UCLouvain

École polytechnique de Louvain

Hardware Implementation of Unlimited Sampling for the Analog-to-Digital Conversion of High Dynamic Range Signals

Author: **Jade LEEMANS**
Supervisors: **Denis FLANDRE**
Readers: **Martin ANDRAUD, Sylvain FAVRESSE , Thomas FEUILLEN**
Academic year 2025-2026
Master [120] in Electrical Engineering

Acknowledgements

Before anything else, I would like to sincerely thank Prof. Denis Flandre, who has been a true inspiration throughout my studies and whose courses sparked my strong interest in analog electronics. When first looking through the list of Master's thesis topics, I was afraid that I would not find a project focused on analog electronics and real hardware implementation, the fields I was most passionate about. Thankfully, you gave me the freedom to shape this project in that direction and allowed me to design and implement a completely new hardware solution on a PCB. Thank you for sharing your expertise throughout this project and for introducing me to a new dimension of engineering work that has been both challenging and rewarding.

I would also like to express my sincere gratitude to Dr. Thomas Feuillen for helping me understand all the nuances behind modulo theory. Even after reading the papers, many concepts still remained unclear to me, and your explanations were always incredibly valuable. Thank you as well for your help with the implementation of the reconstruction algorithm, which was especially appreciated given my sometimes complicated relationship with coding. Beyond the technical aspects, your advice and shared experiences were also tremendously helpful in introducing me to the professional world in Luxembourg.

I would also like to thank Sylvain Favresse, who was a great help in understanding the practical and technical challenges involved in translating theory into a real hardware implementation and evaluating its performance. Thank you as well for your patience in helping me become familiar with all the equipment in the Welcome laboratory, and especially for reassuring me during the PCB soldering stages, when I was genuinely afraid of damaging everything because I was so stressed.

I am deeply grateful to the three of you for your flexibility and understanding while I was on the other side of the world in Australia. Thank you for adapting meeting schedules to the time difference since August, and for your patience during periods when my internet connection was unreliable and communication could only happen through email.

I would also like to thank Prof. Martin Andraud for accepting to be a member of the jury and for taking the time to read and evaluate nearly a full year of work.

I will forever be grateful to my dad. Without him, I would never have pursued these studies. You have been my inspiration since I was little, when I used to jokingly say that one day I would become an electrical engineer, and now I can truly say that I did it. Thank you for always taking the time to share your knowledge with me, for reassuring me whenever I doubted my work, and for patiently reviewing what I had done simply to help me gain confidence. I would not have been able to accomplish this without you.

Finally, I cannot forget to thank my boyfriend, Martin Clotuche, with whom I had the chance to experience the exchange program in Australia. Thank you for being there whenever my anxiety became overwhelming, for trying to help me when I was stuck with simulations or code, and for driving countless hours so that I could continue working while traveling. Thank you for helping me balance discovering the other side of the world with keeping up with my thesis schedule, sometimes even at the expense of your own work. Without you, I most likely would never have gone to Australia, and I therefore would never have had the opportunity to follow the PCB design course at The University of Melbourne, which proved to be an invaluable help for the design of the PCB developed during this work. I will also forever be grateful for these studies, without which I would never have met you.

Once again, my deepest gratitude goes to Prof. Denis Flandre, Dr. Thomas Feuillen, and Sylvain Favresse, without whom this work would never have existed and would never have become such a meaningful and memorable experience.

Jade Leemans, 2025-2026.

Abstract

Nearly every device interacting with real-world information relies on an Analog-to-Digital Converter (ADC), whose role is to convert analog signals into discrete binary values. However, when the input signal exceeds the limited dynamic range of the ADC, saturation occurs and the signal becomes clipped, resulting in a significant loss of information.
To address this limitation, the Unlimited Sampling (US) framework has emerged in recent years. By combining a Modulo-ADC with dedicated reconstruction algorithms, the input signal can be folded into a fixed dynamic range compatible with the ADC operating range, while still allowing recovery of the original signal from its modulo samples.

This work investigates a hardware implementation of a modulo-ADC architecture with the objective of maximizing both the allowable input dynamic range and the maximum input frequency.
Existing implementations proposed in the literature are first reviewed in order to design a new hardware architecture acting as a front-end stage placed before a conventional ADC. The proposed system relies on comparators that monitor the signal amplitude and detect threshold crossings, while a dedicated control block generates the correction signals required to wrap the input back into the desired range whenever necessary. The implementation of each block is described in detail, including the underlying theory, component selection and sizing, as well as LTspice simulations.

Once the PCB is fabricated and assembled, the functional behavior of the prototype is verified experimentally and several performance measurements are conducted.
For this first prototype, the input range is limited to 9.2V peak-to-peak due to the supply voltage constraints of the selected components. With folding thresholds fixed to ± 1V, the system is capable of performing up to two folds while enabling successful post-processing reconstruction of the original signal. The measured folding delay is approximately 200ns, allowing maximum input frequencies ranging from 75kHz for a 9V peak-to-peak input signal up to 270kHz for a 3V peak-to-peak input signal, while maintaining a folded output contained within the interval $[-1.25, 1.25]$V.
Finally, several improvements are proposed to further increase both the achievable dynamic range and the maximum operating frequency, mainly through the replacement of certain components.

The obtained results confirm the feasibility of a hardware implementation of a modulo-ADC for the acquisition and processing of high-dynamic-range signals, while also highlighting challenges related to propagation delays and supply voltage limitations.

Acronyms

ADC Analog-to-Digital Converter

AGND Analog Ground

BNC Bayonet Neill-Concelman

CPLD Complex Programmable Logic Device

CRT Chinese Remainder Theorem

DFT Discrete Fourier Transform

DGND Digital Ground

DNL Differential nonlinearity error

DVG Digital Voltage Generator

EMI Electromagnetic Interference

ENOB Effective Number Of Bits

FFT Fast Fourier Transform

FPGA Field Programmable Gate Array

FSM Finite State Machine

HDR High Dynamic Range

I/V Current to Voltage

ICs Integrated Circuits

INL Integral nonlinearity error

LDR Low Dynamic Range

LSB Least Significant Bit

LVCMOS Low-Voltage Complementary Metal-Oxide-Semiconductor

MCU Micro-Controller Unit

MEMS Micro-Electromechanical Systems

MUX Multiplexer

PCB Printed Circuit Board

PSD Power Spectral Density

PVG Programmable Voltage Generator

S/H Sample-and-Hold

SINAD Signal-to-Noise and Distortion Ratio

SNR Signal-to-Noise Ratio

SRER Signal-to-Reconstruction Error Ratio

STs Schmitt triggers

UDR Unlimited Dynamic Range

US Unlimited Sampling

V/I Voltage to Current

VCRO Voltage-Controlled Ring Oscillator

List of Figures

Contents

1

Introduction

Over the last few decades, the demand for data acquisition and signal processing has increased significantly. The rapid expansion of digital communication technologies, including mobile networks, satellite systems, and the Internet, has led modern systems to rely increasingly on efficient information processing. Furthermore, the widespread integration of sensors in applications such as automotive systems, medical devices, and the rapidly expanding Internet of Things (IoT) has resulted in a substantial increase in the amount of generated and processed data.
Most information originating from the physical world exists as continuously varying signals, whose amplitudes change continuously over time. Such signals are referred to as analog signals.
In contrast, modern processing systems operate using discrete binary representations.
Consequently, before real-world information can be processed digitally, analog signals must first be converted into a digital representation.

