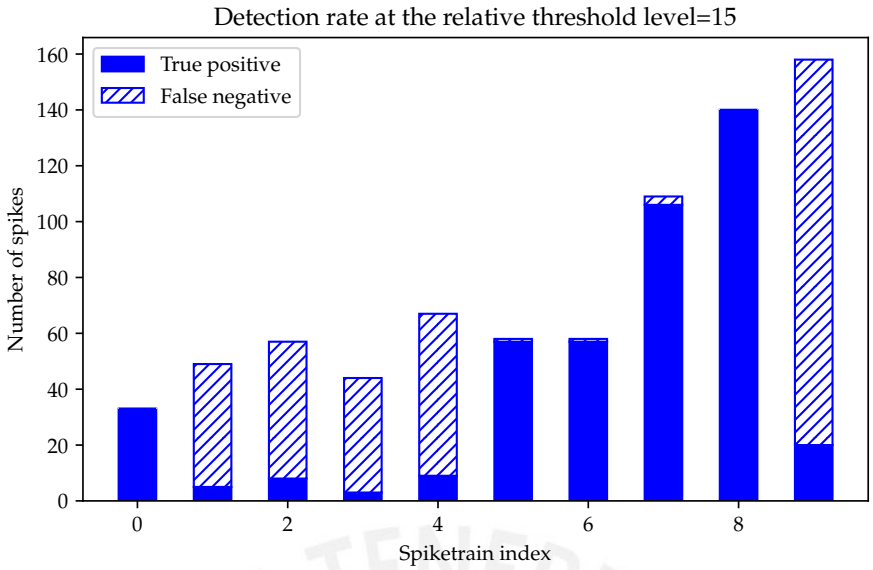


Figure 5.8: Detection values obtained with the signal in channel 27 with 5 µV noise level, sampled at 20 kHz, and with a relative threshold level of $\frac{15}{50}A_{\max}$, where $A_{\max}$ represents the maximum amplitude of the recording signal.

the visible spikes in the recording (or where we expect spikes), match exactly with the positions of the selected spike trains.

5.2.1.4 NEO evaluation with fixed parameters of resolution, bandwidth, and continuous sampling mode

With these settings, we evaluate the NEO operator employing the ROC curves which compare the true positive rate (TPR) vs. the false positive rate (FPR). Figure 5.10 shows the results for 5 µV and 25 µV noise levels. Here, we only plot the evaluation for the spike trains 5, 6, 7, and 8.

The ideal ROC curve for a perfect classifier looks like a corner, with minimum values of FPR and higher values (almost 1) for TPR along the range of FPR. This is because when the threshold has a low value, it will consider all the windows as spikes, and then the rate of false positives and true positives will be one in each case. Then, as the threshold increases in its value, fewer windows are interpreted as positives, and ideally, it will remain to detect all the spikes keeping the value of TPR at 1. The tendency is to reduce the FPR to a minimum value of almost 0, while FPR remains at 1 until the threshold level is higher enough that could not detect any spike, going to TPR and FPR equal to 0.

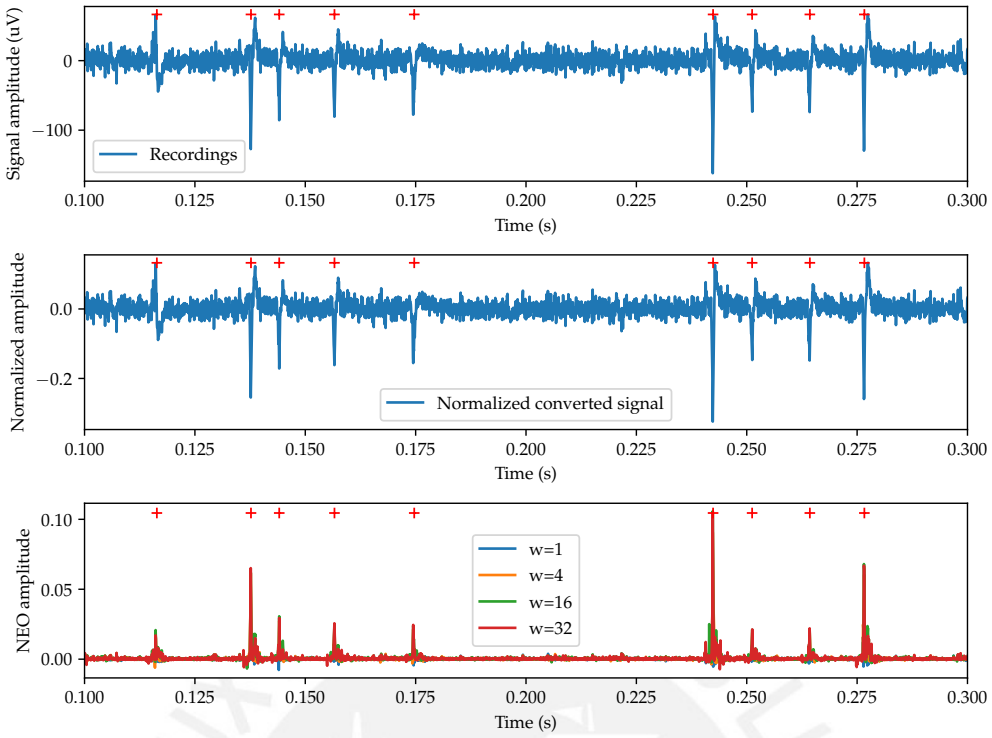


Figure 5.9: Signal in electrode 27 with 5 µV noise level sampled at 20 kHz. The red crosses over the signals represent the position of the maximum values for the spike trains 0, 5, 6, 7, and 8, identified in the detection rate plot.

However, there is no perfect classifier for non-deterministic signals, such as the recordings of neural activity. Then, we won't have a perfect corner in the ROC curve, but we expect it could follow that tendency.

In Figure 5.10, we can appreciate that there is a clear difference between ROC curves for 5 µV and 25 µV levels of noise. While with low noise, the curves are more similar to the ideal classifier, with high noise, the TPR is affected, decreasing its value. On the other hand, the ROC curve is different for different spike trains. Although these differences are not too evident for 5 µV of noise; for 25 µV, it shows how the performance decreases considerably, being worst for spike train 7 (more similar to a line, which is a poor classifier) and minor for spike train 8. This result is consistent with the characteristics of spike trains because spike train 8 is the spike train with more relative amplitude and is expected to be classified better.

On the other hand, Figure 5.11 shows the accuracy for each spike train analyzed and for two different noise levels. The plots present differences between NEO detection and recording/normalized signal detection. In the first case, the maximum detection accuracy is achieved at lower relative threshold values, while in the recordings, this value is higher.

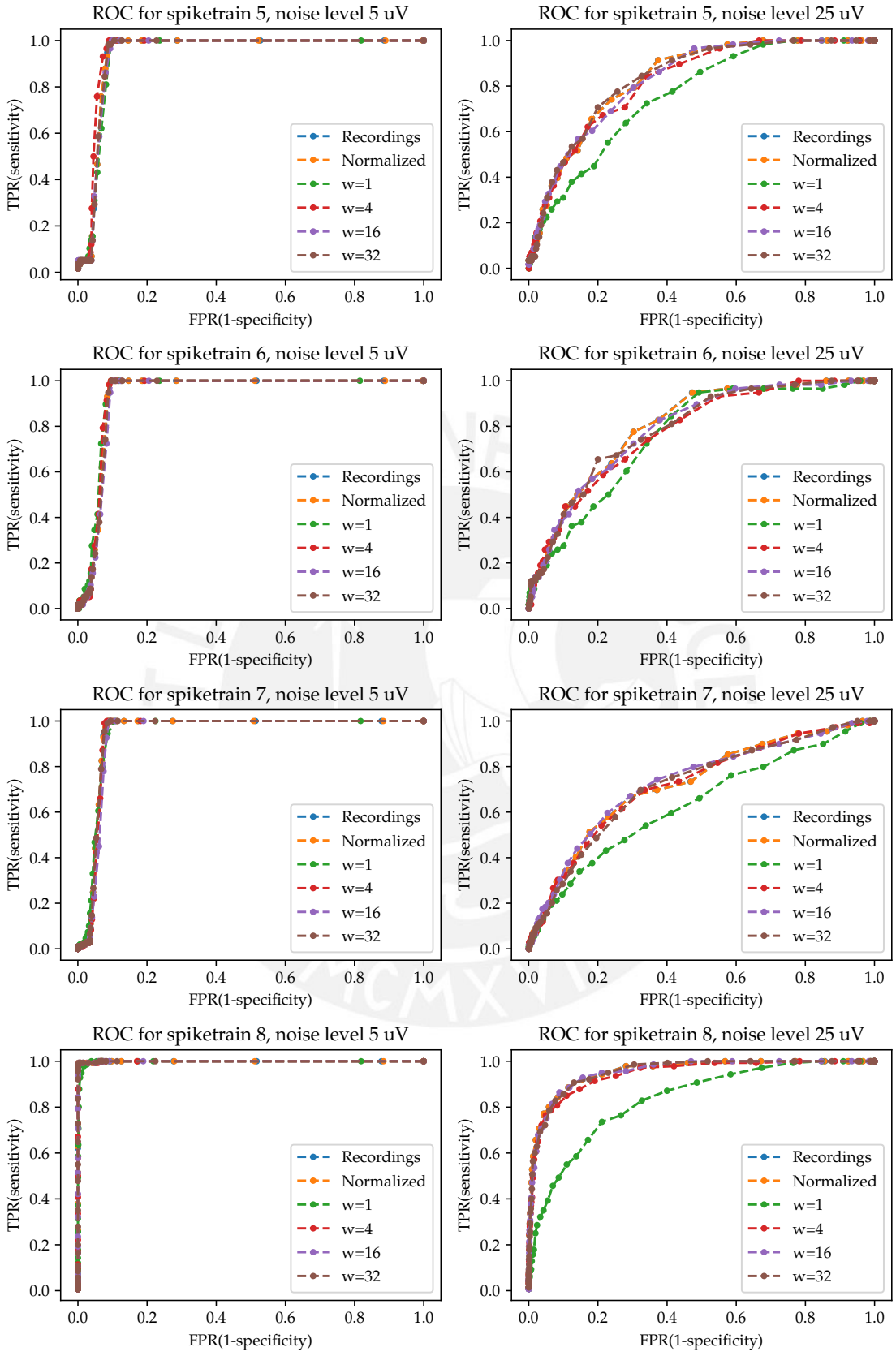


Figure 5.10: Comparison between resulting ROC curves for both 5 µV and 25 µV noise levels in channel 27 sampled at 20 kHz.

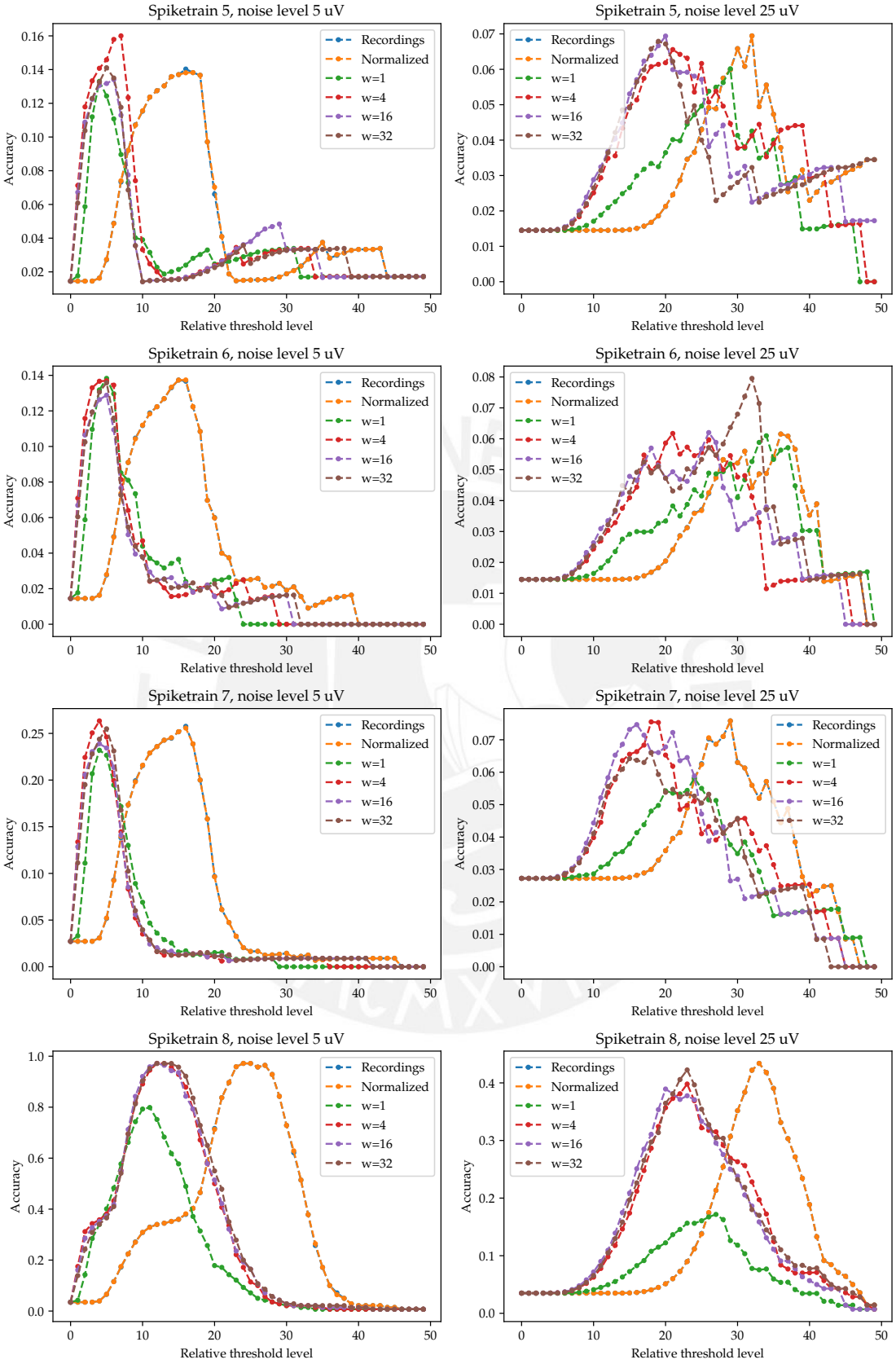


Figure 5.11: Comparison between resulting accuracy for both 5 µV and 25 µV noise levels in channel 27 sampled at 20 kHz.

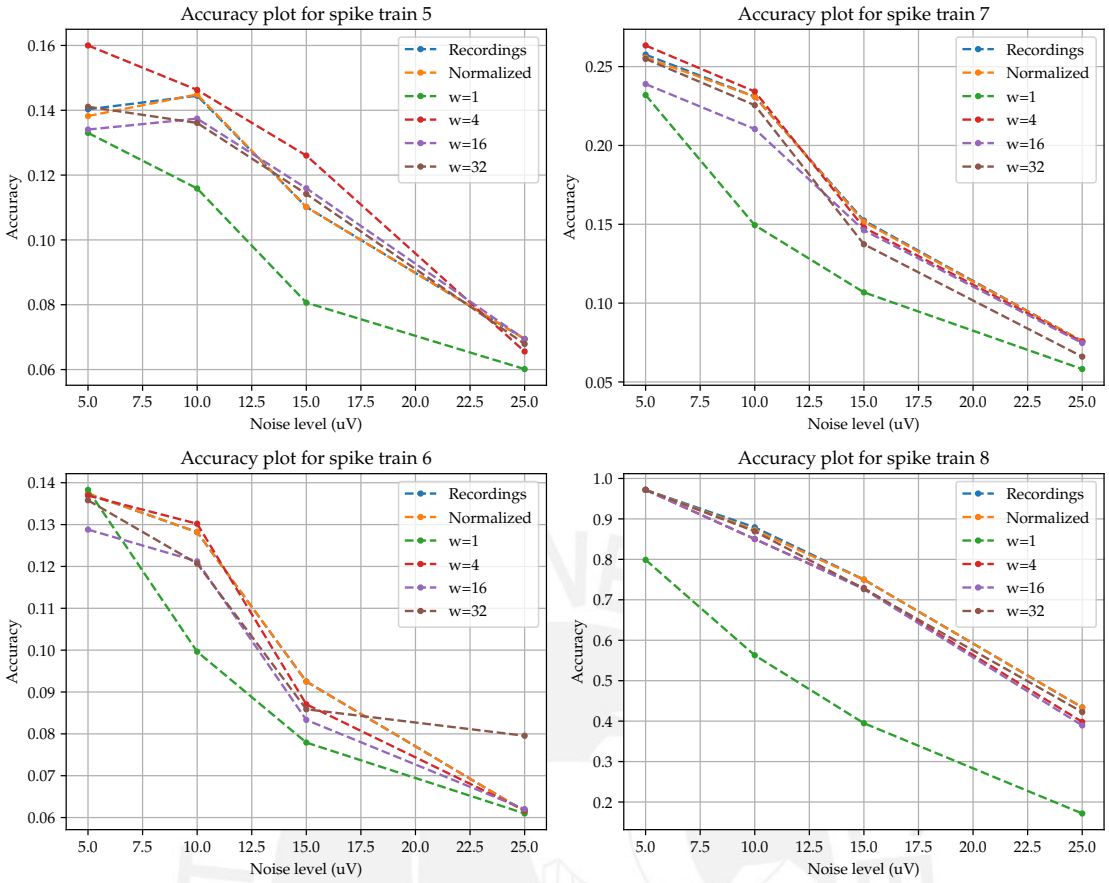


Figure 5.12: Variation of accuracy for different values of the noise level.

In addition, the maximum accuracy value is different for each spike train. For a noise level of 5 µV, the spike train 8 has a maximum value near 1, while the others have values significantly lower (0.25, 0.15, 0.16).

All the accuracy curves have a determined level which produces the higher value. These maximum values can be compared between the different signals: the original recording, the converted with 12 bits of resolution, and the NEO signals for different width values. Figure 5.12 shows those curves. In this case, NEO curves and the original signal with threshold level detection show similar performance at different noise levels, being better with the NEO only in particular spike trains with specific noise levels.

According to the results, the NEO has not too much influence in detecting spikes, at least concerning the false-positive. However, regarding the relative threshold level, NEO could be more selective than the original classification applying the threshold directly to the recording signals.

In addition, the width in the NEO improves the detection, being better for high values, against the original NEO, which takes contiguous samples for the energy calculation.

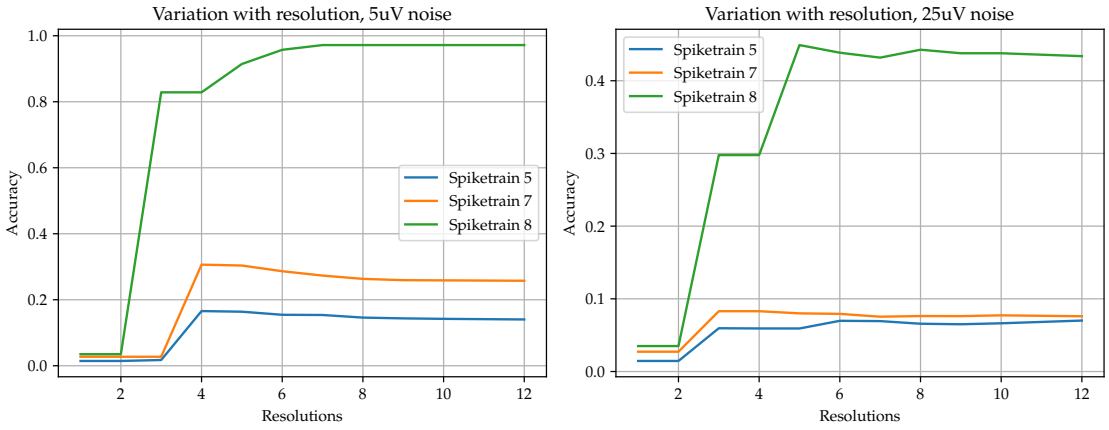


Figure 5.13: Accuracy variation for different resolution values. The left plot is for signals with 5 µV, while the right plot is for signals with 25 µV. All of them were sampled at 20 kHz.

5.2.2 Experiment 2: Evaluation of resolution variation

In the previous experiment, all the converted signals have a fixed resolution value (12 bits) as employed in different studies. However, this thesis intends to demonstrate that different lower-resolution values could maintain the performance for the detection spike. Then, we evaluate different resolutions to verify this characteristic.

The process followed in the experiment is the same as the previous experiment, and only it will increase the number of evaluations for each resolution value.

Originally, the evaluation was limited to two values of resolution (12 and 8 bits) which compared the state-of-the-art resolution against the proposed resolution in this work. However, those were extended to more resolution values, ranging from 10 to 1 bit. This is to appreciate the impact of lower resolutions than the proposed one.

Figure 5.13 shows the effect on the accuracy when the converted signal is quantized at different resolution values. The figure for lower noise level and spike train with higher amplitude (spike train 8) depicts that decreasing the resolution could affect the metrics in the accuracy. Then, for values lower than 7 bits, metrics obtained could not properly reflect the detectable spikes with a specific noise reference. The second graphic exhibits that influence. Apart from decreasing the maximum accuracy as expected at this noise level, decreasing the resolution produces a fluctuation in the accuracy value, which does not correspond to the actual characteristic of detectable spikes in the noisy signal.

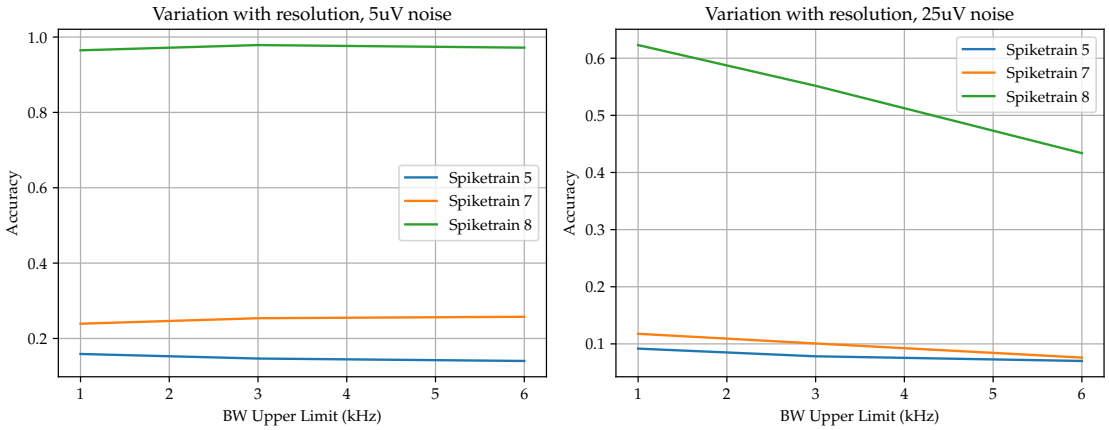


Figure 5.14: Variation of accuracy for different upper limits for bandwidth. The left plot is for signals with 5 µV, while the right plot is for signals with 25 µV. All of them were sampled at 20 kHz. The resolution for both is 12 bits.

5.2.3 Experiment 3: Evaluation of bandwidth variation

All the recordings employed till now had a limited band from 300Hz to 6kHz, where we expect it concentrates most of the signal energy. However, we aim to evaluate narrower bandwidths.

The original proposal from the literature is to reduce the bandwidth to 0.3-1kHz, which the authors claim is enough to achieve adequate performance levels. In this work we evaluate it against two other widths: the complete bandwidth for the neural signal 0.3-6kHz, an intermediate value of 0.3-3kHz, and the suggested value of 0.3-1kHz. All of them have the same lower limit for the band as the known limit that separates the single-unit activity from the local field potential (LFP), minor to 300Hz, which is not considered in this work.

The recording generation stage includes the filter to select the bandwidths. Therefore, we generate new recording files for the new required bands specifying the upper limit for the band in the corresponding configuration file. In the developed software, we enter the change programmatically. We apply the usual conversion process for each of the files obtained, including NEO preprocessing and threshold level crossing.

Figure 5.14 shows the influence of the variation of bandwidth on the accuracy. It shows that when the noise level is low (i.e., 5 µV), the effect on the accuracy is almost undetectable, remaining in the same value as with the full bandwidth. However, when the noise amplitude increases, as shown in 25 µV, the accuracy increases by approximately 0.2 for the spike train with higher amplitude (spike train 8). Other spike trains present the

same behavior, although the increment in the accuracy is minimal as the accuracy with full bandwidth is lower.

The presented results coincide with the literature, where proposes that considering the SBP band (0.3-1kHz) could be enough to achieve an adequate detection rate.

5.2.4 Experiment 4: Evaluation of non-continuous sampling

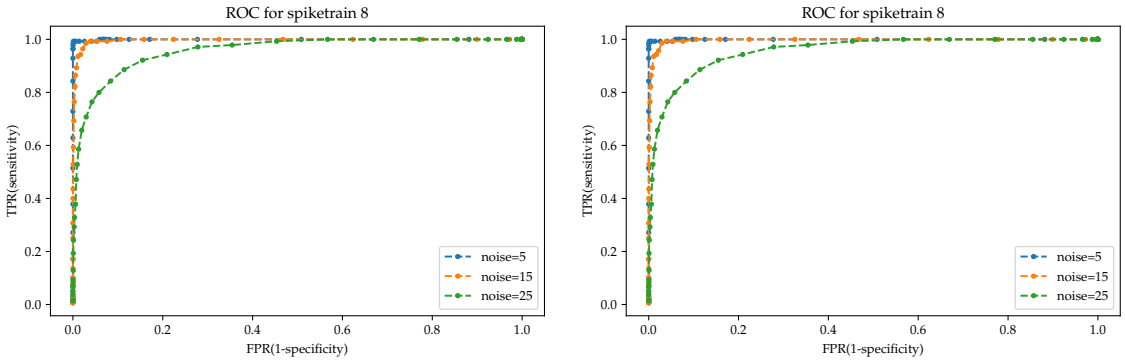
All the previous experiments in the conversion stage used a converter model for continuous sampling. It means that each value in the recording signal has a corresponding value in the encoded amplitude signal, which is the output of the ADC model. However, another evaluation stated in this work is related to the type of conversion. In this case, the proposed architecture is level-crossing, which only takes samples when the amplitude difference relative to the previous sample is higher than a predefined value. The software model for this ADC takes this idea to compare the samples against the previous converted value and then, according to this difference, decides whether the current sample must be included in the output signal. This type of sampling leads to a non-continuous output. Then, apart from the output signal, it includes the indexes corresponding to the original samples in the recording, besides, it could serve as a time reference for the posterior signal reconstruction.

In this case, we apply this converted model for each recording with a determined bandwidth, noise level, or sampling frequency, and obtain the corresponding non-continuous values and their time references. Although the converted values and the time references have the same length, it is lower than the original. It will impose some changes in the storing procedure for these signals and some challenges to the evaluation stage.

Then, in the NEO processing stage, we apply the same formula of the simple NEO and the spaced NEO, despite knowing the samples are non-continuous. We aim to evaluate precisely how the NEO could perform with samples not equally distanced in the time. The output signal will have the same length as the converted signal.

In the threshold crossing level stage, the procedure is similar to the previous experiment and will result in a vector of zeros and ones with the same length as the converter output. However, it will match the indexes, which must be considered in the evaluation stage.

Figure 5.15 shows the result for a continuous sampling conversion and the non-continuous sampling conversion. The obtained plots are almost identical, which suggests that the style of conversion in LCADC could not affect the performance of detection. However, this be-



(a) Continuous samples

(b) Non-continuous samples

Figure 5.15: Comparison between ROC curves for the same spike train but different conversion modes: continuous and non-continuous (LCADC model). Both models are for 12 bits of resolution applied to a signal with different noise levels.

havior could change with low-resolution values. That comparison will be evaluated in the next experiment.

5.2.5 Experiment 5: Evaluation of combined effects of resolution, bandwidth, and sampling method

In previous results, we have evaluated the impact of individual variations of resolution, bandwidth, and sampling method. Now, we will test a combined effect of those parameters. As the proposal for this thesis is using another strategy of sampling and comparing against conventional, we will compare the parameters taking both cases, the uniform and level-crossing sampling method.

In addition, we will add the effect of noise level and evaluate how the variation is affected by it. We decided to use two levels of noise for each test, which we consider representative: 5 µV, and 25 µV, standing for a low and high noise level.

Other parameters not being evaluated in a specific test will remain in their conventional value: the resolution of 12 bits, the bandwidth up to 6 kHz and the sampling frequency of 20 kHz.

The measurement considered for all the following tests is the maximum accuracy achieved for different thresholds, applied to the spike train 8, which is shown to be more prevalent in the chosen channel than the others. This measurement was also used in Figure 5.13 and Figure 5.14.

5.2.5.1 Variation of resolution

We selected the results for a full bandwidth and a sampling frequency of 20 kHz. Then, we group them by the noise level (selecting the two levels chosen for our analysis) and by the sampling method. In Figure 5.16, we can see the results of the effect of resolution for each scenario.

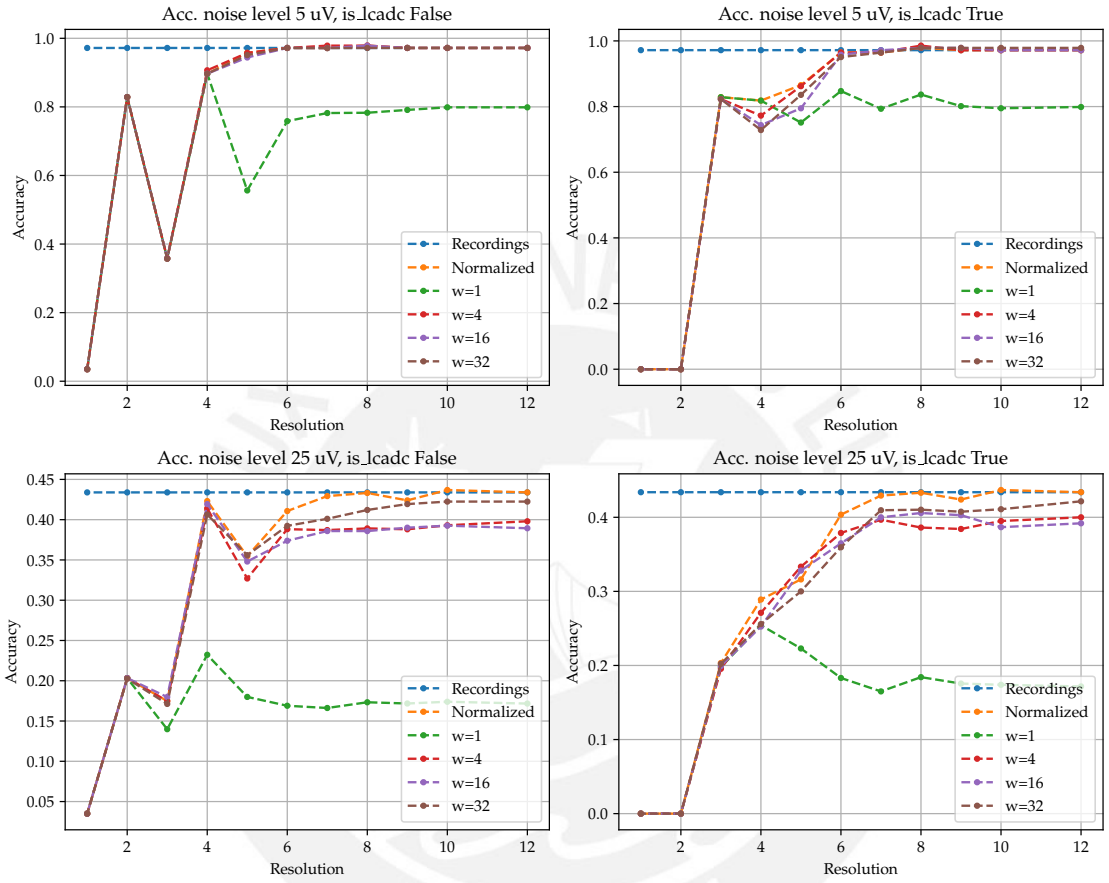


Figure 5.16: Comparison of maximum accuracy against resolution variation. The plots on the left represent the accuracies for uniform sampling, while the plots on the right represent the corresponding level crossing sampling. The plots at the top are the results for 5 µV, while the ones at the bottom are the results for 25 µV. In this graphic, we show the variation of maximum accuracy for each signal variant, original recording, the normalized quantized output, and the NEO with different widths, as used in previous evaluations.

From the graphic, we can identify that there is a minimum resolution that can achieve almost the same level of accuracy as higher values, and it is between 6 and 7 bits. This reference is valid even at different noise levels, being more differentiated when the noise increases. From this resolution to the below values, we cannot reach the maximum accuracy we can obtain from a determined noise level. In the case of the NEO operator, we

also confirm that windows from $w = 4$ can achieve almost the same level of accuracy as bigger windows, discarding the value of $w = 1$, which gives the worst results.

This means we can recover enough information to detect the spikes corresponding to a spike train up to a certain resolution limit. Beyond that resolution, we are not able to extract such information. On the other hand, we observe an outlier in the uniform sampling method, which doesn't follow the tendency if we compare the result at resolution 3 and resolution 2. Here, instead of decreasing with lower resolution, it increases that. We can explain the higher accuracy in resolution 2 as the quantization levels can also act as a filter to noise. If the noise is lower than a certain value, it could be lowered to 0 in the quantization result, decreasing the number of false positives. In this case, the quantization favors the isolation of higher magnitude spikes, but in the case of resolution 3, it acts against it. While it is true that quantization could lower undesired peaks, if the levels are lower enough, as in the case of the higher order of resolution, the rounding process could pull up to higher values and consequently create more false positives. We also must consider that we are evaluating against a particular spike train, but the neural signal is composed of several spike trains; then, the rounding process in the quantization could also enhance lower energy spikes from other spike trains and increase the number of false positives. The level crossing doesn't present this effect as the signal must cross the level to acquire a sample and doesn't have any rounding process included.

5.2.5.2 Variation of bandwidth

In this case, we maintain a fixed value of the resolution (12 bits) and sampling frequency (20 kHz) while varying the bandwidth. In Figure 5.17 we can see the results of the change in the upper limit for three values 6 kHz, 3 kHz, and 1 kHz.

From the graphic, we appreciate that the accuracy level is almost the same for low noise levels, either for a uniform or a level crossing sampling method. However, when the noise increases, we observe an increased maximum accuracy for lower upper limits (1 kHz) compared to the other. Concerning the NEO, we observe that, following the tendency of previous results, we obtain the worst results for $w = 1$ while for widths starting from $w = 4$ are similar to higher-order windows.

We can explain that result if we consider that the spike energy is more concentrated around lower frequencies, specifically around 1 kHz. Then, if we suppress higher frequencies, although we can also suppress part of the spikes' energy, the most affected is the noise. It also is reflected in the spikes that can pass the filter, as false positives generated

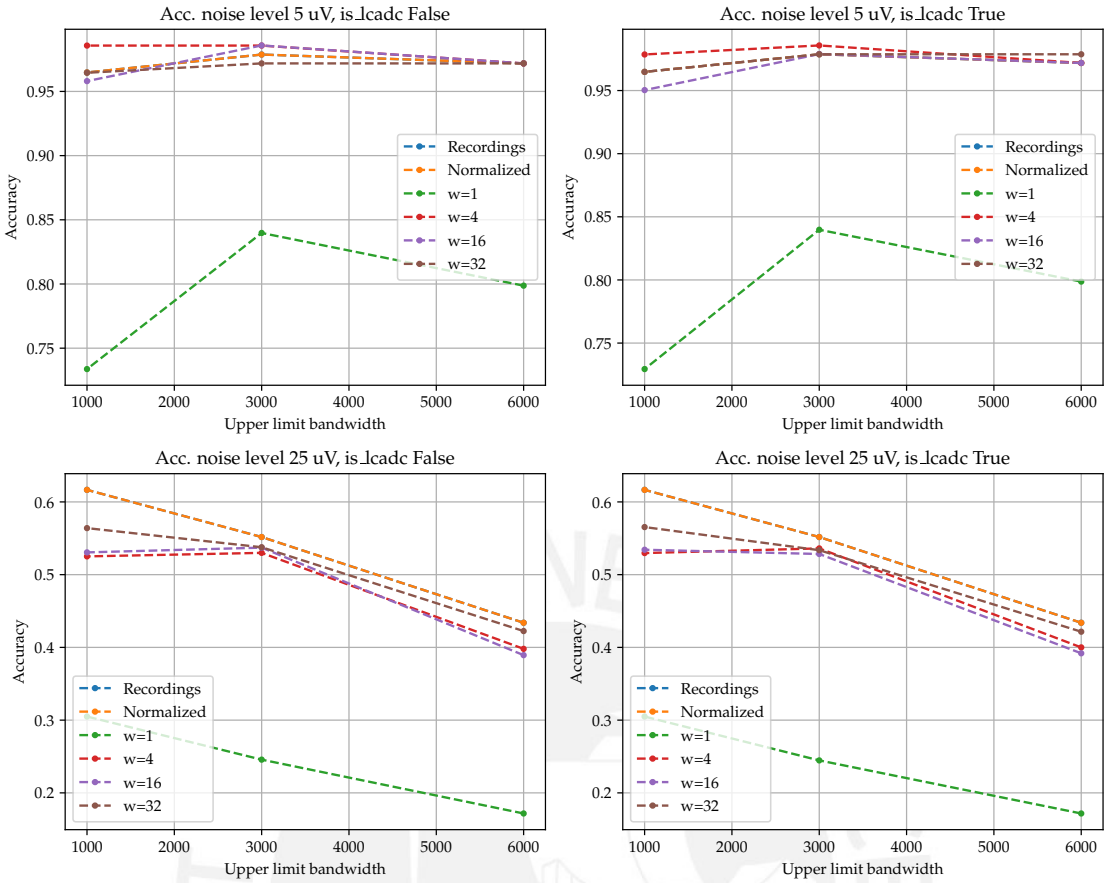


Figure 5.17: Comparison of maximum accuracy against bandwidth upper limit variation. Plots are distributed relative to the sampling method and noise level, equally to the resolution evaluation. We also show the variation of maximum accuracy for each signal variant: the original recording, the normalized quantified output, and the NEO with different widths.

by noise are also suppressed. That's why in the presence of a higher noise level, the benefit of lower bandwidth is more evident than in the case of a lower noise level.

5.2.5.3 Variation of sampling frequency

Although we have considered a fixed frequency of 20 kHz in the previous analysis, we also evaluate the effect of the sampling frequency on the spike detection accuracy. Figure 5.18 shows this effect by taking three values: 20 kHz (used in previous experiments), 24 kHz, and 16 kHz.

In this case, we can appreciate that for low-level noise, the improvement of high sampling frequency is almost not noticeable. The same occurs when we compare the uniform sampling method against the level crossing. However, when the noise increases, higher sampling frequencies can return higher levels of maximum accuracy. That behavior is

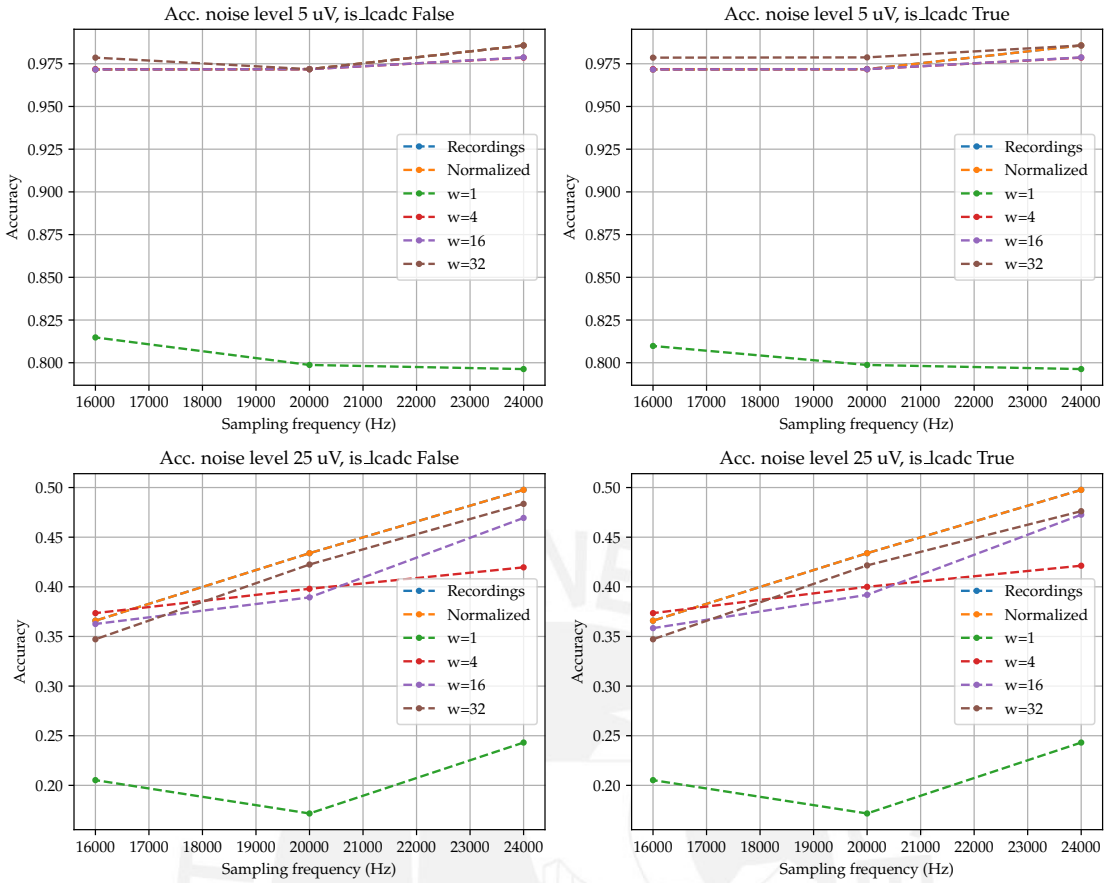


Figure 5.18: Comparison of maximum accuracy against sampling frequency variation. Plots are distributed relative to the sampling method and noise level, equally to the resolution and bandwidth evaluation. It includes the variation of maximum accuracy for each variant of the signal.

replicated either in conventional sampling or in level crossing. In the case of the NEO operator, we identify in the higher noise level that the decrement in performance is different depending on the window size. While a bigger window could return a high accuracy level for high frequencies, it is the opposite for lower sampling frequencies, being better for a window $w = 4$ at a sampling rate of 16 kHz.

We can explain the higher accuracies for higher sampling frequency as it can recover more information from the recordings and help to isolate spikes from the noise, increasing the accuracy in environments of high noise levels. In contrast, in the case of lower noise, the benefit in the accuracy is not remarkable. On the other hand, the differences in the NEO operator for the tendencies with different widths windows, specifically in the case where a low operator order (4) could obtain higher accuracy than higher orders (16 or 32) in a noisy environment. Here, we recall that a same-order window could correspond to a higher window time in low than in high frequencies. The NEO is better at enhancing spikes

while there is a difference in amplitude between the spike sample and the adjacent samples; however, if the samples where the window is taken are not lower enough compared to the spike sample, it cannot properly enhance the amplitude difference. That could happen if the window is too large, and even more in noisy environments where high amplitude noise spikes could reduce the improvement of the NEO. That could explain why wider windows could have lower accuracies. However, we can also notice that the difference is minimal and relatively low compared to the absolute value, as, in general, for lower frequency samples the recovered accuracy is low. We can also conclude that such improvement can be signal-dependent and could not be generalized.

5.2.5.4 Variation of resolution and bandwidth

The last test we focused on was the comparison of accuracies considering both effects, the resolution and the bandwidth, trying to identify if a minimum resolution that allows recovering the information for spike detection, maintains its behavior while reducing the bandwidth. Figure 5.19, shows that effect, taking the same three upper limits we evaluated in previous tests and resolutions from 12 to 1. For all plots, the noise level was 5 µV, and the sampling frequency was 20 kHz.

In that picture, we observe the same tendency as in the evaluations of the individual variations, where the accuracy decreases for lower resolution and for lower bandwidth. However, we can notice something interesting, especially in the lower bandwidth upper limit, that it could require more resolution to achieve the maximum possible accuracy. That means that while for higher bandwidth (6 kHz and 3 kHz), that limit could be up to 6 bits of resolution, in the case of lower bandwidth (1 kHz) the limit could rise up to 7 bits, although the difference in accuracies is minimal. We can also find similar curves between uniform sampling and level-crossing sampling. However, we will appreciate that in the case of lower bandwidth. We will see that the detriment of accuracy starts from one bit higher in the case of level crossing (5 bits) compared to uniform sampling (4 bits). Related to the NEO operator, although the difference is not distinguishable in wider bandwidths when we lower the bandwidth, we can identify that a window of 4 achieve higher accuracies than higher orders of windows.