This conversion is performed by an Analog-to-Digital Converter (ADC), which serves as an interface between the analog and digital domains [1].
As illustrated in Figure 1, the analog-to-digital conversion process can be divided into three main stages. First, the analog signal is sampled at discrete time instants, resulting in a discrete-time signal with continuous amplitude values. The sampled signal is then quantized into a finite number of amplitude levels, creating a discrete-amplitude representation. Finally, these quantized values are encoded into binary words, producing a fully digital signal.

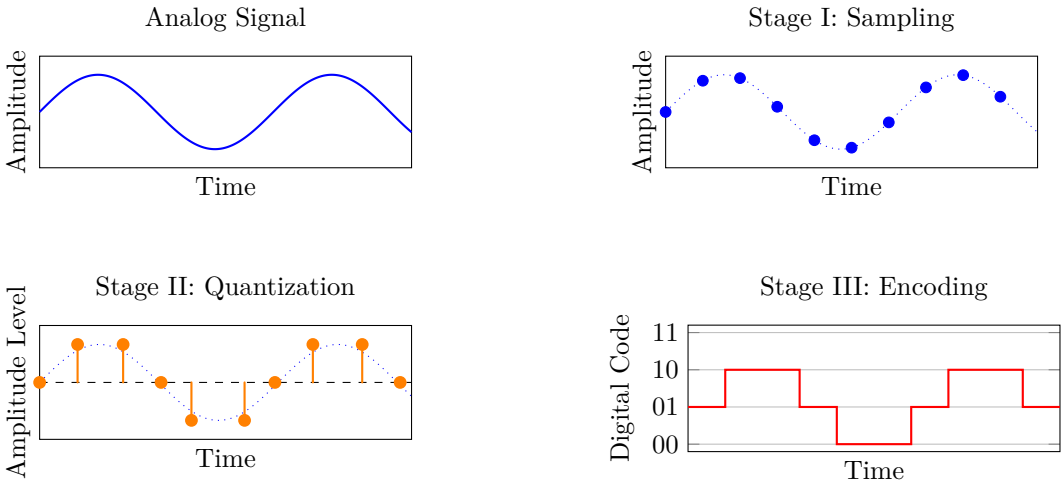


Figure 1: Illustration of the analog-to-digital conversion process

Emerging technologies increasingly require analog-to-digital conversion systems that are fast, energy-efficient, low-cost, and highly accurate.
Although continuous developments in ADC technologies have considerably improved signal acquisition capabilities, one fundamental limitation still restricts their performance.
As described in [2] and [3], the maximum signal amplitude that an ADC can process, known as its input dynamic range, is inherently limited by architectural and power constraints.
Similar limitations exist in other electronic systems. For example, an operational amplifier cannot generate output voltages exceeding its supply rails without entering saturation.
Likewise, when an input signal exceeds the ADC dynamic range, the ADC saturates and the signal gets clipped. Such clipping introduces distortion and permanently removes information from the signal.

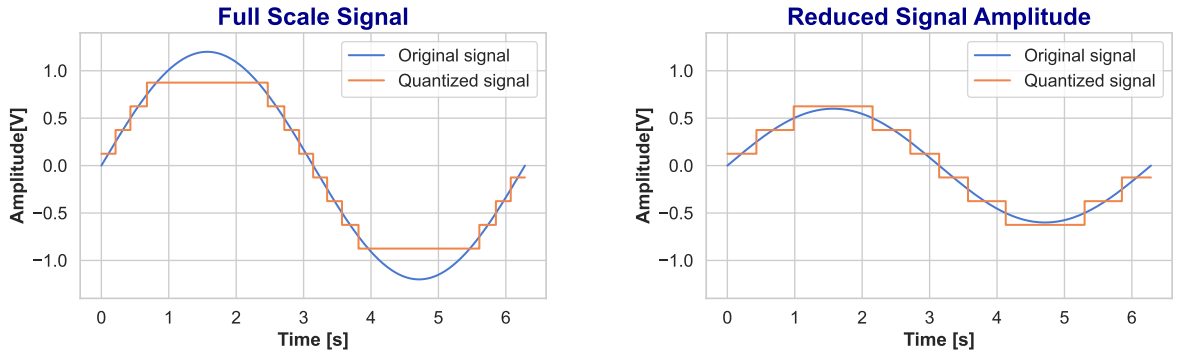
A large dynamic range is essential in many applications where both weak and strong signals must be captured simultaneously. Examples include image sensors recording scenes containing both dark shadows and bright highlights, or radar systems that must detect distant weak reflections while simultaneously receiving strong nearby signals.

One straightforward approach to avoid saturation consists of reducing the signal amplitude prior to conversion so that it remains within the operating range of the ADC.
While this prevents clipping, it introduces another limitation associated with the finite number of quantization levels available in ADCs.
As illustrated in Figure 1, an ADC can only generate a finite number of output levels by dividing its input range into discrete quantization levels. The number of available quantization levels is determined by the ADC resolution, which specifies how finely the input signal can be represented.
Consequently, reducing the signal amplitude causes fewer quantization levels to be effectively used during conversion, thereby increasing the relative quantization error.
Quantization error arises because continuous signal amplitudes are approximated using discrete amplitude levels. When only a small portion of the ADC range is used, each quantization step represents a larger fraction of the signal amplitude, resulting in reduced precision.
This limitation can be particularly problematic in applications requiring the detection of small signal variations. For example, in biomedical signal acquisition systems such as ECG measurements, large quantization errors may obscure subtle signal features containing clinically relevant information.

For illustration, Figure 2 presents two quantized signals obtained using an ADC with eight quantization levels and an input range of [-1,1]V. Since the total dynamic range is 2V, dividing this range into eight levels results in a quantization step size of:

$$\Delta = \frac{2V}{8} = 0.25V \tag{1}$$

The first case corresponds to a signal with an amplitude of 1.2V. Since this exceeds the ADC range, the signal reaches the limits of the converter and spans the full available input range before saturation occurs, therefore using all eight quantization levels.
In the second case, the signal amplitude is reduced by a factor of two, resulting in an amplitude of 0.6V. The signal now varies only between -0.6V and +0.6V and therefore occupies a smaller portion of the ADC input range. As a result, the extreme quantization levels corresponding to the regions [-1,-0.75]V and [0.75,1]V are never reached. Consequently, only six out of the eight available quantization levels are effectively used during conversion. Because the same signal information is represented using fewer quantization levels, the quantization error may become larger relative to the signal amplitude, reducing conversion precision.