We can explain the results as both resolution and bandwidth allow recovering information from the recording; if we lower those parameters, we reduce the opportunities to detect spikes and be more accurate in that task. That explains why we will require more resolution when we lower the bandwidth, if we intend to obtain the maximum accuracy

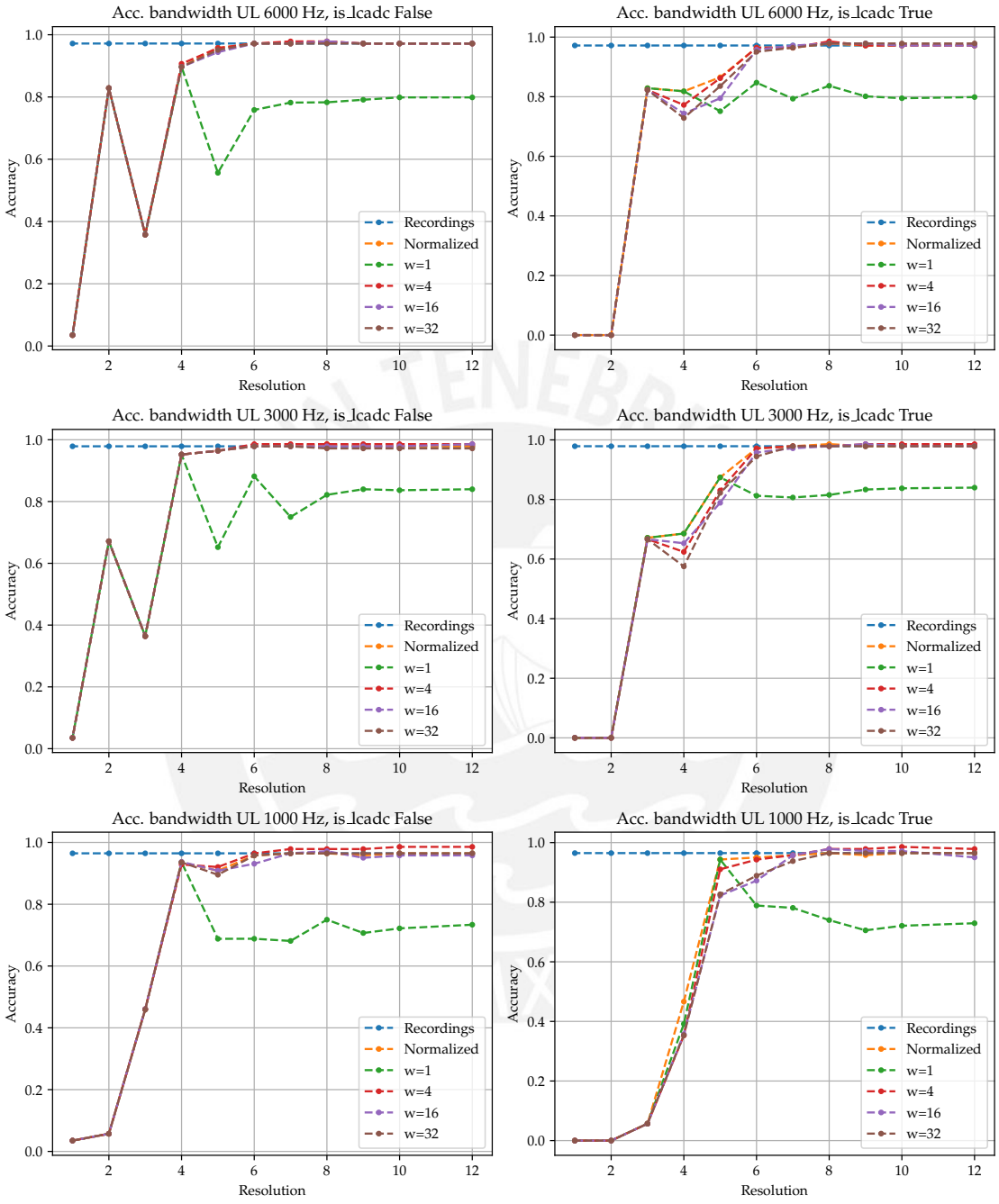


Figure 5.19: Comparison of maximum accuracy against resolution and bandwidth variation. Each row has plots with the same upper limit for the bandwidth, while the columns have the same sampling method: uniform (left) and level crossing (right). Different variants of the neural signal are also compared in each plot, including the NEO outputs.

possible with that bandwidth. On the other hand, although there is no remarkable difference in the case of NEO, the width of 4 gives higher and more consistent results when the bandwidth is lower. This can be explained as in the environment with lower frequencies, there is a low probability that near samples have rapid variations. Then the spikes could be more isolated and better captured with a lower window operator than a wider window. However, we can also consider that the difference is minimum, just in the case of level crossing sampling with resolutions between 5 and 7 bits.

These results suggest a trade-off between bandwidth and resolution; choosing one could impact the other. We can choose a 3 kHz limit for the bandwidth and use a 6-bit resolution or a 1 kHz limit and a 7-bit resolution, obtaining comparable results. Power savings should also be included in that balance. Therefore, if a bit could reduce the power dissipation, a lower bandwidth could also impact other components (amplifier, ADC, etc.), which could save more energy. Ultimately, we opt for the second alternative (1 kHz limit and use 7-bit resolution). At the same time, the NEO operator will have a window of 4 as it is the minimum required to obtain acceptable results and has been demonstrated to be enough to recover the information from the recordings. In the case of sampling type, there is not much difference; then, we can opt for any of the schemes. In this thesis, we will test the level-crossing structure.

5.3 Evaluation of SAR and LCADC circuit design

After we have modeled both sampling schemes via programming, we proceed to design both architectures and simulate them in the Cadence Virtuoso software, the standard tool used for the integrated circuit design.

For that, we will extract the performance metrics of both ADCs, considering both AC and DC characteristics. For the first, we use the obtained results from the design stage, which feeds the ADC with a sinusoidal input, and then measures the SNDR and the ENOB. In the case of the second group of characteristics, we apply a ramp input from which we extract the DNL and INL.

5.3.1 Metrics of SAR design

In Chapter IV, we have developed the design of three versions of the SAR ADC, two for single-ended input and one for differential input. However, we decided to use for this evaluation the differential one, as it is the most used in the industry.

In the test, we use a sinusoidal signal to measure the ENOB, and we have also extracted other dynamic parameters such as the SNDR (or Signal-to-noise-and-distortion ratio (SINAD)), the SNR called signal-to-non-harmonic ratio (SNHR) as well, and the spurious free dynamic range (SFDR) [68]. We explained the process of extracting them in the design chapter, specifically in section 4.2.6.2. To recall it, we have used a non-coherent sampling with 7 complete cycles and 128 samples, while the input frequency is 1.093 75 kHz. After that, we can generate the FFT and measure all the AC parameters using the Cadence tools.

We employed a ramp signal for the DC parameters, such as the INL and DNL, slowly enough to change a code in several samples. As the sampling period is 50 µs, and the number of codes is $2^8 = 256$, we choose a quantity of 5 samples per voltage level, which will give us a resolution of LSB/5, requiring a rise time of $256 * 50\,\mu s * 5 = 64\,ms$. We also employ the Cadence functions to measure the parameters, but, in this case, configuring a step size of $5 * 50\,\mu s = 250\,\mu s$. The output variations relative to this ideal width for each code correspond to the DNL, while the accumulation of those errors corresponds to the INL. Ideally, both values should be 0; however, due to the comparator's non-idealities, such as the delays and offsets, it introduces errors, as well as the resistances in the switches used in the DAC, which affects the linearity of the transfer curve.

The results for AC and DC parameters measured for the differential SAR design under typical process conditions, voltage (1.2 V) and temperature (27 °C), are presented in Table 5.1.

The first two parameters are related to design specifications, such as the 8-bit design requirement for the maximum amplitude resolution and the operation frequency for the conversion, which is set to 20 kHz. The next parameters are related to AC performance from the FFT analysis. We should consider that it uses a lower amplitude than the full scale (0.5 V instead of 0.6 V). While the SNDR and SNHR are similar, implying that the harmonic components are significantly low, the SFDR is higher, which implies that the maximum spurious component is almost 60dB behind the fundamental component.

The following parameters are related to power dissipation. As we can see, the most significant consumption comes from the logic. It makes sense as it is the component with more transitions during the conversion process in each clock cycle, with a frequency equivalent to $8 \times 20\,kHz = 160\,kHz$. The next component that consumes more current is the comparator (lower than 100 nA), which is dynamic and only has a relevant power consumption at the falling edge of the clock. That reduces significantly the component

Table 5.1: Summary of parameters of performance for the designed differential SAR. Includes the DC parameters and AC parameters, as well as the power consumption.

Parameter	**Value**	**Units**
Amplitude resolution	8	bits
Sampling frequency	20k	Hz
ENOB	7.691	bits
SINAD	48.06	dB
SFDR	58.44	dB
SNHR	48.06	dB
Average power total	604.8n	W
Average current total	504.n	A
Average current comparator	98.17n	A
Average current register	4.386n	A
Average current DAC	8.05n	A
Average current logic	349.4n	A
FOM_1 (fs/ENOB)	4.132M	Hz/conv-step
FOM_2 (power vs fs ENOB)	146.4f	J/conv-step
INL max	2.015	LSB
INL min	0	LSB
DNL max	538.6m	LSB
DNL min	-626m	LSB

dissipation. Other components, such as the DAC or the register, have marginal current consumption compared with the logic or the comparator. The total power is 604 nW which gives us an adequate reference to compare against the other design based on level-crossing architecture.

Concerning the DC parameters, we can see that DNL is approximately half of the LSB. Exploring the waveforms, we detect that it is because of a reduced output width for some codes around 700 mV. In the case of the INL, we see that it misses a code during a lapse of time until nearly the end, when it returns to the approximately 0 value.

On the other hand, we want to compare the results obtained against the performance of other ADCs from the literature. To do this, we use the figure of merit (FOM). In the summary table, we find two FOM_1 and FOM_2.

The first figure of merit has the following equation:

$$\mathrm{FOM}_1 = 2^{\mathrm{ENOB}} \times f_s \tag{5.5}$$

It aims to compare at how higher sampling frequency, the ADC can operate without degrading the resulting ENOB. In this case, the higher the number, the better FOM_1.

The second figure of merit has the equation:

$$\mathrm{FOM}_2 = \frac{\mathrm{P}_{\mathrm{dissipation}}}{2^{\mathrm{ENOB}} \times f_s} \tag{5.6}$$

In this case, the objective is to compare the power dissipation concerning the sampling frequency and the ENOB. While the SAR has lower dissipation with a higher sampling frequency or higher ENOB, it will obtain a better figure of merit, which means that a lower value means better performance.

In Table 5.2 we use the second FOM to compare the current results against other SAR ADCs that we used as reference.

Table 5.2: Comparison with general-purpose SAR ADCs. It compares against some references we used to design the current architecture.

Reference	Bonetti [60]	Nazzal, Mahmoud, and Shaker [69]	Li *et al.* [61]	This thesis
Year	2011	2016	2020	2022
Architecture	SAR	SAR	SAR	SAR
Resolution (bits)	10	8	12	8
ENOB (bits)	9.7	7.6	11.36	7.7
Sampling frequency (kS/s)	1280	10	2000	20
Voltage (V)	3.3	1	1.2	1.2
Technology (nm)	350	90	40	180
Power (uW)	140	0.2	356	0.6
FOM (fJ/conv-step)	131.5	103	67.8	144.27

As we can look at the table, we obtained a FOM a little higher than other designs. It is more similar to [60], also used for biomedical signals. However, we need to consider that it uses a higher voltage which will increase the power and, consequently, the FOM. It could have approximately the third part of its current value with a similar voltage to the other designs in the table. On the other hand, our test signal is lower than the full-scale range, which will reduce the ENOB and produces higher values for the FOM. Other designs, such as the [69], intended for biomedical applications, and the [61], a low-power SAR ADC, present lower FOM. We can advert that it could be due to the more actual technology, which can significantly reduce digital consumption, as also presented in [61], where the logic only represents 10%. In comparison, it is almost 70% of the total power dissipation in our case.

After we compared against general SAR designs, we aim to compare against ADCs for neural interfaces to identify how good this design can be considered as a reference. For this, we elaborated the table in Table 5.3.

Table 5.3: Comparison with other ADCs for neural signal interfaces. Here, we extracted the parameters from the publications and in the case of needed, we calculate the FOM with the aim to compare against the current SAR design in this thesis.

Reference	Ranjandish and Schmid [70]	Liu *et al.* [71]	Pazhouhandeh *et al.* [72]	Kakaraparty, Tasneem, and Mahbub [73]	Shui *et al.* [74]	De Dorigo *et al.* [9]	Mora Lopez *et al.* [75]	Shahrokhi *et al.* [76]	Kim *et al.* [77]	Xu *et al.* [78]	This thesis
Year	2021	2021	2021	2020	2019	2018	2017	2010	2018	2020	2022
Architecture	SAR/shared	SAR	Delta-ADC	Compressive sensing SAR	Sigma-Delta	OTA-C Sigma-Delta	SAR	SAR	SAR	Sigma-Delta	SAR
Resolution[1]	10	9	9.7	8	12	8.2	10	6	10.7	14	7.7
Sampling frequency (kS/s)	32	Oversampling 1M	10k (oversampling 10MHz)	10-40KHz	10	20	AP(30kS/s) LPF(2.5 kS/s) x 13 channels	14	32	80	20
Voltage (V)	0.8	1.8	0.6/1.2/3.3	1.8	1.8	1.8	1.2/1.8	3.3	0.8	1	1.2
Technology	180nm	180nm	130nm	180nm	180nm	180nm	130nm	350nm	65nm	130nm	180nm
Power (uW)	1.26	-	0.99	623.55uW/32	26.5	21.04	11.6	2.7	0.8	3.5	0.6
FOM (fJ/conv-step)	38.45	13000.00	119.03	1902.92	646.97	3577.42[2]	108.4	3013.39	15.03	2.67	144.27

[1] Values with decimals are the ENOB reported in their respective article, while the others are the nominal resolution, as the ENOB was not available.
[2] This value is half of the reported FOM for the standalone ADC in the article.

We can notice from this table that most of the developments (even in recent years) use the technology of 180 nm, equal to the employed in this thesis. On the other hand, in most cases, the voltages for the ADCs are higher than 1 V. Also, the architecture most used is the SAR, while the number of bits is commonly higher than 10 bits. At the same time, the sampling frequency is usually lower (except in the cases that use an oversampling technique to obtain higher frequencies).

If we compare the current development in this document, we can find that it has a lower FOM than many of them. It is comparable with [72], although it uses another architecture to reduce power dissipation. It is also comparable with [75]. Other developments that have lower FOM are SAR and Sigma Delta. However, they employ an operating frequency higher than in our case and with a lower voltage, decreasing its value. Also, those developments ([77], [78], or [70]) use technologies with a smaller transistor size, which could help them to reduce the power in the digital side of the architecture.

In general, compared with the other designs presented in the literature, we can consider that ours is competitive with them and can be used as a reference to compare against the other architecture we aim to evaluate.

5.3.2 Metrics of LCADC design

In the chapter of design, in section 4.3.6, we have developed an LCADC with some specifications that will achieve equivalent performance to the obtained from the analysis with the software models, which has an effective resolution of about 7 bits. Although, in the SAR case, we developed an 8-bit as the effective number will be lowered, in the case of the level-crossing, the amplitude resolution is lower as can capture more samples and can grab more information than a fixed sampling ratio such as the SAR. As a reference, previous developments for this architecture, such as the one presented by Li, Zhao, and Serdijn [63], used an amplitude resolution of 6 bits, the same we will use here.

At the end of the design section, we discussed how we chose the simpler version with a single-stage OPAMP for the comparator and an additional logic that could fix wrong codes around the middle level of the window (VM). We also introduced a comparator to fix the accumulative offset that will reset the ADC when the input crosses the VM voltage. In addition to these changes, we added a current bias for the comparators to control their current consumption. With that in mind, we used a base bias of 50 nA, mirrored in the different comparators with a factor of 5, to have a total current bias of 250 nA in their respective differential stages. This strategy helped us significantly reduce

the current consumed by design, as well as provides a more reliable way to polarize the comparators (due to it is more robust to variations in the power supply).

We cannot use the Cadence tools to extract the parameters for the dynamic performance as there is no fixed sampling frequency. For that, we need to apply processing to the samples. This processing includes exporting the waveform and extracting the samples at which the change occurs (level crossing). Then, we use an interpolation algorithm and, finally, a decimation to get a fixed sampled waveform from which we can obtain the FFT, and from it, all the dynamic parameters also considered for the SAR ADC. Something important to notice here is the interpolator we used. For this experiment, we initially made the process in Python; however, we had to move on to MatLab as it had a more variety of algorithms to test. After probing the linear, spline, and Makima, we used the last algorithm as it returned better results, reducing some glitches due to the reset process when the input crosses the VM level. After the interpolation, we oversample the waveform in a factor of $M = 256$, taking as the base frequency 20 kHz. Then, we decimate the resulting signal in a factor of $N = 64$, which returns a final sampling frequency of 80 kHz. That will require 512 samples instead of 128, used in the SAR analysis. For the FFT, we applied a rectangular window to the sampled decimated signal, equivalent to the procedure performed by Cadence to calculate the metrics. With that operation, we obtained the metrics in Table 5.4. One detail to highlight here is that using a different window (kaiser) will return an increased SNDR value (39 dB) with a higher ENOB (6.3). However, as the method used in the SAR employs the rectangular one, we opt to follow the same to make a fair comparison.

In the case of the bandwidth, we performed several simulations with sinusoidal inputs of different frequencies until the output was reduced to 70%. In the test, we found that after 12.5 kHz, the output amplitude is dramatically low because the comparator cannot follow the input and convert it correctly. Then, we consider this value as the bandwidth for the ADC. In contrast, this procedure cannot be performed in the SAR as the amplitude samples are not continuous, and measuring the difference in amplitude cannot be reliable.

Concerning the static performance, as expected, all the values are almost 0. That is because the output cannot have missed codes if the input is slow enough. For this test, we used a 1 ms ramp signal, which is faster than the one used in the SAR experiment, which is 64 ms rise time.

We identified the comparators as the most power-hungry components in the power dissipation analysis. This agrees with their nature as they consume static power, being

Table 5.4: Summary of parameters of performance for the LCADC. As we do not have a sampling frequency, we included the bandwidth from which we calculated the FOM.

Parameter	**Value**	**Units**
Amplitude resolution	6	bits
ENOB	5.927	bits
SINAD	37.4	dB
SFDR	42.6	dB
SNHR	40.5	dB
Bandwidth	12.5k	Hz
Average total power	1.062u	W
Average total current	884.9n	A
Average current comparator CC	191.4n	A
Average current comparator CR	250.9n	A
Average current comparator reset	248.n	A
Average current DAC	960.1p	A
Average current multiplexer	24.21p	A
Average current logic DAC	56.11n	A
Average current logic RLC	23.6n	A
Average current logic RLC reset	16.79n	A
Average current counter	27.87n	A
FOM	711.39f	J/conv-step
DNL max	5.329f	LSB
DNL min	-9.437f	LSB
INL max	222a	LSB
INL min	-9.548f	LSB

active during the conversion process. The other components have minimum consumption, including the logic blocks and even the DAC. That is the main saving compared with the SAR, whose digital logic represents the most significant consumption.

With the power consumption information, we can calculate the figure of merit. For the FOM calculus, we employed a modified version of the used in the SAR case, which replaces the sampling frequency with an equivalent value based on the bandwidth:

$$\mathrm{FOM}_2 = \frac{\mathrm{P}_{\mathrm{dissipation}}}{2^{\mathrm{ENOB}} \times 2 \times \mathrm{ERBW}} \tag{5.7}$$

Finally, to evaluate how these results are compared to the literature, we build Table 5.5. As we can appreciate from the table, we calculated two types of FOM from the developments, depending on the reported parameters. In the cases of the ADCs that use a kind of sampling frequency, we employed the same FOM applied to the SAR architectures. On the other hand, we use the FOM calculated with the bandwidth data, generally, for those that do not have a clock with a fixed sampling frequency. In the table, we opted to separate the FOM as they take different parameters to calculate them and cannot be interchangeable. Therefore, although differences, we will just use the values obtained as a reference to compare the developments based on the LCADC architecture.

Table 5.5: Comparison with other LCADCs developments. Depending on the data available we calculated the FOMs, considering both the sampling frequency and the bandwidth, according to strategy used for the sampling.

Reference	Li, Zhao, and Serdijn [63]		Schell and Tsividis [79]	Li *et al.* [64]	Hou *et al.* [80]	Joshi, Deshmukh, and Patrikar [81]	Weltin-Wu and Tsividis [82]	Das *et al.* [83]	This thesis
Year	2013	2013	2008	2011	2019	2017	2013	2019	2022
Architecture	LCADC	LCADC	LCADC	LCADC	LCADC	LCADC	LCADC	LCADC	LCADC
Resolution (bits)	6	6	8	8	5.6	7	8	8	6
Sampling frequency (kS/s)	-	-	-	51.2	2	14	40	1	-
SNDR (dB)	40	49	47	-	-	-	-	-	37
ENOB (bits)	6.352	7.847	7.515	-	-	-	-	-	5.9
BW (kHz)	3.3	5.1	4	-	-	-	-	-	12.5
Input Signal Freq (kHz)	1.1	1.1	4	10.5	1	1	4	0.125	1.09
Voltage (V)	0.8	0.8	1	0.5	0.5	0.8	0.8	1.8	1.2
Technology	180nm	180nm	90nm	90nm	180nm	180nm	130nm	180nm	180nm
Power consumption (uW)	0.313	0.582	40	0.729	0.22	0.76	6.5	1.895	1.062
FOM (BW)	580.51	247.79	27336.61	-	-	-	-	-	711.39
FOM (fJ/conv-step)	-	-	-	55.62	2267.90	424.11	634.77	7402.34	-

From the data obtained, we can see that our development is competitive against some reported works; however, relative to our main reference in [63], it has a higher FOM, which implies a lower performance. Despite that, we should consider that the FOM uses power dissipation as a parameter whose value includes the power supply voltage. In the case of [63], the power supply is 0.8 V, while in our case is 1.2 V. Consequently, if we scale the FOM values in our case to the voltage that they use, the value will be 474 fJ/conv-step, which is comparable to the first LCADC reported in their article. Another difference we can notice for our design is that in the reference, the ENOB is always higher than the amplitude resolution; however, in our development and others reported, such as the [79], the ENOB has a lower value. That increase significantly reduces their FOM, achieving lower values than our case. On the other hand, we can appreciate that our design shares the same 180 nm technology with other developments, which allows a fairer comparison.

From the table, we can conclude that, although not the best in performance, our circuit is comparable with some designs reported previously. Likewise, if we attempt to compare against our SAR design, we can identify that the power dissipation is higher than it. However, we are considering a general sinusoidal input that has several changes and for which it doesn't take advantage of only obtaining samples when it crosses the amplitude levels, which is the main characteristic of the LCADC.

5.3.3 Evaluation with neural signals

In this section, we apply a generated neural signal to the converters with the same characteristics as those we employed in the NEO evaluation in the first section of the results. The channel selected is the same as the one used in the NEO evaluation (channel 27), with a noise level of 5 uV. In both cases, we use a recording of 10s long.

5.3.3.1 Evaluation of SAR ADC

First, we will probe the SAR ADC with the neural test signal. Due to its length, we split it into chunks to speed up the simulations, and then, we consolidated the results by post-processing in a single continuous signal. We take the samples for the resulting waveform each 50 us, corresponding to the sampling frequency of 20 kHz. The signal is also amplified in the simulation with a gain of 1000, as originally, it has amplitudes of microvolts. In our case, we want a scale of millivolts to take advantage of its dynamic range.

The resulting signal may not be appreciated as has multiple spikes close together. Then, in Figure 5.20 we plot only an interval where the spikes are more visible to appreciate the effect of the conversion.

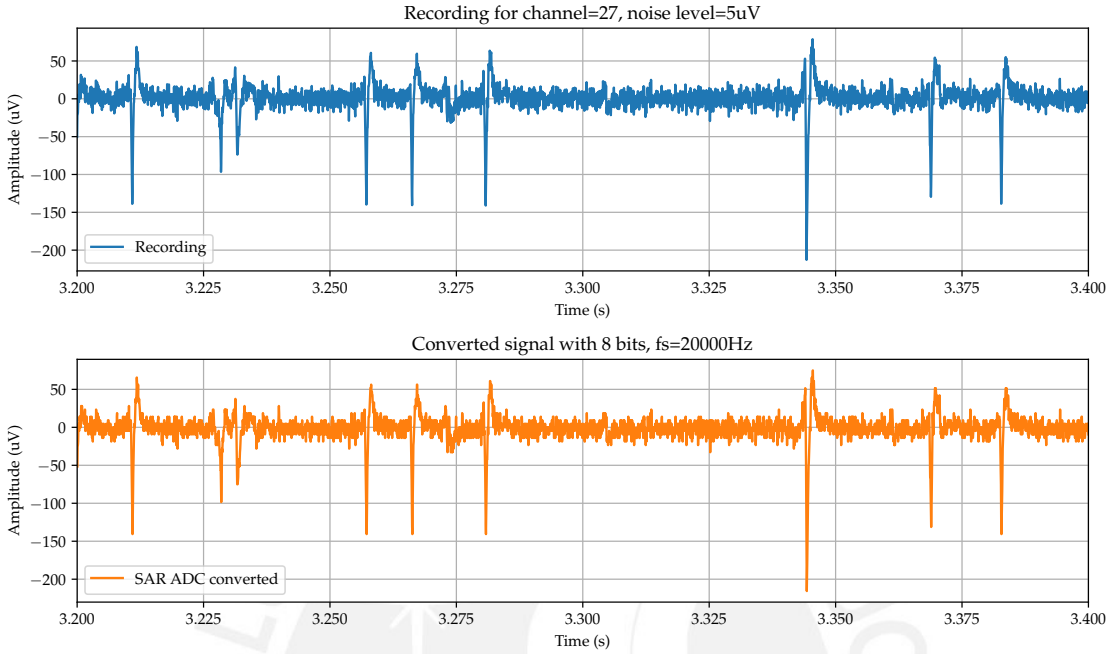


Figure 5.20: Comparison along the time between the original signal vs. the converted one. Both signals are from the time 3.2 s to 3.4 s. Here, we can appreciate several significant amplitude spikes that are converted with the SAR.

In this figure, we can observe that the magnitude is almost identical to the original signal, which means that the resolution employed could reproduce almost exactly the spikes obtained from the recording.

However, the comparison along the time may not be enough to verify that the converted signal recovers the information from the analog one. For that reason, we also obtain the power spectral density of the original signal and compare it against the converted one. This is presented in Figure 5.21.

From this figure, we can appreciate that the spectral density of the original recording is almost identical to the converted signal with the SAR ADC. Also, we can notice that even the noise level is reproduced in the converted one, which implies that the effect of noise in the acquired signal will be equivalent to the original recording.

Regarding power consumption, we can see the distribution for each component in Table 5.6. Here, we appreciate that the most significant power dissipation comes from the digital logic, while the register and the DAC represent the minimum. We also see that the comparator does not represent an important part of the whole dissipation of the full

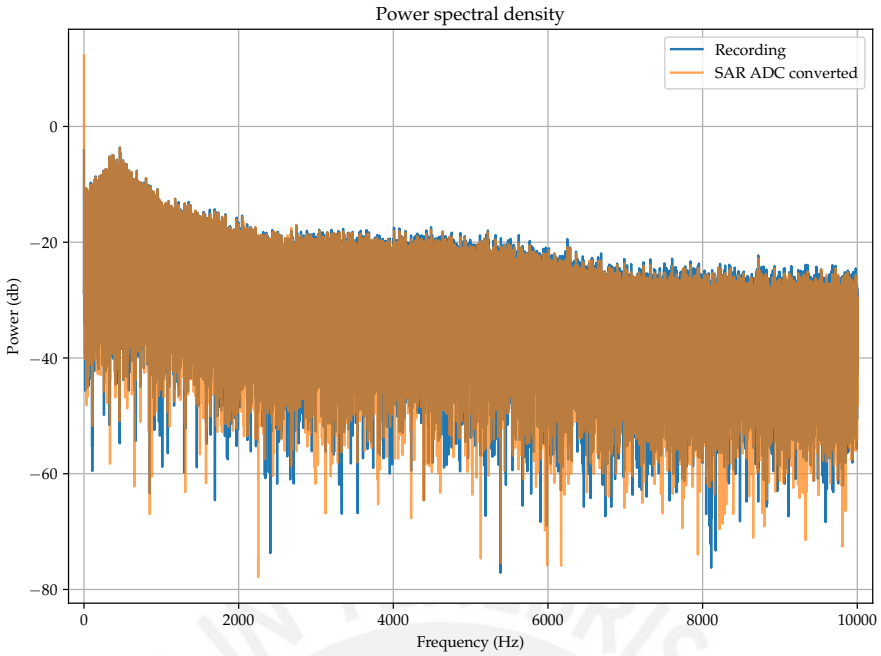


Figure 5.21: Comparison of the power spectral density between original signal vs. converted ADC signal. We build this plot from the FFT of both signals, for which we apply the square power to obtain the power density distribution.

circuit. In addition, the result is even lower than the registered when we use a sinusoidal input, such as in Table 5.1.

Table 5.6: Power consumption for SAR ADC conversion separated by each block component in its design simulating with neural signals.

Output	Nominal	Units
Average current comparator	61.741n	A
Average current DAC	8.0783n	A
Average current logic	350.15n	A
Average current register	4.1345n	A
Average current total	468.24n	A
Average power total	560.69n	W

5.3.3.2 Evaluation of LCADC

In this section, we evaluate the LCADC design with the neural signal used previously with the SAR. As we can recall from the metrics measurement for this circuit, it has an amplitude resolution of 6 bits. Also, it doesn't have a fixed sampling frequency, as the samples are taken each time the signal crosses a level corresponding to a specific digital code.

The experiment also required splitting the signal into chunks which were applied to the converter. Then, the converted signals are joined back together, choosing as samples the instants when the amplitudes change level via a post-processing script.

The resulting signal is also 10 s long; however, similar to the SAR case, we cannot appreciate the difference if we consider all its length; then, we opted to compare just in the interval where we know spikes are present in the neural recording. That result is presented in Figure 5.22.

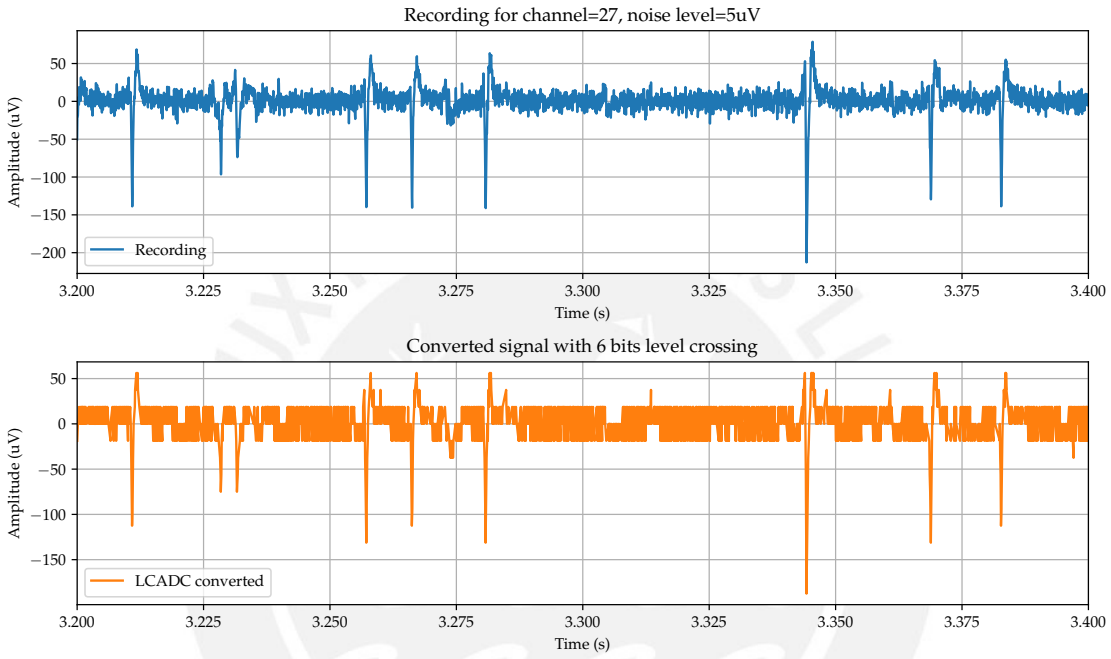


Figure 5.22: Comparison along an interval of time between original recording vs. the converted LCADC signal. We used an interval from 3.2 s to 3.4 s, where we identified spikes with significant amplitude.

From the figure, first, we can appreciate that the amplitude is scaled down in the converted signal due to the gain of the LCADC converter that is lower than 1 (from the original factor obtained in the design step, it is approximately $14/15 = 0.93$). On the other hand, we appreciate that the lower resolution may impact the precision of the spikes' amplitude, as minimal differences can be amplified if it is nearly the threshold for a new reference level. That is why we can appreciate two spikes that may appear similar in the original signal, with different amplitudes in the converted one. Another aspect we need to consider is that the step size for the converted signal is 18.75 mV which makes the amplitude differences appear higher between samples.

Similar to the SAR evaluation, we must also compare the spectral density of the original signal compared to the LCADC-converted one. For this purpose, we apply interpolation

to the nonuniform samples in a factor of 128 and then decimate the resulting signal using two decimators of 16 and 8 to return to the original frequency of 20 kHz. After that process, we apply the FFT to the output signal and compare it against the original. We can appreciate the result in Figure 5.23.

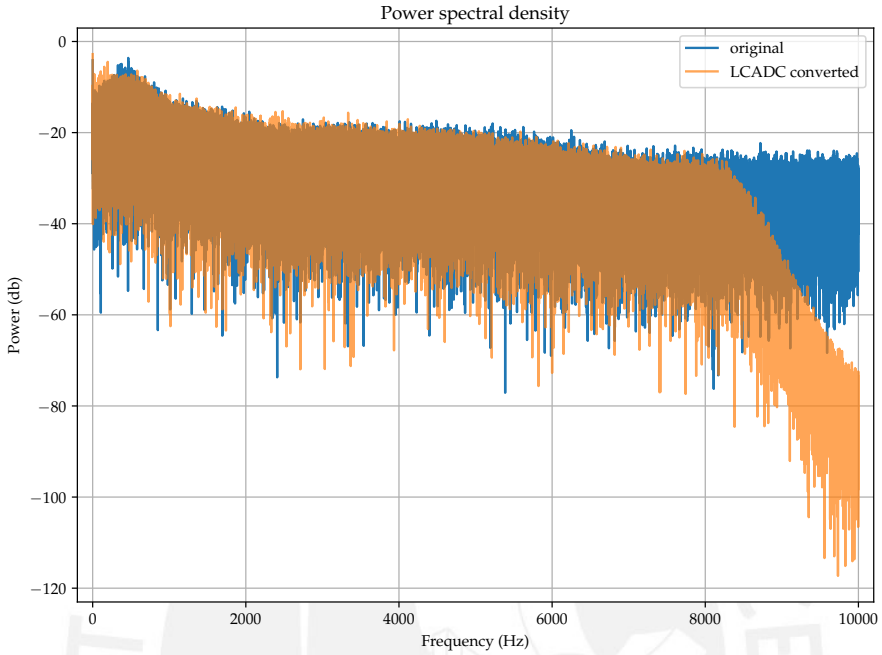


Figure 5.23: Comparison of power spectral density between original signal vs. converted LCADC signal after post-processing that includes interpolation and decimation to a final frequency of 20 kHz.

From here, we can observe a more significant difference relative to the original spectral distribution, principally at higher frequencies. This occurs due to the included filter in the decimator, which acts as a low-pass filter that suppresses frequencies beyond 8 kHz. On the other hand, lower frequencies are more likely the original spectrum with slight differences in some portions, which may be explained as the interpolation effects.

Finally, we can also compare the power dissipation produced by each block in the LCADC architecture when it converts a neural signal. Those values are presented in Table 5.7. Here, we can appreciate that the most representative consumption comes from the comparators. There is also a special relevance of the comparators used for the recurrent code as well as for the reset feature. This increased power consumption in those components comes from a signal oscillating more around the DC value. That is due to the noise in the signal, which is within the scale of one converter step of amplitude. Other blocks, such as the logic circuits, have meaningless dissipation, except for the logic used

in the recurrent code, which consumes more than the others. That is also related to the increased power dissipation in the comparators attached to them.

Table 5.7: Power consumption for LCADC conversion, distributed by each component that constitutes its architecture.

Output	Nominal	Units
Average current comparator CC	164.44n	A
Average current comparator CR	257.3n	A
Average current comparator reset	252.56n	A
Average current counter	4.5026n	A
Average current DAC	202.31p	A
Average current logic DAC	696.76p	A
Average current logic RLC	73.4n	A
Average current logic RLC reset	51.721n	A
Average current multiplexer	34.328p	A
Average total current	810.47n	A
Average total power	972.54n	W

In the end, we can appreciate that the power dissipation is about 972 nW, higher than the presented by the SAR with 560 nW. From that, we can conclude that the LCADC can hardly obtain lower consumption than the SAR, even in cases of neural signals, principally due to the static comparators that are the most power-hungry components in the LCADC architecture. Then, the biggest advantage we can obtain from the LCADC is not the power dissipation, it is the number of samples.

If we compare the fixed width sampling of the SAR, it has a total of 200000 samples for 10 s at 20 kHz, while in the case of LCADC it has only 115567 samples which is almost half of the previous value. Although it may not appear meaningful, we should consider that by applying a kind of digital processing, the power dissipated will be proportional to the number of samples, increasing for the SAR case.

In the next section, we will evaluate it, considering a digital block for post-processing, the NEO operator, also designed in Cadence.

5.4 Evaluation of power consumption

To evaluate the power consumption, we must simulate the ADC and the NEO circuit together. This simulation is applied to both ADC architectures: SAR and the LCADC.

5.4.1 Evaluation of ADC with NEO

We need to consider that the NEO block is a clocked digital circuit; then, it will require a clock signal to execute the arithmetic operation. In the case of the SAR, this clock signal comes from the EOC pulse, which indicates that the conversion has finished. Then, the resulting sampling frequency is 20 kHz. In the case of the LCADC, we use the change code signal, which indicates the change in the value in its internal counter. As it is an asynchronous circuit, it doesn't have a fixed frequency we can estimate as the output.

We added a delay to interface the clock signal used for the blocks in the respective NEO circuits. This delay will ensure that after the new value is loaded in the ADC output, it has enough time to calculate in the multipliers before shifting the value inside the NEO. This additional delay helps the inner one added to the NEO architecture presented in Figure 4.54. Otherwise, the corresponding NEO output will be loaded on the next positive edge clock. In the SAR, this just means a delayed NEO output that can be fixed by delaying the SAR results in the processing stage. However, in the LCADC case, this approach cannot be applied because the times between samples are not equal to those in the fixed sampling scheme. Consequently, the delayed result will significantly change the NEO waveform. Therefore, we prefer adding the delayed clock that helps obtain the correct NEO result for each ADC sample.

We use a recording signal for this test, also employed in the NEO evaluation. Due to the system's complexity, the hardware simulation will take a long time. Therefore, we select a chunk of the whole signal. We continue using channel 27 but only a 1s chunk of the full 10s signal length. As our evaluation target is spike train 8, for which we detected higher accuracy values in this channel, we select the chunk with more spikes. The chunk selected was the fifth, with 20 spikes between 4.9 s and 5.9 s.

After we perform the simulation with input signals of 300 Hz - 6 kHz, 300 Hz - 1 kHz of bandwidth, and 5 µV, 25 µV of noise level, we obtain the results presented in Table 5.8.

We can see from the table two values for NEO. One corresponds to the block applied directly to the ADC (either SAR or LCADC) outputs, and the other to the block applied to the absolute values of the ADC outputs. That is to evaluate how the kind of samples could affect power consumption. For both sampling schemes, the NEO applied to direct samples consumes more power than the applied to absolute ones.

To analyze the results, we will concentrate first on low-noise signals. The increase of the NEO with direct samples is 12%-7% and 38%-48% for the LCADC and SAR samples,

Table 5.8: Power consumption results for ADC and NEO with simulated intracortical signals.

Test input signal		**Component**	**SAR**		**LCADC**	
BW (Hz)	Noise level (µV)		Current (nA)	Power (nW)	Current (nA)	Power (nW)
300-6000	5	ADC	479.5	575.4	805.8	967
		NEO	156.6	187.9	60.06	72.08
		NEO(abs)[1]	113.1	135.7	53.85	64.62
		Total: ADC+NEO	636.1	763.3	865.86	1039.08
		Total: ADC+NEO(abs)	592.6	711.1	859.65	1031.62
300-1000	5	ADC	473.2	567.9	812.3	974.8
		NEO	169.0	202.9	56.1	67.32
		NEO (abs)	114.0	136.8	52.24	62.68
		Total: ADC+NEO	642.2	770.7	868.4	1042.12
		Total: ADC+NEO(abs)	587.2	704.7	864.54	1037.48
300-6000	25	ADC	501.1	601.2	880.4	1056
		NEO	190.3	228.4	240.8	288.9
		NEO (abs)	122.9	147.5	170.1	204.1
		Total: ADC+NEO	691.4	829.6	1121.2	1344.9
		Total: ADC+NEO(abs)	624.0	748.8	1050.5	1260.1
300-1000	25	ADC	497.3	596.8	880.8	1057
		NEO	177.5	213.1	181.7	218
		NEO (abs)	122.8	147.3	128	153.6
		Total: ADC+NEO	674.8	809.9	1062.5	1275
		Total: ADC+NEO(abs)	620.1	744.1	1008.8	1210.6

[1] NEO applied to the absolute values from the ADC output.

with upper limit bandwidths of 6 kHz and 1 kHz, respectively. In this case, the benefit of using absolute values is greater in the fixed-rate scheme and with low bandwidth.

However, compared to the ADC consumption, NEO represents 32%-35% of the SAR consumption and 7.4%-7% of the LCADC consumption for 6 kHz and 1 kHz, respectively. It means that while the SAR reduces the consumption in its architecture, the sampling scheme increases the following digital blocks (in this case, the preprocessor). On the other hand, the LCADC has more power consumption than the SAR but reduces the power consumption of the digital blocks applied to its samples. This is confirmed by comparing the consumption between NEO for the SAR sampling scheme (187.9 nW-202.9 nW) with the LCADC sampling scheme (72 nW-67.32 nW), resulting in a relation of 2.6 and 3 for 6 kHz and 1 kHz, being more beneficial for low bandwidth. This means that digital blocks applied to the fixed sampling rate will consume up to three times the power of those applied to level-crossing samples in intracortical signals.

Analyzing the results for high noise signals, the metrics for SAR are almost identical; however, as the LCADC depends on the quantity of samples, its power increased, especially with a high bandwidth (6 kHz), where the consumption of NEO overpasses its counterpart with the SAR scheme, being less efficient. Something we can highlight here is that using a low bandwidth (1 kHz) is more beneficial when there is a high noise level, particularly to the LCADC scheme. Firstly, because of the reduction of NEO power when applied to absolute values in about 41%, relative to the NEO applied to direct samples. While, the power dissipated by this component is more similar to SAR scheme, increasing only in 2% while for 6 kHz is 26%.

Although results with high noise conditions (25 µV) are not promising, we must consider that this scenario is pessimistic as the RMS of the whole signal (including noise) is equal to 42.3 µV) which means that the RMS value for the isolated signal is comparable with the noise. We included this test because we wanted to show the limits to which the LCADC could reduce the power.

In addition, the table shows that the total power including ADC and NEO is still higher for the LCADC compared with the SAR, even for low noise signals. However, the power saving of the level-crossing scheme is evidenced in the preprocessor consumption. As we can see, it is considerably low for low noise conditions (that guarantees sparse samples), which is also desired to not significantly impact to the full system power. Although it is not enough to balance the intrinsic consumption of LCADC, it gives a clue about the potential power reduction in other components that depend on the sampling rate, such as the wireless transmission. Therefore, the major benefit of the level-crossing scheme is reducing the number of samples with a consequent reduced effective sampling rate, which can compensate for the increase in the LCADC block.

Once the simulation with the implemented circuits calculates the ADC and preprocessor power consumption, we will continue with other blocks that comprise the acquisition chain.

5.4.2 Estimation of analog frontend power consumption

For the analog frontend, there are some approximations we can use to estimate the power dissipation. One of them is the one presented by Even-Chen *et al.* [6], which depends on the power efficiency factor (PEF), the noise bandwidth (Δf), and the RMS noise along

the selected band ($\bar{v}_{in}$). It could be written as:

$$P_A = \text{PEF}\frac{\Delta f \pi U_T 2kT}{\bar{v}_{in}^2}; \qquad U_T = \frac{kT}{q} \tag{5.8}$$

Where $k = 1.38 \times 10^{-23}\,\text{J/K}$ is the Boltzmann constant, $q = 1.6 \times 10^{-19}\,\text{C}$ is the electron charge, and T is the absolute temperature in Kelvin.

For ambient temperature (27 °C), it will be equal to:

$$P_A = \text{PEF}\frac{\Delta f}{\bar{v}_{in}^2}(6.73 \times 10^{-22}) \tag{5.9}$$

We can consider a target of $4\,\mu\text{V}_{\text{rms}}$ for the noise level alongside a band between 300 Hz to 1 kHz. Even-Chen *et al.* consider a PEF = 1.12 V. Then, the approximated power dissipated will be:

$$P_A = 1.12\frac{700}{(4 \times 10^{-6})^2}(6.73 \times 10^{-22}) = 32.97\,\text{nW} \tag{5.10}$$

However, this value is ideal and doesn't consider the additional bias circuits that make the amplifier work, as well as it assumes a linear dependence between the noise band and the power with a noise uniformly distributed alongside the frequencies, which is not necessarily true. That is why we prefer designing a model of analog frontend to have a more precise approximation for the circuit power dissipation.

We based that design on the one proposed by Ranjandish and Schmid [70] for their analog frontend. It comprises three stages: a low-noise amplifier (LNA), a Gm-C filter, and a second-stage amplifier. Although they suggested some values for the design, we adapted them to obtain an LNA with $4\,\mu\text{V}_{\text{rms}}$ for the band between 300 Hz to 1 kHz bandwidth and a filter with 1 kHz cut frequency.

By doing this design, we identified how difficult it is to reduce power consumption while maintaining low noise. The LNA uses the folded cascode topology, which is useful for building a fully differential structure with higher stability. It also includes a self-bias generation to obtain the voltages required and a circuit for the common mode feedback (CMFB) [84]. This circuit must be present in the fully differential architecture to ensure the output will be around a selected common voltage. Approximations like the one presented in (5.8) don't consider these details; however, they are mandatory for the correct circuit operation.

On the other hand, regarding the noise, it has two components: the flicker noise and the thermal noise. The flicker noise (1/f) is the dominant component in low frequencies. It is challenging to reduce as it depends on the sizes and the technology's noise factor.

Table 5.9: Parameters for three-stage analog frontend under typical conditions, adapted from the original design proposed by Ranjandish and Schmid [70].

Component	**Parameter**	**Value**	**Unit**
First stage (low-noise) amplifier	Current Consumption	2.8	uA
	Power Consumption	3.36	uW
	Offset	2	mV
	Phase Margin	111.192	degrees
	Gain Margin	75.46	dB
	Unity-gain Frequency	21.18	KHz
	Integral noise 300-6kHz	8.14	uV
	Integral noise 300-3kHz	6.36	uV
	Integral noise 300-1kHz	4.07	uV
Gm-C Filter	Current Consumption	0.217	uA
	Power Consumption	0.261	uW
	Cut frequency (-3dB)	1.12	KHz
	Gain DC	6.87	dB
	Integral noise 300-6kHz	34.97	uV
	Integral noise 300-3kHz	24.29	uV
	Integral noise 300-1kHz	12.66	uV
Second stage amplifier	Current Consumption	0.643	uA
	Power Consumption	0.772	uW
	Offset	8.24	mV
	Phase Margin	93.18	degrees
	Gain Margin	66.78	dB
	Unity-gain Frequency	22.53	KHz
	Integral noise 300-6kHz	37.82	uV
	Integral noise 300-3kHz	31.59	uV
	Integral noise 300-1kHz	21.69	uV
Total consumption	Current Consumption	3.656	uA
	Power Consumption	4.393	uW

The thermal noise depends on the transconductance, and it directly relates to the circuit's operational current [85].

Due to the low-noise requirements, the first stage cannot reduce their power considering the other operational currents that do not participate directly in the transconductance but are required to make it work. However, reducing the target band helps to reduce the final current, as shown in Table 5.9. For the band of interest, the noise is 4 µV, but for wider bands, the noise is higher. Consequently, the circuit will demand more current to reduce to the same value of 4 µV. In the case of the next circuits, they have relaxed noise value specifications beacuse their value is divided by the gain of the first stage which is around, 180. That condition helps to concentrate the design on other characteristics, such as the cut frequency for the filter (with minimal transconductance and current consumption) and the gain for the second stage, to get output values in the order of hundreds of millivolts required by the ADC.