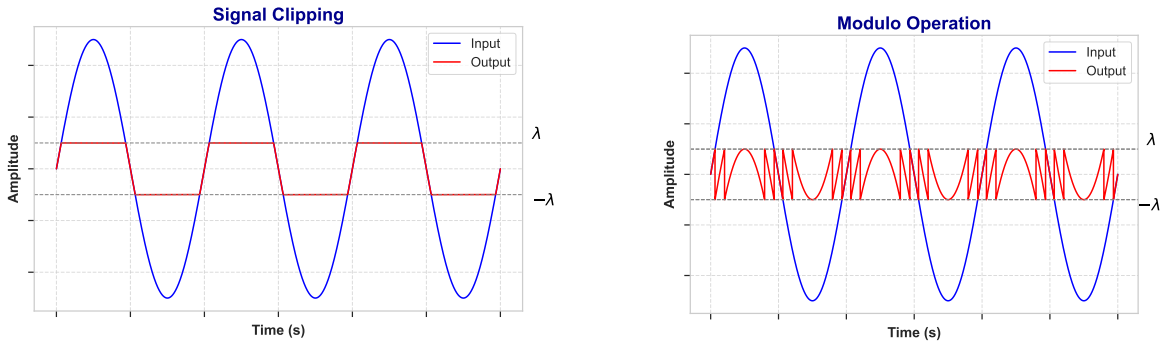
(a) Clipped full-scale signal

(b) Signal with reduced amplitude

Figure 2: Comparison of quantized signals for a clipped full-scale input and a reduced-amplitude input

Conventional ADC architectures therefore face a fundamental trade-off: avoiding saturation often requires sacrificing quantization precision, while preserving precision increases the risk of clipping large-amplitude signals. To overcome these limitations, alternative signal acquisition techniques have been investigated.

One promising approach introduced in recent years is the Unlimited Sampling (US) Theory [2, 3, 4, 5, 6, 9], which relies on the modulo operation. In mathematics, the modulo operation maps values exceeding a predefined interval back into that interval through a wrapping process. A similar principle is employed in Modulo-ADCs.
Unlike conventional ADCs, where signal amplitudes exceeding the available range are clipped, modulo ADCs apply a modulo operation that folds the signal back into a predefined interval rather than discarding information.
The Figure 3 illustrates the difference between a conventional ADC and a Modulo-ADC, where in the case of the moulo-ADC, the signal is wrapped in a defined range so no information is lost and all quantization levels are used.



(a) Conventional ADC (b) Modulo-ADC

Figure 3: Comparison between a clipped signal and a folded signal

The development of the US framework has led to significant research efforts aimed at designing Modulo-ADC architectures capable of folding signals exceeding the ADC threshold, together with efficient reconstruction algorithms able to recover the original signal from its folded representation.
As a result, previous works have demonstrated the capability to process signals with amplitudes significantly larger than the nominal ADC range, with extensions up to approximately sixty times the threshold value. Furthermore, improved robustness against distortion has been reported due to the more efficient use of the ADC resolution [4].

Despite these advantages, practical hardware implementation remains challenging, and existing implementations proposed in the literature still exhibit several limitations.
For instance, some previously proposed Modulo-ADC architectures [5, 11] rely on counters and Digital-to-Analog Converters (DACs), which perform the inverse operation of ADCs by converting digital signals into analog signals. However, these additional components introduce practical constraints. In particular, finite DAC resolution limits the number of achievable folds and consequently restricts dynamic range extension capabilities. Moreover, adapting the folding threshold for different applications often requires substantial modifications to the hardware design.
Other architectures simplify signal reconstruction by encoding additional information, such as the number of folds [6]. However, this can increase the hardware complexity.
Certain implementations also rely on Sample-and-Hold (S/H) circuits prior to the modulo operation [6]. In these designs, the S/H circuit must maintain the sampled value during both folding and quantization processes, resulting in significantly longer hold times than in conventional ADC systems.
Consequently, reported implementations have demonstrated signal folding capabilities only up to approximately 300Hz while requiring input amplitudes to remain below approximately three times the ADC dynamic range. Such limitations are insufficient for numerous practical applications, including audio processing systems where signal frequencies typically range from 20Hz to 20kHz.

The objective of this work is therefore to investigate existing theoretical Modulo-ADC architectures proposed in the literature and evaluate their practical implementations when available. Based on this analysis, this work aims to design a new Modulo-ADC hardware implementation that is simple, fast, and adaptable to various application requirements.

This Master's thesis is organized as follows.
Chapter 2 introduces the Unlimited Sampling (US) Framework, its assumptions and conditions, followed by the theoretical background of the reconstruction algorithms used to recover original signals.
Several Modulo-ADC architectures proposed in the literature are then presented, together with their advantages, limitations, and reported performances.
The architecture selected for this work, based on [6], is subsequently described in detail with emphasis on the challenges associated with hardware implementation.

Chapter 3 focuses on hardware implementation aspects.
First, the implementation proposed in [6] is analyzed and its limitations are highlighted.
The specifications of the architecture developed in this work are then defined with the objective of maximizing both input frequency and input dynamic range.
A global circuit analysis is presented, followed by a detailed study of each individual block including theoretical background, component selection, and LTspice simulations.

Finally, Chapter 4 evaluates the performance of the proposed hardware implementation through several experimental analyses, including basic functionality validation, maximum frequency characterization, and noise and resolution assessments. The chapter concludes with potential improvements for future developments.

2

Theoretical Foundations and Architectures of Modulo ADCs

Nowadays, in most applications such as telecommunications, audio processing, radar systems, and many other fields, analog-to-digital converters are essential for information processing.
Consequently, the design of ADCs has become a crucial aspect of modern signal processing systems, with particular emphasis on mitigating their inherent limitations, such as the finite dynamic range. Moreover, while extending the ADC input range helps avoid saturation, it does not improve overall performance, since it results in reduced digital resolution due to a larger quantization interval.

To address these challenges, numerous studies have led to the development of the US Theory. Within this framework, a modulo operation is employed to fold the signal whenever it exceeds the ADC thresholds. Based on this principle, several mathematical algorithms have been proposed to recover the original signal from the folded samples. This approach enables the processing of High Dynamic Range (HDR) signals using Low Dynamic Range (LDR) ADCs, while minimizing information loss and signal distortion.

This chapter first introduces the US theory and the modulo operation.
It then presents a widely used reconstruction algorithm, which recovers the original signal solely from the modulo samples. Next, several modulo-ADC architectures proposed in the literature are reviewed, together with their advantages and challenges.
Finally, the selected architecture is described in detail, followed by a behavioral simulation in Matlab Simulink to validate its operation and highlight potential limitations and implementation challenges.

2.1 Unlimited Sampling Framework

The US theory is based on a co-design approach that combines a Modulo-ADC architecture with an associated signal reconstruction algorithm.
The following section introduces the fundamental concepts of the US framework, including the modulo folding operation and the main properties of modulo signals that enable reconstruction of the original signal from the folded samples.

2.1.1 Modulo Acquisition and Folding Mechanism

A Modulo-ADC is designed to handle input signals that may exceed the ADC's dynamic range. When the input signal goes beyond this range, the expected output is the digitized version of the folded signal. Conversely, when the input signal remains within the dynamic range, the output should simply be the digitized version of the original signal, without any folding.