In conclusion, only considering the noise value for the power estimation will return a limited approximation. The acquisition requires other functions, such as filtering and amplification, to obtain useful signals for digital conversion, which will increase power consumption. In this case, we will consider the values obtained from our circuit designs to estimate the full power dissipation. In addition, the value obtained is similar to the value in other development [86] (although with lower noise, but with a single-ended simplified topology), and it is in the range expected according to literature according to [6] from 0.5 µW to 10 µW.

5.4.3 Estimation of wireless transmission power consumption

The last component in the BCI is the wireless transmission. We will not design this circuit because of its complexity. That will force us to use an estimation for the power dissipation. Even-Chen *et al.* [6] use an approximation that employs the transmitter efficiency representing the amount of power per bit E_b. Their calculus employs a value of 8.5 pJ for this parameter. It comes from a highly efficient ultra-wideband (UWB) transmitter for neurorecording systems, presented by Miranda and Meng [87]. However, we also need to consider the static power (P_s) required by this transmitter which is 13 µW for 32 neural channels. Finally, the transmitter power per channel will be expressed as:

$$P_{\mathrm{TX}} = E_b R + \frac{P_s}{\mathrm{N\ channels}} \tag{5.11}$$

Where R is the transmission bit data rate.

We need to obtain the total number of bits and the rate at which they are transmitted to calculate this data rate. From the results of NEO application to LCADC samples, for the reduced signal length, we get the characteristics shown in Table 5.10

In this case, we have improved the output for the LCADC to avoid creating unnecessary samples when the value is around VCM voltage, by adding a register to the output which holds the sample for a certain time before changing to the next. That is why we reduced the previous number of samples from the LCADC results.

Regarding the results in Table 5.10, we can observe two differentiated scenarios. One for low noise and another for high noise signal. In the first case, the worst number of samples is 3508, which is considerably low compared with a fixed sampling scheme with a total of 20000 samples for 1 s signal length. It can even be reduced considering NEO output and low bandwidth (1645 samples), which represents almost 10% of the number with the SAR. In the other case, with high noise signals, the behavior changes, increasing

Table 5.10: Sampling results from the LCADC and NEO application to a neural signal. The statistical measurements for count, mean, standard deviation, min, median, and max are calculated for the differential times between samples.

BW (Hz)	**Noise** (µV)	**Result type**	**Count**	**Mean** (µs)	**Std dev** (µs)	**Min** (µs)	**Median** (µs)	**Max** (ms)
300-6000	5	ADC out	3508	285	761	0.125	102	10.828
		NEO	2485	403	922	0.14	120	11.315
		NEO(abs)	2269	441	982	0.14	121	11.315
300-1000	5	ADC out	2413	416	943	0.126	105	12.048
		NEO	1705	589	1163	0.159	134	12.141
		NEO(abs)	1645	610	1192	0.159	134	12.141
300-6000	25	ADC out	34289	29	26	0.493	21	0.470
		NEO	28087	36	37	0.493	23	0.655
		NEO(abs)	27079	37	45	0.493	23	0.780
300-1000	25	ADC out	24468	41	36	0.254	29	0.436
		NEO	18607	54	59	0.254	33	0.816
		NEO(abs)	14416	70	106	0.254	34	1.653

the quantity of samples by a factor of around 10. This increase is worst with a high bandwidth 6 kHz overpassing in more than 70% of the reference number of SAR. It is only comparable with low bandwidth 1 kHz being even lower in the best case, considering NEO outputs applied to absolute values. This is useful to identify the benefits of including the NEO block, especially for LCADC scheme, which helps reduce the number of samples even in high-noise environments.

On the other hand, relative to the differential times, we see that the standard deviation in most cases is higher than the mean, which would not return useful values to estimate a representative measurement for the differential times. It also happens with the mean value, as not uniformly distributed values constitute the differential times; several long differences will overestimate the mean measurement, although the number of short differences was higher. That is why we opt to use the median value for our representative difference.

In the case of low noise signals, the median sampling count is 102 µs-134 µs and 21 µs-34 µs for low and high noise respectively. Concerning the resolution in time, we observe that the minimum value is 0.125 µs. However, it will excessively increase the number of bits for our differential value. Finally, we employed a base value of 0.5 µs, which will be used in all the cases. It will give a number of bits of 8 and 6 for 5 µV and 25 µV of noise, respectively, for the representative time difference, which will be used to estimate the number of bits required for the transmission.

First, we will estimate the power for our reference design with fixed rate sampling, which is the output of the SAR. In this case, the stream length is agnostic of input signal noise or bandwidth after the conversion with the ADC. Therefore, the results will be the

same for any of the conditions considered in the LCADC case. The computed values are presented in Table 5.11.

Table 5.11: Transmission power estimation for fixed rate sampling scheme outputs.

Parameter	**SAR**	**NEO(SAR)**
ADC frequency (KS/s)	20	20
Bits	8	16
R_{ADC} (Kbits/s)	160	320
R_{TX} (Kbits/s)	160	320
Dynamic TX power (µW)[1]	1.36	2.72
Total TX power (µW)[2]	**1.766**	**3.126**

[1] Dynamic power obtained from $E_b \times R_{\text{TX}}$.
[2] Total power adds 13 µW/32 of static power to the dynamic power, from value reported by [87].

After that, we estimate the bits for the transmission stream and the corresponding power for different level-crossing schemes described in Table 5.12.

Table 5.12: Transmission power estimation for different bit stream schemes of level crossing sampling. It includes variations in noise level, bandwidth, and the type of samples to which the NEO is applied.

Test input signal		**Parameter**	**ADC Samples**		**NEO samples**		**NEO(abs) samples**	
BW (Hz)	Noise (µV)		LCADC	LCADC +dtime[3]	NEO (LCADC)	NEO (LCADC) +dtime	NEO (LCADC)	NEO (LCADC) +dtime
300-6000	5	ADC frequency (KS/s)	3.5	3.5	2.485	2.485	2.27	2.27
		Bits	6	14	12	20	12	20
		R_{ADC} (Kbits/s)	21	49	29.82	49.7	27.24	45.4
		R_{TX} (Kbits/s)	21	49	29.82	49.7	27.24	45.4
		Dynamic TX power (µW)[1]	0.179	0.417	0.253	0.422	0.232	0.386
		Total TX power (µW)[2]	**0.585**	**0.823**	**0.660**	**0.829**	**0.638**	**0.792**
300-1000	5	ADC frequency (KS/s)	2.41	2.41	1.7	1.7	1.645	1.645
		Bits	6	14	12	20	12	20
		R_{ADC} (Kbits/s)	14.46	33.74	20.4	34	19.74	32.9
		R_{TX} (Kbits/s)	14.46	33.74	20.4	34	19.74	32.9
		Dynamic TX power (µW)	0.123	0.287	0.173	0.289	0.168	0.280
		Total TX power (µW)	**0.529**	**0.693**	**0.580**	**0.695**	**0.574**	**0.686**
300-6000	25	ADC frequency (KS/s)	34.3	34.3	28	28	27	27
		Bits	6	12	12	18	12	18
		R_{ADC} (Kbits/s)	205.8	411.6	336	504	324	486
		R_{TX} (Kbits/s)	205.8	411.6	336	504	324	486
		Dynamic TX power (µW)	1.749	3.499	2.856	4.284	2.754	4.131
		Total TX power (µW)	**2.156**	**3.905**	**3.262**	**4.690**	**3.160**	**4.537**
300-1000	25	ADC frequency (KS/s)	24.46	24.46	18.6	18.6	14.41	14.41
		Bits	6	12	12	18	12	18
		R_{ADC} (Kbits/s)	146.76	293.52	223.2	334.8	172.92	259.38
		R_{TX} (Kbits/s)	146.76	293.52	223.2	334.8	172.92	259.38
		Dynamic TX power (µW)	1.247	2.495	1.897	2.846	1.470	2.205
		Total TX power (µW)	**1.654**	**2.901**	**2.303**	**3.252**	**1.876**	**2.611**

[1] Dynamic power obtained from $E_b \times R_{\text{TX}}$.
[2] Total power adds 13 µW/32 of static power to the dynamic power, from value reported by [87].
[3] dtime stands for differential time between samples.

Here, in the first column, we present a special case when it only transmits the LCADC voltage values. This will require the reception times to be used to estimate the intervals corresponding to each sample. However, due to the nature of wireless links this scheme is

not reliable. That is why, in the second column, we include the time interval as a digital value, with representative bits according to our estimation. This scheme only transmits the required number of bits for each interval, as transmitting the same number all the time will produce an excessive overhead that would waste unnecessary power. The other four columns are the corresponding schemes that transmit the NEO values applied to the level crossing output, and this information, plus time interval info, for both direct and absolute values NEO(abs).

From the dynamic power values, we identify that the reduced number of samples considerably decreases the power. For low noise signal inputs (5 µV), the reduction is such that even with the additional time information, the power in the level-crossing sampling scheme for NEO output (0.422 µW) is still lower than the fixed one sending only converted voltage values (1.36 µW). This reduction is improved with low bandwidth signals and using absolute signal values as NEO input, reaching up to 0.288 µW.

In contrast, for high signal inputs (25 µV), the number of samples increases significantly. That produces an increase in the effective sampling frequency and, consequently, in the power dissipation. As anticipated by the results of the NEO block Table 5.8, it overpasses the consumption estimated for a fixed rate sampling scheme (3.126 µW). However, we can also identify that for low bandwidth (1 kHz), the total transmission power is comparable (3.252 µW) and even lower (2.611 kHz) when using the absolute voltage values as NEO input.

5.4.4 Estimation of full acquisition chain

If we combine the power dissipated by the transmission in Table 5.11 and Table 5.12, plus the ADC and preprocessor power in Table 5.8, plus the frontend in Table 5.9, we would obtain the total power dissipated by the acquisition chain, which is presented in the Table 5.13.

According to Table 5.13, we appreciate that the fixed rate sampling scheme has almost no variation in the power consumption for different input bandwidths and noise levels, while level-crossing is highly dependent on them, especially on the input noise level. Besides that, the low bandwidth helps reduce the power, particularly for high noise levels. In addition, the NEO applied to absolute ADC output values also contributes to reducing the power, as identified in previous estimations and NEO measurements. From this, we can conclude that using lower bandwidth and the absolute output values for NEO calculation

Table 5.13: Power dissipation for each component in the acquisition chain. Comparison between fixed-rate (SAR) and level-crossing sampling schemes.

Test input signal		**Component**	**Fixed rate sampling**		**Level-crossing sampling**	
BW (Hz)	Noise (µV)		with NEO	with NEO(abs)	with NEO	with NEO(abs)
300-6000	5	Frontend (µW)[1]	4.393	4.393	4.393	4.393
		ADC (µW)	0.575	0.575	0.967	0.967
		Preprocessor (µW)	0.187	0.135	0.072	0.064
		TX (µW)	3.126	3.126	0.829	0.792
		Total (µW)	**8.281**	**8.229**	**6.261**	**6.216**
300-1000	5	Frontend (µW)	4.393	4.393	4.393	4.393
		ADC (µW)	0.568	0.568	0.974	0.974
		Preprocessor (µW)	0.203	0.137	0.067	0.062
		TX (µW)	3.126	3.126	0.695	0.686
		Total (µW)	**8.290**	**8.224**	**6.129**	**6.115**
300-6000	25	Frontend (µW)	4.393	4.393	4.393	4.393
		ADC (µW)	0.601	0.601	1.056	1.056
		Preprocessor (µW)	0.228	0.147	0.289	0.204
		TX (µW)	3.126	3.126	4.69	4.537
		Total (µW)	**8.348**	**8.267**	**10.428**	**10.190**
300-1000	25	Frontend (µW)	4.393	4.393	4.393	4.393
		ADC (µW)	0.597	0.597	1.057	1.057
		Preprocessor (µW)	0.213	0.147	0.218	0.153
		TX (µW)	3.126	3.126	3.252	2.611
		Total (µW)	**8.329**	**8.263**	**8.920**	**8.214**

[1] For a matter of simplicity, we considered the same analog frontend for high and low bandwidth. However, the values correspond to low bandwidth, as for a higher one, it will require more power to keep the same input referred noise level.

gives the best conditions to reduce power, principally for the level-crossing scheme that could benefit most in conditions of low input noise.

In this case, although the intrinsic power dissipation of LCADC is higher than SAR, the level-crossing sampling scheme has lower power dissipation (8.22 µW) than the fixed rate (6.12 µW). The difference is mainly due to the reduced power transmission consequence of the effectively reduced stream in the level-crossing generating fewer samples. We should also consider that we are using an extremely efficient transmitter, which significantly decreases the transmission power. It means that if it had higher power per bit, the difference between both schemes would be higher, favoring the level-crossing one.

On the other hand, high noise input levels degrade the performance of the level-crossing scheme, causing, in some cases, overpassing the reference value obtained with the fixed rate scheme. However, considering the low bandwidth (1 KHz) and the NEO applied to absolute values, the power obtained is similar for both cases, being slightly lower for the level-crossing one (8.26 µW and 8.21 µW, respectively). We must also recall that the noise

used in the test (25 µV) is pessimistic and unlikely to happen. It just gives an idea of the limits for which the level-crossing scheme could be beneficial relative to the fixed rate one.

We can refer to the literature to use a more typical noise value. For example, Lempka *et al.* [88] and Ludwig *et al.* [89] report a noise of around 12.1 µV-13.1 µV. Considering that the power depends on the square noise level $P \propto \eta^2$, we can make a gross approximation to find the power consumption for this representative noise value. Performing a simple interpolation, we obtain that for 13.1 µV the power dissipation will be 6.63 µW, which is still appreciably lower than the fixed rate sampling.

We can compare the results obtained with other developments in the academy and industry to know how they are relative to them. That comparison is presented in Table 5.14.

Table 5.14: Power dissipation comparison of the current development in this thesis against other acquisition devices in the literature.

Parameter	Lopez [75]	De Dorigo [9]	NeuroPixel [90]	NeuroSeeker [91]	HermesE [92]	SiNAPS [91]	This thesis	
							FR scheme	LC scheme
Input noise (µV)	6.36	10.46	5.5	31	2.2	7.5	4.5	4.5
f_0 (kHz)[1]	0.3	0.3	0.3	0.3	0.28	0.3	0.3	0.3
f_1 (kHz)[1]	10	10	10	7.5	10	7.5	1	1
F_s (kS/s)	30	20	30	40	31.25	25	20	adaptive
Bits	10	11	10	10	10	12	8	6
Power (µW)	49.06	39.14	45.57	45	68	6[2]	8.24[3]	6.12-8.21[4]

[1] f_0 and f_1 are the limits for the bandwidth used for each acquisition circuit.
[2] This value comes from the article, however, it may not include the ADC power dissipation because they mention it is off-chip.
[3] As the variation for fixed rate is minimal, we used the mean as a representative value.
[4] Values obtained for low (5 µV) and high (25 µV) noise levels.

As presented in the table, we notice that our development is competitive with other acquisition devices. However, it is expected as the bandwidth and resolution will have reduced values. We achieve lower power dissipation even with a non-optimized analog frontend, such as the one used in our sample circuit. Also, we see that the level-crossing scheme could benefit more from low noise signals giving a reduced value compared with others.

The objective of reducing power dissipation is to increase the number of channels, as they would provide more valuable information than fewer channels with higher resolution, according to Even-Chen *et al.* [6]. This is because of the maximum power limit that could have an implanted device. According to Marblestone *et al.* [93], this limit is around 40 mW. That is to avoid a temperature change greater than 2 °C that could damage the brain tissue. Given a margin of 5 mW, the fixed rate scheme can scale up to 4200 electrodes. On its side, the level crossing could be scaled up to 5200 electrodes, considering

the power dissipation estimated previously for a typical noise value. Those values are in the ranges stated by Schwarz *et al.* [94] (> 2500 channels) that could be useful for limb movement restoration.

5.4.5 How about the accuracy?

Until now, we concentrated on estimating the power consumption and reducing it to the minimum. We set the target specifications based on the results we obtained from the ADC and the NEO software models. However, as we also have the hardware models at the transistor level for those components, we consider it useful to obtain the accuracy level for them in spike detection.

For this purpose, we employ the obtained waveforms from the previous simulation of 1 s length. Then we applied an oversampling, following the estimated resolution of 0.5 us for the time. After that, we applied several threshold levels against which we calculated the true positives (TN), false positives (FP), false negatives (FN), and true negatives (TN), following the same strategy used with the software results. Finally, we convert them to accuracy levels.

Something different from the software approach is the selection of thresholds. Previously we used fractions of the maximum value. However, trying to approximate more to a real scenario of detection, we use scaled values of the RMS value (as suggested by Even-Chen *et al.* [6]), in this case, from a factor of 1 to 12 with steps of 0.5.

From Figure 5.24, for low noise signals, we can see that either with the LCADC samples or the NEO samples, we can obtain the same maximum accuracy for both bands. For high noise signals, we obtain a bit more accuracy with only LCADC samples with low bandwidth.

On the other hand, taking as a reference the RMS for scaled thresholds, the maximum accuracy for NEO occurs at a higher scaled value than for LCADC. That makes sense as NEO can isolate the spikes, enhancing them but with a reduced spike width, lowering the RMS value. In addition, we can obtain the maximum accuracy for the LCADC samples in a reduced window of scale RMS values. At the same time, we have a wider threshold window with the NEO results to achieve maximum accuracy. This is an advantage of using NEO samples, as in a real scenario, we couldn't know which is the better threshold that provides the maximum accuracy, and having a wider range of threshold values to obtain it, increases the possibility of detecting spikes accurately. On the other hand, using the absolute samples for the NEO doesn't have a meaningful difference relative to using the

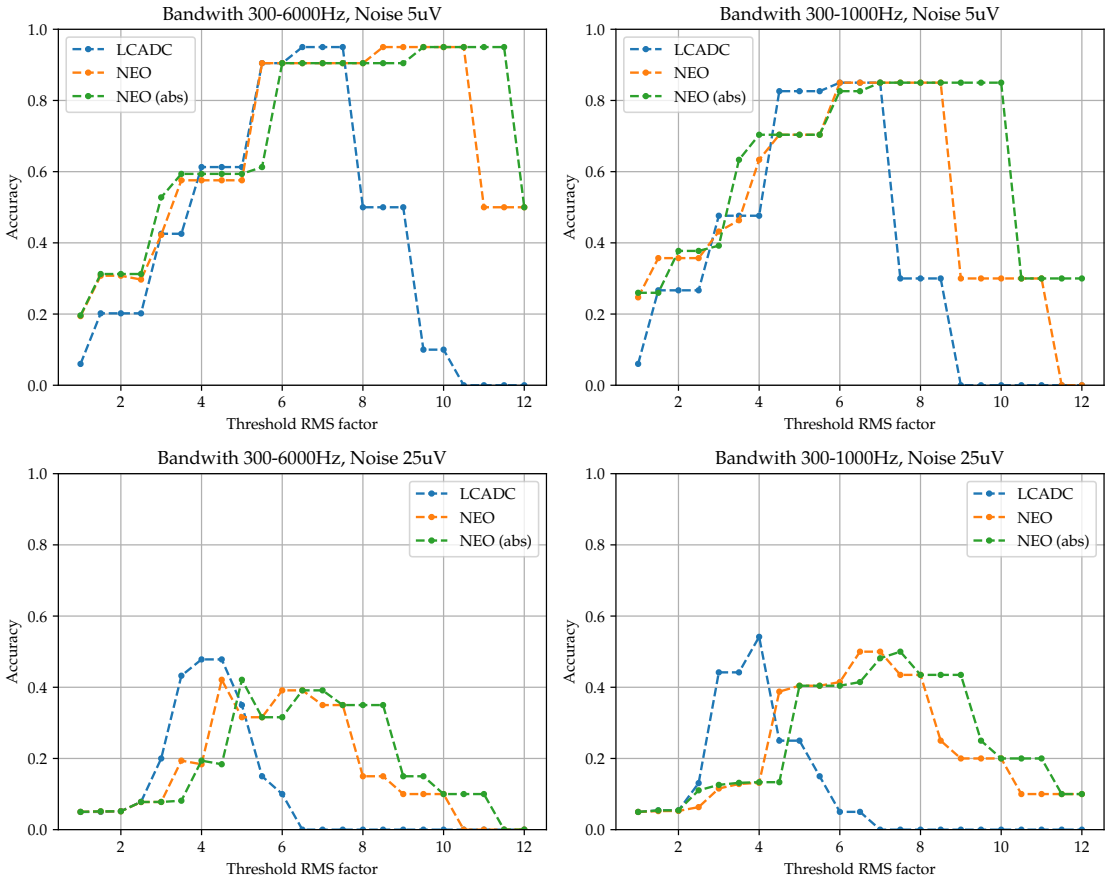


Figure 5.24: Results of accuracy for 1s length, with 5 µV and 25 µV of noise-level. The left side corresponds to the full neural bandwidth, while the right side corresponds to the selected band we use for our device. The values in the x-axis represent the factor multiplied by the RMS value to obtain the threshold level.

normal samples. Consequently, we can decide to use it according to our system in the power consumption section. That behavior is reproduced either with low or high noise signal levels.

If we compare the effect of bandwidth, we see that lower bandwidth reduces the maximum accuracy that can be obtained from the samples. That is expected as reducing the bandwidth will reduce the information that can be obtained from the signal.

Taking as reference the first threshold that achieves the maximum accuracy, we can build Table 5.15. The threshold in LCADC is higher than the NEO, as the RMS value is lower for it (considering also that it is the square of a small voltage). On the other hand, the scale is more distant in the full bandwidth than the lower bandwidth (1kHz), which means that with reduced bandwidth signals we can obtain the maximum accuracy with lower a lower scaled threshold value.

Table 5.15: Results for first threshold level with maximum accuracy. TP is the true positive number, FP is the false positive, and FN is the false negative number.

Test Input signal		**Result type**	**Threshold**		**Detection**			**Accuracy**
BW (Hz)	Noise (µV)		Scale	Value	TP	FP	FN	
300-6000	5	ADC out	6.5	0.023677	19	0	1	0.950
		NEO	8.5	0.000234	19	0	1	0.950
		NEO(abs)	9.5	0.000233	19	0	1	0.950
300-1000	5	ADC out	6	0.019259	17	0	3	0.850
		NEO	6	0.000153	17	0	3	0.850
		NEO(abs)	7	0.00015	17	0	3	0.850
300-6000	25	ADC out	4	0.032888	11	3	9	0.478
		NEO	4.5	0.000295	16	18	4	0.421
		NEO(abs)	5	0.000294	16	18	4	0.421
300-1000	25	ADC out	4	0.025121	13	4	7	0.542
		NEO	6.5	0.000236	13	6	7	0.500
		NEO(abs)	7.5	0.000244	13	6	7	0.500

In addition, concerning the disaggregated values for accuracy, we can analyze two cases, when the signal has low noise and when it has high noise. In the first case, we aim to evaluate the maximum detectability with appropriate conditions, while in the second, we aim to evaluate the performance degradation while increasing the noise to a pessimistic scenario. With low noise signals, the false positives are 0 (FP=0), which means there isn't any wrong detected spike. However, the false negatives reduce the accuracy, which is worse in the case of lower bandwidth. This means that some spikes weren't detected. The explanation for this is that the amplitude of these spikes was not enough to make them distinguishable from other spikes from other neurons. Also, we need to consider that the maximum number of spikes is 20, and consequently, with this test, we can only achieve resolutions of 0.05 in accuracy, which ranges from 0 to 1. Having more samples could reduce the margin of difference. However, it will take longer simulation times.

In the other case, with a high noise signal, we definitely expect a decreased accuracy, as an ideal detector (employing a software model with the same amplitude resolution) would only obtain a maximum of 63% of accuracy. Although it is lower than that value, it can still detect spikes even in a high-noise environment. On the other hand, we also recognize the benefit of using a low bandwidth, as it could obtain higher detection accuracy than the full bandwidth signal. Observing the disaggregated detection information, we identify that the main reason for the lower detection values in the full bandwidth is due to the increased number of false positives. That means that many high-amplitude noise samples are being detected as spikes of the neural signal. It is especially noticeable with the NEO output.

It is explained that this operator has the aim to enhance changes in amplitude, expected to come from intracortical spikes, but in a noise environment also enhances the spurious peaks of amplitude introduced by it. However, this is attenuated, reducing the bandwidth, as it helps decrease the high-frequency amplitude changes that can be misinterpreted as part of a spike train, even when it is applied to the NEO, dramatically reducing the number of false positives and increasing the accuracy. As a side effect, it increases the number of false negatives (spikes not detected); however, the benefit of avoiding spurious spikes compensates for it.

The reduction in accuracy relative to the ideal model for both cases of low and high noise input signals can be explained due to the errors introduced by the hardware model circuit. However, this difference of approximately 10% of accuracy can be tolerated, as it is more realistic and approaches to an implementable design.

Finally, although the accuracy for the low noise case is a bit lower (85%) than expected for reduced bandwidth, we consider it is still appropriate to be used to detect spikes, as it may be compensated with the increase of channels, thanks to the reduction of power dissipation.

Chapter VI

Conclusions

Through this thesis, we explored one of the main components in the acquisition system for an intracortical system: the ADC. We first worked with a software model for the converter and NEO preprocessor to identify the minimum requirements for adequate detection accuracy. Then, we designed the ADC, first with the conventional SAR architecture and with the level-crossing. Also, we designed the NEO preprocessor digital block and measured power consumption with the ADC. Finally, we drafted a sample circuit for the analog frontend and estimated the power required for the transmission of the data stream, calculating the total power needed for the channel signal acquisition and comparing it with other developments in the literature.

From the software modeling of NEO and its evaluation, we have found that the level-crossing scheme could carry almost the same information as the fixed rate sampling scheme, achieving similar accuracies. On the other hand, there is a trade-off between the bandwidth and the resolution regarding the information that can be recovered. Lowering the bandwidth will require more bits to achieve similar accuracy values. We obtained a good balance with a 7-bit resolution for the ADC and a 1kHz for the signal bandwidth's upper limit, achieving accuracies higher than 90% for the most significant spike train in the evaluated recording channel. Also, we found that for the level-crossing scheme, a NEO preprocessor with a 4-sample window would return better results than wider windows in terms of accuracy.

From the ADC design, considering both SAR and LCADC, we found that in terms of performance, the SAR circuit could achieve a bit higher ENOB than the LCADC relative to their respective amplitude resolution, obtaining better values for the FOM. Compared with other designs, ours is average, better than some reported in the literature. When

testing with neural recordings, we appreciate an improvement in the LCADC, which tends to reduce power consumption. However, intrinsic results still obtain lower dissipation for SAR. In addition, evaluating the LCADC requires preprocessing to use the samples, and this procedure shows the behavior of a low-pass filter in the spectral distribution result, which can help reduce the noise in high frequencies.

From power consumption evaluation and their distribution in the components of the ADCs, we concluded that it is challenging that the intrinsic power dissipation of the LCADC would be lower than the SAR. This is for the high efficiency of the dynamic comparator, which with lower sampling frequency, can significantly reduce the consumption compared with a minimalistic continuous time static comparator.

However, despite this fact, the greatest advantage the LCADC provides is the sampling method and the reduced number of samples which can dramatically minimize the power consumption of following digital blocks, such as the NEO preprocessor, especially with low noise signals. The behavior observed with the NEO anticipates the posterior estimation in the wireless transmission component. Although it required more bits for the transmission of time differences with the level-crossing sampling scheme, the significantly lower number of samples considerably reduces the total power for the transmission, which also reduces the total acquisition chain dissipation. The same would apply to any other digital block that could work with asynchronous samples and whose power will be reduced due to the spare sampling characteristic.

In this stage, we also observed the dependence of the transmission power with the input noise, the bandwidth, and the type of NEO input samples. According to the results, noise could significantly impact the level-crossing scheme, especially for high values, expected due to the increase in sampling rate. However, we could compensate for that effect by choosing a low bandwidth (1 kHz) and absolute values for the NEO input, obtaining values even slightly lower than the fixed rate sampling.

On the other hand, although not optimized, our analog frontend showed that the low-noise amplifier is one of the most power-hungry components because it requires high currents to achieve the target input-referred noise. However, reducing the band of interest helps minimize the total current reaching the input noise needed with a lower current requirement.

Consolidating the results and comparing the complete acquisition chain power dissipation with others from the academy or industry, we confirm that relaxed resolution and bandwidth specifications help save power requirements in each stage for the final biomed-

ical device and the level-crossing scheme could leverage those savings in scenarios with low input noise levels.

Finally, reducing the power consumption relaxes the power budget for a single channel allowing the increase of the number of channels in the acquisition system. The literature shows that higher electrode counts would provide more valuable information than lower channel numbers with higher resolution. With the estimation we have done, and based on studies for large-scale neural recordings, we predict that our proposed acquisition chains could be helpful to achieve more than 4000 recording channels, with the implications to facilitate the acquisition of similar orders of neurons and its posterior use as an aid device for restoring movement or communication abilities. Concerning accuracy, although lower for the band of 300 Hz-1 kHz, it is still competitive (85 %), and can be compensated with a higher electrode count, available thanks to the power saving of the reduced bandwidth.

Chapter VII

Recommendations

This thesis has shown the design and evaluation of an LCADC of low resolution and low bandwidth that could be applied as part of a BCI device, comparing its performance with a conventional SAR architecture. Also, we developed a software to obtain the minimal specifications to detain an adequate accuracy at spike detection. In addition, we estimated the power dissipation of a channel of the acquisition chain by designing a conventional analog frontend for the target specifications and approximating the calculus of the wireless transmission.

The software developed for evaluating spike detection with different resolutions, bandwidths, and preprocessor windows was used with synthesized signals generated by open-source software. However, applying it to real signals would be worthwhile to confirm that our obtained values achieve an adequate accuracy level.

On the other hand, the level-crossing circuit design is based on another from the literature that showed a good performance in terms of low power dissipation. However, we designed and compared it with SAR; and the intrinsic power can hardly compete with the synchronous one. Therefore, we suggest exploring alternatives for implementing the level-crossing sampling scheme, for example, with a floating comparison window (in this thesis, we used the fixed window). An architecture that combines a folding structure, an n-bit DAC, and a dynamic comparator could probably be more effective than using a 1-bit DAC and a continuous time comparator (used in our analysis). In addition, our evaluation reached the schematic design as it was the scope we proposed. However, the circuit should go to the physical stage to be implementable. It should be tested across process, temperature, voltage, and mismatch variation to test it in the possible conditions that it would work.

Finally, regarding the other elements that comprise the acquisition chain, in the case of the analog frontend, we designed a generic LNA and filter, obtaining good results regarding power dissipation for the band of interest. However, there is an opportunity for improvement by employing more efficient architectures for those components and, consequently, reducing the power dissipation. On the other hand, regarding the wireless transmission, we just modeled the power consumption, and we didn't approach the design, as it is much more complex and deserves more dedicated effort for its development. However, we consider that, to have the device for production; this will be a crucial component for which the reference circuit employed for the dissipation model could be used as a starting point to design it.

Bibliography

[1] National Institute of Neurological Disorders and Stroke. (2021). Amyotrophic Lateral Sclerosis (ALS) Fact Sheet, [Online]. Available: https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Amyotrophic-Lateral-Sclerosis-ALS-Fact-Sheet (visited on 11/23/2021).

[2] WHO. (2013). Spinal cord injury, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury (visited on 11/23/2021).

[3] M. W. Slutzky, "Brain-Machine Interfaces: Powerful Tools for Clinical Treatment and Neuroscientific Investigations," *The Neuroscientist*, vol. 25, no. 2, pp. 139–154, Apr. 1, 2019, ISSN: 1073-8584. DOI: 10.1177/1073858418775355. [Online]. Available: https://doi.org/10.1177/1073858418775355 (visited on 11/23/2021).

[4] J. E. Huggins, P. A. Wren, and K. L. Gruis, "What would brain-computer interface users want? Opinions and priorities of potential users with amyotrophic lateral sclerosis," *Amyotrophic Lateral Sclerosis*, vol. 12, no. 5, pp. 318–324, Sep. 2011, ISSN: 1748-2968. DOI: 10/bzn42p. pmid: 21534845. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3286341/ (visited on 11/30/2021).

[5] J. D. Simeral, T. Hosman, J. Saab, S. N. Flesher, M. Vilela, B. Franco, J. Kelemen, D. M. Brandman, J. G. Ciancibello, P. G. Rezaii, E. N. Eskandar, D. M. Rosler, K. V. Shenoy, J. M. Henderson, A. V. Nurmikko, and L. R. Hochberg, "Home Use of a Percutaneous Wireless Intracortical Brain-Computer Interface by Individuals With Tetraplegia," *IEEE Transactions on Biomedical Engineering*, no. c, 2021, ISSN: 15582531. DOI: 10.1109/TBME.2021.3069119.

[6] N. Even-Chen, D. G. Muratore, S. D. Stavisky, L. R. Hochberg, J. M. Henderson, B. Murmann, and K. V. Shenoy, "Power-saving design opportunities for wireless intracortical brain–computer interfaces," *Nature Biomedical Engineering*, vol. 4,

no. 10, pp. 984–996, 2020, ISSN: 2157846X. DOI: 10.1038/s41551-020-0595-9. [Online]. Available: http://dx.doi.org/10.1038/s41551-020-0595-9.

[7] M. W. Slutzky, "Increasing power efficiency," *Nature Biomedical Engineering*, vol. 4, no. 10, pp. 937–938, 2020, ISSN: 2157846X. DOI: 10.1038/s41551-020-00631-7. [Online]. Available: http://dx.doi.org/10.1038/s41551-020-00631-7.

[8] S. R. Nason, A. K. Vaskov, M. S. Willsey, E. J. Welle, H. An, P. P. Vu, A. J. Bullard, C. S. Nu, J. C. Kao, K. V. Shenoy, T. Jang, H. S. Kim, D. Blaauw, P. G. Patil, and C. A. Chestek, "A low-power band of neuronal spiking activity dominated by local single units improves the performance of brain–machine interfaces," *Nature Biomedical Engineering*, vol. 4, no. 10, pp. 973–983, 2020, ISSN: 2157846X. DOI: 10.1038/s41551-020-0591-0. [Online]. Available: http://dx.doi.org/10.1038/s41551-020-0591-0.

[9] D. De Dorigo, C. Moranz, H. Graf, M. Marx, D. Wendler, B. Shui, A. Sayed Herbawi, M. Kuhl, P. Ruther, O. Paul, and Y. Manoli, "Fully immersible subcortical neural probes with modular architecture and a delta-sigma ADC integrated under each electrode for parallel readout of 144 recording sites," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 11, pp. 3111–3125, 2018, ISSN: 00189200. DOI: 10.1109/JSSC.2018.2873180.

[10] J. Van Assche and G. Gielen, "Power Efficiency Comparison of Event-Driven and Fixed-Rate Signal Conversion and Compression for Biomedical Applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 746–756, 2020, ISSN: 19409990. DOI: 10.1109/TBCAS.2020.3009027.

[11] H. Zaer, A. Deshmukh, D. Orlowski, W. Fan, P. H. Prouvot, A. N. Glud, M. B. Jensen, E. S. Worm, S. Lukacova, T. W. Mikkelsen, L. M. Fitting, J. R. Adler, M. B. Schneider, M. S. Jensen, Q. Fu, V. Go, J. Morizio, J. C. H. Sørensen, and A. Stroh, "An Intracortical Implantable Brain-Computer Interface for Telemetric Real-Time Recording and Manipulation of Neuronal Circuits for Closed-Loop Intervention," *Frontiers in Human Neuroscience*, vol. 15, pp. 1–14, February 2021, ISSN: 16625161. DOI: 10.3389/fnhum.2021.618626.

[12] Y. Yang, S. Boling, and A. J. Mason, "A Hardware-Efficient Scalable Spike Sorting Neural Signal Processor Module for Implantable High-Channel-Count Brain Ma-

chine Interfaces," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 4, pp. 743–754, 2017, ISSN: 19324545. DOI: 10.1109/TBCAS.2017.2679032.

[13] M. Zamani, D. Jiang, and A. Demosthenous, "An Adaptive Neural Spike Processor with Embedded Active Learning for Improved Unsupervised Sorting Accuracy," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 3, pp. 665–676, 2018, ISSN: 19324545. DOI: 10.1109/TBCAS.2018.2825421.

[14] S. Mansour, K. K. Ang, K. P. Nair, K. S. Phua, and M. Arvaneh, "Efficacy of Brain–Computer Interface and the Impact of Its Design Characteristics on Poststroke Upper-limb Rehabilitation: A Systematic Review and Meta-analysis of Randomized Controlled Trials," *Clinical EEG and Neuroscience*, vol. 53, no. 1, pp. 79–90, 2022, ISSN: 21695202. DOI: 10.1177/15500594211009065. pmid: 33913351.

[15] NCT00912041. (2021). BrainGate2: Feasibility Study of an Intracortical Neural Interface System for Persons With Tetraplegia, [Online]. Available: https://clinicaltrials.gov/ct2/show/NCT00912041 (visited on 11/23/2021).

[16] C. Pandarinath, P. Nuyujukian, C. H. Blabe, B. L. Sorice, J. Saab, F. R. Willett, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, "High performance communication by people with paralysis using an intracortical brain-computer interface," *eLife*, vol. 6, pp. 1–27, 2017, ISSN: 2050084X. DOI: 10.7554/eLife.18554. pmid: 28220753.

[17] F. R. Willett, D. T. Avansino, L. R. Hochberg, J. M. Henderson, and K. V. Shenoy, "High-performance brain-to-text communication via handwriting," *Nature*, vol. 593, no. May, 2021, ISSN: 1476-4687. DOI: 10.1038/s41586-021-03506-2. [Online]. Available: http://dx.doi.org/10.1038/s41586-021-03506-2.

[18] NCT01364480. (2021). Microelectrode Brain-Machine Interface for Individuals With Tetraplegia, [Online]. Available: https://clinicaltrials.gov/ct2/show/NCT01364480 (visited on 11/23/2021).

[19] J. E. Downey, N. Schwed, S. M. Chase, A. B. Schwartz, and J. L. Collinger, "Intracortical recording stability in human brain–computer interface users," *Journal of Neural Engineering*, vol. 15, no. 4, p. 046 016, May 2018, ISSN: 1741-2552. DOI: 10.1088/1741-2552/aab7a0. [Online]. Available: https://doi.org/10.1088/1741-2552/aab7a0 (visited on 11/23/2021).

[20] S. C. Colachis, C. F. Dunlap, N. V. Annetta, S. M. Tamrakar, M. A. Bockbrader, and D. A. Friedenberg, "Long-term intracortical microelectrode array performance in a human: A 5 year retrospective analysis," *Journal of Neural Engineering*, vol. 18, no. 4, p. 0460d7, Aug. 2021, ISSN: 1741-2552. DOI: 10.1088/1741-2552/ac1add. [Online]. Available: https://doi.org/10.1088/1741-2552/ac1add (visited on 11/23/2021).

[21] C. H. Blabe, V. Gilja, C. A. Chestek, K. V. Shenoy, K. D. Anderson, and J. M. Henderson, "Assessment of brain–machine interfaces from the perspective of people with paralysis," *Journal of Neural Engineering*, vol. 12, no. 4, p. 043 002, Jul. 14, 2015, ISSN: 1741-2552. DOI: 10/gmf32r. [Online]. Available: http://iopscience.iop.org/article/10.1088/1741-2560/12/4/043002/meta (visited on 12/01/2021).

[22] M. P. Branco, E. G. M. Pels, R. H. Sars, E. J. Aarnoutse, N. F. Ramsey, M. J. Vansteensel, and F. Nijboer, "Brain-Computer Interfaces for Communication: Preferences of Individuals With Locked-in Syndrome," *Neurorehabilitation and Neural Repair*, vol. 35, no. 3, pp. 267–279, Mar. 2021, ISSN: 1545-9683. DOI: 10/gmmbvn. pmid: 33530868. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7934157/ (visited on 11/30/2021).

[23] E. Musk and Neuralink, "An integrated brain-machine interface platform with thousands of channels," *bioRxiv*, 2019. DOI: 10.1101/703801. eprint: https://www.biorxiv.org/content/early/2019/08/02/703801.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2019/08/02/703801.

[24] Neuralink. (2019). The first fully-implanted 1000+ channel brain-machine interface - Neuralink, [Online]. Available: https://neuralink.com/blog/monkey-mindpong/ (visited on 09/05/2021).

[25] Blackrock Neurotech. (2021). Brain-Computer Interfaces – Blackrock Neurotech, [Online]. Available: https://blackrockneurotech.com/brain-computer-interfaces/ (visited on 09/05/2021).

[26] BrainGate. (2021). BrainGate | Turning Thought Into Action, [Online]. Available: https://www.braingate.org/ (visited on 09/05/2021).

[27] K. E. Barrett, "Ganong's review of medical physiology," pp. 218–219, 2019.

[28] J. J. Feher, *Quantitative human physiology: an introduction*. Academic press, 2017, pp. 365–386.

[29] The McGill Physiology Virtual Lab. (2023). Background: Biphasic, extracellular recording, [Online]. Available: https://www.medicine.mcgill.ca/physio/vlab/Other_exps/CAP/recording.htm (visited on 06/29/2023).

[30] McGill Physiology Virtual Lab. (2005). Background: Biphasic, extracellular recording, [Online]. Available: https://www.medicine.mcgill.ca/physio/vlab/Other_exps/CAP/recording.htm (visited on 09/10/2021).

[31] V. Viswam, M. E. J. Obien, F. Franke, U. Frey, and A. Hierlemann, "Optimal electrode size for multi-scale extracellular-potential recording from neuronal assemblies," *Frontiers in Neuroscience*, vol. 13, no. APR, 2019, ISSN: 1662453X. DOI: 10.3389/fnins.2019.00385. [Online]. Available: www.frontiersin.org.

[32] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-Computer Interfaces in Medicine," *JMCP*, vol. 87, no. 3, pp. 268–279, 2012. DOI: 10.1016/j.mayocp.2011.12.008. [Online]. Available: www.mayoclinicproceedings.org.

[33] K. M. Szostak, L. Grand, and T. G. Constandinou, "Neural interfaces for intracortical recording: Requirements, fabrication methods, and characteristics," *Frontiers in Neuroscience*, vol. 11, no. DEC, 2017, ISSN: 1662453X. DOI: 10.3389/fnins.2017.00665.

[34] C. Ratametha, S. Tepwimonpetkun, and W. Wattanapanitch, "A 2.64-μW 71-dB SNDR Discrete-Time Signal-Folding Amplifier for Reducing ADC's Resolution Requirement in Wearable ECG Acquisition Systems," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 1, pp. 48–64, 2020, ISSN: 19409990. DOI: 10.1109/TBCAS.2019.2957030.

[35] C. Gold, D. A. Henze, C. Koch, and G. Buzsáki, "On the origin of the extracellular action potential waveform: A modeling study," *Journal of Neurophysiology*, vol. 95, no. 5, pp. 3113–3128, 2006, ISSN: 00223077. DOI: 10.1152/jn.00979.2005.

[36] F. Maloberti, *Data converters*. 2007, pp. 1–440, ISBN: 0387324852. DOI: 10.1007/978-0-387-32486-9.

[37] C. Perumal, "Design of a successive approximation (SAR) ADC in 65 nm technology," Lund University, 2011. [Online]. Available: https://www.eit.lth.se/sprapport.php?uid=462.

[38] W. Sansen, *Analog Circuit Design*, M. Steyaert, A. van Roermund, and A. Baschirotto, Eds., 11. Dordrecht: Springer Netherlands, 2012, vol. 5, pp. 3–18, ISBN: 978-94-007-1925-5. DOI: 10.1007/978-94-007-1926-2. [Online]. Available: http://link.springer.com/10.1007/978-94-007-1926-2%20http://link.springer.com/10.1007/978-94-007-1926-2_1.

[39] B. Razavi, "AnalogtoDigital Converter Architectures," *Principles of Data Conversion System Design*, vol. 96, pp. 96–152, 2010. DOI: 10.1109/9780470545638.ch6.

[40] Y. Tsividis, "Event-driven data acquisition and digital signal processing-A tutorial," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 8, pp. 577–581, 2010, ISSN: 15497747. DOI: 10.1109/TCSII.2010.2056012.

[41] J. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, IEEE, 1990, pp. 381–384. DOI: 10.1109/ICASSP.1990.115702. [Online]. Available: http://ieeexplore.ieee.org/document/115702/.

[42] S. Mukhopadhyay and G. C. Ray, "A new interpretation of nonlinear energy operator and its efficacy in spike detection," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 2, pp. 180–187, 1998, ISSN: 00189294. DOI: 10.1109/10.661266.

[43] E. Koutsos, S. E. Paraskevopoulou, and T. G. Constandinou, "A 1.5 μw NEO-based spike detector with adaptive-threshold for calibration-free multichannel neural interfaces," *Proceedings - IEEE International Symposium on Circuits and Systems*, pp. 1922–1925, 2013, ISSN: 02714310. DOI: 10.1109/ISCAS.2013.6572243.

[44] J. C. S. Pumarica, "Sistemas de detecção e classificação de impulsos elétricos de sinais neurais extracelulares," 2016. [Online]. Available: http://www.teses.usp.br/teses/disponiveis/3/3140/tde-19122016-133542/.

[45] J. Saldana-Pumarica and E. Del-Moral-Hernandez, "A CMOS Squarer Based Nonlinear Filter for Spike Detection," *2018 25th IEEE International Conference on Electronics Circuits and Systems, ICECS 2018*, pp. 121–124, 2019. DOI: 10.1109/ICECS.2018.8617919.

[46] S. Gibson, J. W. Judy, and D. Marković, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 124–143, 2012, ISSN: 10535888. DOI: 10.1109/MSP.2011.941880.

[47] H. Hao, J. Chen, A. Richardson, J. Van Der Spiegel, and F. Aflatouni, "A 10.8 μw Neural Signal Recorder and Processor with Unsupervised Analog Classifier for Spike Sorting," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, no. 2, pp. 351–364, 2021, ISSN: 19409990. DOI: 10.1109/TBCAS.2021.3076147.

[48] A. P. Buccino and G. T. Einevoll, "Mearec: A fast and customizable testbench simulator for ground-truth extracellular spiking activity," *Neuroinformatics*, pp. 1–20, 2020.

[49] N. T. Carnevale and M. L. Hines, "The NEURON book," *The NEURON Book*, pp. 1–457, 2006. DOI: 10.1017/CBO9780511541612.

[50] H. Lindén, E. Hagen, S. Łeski, E. S. Norheim, K. H. Pettersen, and G. T. Einevoll, "LFPy: A tool for biophysical simulation of extracellular potentials generated by detailed model neurons," *Frontiers in Neuroinformatics*, vol. 7, no. JAN, pp. 1–15, 2014, ISSN: 16625196. DOI: 10.3389/fninf.2013.00041.

[51] A. P. Buccino, C. L. Hurwitz, S. Garcia, J. Magland, J. H. Siegle, R. Hurwitz, and M. H. Hennig, "Spikeinterface, a unified framework for spike sorting," *eLife*, vol. 9, pp. 1–24, 2020, ISSN: 2050084X. DOI: 10.7554/eLife.61834.