To fold the signal into a predefined range $[-\lambda, \lambda]$, where λ denotes the threshold of the ADC, the following mapping is applied [4]:

$$M_\lambda : f \mapsto 2\lambda \left(\left[\!\!\left[\frac{f}{2\lambda} + \frac{1}{2} \right]\!\!\right] - \frac{1}{2} \right), \; [\![f]\!] := f - \lfloor f \rfloor \tag{2}$$

Here, $\lfloor f \rfloor$ denotes the *floor function*, defined as the greatest integer less than or equal to f. The quantity $[\![f]\!]$ represents the fractional part of f, i.e., the difference between f and its floor.
The mapping $M_\lambda(f)$ performs a centered modulo operation with period 2λ. In other words, it subtracts an appropriate multiple of 2λ from the input f so that the result lies within the interval $[-\lambda, \lambda]$. Whenever f exceeds this interval, it is "wrapped" back into it rather than being clipped.

According to the US Theorem, a signal can be perfectly recovered using only its modulo samples, without requiring any side-information such as the number of folds or the sign of the fold. This property is crucial when selecting an appropriate hardware architecture, as some designs generate additional information beyond the modulo samples themselves. While this extra information may facilitate reconstruction, it also increases circuit complexity.

More specifically, the US Theorem can be summarized as follows [2]:
Let $g(t)$ be a finite-energy bandlimited signal with maximum angular frequency $\Omega = 2\pi f$, and let $y[k]$, $k \in \mathbb{Z}$, be the modulo samples of $g(t)$ taken with sampling period T. A sufficient condition for perfect recovery of $g(t)$ from $\{y[k]\}_k$ (up to an additive multiple of 2λ) is

$$T < \frac{1}{2\Omega e} \tag{3}$$

where e denotes Euler's constant.

This condition is more restrictive than the classical Nyquist sampling criterion. For a signal with highest frequency component $f_{\max}$, the Nyquist condition requires:

$$f_s = \frac{1}{T} \geq 2f_{\max} \tag{4}$$

In contrast, the Unlimited Sampling framework requires additional oversampling. Since $\Omega = 2\pi f_{\max}$, the sampling frequency must satisfy:

$$f_s = \frac{1}{T} \geq 2\Omega e \tag{5}$$

Thus, an additional oversampling factor proportional to $2\pi e$ is required to guarantee perfect reconstruction. It is important to note that this condition is sufficient but not necessary, and successful recovery may still be possible at lower sampling rates.

2.1.2 Reconstruction from Modulo Samples

The recovery of the original signal from modulo samples relies on exploiting the structure introduced by the folding operation. A key insight is provided by the Modular Decomposition Property [4], which states that any signal $g(t)$ can be decomposed as:

$$g(t) = y(t) + \epsilon_g(t) \tag{6}$$

where $y(t) = M_\lambda(g(t))$ is the folded signal, and $\epsilon_g(t)$ is a residual discontinuous function that takes values in $2\lambda\mathbb{Z}$. This residual term accounts for the multiples of 2λ that are removed by the modulo operation and is piecewise constant, with jumps occurring at the folding instants.

An example for an arbitrary function $g(t) = 4\sin(2\pi \times 3t) + 0.5\sin(2\pi \times 12t)$ is shown in Figure 4, where the threshold λ is fixed to 1. In this case, the residual function $\epsilon_g(t)$ takes values that are integer multiples of $2\lambda = 2$.

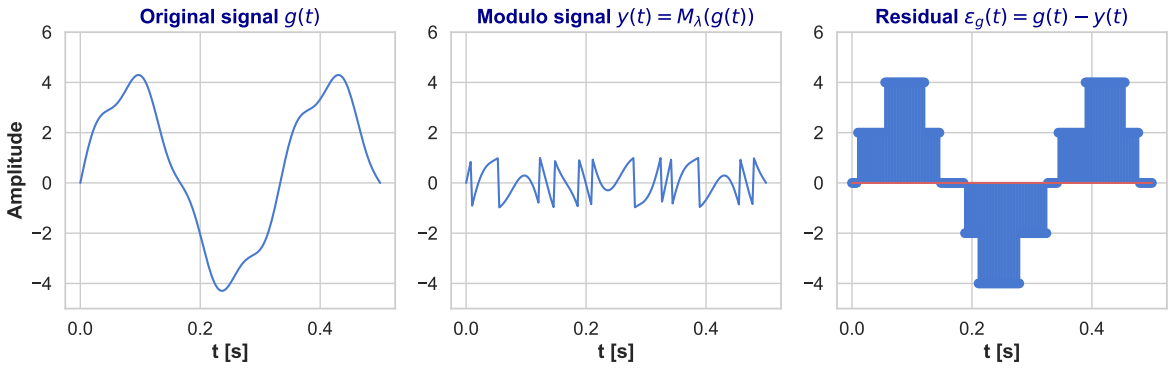


Figure 4: Python Simulation: Modular Decomposition Property

Under this decomposition, reconstructing $g(t)$ from $y(t)$ is equivalent to estimating the residual function $\epsilon_g(t)$, which captures the ambiguity introduced by the modulo operation.
This structure is the key for reconstruction when applying finite-difference operators.

The first-order finite difference is defined as:

$$(\Delta y)[k] \triangleq y[k+1] - y[k] \tag{7}$$

and higher-order differences are obtained recursively as:

$$\Delta^N y = \Delta^{N-1}(\Delta y) \tag{8}$$

Applying the finite-difference operator to the decomposition yields:

$$\Delta^N g = \Delta^N y + \Delta^N \epsilon_g \tag{9}$$

Since $\epsilon_g(t)$ takes values in $2\lambda\mathbb{Z}$ and is piecewise constant, its finite differences also take values in $2\lambda\mathbb{Z}$. Therefore, applying the modulo operator removes this term:

$$M_\lambda(\Delta^N \epsilon_g) = 0 \tag{10}$$

Applying the modulo operator to (9) gives:

$$M_\lambda(\Delta^N g) = M_\lambda(\Delta^N y) \tag{11}$$

Furthermore, it can be shown [4] that if the sampling period satisfies $T < \frac{1}{\Omega e}$, then the finite differences of $g(t)$ are bounded as:

$$\|\Delta^N g\|_\infty < \lambda \tag{12}$$

Under this condition, $\Delta^N g$ lies within the interval $[-\lambda, \lambda]$, and therefore:

$$M_\lambda(\Delta^N g) = \Delta^N g \tag{13}$$

Combining the above results, we obtain:

$$\Delta^N g = M_\lambda(\Delta^N y) \tag{14}$$

which shows that the finite differences of the original signal can be recovered from the modulo samples.

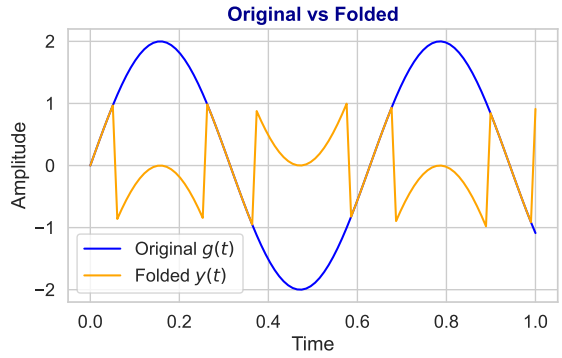
The original signal can then be reconstructed by inverting the finite-difference operator, which corresponds to discrete summation. For the first-order case, this is given by:

$$S : \{s[k]\}_{k\in\mathbb{Z}^+} \mapsto \sum_{m=1}^{k} s[m] \tag{15}$$

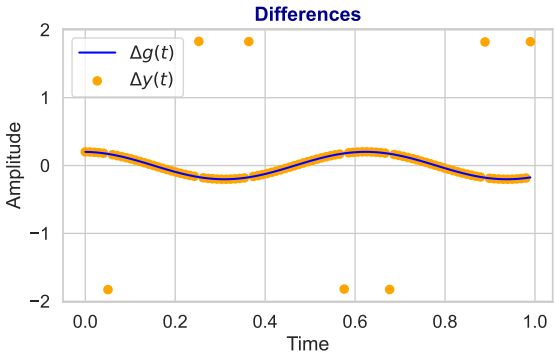
Therfore, the original signal is obtained through:

$$g[k] = S^N \left(M_\lambda(\Delta^N y[k])\right) \tag{16}$$

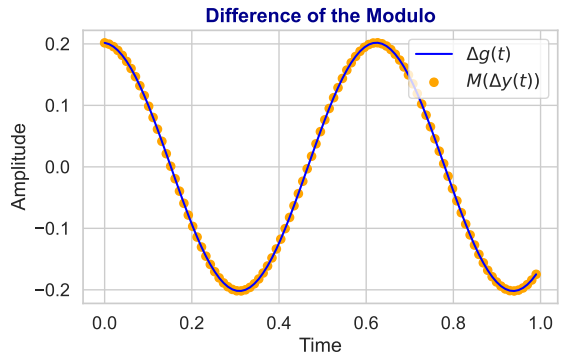
Figure 5 illustrates the application of the reconstruction algorithm on a sinusoidal signal.
In particular, Figure 5c shows that the finite differences of the original signal remain within the interval $[-\lambda, \lambda]$. As a result, the modulo operator does not alter these values, thereby verifying (14).
The original signal is then successfully reconstructed by applying the inverse finite-difference operation, as shown in Figure 5d.



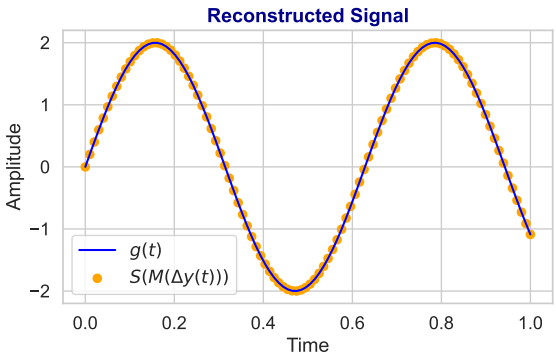
(a) Original input signal $g(t) = 2\sin(2\pi \times 10t)$ together with its modulo-folded version $y(t) = M_\lambda(g(t))$



(b) First-order finite differences computed from the original signal $g(t)$ and from the modulo signal $y(t)$



(c) Comparison between the first-order finite difference of $g(t)$ and the modulo operation applied to the first-order finite difference of $y(t)$



(d) Comparison between the original signal and the reconstructed signal obtained after applying the reconstruction algorithm

Figure 5: Python Simulation: Unlimited Sampling Theorem Reconstruction

2.2 Modulo-ADC Architectures

A variety of Modulo-ADC architectures have been proposed in the literature. While all of them are designed to acquire samples of the folded signal, they differ mainly in the way the modulo operation is implemented. In some architectures, the modulo block is placed at the analog front-end prior to conventional ADC processing, whereas in others it is integrated into the conversion chain together with the sampling and/or quantization stages.
Furthermore, although these architectures are often presented as offering an unlimited dynamic range, their practical dynamic range remains finite. Unlike conventional ADCs, where the limitation mainly originates from the quantization stage, the constraints in Modulo-ADCs are primarily imposed by circuit-level limitations such as supply voltage bounds and component operating ranges.
The purpose of this section is to review several Modulo-ADC architectures reported in the literature and to discuss their main advantages and associated challenges.

2.2.1 Unlimited Dynamic Range (UDR) ADC

In Figure 6, the Unlimited Dynamic Range (UDR) ADC architecture proposed in [6] is illustrated.
In this design, the modulo operation is performed sequentially within the ADC block: the sampling is performed first, followed by the modulo operation, and finally the quantization. The sampling and quantization stages are implemented using conventional blocks such as a S/H circuit [7] and a standard quantizer, respectively, while the modulo operation relies on a counter-based circuit combined with a feedback mechanism.

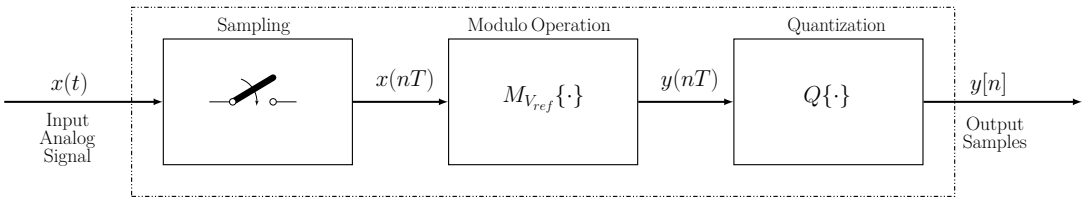


Figure 6: Block Diagram of the proposed UDR-ADC in [6]

In the proposed modulo circuit, whenever the sampled input exceeds the threshold, a reset event is triggered: the internal signal is wrapped back into the admissible range by subtracting or adding an appropriate multiple of the threshold level. A counter keeps track of these reset events and encodes this information into two additional bits.
Although this side information is not theoretically required for signal recovery, the authors use these extra bits to simplify and improve the efficiency of the reconstruction process, at the cost of a higher data rate per sample. In particular, the additional bits indicate whether a wrap event occurred and its direction (positive or negative). Based on this information, the authors developed a wavelet-based reconstruction algorithm.
The proposed approach was validated through a hardware prototype together with the associated reconstruction algorithm.
One advantage of the implementation is that the threshold voltage is configurable and provided as an input parameter rather than being fixed by the architecture. However, the number of possible folds is limited to three, which restricts the input amplitude range to 4λ. In addition, the paper does not discuss possible extensions for supporting a larger number of folds.
Moreover, the paper does not provide details regarding the maximum supported input signal frequency.
The hardware implementation relies on an Micro-Controller Unit (MCU), namely the ATmega328P [8], which implies that part of the processing is software-based.
This design choice may introduce additional processing delays compared to a fully hardware-based implementation, potentially limiting the achievable operating speed, as discussed in Section 2.3.3.