[52] T. Kobayashi, K. Nogami, T. Shirotori, Y. Fujimoto, and O. Watanabe, "A current-mode latch sense amplifier and a static power saving input buffer for low-power architecture," in *1992 Symposium on VLSI Circuits Digest of Technical Papers*, Jun. 1992, pp. 28–29. DOI: 10.1109/VLSIC.1992.229252.

[53] B. Razavi, "The StrongARM latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015, ISSN: 19430582. DOI: 10.1109/MSSC.2015.2418155.

[54] ——, "The Design of a Comparator [The Analog Mind]," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 4, pp. 8–14, 2020, ISSN: 1943-0590. DOI: 10.1109/MSSC.2020.3021865.

[55] ——, *Design of Analog CMOS Integrated Circuits 2nd Edition*. 2017, 100-126, ISBN: 978-0-07-252493-2. pmid: 9917832.

[56] T. O. Anderson, "Optimum Control Logic for Successive Approximation Analog-to-Digital Converters," *Deep Space Network Progress Report*, vol. 13, pp. 168–176, Nov. 1, 1972. [Online]. Available: https://ui.adsabs.harvard.edu/abs/1972DSNPR..13..168A (visited on 05/29/2022).

[57] T. Bos, K. Badami, W. Dehaene, and M. Verhelst, "Architecture optimization for energy-efficient resolution-scalable 8–12-bit SAR ADCs," *Analog Integrated Circuits and Signal Processing*, vol. 97, no. 3, pp. 437–448, Dec. 1, 2018, ISSN: 1573-1979. DOI: 10.1007/s10470-018-1235-0. [Online]. Available: https://doi.org/10.1007/s10470-018-1235-0 (visited on 03/04/2022).

[58] H. Li and J. Hu, "The Research on SAR ADC Integrated Circuit," *Journal of Physics: Conference Series*, vol. 1314, no. 1, p. 012 022, Oct. 1, 2019, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1314/1/012022. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1314/1/012022 (visited on 03/05/2022).

[59] A. Ahuja, K. Badami, C. Barbelenet, and S. Emery, "Comparison of Capacitive DAC Architectures for Power and Area Efficient SAR ADC Designs," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401768.

[60] A. Bonetti, "Low Power and Compact Successive Approximation ADC for Bioelectronic Chips," Apr. 23, 2012. DOI: 10.13140/RG.2.1.3850.0326.

[61] S. Li, J. Chen, B. Liang, and Y. Guo, "Low Power SAR ADC Design with Digital Background Calibration Algorithm," *Symmetry*, vol. 12, no. 11, p. 1757, 11 Nov. 2020, ISSN: 2073-8994. DOI: 10.3390/sym12111757. [Online]. Available: https://www.mdpi.com/2073-8994/12/11/1757 (visited on 04/07/2022).

[62] B. Razavi, *Fundamentals of Microelectronics*. John Wiley & Sons, Apr. 8, 2013, 934 pp., ISBN: 978-1-118-15632-2. Google Books: zpMYAgAAQBAJ.

[63] Y. Li, D. Zhao, and W. A. Serdijn, "A Sub-Microwatt Asynchronous Level-Crossing ADC for Biomedical Applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, no. 2, pp. 149–157, Apr. 2013, ISSN: 1940-9990. DOI: 10.1109/TBCAS.2013.2254484.

[64] Y. Li, D. Zhao, M. N. van Dongen, and W. A. Serdijn, "A 0.5V signal-specific continuous-time level-crossing ADC with charge sharing," in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Nov. 2011, pp. 381–384. DOI: 10.1109/BioCAS.2011.6107807.

[65] Y. Li, "Level-Crossing ADCs and Their Applications in Biomedical Readout Systems," 2015. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A69f7fbd3-4922-4526-b92d-0f3a6bbe6ba8 (visited on 05/14/2022).

[66] X. Qian and T. H. Teo, "A low-power comparator with programmable hysteresis level for blood pressure peak detection," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, Jan. 2009, pp. 1–4. DOI: 10.1109/TENCON.2009.5396125.

[67] N. Sayiner, H. Sorensen, and T. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 4, pp. 335–339, Apr. 1996, ISSN: 1558-125X. DOI: 10.1109/82.488288.

[68] IEEE, "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)*, pp. 1–139, Jan. 2011. DOI: 10.1109/IEEESTD.2011.5692956.

[69] T. B. Nazzal, S. A. Mahmoud, and M. O. Shaker, "A 200-nW 7.6-ENOB 10-KS/s SAR ADC in 90-nm CMOS for Portable Biomedical Applications," *Microelectronics Journal*, vol. 56, pp. 81–96, Oct. 2016, ISSN: 00262692. DOI: 10.1016/j.mejo.2016.08.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0026269216303470 (visited on 04/12/2022).

[70] R. Ranjandish and A. Schmid, "Walsh-Hadamard-Based Orthogonal Sampling Technique for Parallel Neural Recording Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 4, pp. 1740–1749, 2021, ISSN: 15580806. DOI: 10.1109/TCSI.2021.3055484.

[71] X. Liu, H. Zhu, T. Qiu, S. Y. Sritharan, D. Ge, S. Yang, M. Zhang, A. G. Richardson, T. H. Lucas, N. Engheta, and J. Van Der Spiegel, "A Fully Integrated Sensor-Brain-Machine Interface System for Restoring Somatosensation," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4764–4775, 2021, ISSN: 15581748. DOI: 10.1109/JSEN.2020.3030899.

[72] M. R. Pazhouhandeh, H. Kassiri, A. Shoukry, I. Weisspapir, P. L. Carlen, R. Genov, and S. Member, "Opamp-Less Sub-uW/Channel d-Modulated Neural-ADC With Super-G Input Impedance," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 5, pp. 1565–1575, 2021.

[73] K. Kakaraparty, N. Tasneem, and I. Mahbub, "A low-power front-end with compressive sensing circuit for neural signal acquisition designed in 180 nm CMOS Process," *Proceedings of the 2020 IEEE Dallas Circuits and Systems Conference, DCAS 2020*, 2020. DOI: 10.1109/DCAS51144.2020.9330674.

[74] B. Shui, D. D. Dorigo, A. Sayed Herbawi, P. Ruther, O. Paul, Y. Manoli, and M. Kuhl, "A Slim Needle Neural Probe with 160 Active Recording Sites and Selectable ADCs," *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 2019–2022, 2019. DOI: 10.1109/BIOCAS.2019.8919180.

[75] C. Mora Lopez, J. Putzeys, B. C. Raducanu, M. Ballini, S. Wang, A. Andrei, V. Rochus, R. Vandebriel, S. Severi, C. Van Hoof, S. Musa, N. Van Helleputte, R. F. Yazicioglu, and S. Mitra, "A Neural Probe with Up to 966 Electrodes and Up to 384 Configurable Channels in 0.13 um SOI CMOS," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 3, pp. 510–522, 2017, ISSN: 19324545. DOI: 10.1109/TBCAS.2016.2646901.

[76] F. Shahrokhi, K. Abdelhalim, D. Serletis, P. L. Carlen, and R. Genov, "The 128-Channel Fully Differential Digital Integrated Neural Recording and Stimulation Interface," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, no. 3, pp. 149–161, Jun. 2010, ISSN: 1940-9990. DOI: 10.1109/TBCAS.2010.2041350.

[77] C. Kim, S. Joshi, H. Courellis, J. Wang, C. Miller, and G. Cauwenberghs, "A 92dB dynamic range sub-uVrms-noise 0.8uW/ch neural-recording ADC array with predictive digital autoranging," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb. 2018, pp. 470–472. DOI: 10.1109/ISSCC.2018.8310388.

[78] J. Xu, A. T. Nguyen, T. Wu, W. Zhao, D. K. Luu, and Z. Yang, "A Wide Dynamic Range Neural Data Acquisition System With High-Precision Delta-Sigma ADC and On-Chip EC-PC Spike Processor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 3, pp. 425–440, Jun. 2020, ISSN: 1940-9990. DOI: 10.1109/TBCAS.2020.2972013.

[79] B. Schell and Y. Tsividis, "A Continuous-Time ADC/DSP/DAC System With No Clock and With Activity-Dependent Power Dissipation," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 11, pp. 2472–2481, Nov. 2008, ISSN: 1558-173X. DOI: 10.1109/JSSC.2008.2005456.

[80] Y. Hou, J. Qu, Z. Tian, M. Atef, K. Yousef, Y. Lian, and G. Wang, "A 61-nW Level-Crossing ADC With Adaptive Sampling for Biomedical Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 1, pp. 56–60, Jan. 2019, ISSN: 1558-3791. DOI: 10.1109/TCSII.2018.2841037.

[81] A. Joshi, R. Deshmukh, and R. Patrikar, "A Hysteresis Controlled Resolution for Level Crossing Sampling ADC," in *2017 14th IEEE India Council International Conference (INDICON)*, Dec. 2017, pp. 1–5. DOI: 10.1109/INDICON.2017.8488136.

[82] C. Weltin-Wu and Y. Tsividis, "An Event-driven Clockless Level-Crossing ADC With Signal-Dependent Adaptive Resolution," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 9, pp. 2180–2190, Sep. 2013, ISSN: 1558-173X. DOI: 10.1109/JSSC.2013.2262738.

[83] A. Das, S. Rout, A. Urso, and W. A. Serdijn, "Activity Dependent Multichannel ADC Architecture using Level Crossing Quantisation for Atrial Electrogram Recording," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2019, pp. 1–4. DOI: 10.1109/BIOCAS.2019.8919156.

[84] P. R. Gray, Ed., *Analysis and Design of Analog Integrated Circuits*, 5th ed. New York: Wiley, 2009, 881 pp., ISBN: 978-0-470-24599-6.

[85] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*, 3rd ed, ser. The Oxford Series in Electrical and Computer Engineering. New York ; Oxford: Oxford University Press, USA, 2012, 757 pp., ISBN: 978-0-19-976507-2.

[86] P. Gupta and S. L. Tripathi, "Low power design of bulk driven operational transconductance amplifier," in *2017 Devices for Integrated Circuit (DevIC)*, Mar. 2017, pp. 241–246. DOI: 10.1109/DEVIC.2017.8073944.

[87] H. Miranda and T. H. Meng, "A programmable pulse UWB transmitter with 34% energy efficiency for multichannel neuro-recording systems," in *IEEE Custom Integrated Circuits Conference 2010*, Sep. 2010, pp. 1–4. DOI: 10.1109/CICC.2010.5617608.

[88] S. F. Lempka, M. D. Johnson, M. A. Moffitt, K. J. Otto, D. R. Kipke, and C. C. McIntyre, "Theoretical analysis of intracortical microelectrode recordings," *Journal of neural engineering*, vol. 8, no. 4, p. 045 006, Aug. 2011, ISSN: 1741-2560. DOI:

10.1088/1741-2560/8/4/045006. pmid: 21775783. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3196618/ (visited on 09/03/2023).

[89] K. A. Ludwig, J. D. Uram, J. Yang, D. C. Martin, and D. R. Kipke, "Chronic neural recordings using silicon microelectrode arrays electrochemically deposited with a poly(3,4-ethylenedioxythiophene) (PEDOT) film," *Journal of Neural Engineering*, vol. 3, no. 1, pp. 59–70, Mar. 1, 2006, ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2560/3/1/007. [Online]. Available: https://iopscience.iop.org/article/10.1088/1741-2560/3/1/007 (visited on 09/03/2023).

[90] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın, M. Barbic, T. J. Blanche, V. Bonin, J. Couto, B. Dutta, S. L. Gratiy, D. A. Gutnisky, M. Häusser, B. Karsh, P. Ledochowitsch, C. M. Lopez, C. Mitelut, S. Musa, M. Okun, M. Pachitariu, J. Putzeys, P. D. Rich, C. Rossant, W.-l. Sun, K. Svoboda, M. Carandini, K. D. Harris, C. Koch, J. O'Keefe, and T. D. Harris, "Fully Integrated Silicon Probes for High-Density Recording of Neural Activity," *Nature*, vol. 551, no. 7679, pp. 232–236, Nov. 8, 2017, ISSN: 0028-0836. DOI: 10.1038/nature24636. pmid: 29120427. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5955206/ (visited on 05/05/2023).

[91] G. N. Angotzi, F. Boi, A. Lecomte, E. Miele, M. Malerba, S. Zucca, A. Casile, and L. Berdondini, "SiNAPS: An implantable active pixel sensor CMOS-probe for simultaneous large-scale neural recordings," *Biosensors and Bioelectronics*, vol. 126, pp. 355–364, Feb. 1, 2019, ISSN: 0956-5663. DOI: 10.1016/j.bios.2018.10.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0956566318308273 (visited on 05/05/2023).

[92] H. Gao, R. M. Walker, P. Nuyujukian, K. A. A. Makinwa, K. V. Shenoy, B. Murmann, and T. H. Meng, "HermesE: A 96-Channel Full Data Rate Direct Neural Interface in 0.13 μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 4, pp. 1043–1055, Apr. 2012, ISSN: 1558-173X. DOI: 10.1109/JSSC.2012.2185338.

[93] A. Marblestone, B. Zamft, Y. Maguire, M. Shapiro, T. Cybulski, J. Glaser, D. Amodei, P. B. Stranges, R. Kalhor, D. Dalrymple, D. Seo, E. Alon, M. Maharbiz, J. Carmena, J. Rabaey, E. Boyden**, G. Church**, and K. Kording**, "Physical principles for scalable neural recording," *Frontiers in Computational Neuroscience*,

vol. 7, 2013, ISSN: 1662-5188. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2013.00137 (visited on 05/22/2023).

[94] D. A. Schwarz, M. A. Lebedev, T. L. Hanson, D. F. Dimitrov, G. Lehew, J. Meloy, S. Rajangam, V. Subramanian, P. J. Ifft, Z. Li, A. Ramakrishnan, A. Tate, K. Z. Zhuang, and M. A. L. Nicolelis, "Chronic, wireless recordings of large-scale brain activity in freely moving rhesus monkeys," *Nature Methods*, vol. 11, no. 6, pp. 670–676, Jun. 2014, ISSN: 1548-7105. DOI: 10.1038/nmeth.2936. pmid: 24776634.

[95] W. R. Bennett, "Spectra of quantized signals," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 446–472, Jul. 1948, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01340.x.

[96] W. Kester, "Taking the Mystery out of the Infamous Formula, "SNR = 6.02N + 1.76dB," and Why You Should Care," Analog Devices, Inc., MT-001 Tutorial, 2009, p. 7. [Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/mt-001.pdf (visited on 05/17/2021).

[97] M. Trakimas and S. R. Sonkusale, "An Adaptive Resolution Asynchronous ADC Architecture for Data Compression in Energy Constrained Sensing Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 5, pp. 921–934, May 2011, ISSN: 1558-0806. DOI: 10.1109/TCSI.2010.2092132.

[98] K. Kozmin, J. Johansson, and J. Delsing, "Level-Crossing ADC Performance Evaluation Toward Ultrasound Application," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 8, pp. 1708–1719, Aug. 2009, ISSN: 1558-0806. DOI: 10.1109/TCSI.2008.2010094.

[99] J. Mark and T. Todd, "A Nonuniform Sampling Approach to Data Compression," *IEEE Transactions on Communications*, vol. 29, no. 1, pp. 24–32, Jan. 1981, ISSN: 1558-0857. DOI: 10.1109/TCOM.1981.1094872.

[100] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*. Cengage Learning, Oct. 27, 2014, 950 pp., ISBN: 978-1-111-79878-9. Google Books: lTgGAAAAQBAJ.

[101] W. Kester and A. D. I. Engineering, *Data Conversion Handbook*. Newnes, 2005, 977 pp., ISBN: 978-0-7506-7841-4. Google Books: 0aeBS6SgtR4C.

[102] J. Blair, "Histogram measurement of ADC nonlinearities using sine waves," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 3, pp. 373–383, Jun. 1994, ISSN: 1557-9662. DOI: 10.1109/19.293454.

[103] Maxim Integrated. (Mar. 29, 2002). Coherent Sampling vs. Window Sampling, [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/1040.html (visited on 05/18/2022).

[104] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, Apr. 1999, ISSN: 1558-0008. DOI: 10.1109/49.761034.

[105] G. Geelen, "A 6 b 1.1 GSample/s CMOS A/D converter," in *2001 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC (Cat. No.01CH37177)*, Feb. 2001, pp. 128–129. DOI: 10.1109/ISSCC.2001.912572.

[106] D. Draxelmayr, "A 6b 600MHz 10mW ADC array in digital 90nm CMOS," in *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)*, Feb. 2004, 264–527 Vol.1. DOI: 10.1109/ISSCC.2004.1332695.

[107] B. E. Jonsson. (Jan. 28, 2011). Generic ADC FOM classes, Converter Passion, [Online]. Available: https://converterpassion.wordpress.com/generic-adc-fom-classes/ (visited on 05/18/2022).

[108] H. H. Ku, *Notes on the Use of Propagation of Error Formulas*. National Bureau of Standards, 1966. [Online]. Available: http://archive.org/details/jresv70Cn4p263 (visited on 10/21/2023).

[109] National Institute of Standards and Technology. (2012). 2.5.5. Propagation of error considerations, NIST/SEMATECH e-Handbook of Statistical Methods, [Online]. Available: https://www.itl.nist.gov/div898/handbook/mpc/section5/mpc55.htm (visited on 10/21/2023).

Appendix A

Fundamentals on ADC's parameters of performance

To evaluate both proposed designs properly (SAR and LCADC) we need to consider some parameters of performance and understand how they work to make an effective comparison between them. Here, we choose some parameters and explain how they could be interpreted when applied to the selected schemes.

A.1 ENOB and quantization noise

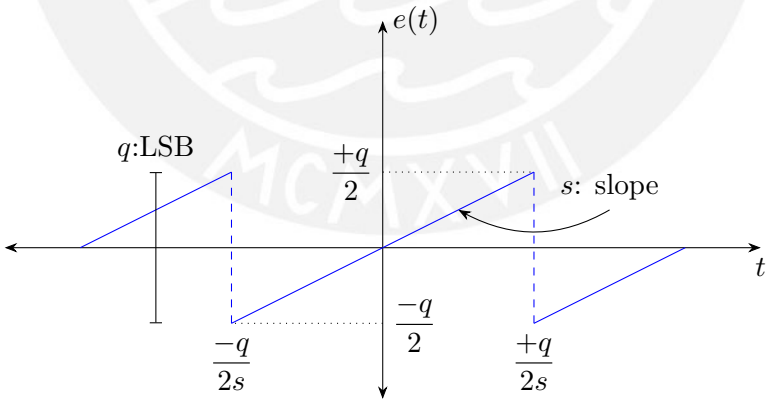


Figure A.1: Error function for a slow ramp input signal. Here the quantization amplitude is represented by q, equivalent to the LSB in an multi-bit scheme. The error ranges from $\frac{-q}{2}$ and $\frac{q}{2}$, while the time will depend on the slope input signal.

When a linear input is quantized, the quantization error signal is similar to the presented in Figure A.1. This image shows that the error varies between two values $\frac{q}{2}$ and

$\frac{-q}{2}$ where q is equal to the LSB [95], [96]. The equation of this function is:

$$e(t) = st; \qquad \frac{-q}{2s} < t < \frac{q}{2s} \tag{A.1}$$

where s represents the slope of the signal. Then, the mean square value of this signal can be calculated from the integral of the function over a period:

$$\overline{e^2(t)} = \frac{s}{q}\int_{-\frac{q}{2s}}^{+\frac{q}{2s}} (st)^2\, dt = \frac{1}{q} \times \frac{(st)^3}{3}\Big|_{-q/2s}^{+q/2s} = \frac{q^2}{12} \tag{A.2}$$

Then the root-mean-square quantization error is:

$$\sqrt{\overline{e^2(t)}} = \frac{q}{\sqrt{12}} \tag{A.3}$$

In general, the quantization noise is uncorrelated from the input, and its RMS value remains approximately around $q/\sqrt{12}$, as pointed out by Kester [96]. It is also interesting to point out that this error doesn't depend on the sampling frequency, being also determined by the resolution in amplitude. Then, considering an input of a full-scale sine wave, whose equation is:

$$v(t) = \frac{q \cdot 2^N}{2}\sin(2\pi f t) \tag{A.4}$$

Here, as q is the LSB, then the full range will be $q2^N$, however, as the sine function oscillates between the amplitude with positive and negative values, then to cover the full range, the amplitude must be half of the full range.

Calculating the RMS value for a sinusoidal function it is known that the value is equal to the amplitude divided by the $\sqrt{2}$ and could be demonstrated by:

$$\overline{v^2(t)} = \frac{A^2}{T}\int_0^T \sin^2(2\pi f t)\, dt = \frac{A^2}{8\pi}\int_0^T 1 - \cos(4\pi f t)\, d(4\pi f t) \tag{A.5}$$

$$= \frac{A^2}{8\pi}\left(4\pi f t - \sin(4\pi f t)\right)\Big|_0^T = \frac{A^2}{2} \tag{A.6}$$

$$\sqrt{\overline{v^2(t)}} = \frac{A}{\sqrt{2}} = A\frac{\sqrt{2}}{2} \tag{A.7}$$

In the case of the (A.4), the amplitude A is equal to $q2^N/2$, then the final RMS value will be:

$$\sqrt{\overline{v^2(t)}} = \frac{q2^N}{2\sqrt{2}} \tag{A.8}$$

With this value it is possible to obtain the SNR value for an ideal N-bit converter, which will be [96]:

$$\text{SNR} = 20\log_{10}\frac{\text{rms value of signal}}{\text{rms value of noise}} = 20\log_{10}\left[\frac{q2^N/2\sqrt{2}}{q/\sqrt{12}}\right] \tag{A.9}$$

$$= 20\log_{10}2^N + 20\log_{10}\sqrt{\frac{3}{2}} \tag{A.10}$$

$$\text{SNR} = 6.02N + 1.76\,\text{dB}, \qquad \text{over the DC to } f_s/2 \text{ bandwidth} \tag{A.11}$$

This relation is frequently used to calculate the ENOB, which in this formula is the N quantity, whose classical equation is [36]:

$$\text{ENOB} = \frac{\text{SNR} - 1.76}{6.02} \tag{A.12}$$

However, it assumes that the input signal is full-scale. It means that if the input amplitude is affected by a scale factor α, then the SNR will be:

$$\text{SNR} = 6.02N + 1.76\,\text{dB} + 20\log_{10}\alpha \tag{A.13}$$

Then, if the scale factor is lower than 1, the SNR will be less, and it will reduce the ENOB value.

In fact, the ENOB means how many bits can be extracted from an output with a certain level of SNR and it compares how accurate the conversion is. However, if the input conditions differ, comparing two systems by their ENOBs could not be the fairest method. Then, we must fix the previous ENOB formula to attenuate the variation input level. We will call it extrapolated ENOB, meaning how many bits would have if the input was a full-scale sinusoidal signal. Then, we introduced that scale factor to update the ENOB formula as follows:

$$\text{ENOB}_{\text{ext}} = \frac{\text{SNR} - 1.76 - 20\log_{10}\alpha}{6.02} \tag{A.14}$$

A.2 Oversampling

Usually, when converting a signal, it is sampled at a greater frequency than the Nyquist (twice the signal bandwidth). Although the quantization noise remains the same, it is extended over a large range of frequencies. Then, If we filter only the bandwidth of interest, it will increase the resulting SNR.