2.2.2 Integrator-based Modulo-ADC

The architecture proposed in [4] uses an integrator and a pair of Schmitt triggers (STs) to implement a simple, low cost, and adaptive signal folding mechanism. In contrast to the previously presented architecture, the modulo operation is performed as an analog front-end stage, prior to the conventional ADC processing chain.

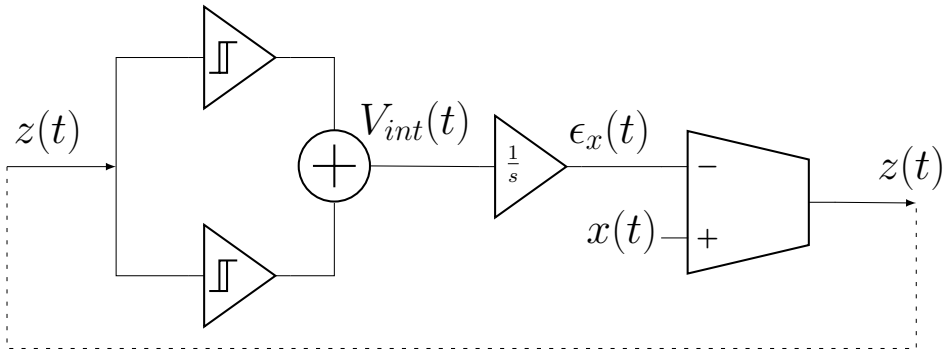


Figure 7: Block Diagram of the proposed integrator-based modulo ADC in [4]

The design relies on the Modular Decomposition Property introduced in Section 2.1.2.
Indeed, the design reposes on the fact that the input signal $x(t)$ can be expressed as:

$$x(t) = z(t) + \epsilon_x(t),\ z(t) := M_\lambda(x(t)) \tag{17}$$

where $M_\lambda(\cdot)$ denotes the modulo operation with threshold λ, and $\epsilon_x(t)$ is the residue function.

An instrumentation amplifier, which is a precision differential amplifier with a scaling gain, is used to extract the modulo signal from the decomposition of $x(t)$ by performing a subtraction:

$$z(t) = x(t) - \epsilon_x(t) \tag{18}$$

The residue function $\epsilon_x(t)$ is obtained by using an operational amplifier configured as an integrator:

$$\epsilon_x(t) = -\frac{1}{RC}\int_0^t V_{int}(\tau)d\tau + \epsilon_x(0) \tag{19}$$

where the integrator input $V_{int}(t)$ is obtained as the sum of the outputs of a pair of STs, whose transfer characteristics are shown in Figure 8.

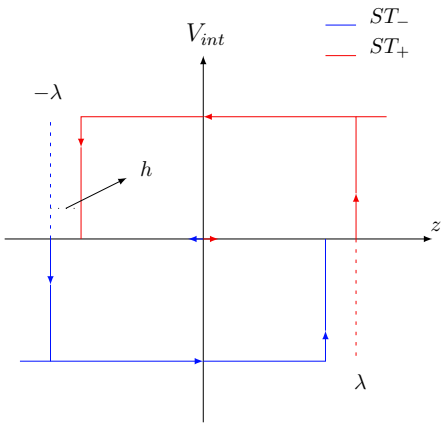


Figure 8: Transfer function of the Schmitt trigger pair and adder from [4]

Whenever $|z(t)| > \lambda$, the ST_+ generates a positive voltage, while the ST_- remains zero. The resulting sum drives the integrator, causing $\epsilon_x(t)$ to increase.
Once $z(t)$ falls below $(h - \lambda)$, where h is the hysteresis parameter, ST_+ returns to zero and the integrator stabilizes to a constant voltage which completes the folding cycle.

The proposed implementation offers several advantages.
By relying solely on analog components, it eliminates the need for digital blocks such as DACs or counters, resulting in reduced circuit complexity and lower power consumption [4].
Furthermore, the recovery process is straightforward since the residual function is already known.
They demonstrated the feasibility of their design through a hardware implementation capable of capturing HDR signals with 60 folds for a threshold $\lambda = 1$V.
However, the architecture also presents some limitations. In particular, the Schmitt trigger hysteresis parameter h must be carefully tuned to avoid oscillations and unintended transitions.
Moreover, little information is provided regarding the input frequency range or the hardware implementation itself, since no details are given about the components used or the supply voltages.

2.2.3 Multi-Channel (CRT) Modulo-ADC

Single-channel modulo-ADCs are the most common architecture. However, in noisy environments, single-channel modulo-ADC systems can require high sampling rates and significant computational costs [12].
The architecture presented in Figure 9 extends this idea by using an L-channel modulo-ADC based on the Chinese Remainder Theorem (CRT). The core idea of the CRT is that a number can be reconstructed from the residuals of its divisions by several coprime numbers (i.e., numbers that do not share common divisors other than 1). In this approach, multiple parallel acquisition channels operate with different modulo thresholds, producing a set of residues of the same input signal. The CRT is then used to reconstruct the original signal from these residues, provided that the chosen thresholds are pairwise compatible, as described in [12].

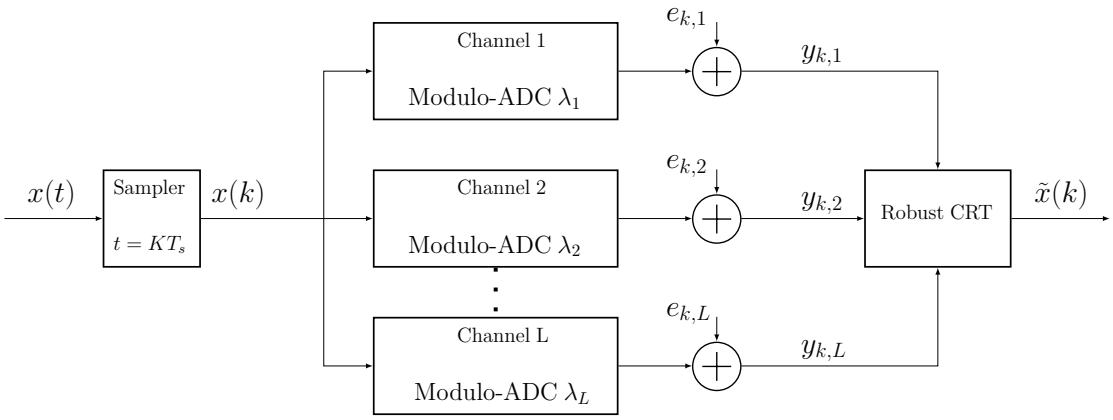


Figure 9: Block Diagram of the Multi-Channel modulo-ADC proposed in [12]

Similar to the UDR-ADC, the input signal $x(t)$ is first sampled, and each sample is then fed into L parallel modulo-ADCs with different thresholds $\lambda_1, \lambda_2, \ldots, \lambda_L$ defined by:

$$\lambda_l = \epsilon\tau_l \tag{20}$$

where ϵ is a positive floating-point number, and τ_l are pairwise coprimes integers, as required by CRT theory The terms $e_{k,i}$ denote additive measurement noise affecting channel i at time instant k. The CRT-based reconstruction can recover the original signal as long as the modulo-ADC errors stay below $|e| < \frac{\epsilon}{4}$.

Simulation results demonstrate that the multi-channel modulo-ADC can significantly reduce the required sampling frequency compared to a conventional ADC or a single-channel modulo-ADC.
However, the study does not address circuit-level feasibility or practical implementation, leaving open questions regarding the difficulty of properly setting the parameters for each channel.

2.2.4 Phase-Domain (Ring Oscillator) Modulo ADC

The architecture illustrated in Figure 10 is derived from [13], and was adopted in the previous Master's thesis [10].
This design employs a Voltage-Controlled Ring Oscillator (VCRO), which is a ring oscillator whose oscillation frequency is controlled by the input voltage, to perform the modulo operation and quantization simultaneously. A ring oscillator is a circuit composed of an odd number of cascaded inverters connected in a closed loop, producing a self-sustained oscillating signal due to the propagation delay of each stage. The key idea is to exploit the inherent modulo-2π nature of phase in oscillatory systems. Specifically, the ring oscillator maps the input voltage to a phase value, and by monitoring the state transistions of the oscillator over time, the modulo of the integrated input signal can be effectively captured.
To extract only the modulo component of the input, a finite-difference operation is applied in software during post-processing.

The main advantage of this approach is that it combines modulo operation and quantization into a single step, reduces the number of required analog and digital components, and improves energy efficiency. However, this architecture also introduces certain challenges, since it relies on accurate modeling of phase evolution, and the non-linearity of the ring oscillator can degrade performance at high operating rates [10].

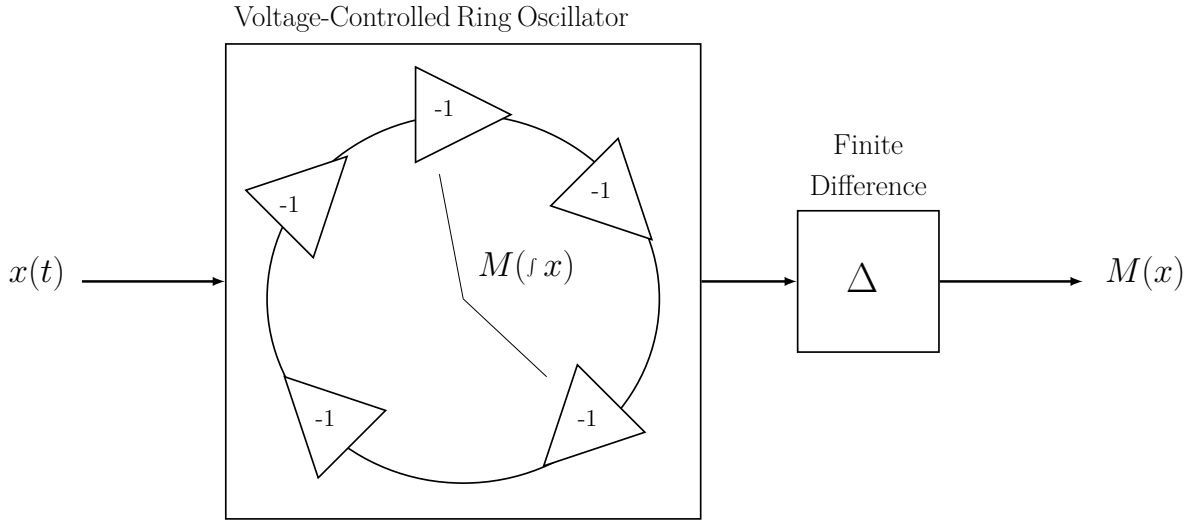


Figure 10: Block Diagram of the Phase Domain modulo-ADC proposed in [13]

2.3 Proposed Modulo-ADC Architecture

The architecture proposed in this work is illustrated in Figure 11. Similar to the integrator-based modulo architecture [4], it implements a modulo front-end block placed before the conventional ADC.

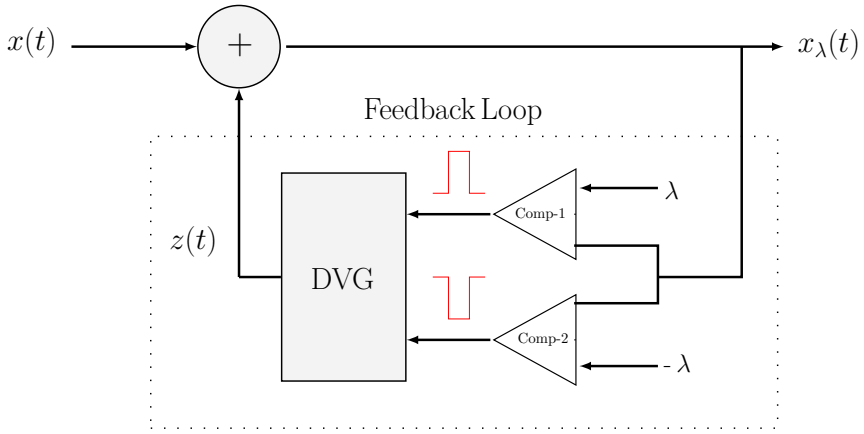


Figure 11: Block Diagram of the Wideband High-Dynamic Range ADC from [11]

The conceptual system architecture is derived from the work presented in [11], entitled *"A Hardware Prototype of Wideband High-Dynamic-Range ADC"*, in which the modulo operation is applied directly to the input signal prior to any sampling or quantization.

As discussed in the following sections, this architecture is particularly well suited to the objective of this work, namely the realization of a Printed Circuit Board (PCB) aimed at increasing the dynamic range of an ADC. Since the modulo operation is implemented as a front-end block, it can be directly inserted before an existing ADC and an existing digital post-processing system.
Moreover, its fundamental building blocks can be readily implemented using standard components such as operational amplifiers (op-amps), making the design practical and achievable within the scope of a Master's thesis. Finally, this approach facilitates the development of a hardware prototype that can be easily modified or adapted in the future, for example by adjusting threshold values to meet the specific requirements of different applications.

2.3.1 Operating Principle

The design applies the folding operation directly to the continuous-time input signal $x(t)$.
The operating principle relies primarily on a feedback mechanism composed of three main blocks:

- **Two comparators** that continuously monitor the input signal.
- **A Digital Voltage Generator (DVG)** that produces a controlled voltage signal $z(t)$ based on the triggering of the comparators.
- **A summing block** that combines the input signal $x(t)$ with the output of the DVG, thereby closing the feedback loop.