From the previous formula (A.2), the mean square value, equivalent to the noise power, is approximated to $q^2/12$, almost independent of the sampling frequency. Then, when

increasing the sampling, this value will remain the same but will cover a larger range. Therefore, the amplitude for each component of frequency will be:

$$F^2(f) = \frac{q^2}{12} \times \frac{2}{f_s}; \qquad \overline{F^2(t)} = \int_0^{\frac{f_s}{2}} \frac{q^2}{12} \times \frac{2}{f_s}\, df = \frac{q^2}{12} \tag{A.15}$$

As the area for the full range of the frequencies from DC to $f_s/2$ will be equal to the previously obtained value for the power of noise. However, if we filter the spectrum to only the band of interest (BW), we can also reduce the total noise power in the band, as Figure A.2 shows.

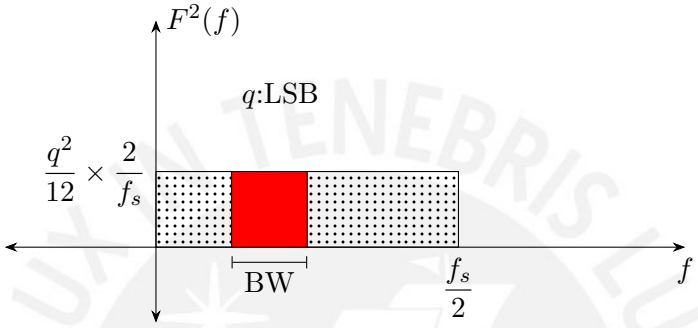


Figure A.2: Power spectral density measured from DC to $\frac{f_s}{2}$. Here, the band selected after the oversampling is represented by BW, which can contain only low frequencies. This band filtering is the basis for the benefits of oversampling. Adapted from [96].

Then, the final noise power will be:

$$\overline{F^2(t)}_{\text{BW}} = \frac{q^2}{12} \times \frac{2}{f_s} \times \text{BW} \tag{A.16}$$

and the RMS value will be:

$$\sqrt{\overline{F^2(t))}_{\text{BW}}} = \frac{q}{\sqrt{12}} \sqrt{\frac{2\text{BW}}{f_s}} \tag{A.17}$$

Calculating the SNR based on the equation shown in (A.9), it results:

$$\text{SNR} = 20 \log_{10} \frac{\text{RMS value of signal}}{\text{RMS value of noise}} = 20 \log_{10} \left[\frac{q2^N/2\sqrt{2}}{q/\sqrt{12}} \sqrt{\frac{f_s}{2\text{BW}}} \right] \tag{A.18}$$

$$\text{SNR} = 6.02N + 1.76\,\text{dB} + 10 \log_{10} \frac{f_s}{2\text{BW}} \tag{A.19}$$

The last term, is identified also as a process gain, and it adds this correction factor to the original expression of SNR. The relationship between f_s and $2BW$ is also named as the oversampling ratio (OSR) [36]:

$$\text{OSR} = \frac{f_s}{2\text{BW}} \tag{A.20}$$

Applying properties of logarithms, we can also express this relationship as:

$$\mathrm{SNR} = 6.02N + 1.76\,\mathrm{dB} + 0.5(\underbrace{20\log_{10} 2}_{6.02})\,\underbrace{\frac{\log_{10}\mathrm{OSR}}{\log_{10} 2}}_{\log_2 \mathrm{OSR}} \tag{A.21}$$

$$\mathrm{SNR} = 6.02\left[\underbrace{N + 0.5\log_2 \mathrm{OSR}}_{\mathrm{ENOB}}\right] + 1.76\,\mathrm{dB} \tag{A.22}$$

This means that when the sampling frequency is four times 2BW, the ENOB increases in 1 bit. However, taking advantage of this increase requires filtering the spectrum previously to the required in-band signal, after the oversampling process.

A.3 Level crossing sampling

The level crossing sampling consists of acquiring the samples of an input signal at almost the exact instant where it crosses a predefined set of quantization levels. It has the advantage of increasing the fidelity by taking samples at a higher rate but with fewer samples than common oversampling schemes, less power dissipation, and with a relatively more complex reconstruction process to achieve higher resolution [67].

Figure A.3 shows the basic architecture for the sampling and reconstruction process of the non-uniform sampling method. As Initially proposed, it consists of a cross-level quantizer that detects when the sample crosses a defined level and the time instant when it occurs, approximated by a timer with a frequency (f_{clk}) [97]. Although it may not be closest to the actual implementation with asynchronous comparators, it introduces the final representation when those samples must feed another digital circuit. Once the ADC takes the samples, this non-uniformly sampled signal is interpolated using polynomial functions between each sample, generating uniformed samples at a specified rate (f_{out}). Finally, as shown in the figure, it passes to a decimator to reduce to the desired conversion frequency, that could be the Nyquist frequency.

When acquiring via this method, the process introduces two error types: in amplitude and time. The first one is related to the quantization levels uncertainty, while the second one comes from the finite time resolution of the sampler. These two errors produce a total RMS error that can be expressed as [67]:

$$\sqrt{E[e_{\mathrm{total}}^2]} = \sqrt{E[e_a^2] + E[e_t^2]} \tag{A.23}$$

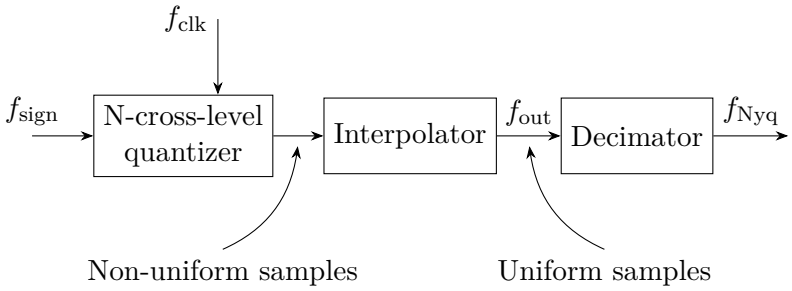


Figure A.3: Conceptual architecture for level-crossing sampling. It describes the general signal flow in this sampling scheme, obtaining first a set of non-uniform samples. Then, via a procedure of interpolation and decimation, it can be returned to the required sampling frequency, here, considered as the Nyquist frequency. Based on [67].

where $E[e_a^2]$ represents the expected quadratic error due to uncertainties in amplitude e_a, while $E[e_t^2]$ represents the expected quadratic error due to uncertainties in time e_t. It assumes that both errors are uncorrelated, which is valid for most physical signals.

From both, the most relevant is the error in time, as the samples are taken in almost exact levels but will be affected by the resolution of the timer [97], [98]. Figure A.4 shows the process of approximation. The uncertainty in time could be expressed as δ_t (segment $\overline{DB}$) where $|\delta_t| \leq \frac{t_q}{2}$, being t_q the quantized time, which is the inverse of the sampling frequency f_q. Then, it can be translated to a quantization error in amplitude (segment $\overline{DC}$) by considering it approximately equal to $s\delta_t$, where s is the slope of the signal. This slope can be obtained by the input function $v(t)$ derivative in the time t_j.

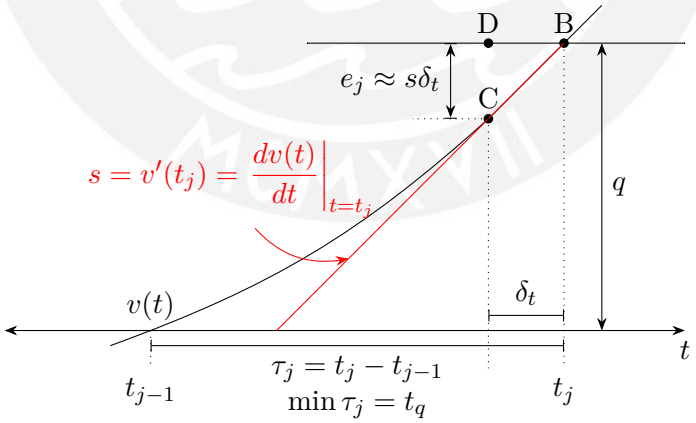


Figure A.4: Approximation of quantization noise due to variations in time in a level crossing scheme. It represents the interval between two samples in the non-uniform sampling scheme. Then the variation in the sampling time is represented as δ_t, which can be approximated to an amplitude variation employing the signal slope s. Based on [99]

Then, assuming that both, the derivative and the δ_t, are statistically independent and also the different values of δ_t are zero mean and uniformly distributed in the interval

$(-\frac{t_q}{2}, \frac{t_q}{2})$, the variance of error can be expressed as [99]:

$$E[e_t^2] = E\{\left[v'(t)\right]^2\}E\{[\delta_t]^2\} \tag{A.24}$$

For the $E\{[\delta_t]^2\}$ we can calculate it by [100]:

$$E\{[X]^2\} = \int_{-\infty}^{\infty} x^2 f(x)\, dx \tag{A.25}$$

where $f(x)$ is the probability density function of x.

In our case, δ_t is uniformly distributed in $(-\frac{t_q}{2}, \frac{t_q}{2})$; thus, the $f(x)$ equals $1/t_q$. Then:

$$E\{[\delta_t]^2\} = \frac{1}{t_q}\int_{-\frac{t_q}{2}}^{\frac{t_q}{2}} {\delta_t}^2\, d\delta_t = \frac{t_q^2}{12} \tag{A.26}$$

Replacing in the equation (A.24), the resulting expected squared error is:

$$E[e_t^2] = E\{\left[v'(t)\right]^2\}\frac{t_q^2}{12} \tag{A.27}$$

Sayiner, Sorensen, and Viswanathan [67], considers another interval for δ_t, where it is within $[0, t_q]$. It produces a different expected error value:

$$E[e_t^2] = E\{\left[v'(t)\right]^2\}\frac{t_q^2}{3} \tag{A.28}$$

This is because it also includes its mean value, which is $\frac{t_q}{2}$ and not 0 as in the previous assumption. However, as we consider here a timer whose output value could be before or after the exact one, for the posterior analysis, we will continue with (A.27).

Afterwards, for a sinusoidal input which is $v(t) = A\sin(2\pi ft)$, the corresponding slopes will be $v'(t) = 2\pi fA\cos(2\pi ft)$. Although this function is deterministic, the instant of times is variable. Then, to calculate the expected value for the $[v'(t)]^2$, we can apply [100]:

$$E\{v[X]\} = \int_{-\infty}^{\infty} v[x] f(x)\, dx \tag{A.29}$$

As the time could be considered distributed within $[-\frac{T}{2}, \frac{T}{2}]$ where $T = \frac{1}{f}$. Assuming all times in the interval equally probable, the distribution could be uniform with a density of probability equal to $\frac{1}{T}$. Applying to the sinusoidal function:

$$E\{[v'(t)]^2\} = \int_{-\infty}^{\infty} [2\pi fA\cos(2\pi ft)]^2 f(t)\, dt \tag{A.30}$$

$$E\{[v'(t)]^2\} = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} [2\pi f A\cos(2\pi f t)]^2\, dt$$
$$= \frac{(2\pi f A)^2}{T} \times \frac{1}{2}\int_{-\frac{T}{2}}^{\frac{T}{2}} 1+\cos(4\pi f t)\, dt$$
$$= \frac{(2\pi f A)^2}{T} \times \frac{1}{2}\left[t + \frac{\sin(4\pi f t)}{4\pi f}\right]\Bigg|_{-\frac{T}{2}}^{\frac{T}{2}}$$
$$E\{[v'(t)]^2\} = 2(\pi f A)^2 \tag{A.31}$$

Replacing this result in (A.27), we can obtain that:

$$E[e_t^2] = 2(\pi f A)^2 \frac{t_q^2}{12} = \frac{1}{6}(\pi A)^2 \left(\frac{f}{f_q}\right)^2 \tag{A.32}$$

The literature [67], [97], [98] usually names the factor f_q/f as R which is called the timer resolution ratio. Compared with the previously defined OSR (A.20), it will be the half value:

$$\text{OSR} = \frac{R}{2} \tag{A.33}$$

Neglecting the $E[e_a^2]$, the total RMS value for error can be obtained by combining (A.23) and (A.32):

$$\sqrt{E[e_{\text{total}}^2]} \approx \sqrt{E[e_t^2]} = \frac{1}{\sqrt{6}}\pi A \frac{1}{R} \tag{A.34}$$

Now, applying the formula for SNR in (A.9), for a sinusoidal input we can obtain:

$$\text{SNR} = 20\log_{10}\frac{\text{rms value of signal}}{\text{rms value of noise}} = 20\log_{10}\left[\frac{A/\sqrt{2}}{\pi A/(\sqrt{6}R)}\right] \tag{A.35}$$
$$= 20\log_{10}(\frac{\sqrt{3}}{\pi}) + 20\log_{10} R \tag{A.36}$$
$$\text{SNR} = 20\log_{10} R - 5.17\,\text{dB} \tag{A.37}$$

Other references [97], [98] obtains a different constant value (11.2 dB instead of 5.17 dB), however, it is due the different interval for the δ_t, as commented for (A.28).

Either case, which is more relevant for the analysis is that, considering negligible variations in reference levels for quantization, the SNR only depends on the resolution of the timer. On the other hand, while the conventional oversampling scheme, whose formulas (A.19) and (A.22) show an increment of 3 dB each time the sampling frequency is twice the bandwidth, which is equivalent to 0.5 bit, in the level crossing scheme, the increment is 6 dB which means 1bit [98]. In other words, with less increment of resolution ratio in the level-crossing it could achieve more increment on SNR than in the conventional oversampling.

Some important findings obtained by Sayiner, Sorensen, and Viswanathan [67], is that:

- The process accuracy of level crossing sampling is determined by three factors: the number of quantization levels, the amplitude resolution, and the time resolution.
- If R were infinity, the SNR would increase by 15dB each time the quantization levels is doubled. However, if there is quantization in time, a certain R imposes a threshold where the SNR cannot be higher despite increasing the quantization levels.
- The threshold for quantization levels increases with increasing the R.
- If the accuracy of resolution is L, with:

$$L = -\log_2(2\delta_a) \tag{A.38}$$

  where δ_a is the variation in amplitude for the quantization levels. Each increment of L will increase in 6 dB for R infinity. Whereas with time quantization, the SNR will not increase beyond a threshold.
- With no level uncertainty (L is infinity), the SNR increases about 6 dB, each doubling the value of R, confirming the tendency shown in (A.37).
- The order of the interpolator has not too much impact improving the SNR. Low order values, such as 2 or 3, are enough to reach the maximum SNR level for a determined value of R.
- The increase of SNR due to the decimation factor (DF) will depend on how much the frequency of interest is close to the signal bandwidth. As long as they are closest, it will favor a greater value of DF and, consequently, increase the value of the resolution, although limited by R.

A.4 Coherent sampling

The FFT is a helpful tool employed to obtain the discrete spectral representation of a signal. This representation is constituted by samples of the actual spectrum, thus, it should be used carefully to obtain properly results. When analyzing ADCs performance we use the FFT to obtain the frequency components of the output signal, and from it, get the SNR, SNDR, and other measurements [101]. Typically, the test is performed with a sinusoidal input with a known frequency. However, this frequency must be sufficiently low to capture the information on transition levels and make dynamic errors negligible [102]. Likewise, the recorded signal must have an integer number of cycles, being also prime

to the number of samples. This guarantees that the records are uniformly distributed in phases from 0 to 2π.

The technique that collects these conditions is called coherent sampling [101]. The noticeable effect in the spectrum is that the fundamental component and its harmonics fall in single frequency bins. The rule to achieve this can be summarized in the following formula:

$$\frac{f_{in}}{f_s} = \frac{M_C}{M} \tag{A.39}$$

where f_{in} is the frequency of the input sine wave, f_s is the sampling frequency, M_C is the number of complete sampled cycles of the input signal, and M is the number of samples which is a power of two.

If we do not follow this rule, the output will have the fundamental and harmonics in adjacent bins, which will conduce to measurement errors. For this, other strategies include using windows to minimize the effects of spectral leakage [103].

When obtaining the spectral representation of the outputs from the designed ADCs, we will use this technique to facilitate the calculation of the SNR, both via a software algorithm (for interpolated signals) and via the included tools from Cadence.

A.5 Figure of merit

The FOM is a quantity used to evaluate the performance of an ADC. Walden [104], in his survey, presents two FOMs used to measure it. The first FOM includes the product of the ENOB times the sample rate:

$$\mathrm{FOM}_1 = 2^{\mathrm{ENOB}} \times f_s \tag{A.40}$$

The second FOM takes into account the power dissipation and can be written as:

$$\mathrm{FOM}_2 = \frac{\mathrm{P}_{\mathrm{dissipation}}}{2^{\mathrm{ENOB}} \times f_s} \tag{A.41}$$

From it, Geelen [105] defined a slightly different form which considers the effective resolution bandwidth (ERBW):

$$\mathrm{FOM}_2 = \frac{\mathrm{P}_{\mathrm{dissipation}}}{2^{\mathrm{ENOB}} \times 2 \times \mathrm{ERBW}} \tag{A.42}$$

Later, Draxelmayr [106] included a variation for this formula which considers the sinusoidal input frequency f_{in} instead of the ERBW:

$$\mathrm{FOM}_2 = \frac{\mathrm{P}_{\mathrm{dissipation}}}{2^{\mathrm{ENOB}} \times 2 \times f_{in}} \tag{A.43}$$

For other forms of this parameter of performance, Jonsson [107] presents an interesting list which includes the original one and the different variations employed to compare ADC proposal designs in the literature.

Depending on the case, we will use any of the three forms of FOM_2 (A.41), (A.42) y (A.43). The first one for the uniform sampling scheme in the SAR ADC as we know the sampling frequency (f_s), while the two other forms will be applied to the non-uniform scheme in the LCADC. In addition, for simplicity, we will refer to the FOM_2 as just FOM.

Appendix B

Project repositories

Across the development process of this thesis, we have created some repositories that hold the designs and software we have used to test the neural recording signals. These repositories can be useful to reproduce the results we presented in this document and use them as a starting point for further developments.

B.1 Software development

- **eappprocessor** – `https://github.com/LuighiV/eapprocessor`.
  This repository contains all the software models developed to generate and evaluate the detection of spikes with recordings, converted signals, and NEO. Apart from the Python package, which we named `eappprocesor`, there are many Jupyter Notebooks that contain the results for each of our performed evaluations. Those notebooks are ordered according to the flow of development followed by the thesis. The reader can follow that suggested order to find the results of interest or reproduce them.

- **extract-signal** – `https://github.com/LuighiV/extract-signal`.
  Here, we developed a simple script to pre-process results obtained from Cadence to delete duplicated samples, especially for the level-crossing scheme. It is intended to run directly on the server. For that reason, it avoids using third-party libraries to prevent compatibility issues with different versions or installing them when we don't have access to it.

B.2 Circuit design

- **lcadc-design** – `https://github.com/LuighiV/lcadc-design`.

  This repository contains the database of the hardware design developed on Cadence. The main lib is `tesis2020`, which contains all the blocks used in both ACDs and the corresponding testbenches for them. Although the name only mentions the LCADC, it also contains the SAR design, the NEO, and the sample analog frontend.

B.3 Third-party tools

- **MEArec** – `https://github.com/SpikeInterface/MEArec`.

  As commented in the document, our main software uses a third-party tool for the neural recording generation which is called `MEArec`. We are also happy to have added some snippets of code to facilitate the integration with our software and to be recognized as contributors to the main repository.

Appendix C

Capacitance error propagation to DAC output

From the Monte Carlo simulation, we obtained a standard deviation for the capacitance variation of 0.054% relative to the nominal value. This variation is propagated to the output voltage obtained in the DAC, as it depends on the relation between the array of capacitors controlled by the DAC logic.

Recalling the general relationship for the DAC output V_{out}, it can be expressed as:

$$V_{out} = V_{REF}\frac{C_U}{C_D + C_U} \tag{C.1}$$

Where $C_D + C_U = 256C$ and C is the unit capacitor, we used to build the array.

According to Ku [108] and the National Institute of Standards and Technology [109], the error propagation for a function that depends on non-correlated variables is equal to:

$$\Delta V = \sqrt{\left(\frac{\partial V}{\partial X}\right)^2 (\Delta X)^2 + \left(\frac{\partial V}{\partial Y}\right)^2 (\Delta Y)^2 + \left(\frac{\partial V}{\partial Z}\right)^2 (\Delta Z)^2 + \ldots}, \quad V = f(X, Y, Z, \ldots) \tag{C.2}$$

Applying it to (C.1), we can obtain the following:

$$\Delta V_{out} = \sqrt{\left(\frac{\partial V_{out}}{\partial V_{REF}}\right)^2 (\Delta V_{REF})^2 + \left(\frac{\partial V_{out}}{\partial C_U}\right)^2 (\Delta C_U)^2 + \left(\frac{\partial V_{out}}{\partial C_D}\right)^2 (\Delta C_D)^2} \tag{C.3}$$

If we want only the variation due to capacitors, we can consider $\Delta V_{REF} = 0$. Then, the terms of the variation can be expressed as:

$$\left(\frac{\partial V_{out}}{\partial C_U}\right)^2 (\Delta C_U)^2 = (V_{REF})^2 \frac{(C_D)^2}{(C_D + C_U)^4} (\Delta C_U)^2 \tag{C.4}$$

$$\left(\frac{\partial V_{out}}{\partial C_D}\right)^2 (\Delta C_D)^2 = (V_{REF})^2 \frac{(C_U)^2}{(C_D + C_U)^4} (\Delta C_D)^2 \tag{C.5}$$

Replacing (C.4) and (C.5) in (C.3), we can obtain:

$$\Delta V_{out} = \frac{V_{REF}}{(C_D + C_U)^2}\sqrt{(C_D\Delta C_U)^2 + (C_U\Delta C_D)^2} \tag{C.6}$$

Taking $C_U = kC$ and $C_D = (256-k)C$ we will have the variations equal to $\Delta C_U = \sqrt{k}\Delta C$ and $\Delta C_D = \sqrt{256-k}\Delta C$. Replacing in the previous expression, we obtain:

$$\Delta V_{out} = \frac{V_{REF}}{(256)^2}\frac{\Delta C}{C}\sqrt{(256-k)^2k + k^2(256-k)} = \frac{V_{REF}}{(256)^2}\frac{\Delta C}{C}\sqrt{256}\sqrt{(256-k)k} \tag{C.7}$$

To obtain the maximum value of this variation, we can calculate the derivative of the term inside $\sqrt{k(256-k)}$ and obtain the k value that makes it zero:

$$\frac{\mathrm{d}}{\mathrm{d}k}\{k(256-k)\} = 256 - 2k = 0, \quad \Rightarrow k = 128 \tag{C.8}$$

This value corresponds to when the output targets $V_{REF}/2$. Replacing the result in (C.8) in (C.7), we can obtain:

$$\Delta V_{out} = \frac{V_{REF}}{(256)^2}\frac{\Delta C}{C}\sqrt{256}(128) = \frac{V_{REF}}{256}\frac{\Delta C}{C}(8) = 8\frac{\Delta C}{C}\mathrm{LSB} \tag{C.9}$$

According to our simulations, $\Delta C/C = 0.54\% = 0.0054$, then the output variation in LSB units will be:

$$\Delta V_{out} = 8(0.0054)\mathrm{LSB} = 0.0432\mathrm{LSB} \tag{C.10}$$

This result is lower than 10% of the limit value (0.5LSB) to avoid introducing a code error in the result. Hence, we consider the value for the capacitance adequate for this DAC at the specified resolution.

Just for reference, we can generalize the expression (C.9) in terms of the resolution N, replacing 256 by 2^N in (C.8) in (C.7). Doing that, we could obtain a $k = 2^{N-1}$, then the result will be:

$$\Delta V_{out} = \frac{V_{REF}}{(2^N)^2}\frac{\Delta C}{C}\sqrt{2^N}(2^{N-1}) = \frac{\sqrt{2^N}}{2}\frac{\Delta C}{C}\mathrm{LSB} \tag{C.11}$$

This expression shows that as long as we increase the resolution, the mismatch on capacitors will affect the voltage output more.