The modulo operation implemented in this design is expressed mathematically as follows:

$$x_\lambda(t) = M_\lambda(x(t)) = \big((x(t) + \lambda) \bmod 2\lambda\big) - \lambda \tag{21}$$

This operation first shifts the signal $x(t)$ upward by λ, moving the symmetric interval $[-\lambda, \lambda]$ to the positive range $[0, 2\lambda]$.
A modulo 2λ operation is then applied, meaning that any value of $x(t) + \lambda$ that exceeds 2λ will be wrapped around to the range $[0, 2\lambda]$. Finally, the result is shifted back down by λ to restore the original symmetric interval $[-\lambda, \lambda]$.
This change of interval is necessary because the modulo operation is defined over intervals starting at zero.

To better understand the operating principle, the behavior of the folding block is illustrated in Figure 12 and Figure 13.

- For an instant t_1 such that $t < t_1$, we have $|y(t)| < \lambda$. Therefore, $y_\lambda(t) = y(t)$ and the output of the DVG is $z(t) = 0$, since no comparator is triggered.
- At $t = t_1$, let $|y(t)|$ cross λ. If $y(t_1) > \lambda$ (Figure 12), then comparator 1 generates a positive trigger. Conversely, if $y(t_1) < -\lambda$ (Figure 13), comparator 2 generates a negative trigger.
- If the DVG receives a positive trigger, its output decreases by 2λ (Figure 12). If it receives a negative trigger, its output increases by 2λ (Figure 13).
- At a later instant $t = t_2$, the signal may again cross one of the thresholds. Depending on the crossing direction, the appropriate comparator is triggered once more, generating either a positive or negative correction signal.
- The DVG responds accordingly, applying the necessary voltage adjustment so that, in both cases (Figure 12 and Figure 13), the folded output returns within the range $[-\lambda, \lambda]$, restoring the condition $y_\lambda(t) = y(t)$ and resulting in $z(t) = 0$.

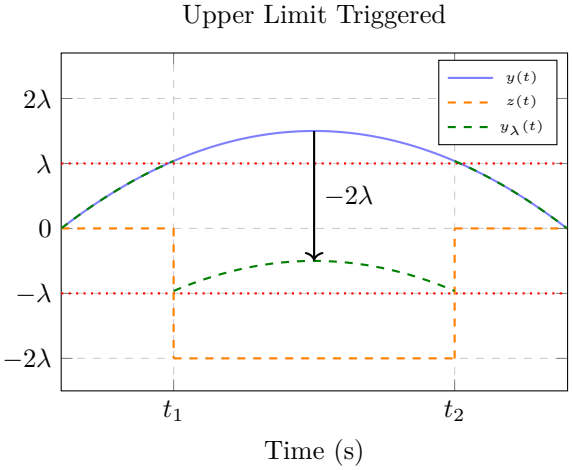


Figure 12: Circuit behavior when the upper threshold is reached.

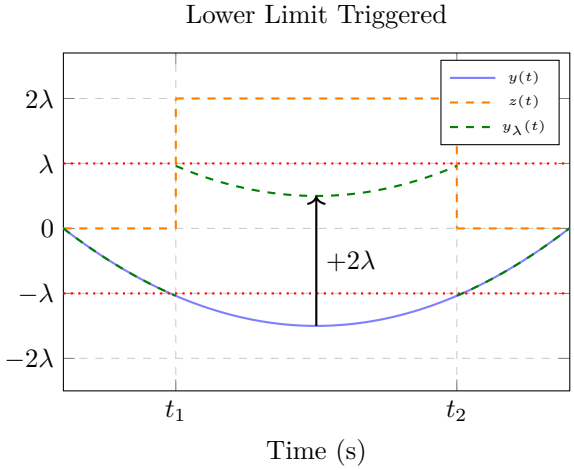


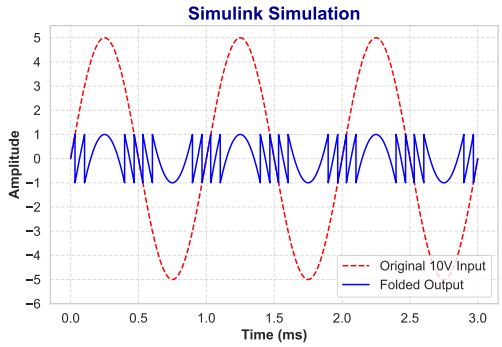
Figure 13: Circuit behavior when the lower threshold is reached.

In other words, when the input signal exceeds the ADC dynamic range, a comparator detects this condition and generates a trigger signal. This trigger activates the DVG, which adds or subtracts a voltage offset to bring the signal back within the valid dynamic range. The voltage shift introduced by the DVG must be a constant multiple of 2λ. Specifically, the DVG increases or decreases the input signal by 2λ in response to positive or negative triggers, respectively.

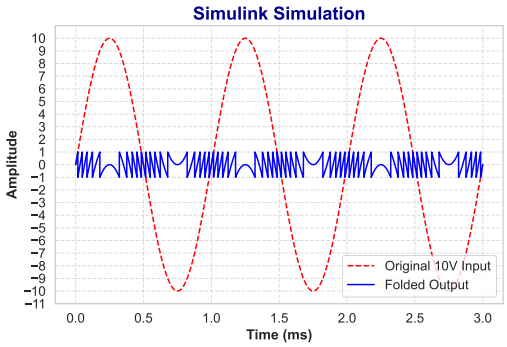
2.3.2 Matlab Simulink Behavorial Simulation

An important step in the design process is to understand the mathematical behavior of the system in order to analyze its operation, identify potential hardware implementation constraints, and highlight possible design limitations that must be considered. To this end, a behavorial simulation of the block diagram shown in Figure 11 is implemented in Matlab Simulink.

To verify the operation of the folding block, the thresholds are set to $\lambda = \pm 1$V, and sinusoidal input signals with amplitudes of 5V and 10V are applied, corresponding to Figure 14a and Figure 14b, respectively. The simulations are performed under ideal conditions where the feedback-loop delay is negligible compared to the signal period (e.g., a 1 kHz input signal with a period of 1 ms and a delay of 1 ns). The results show that whenever the input signal exceeds the thresholds, the output is correctly folded back into the range $[-1, 1]$V.



(a) Folded output for a 5V sinusoidal input signal



(b) Folded output for a 10V sinusoidal input signal

Figure 14: Matlab Simulation: Folded output for different input amplitudes

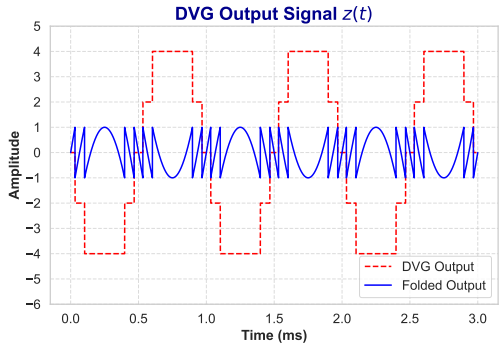
2.3.3 Limitations

This first system-level simulation allowed validation of the operating principle of the proposed architecture. More importantly, it highlighted several limitations that must be considered for the hardware implementation and experimental validation.
Based on these limitations, it is also possible to define realistic design specifications in the context of a first prototype PCB. These specifications will be discussed further in the next chapter dedicated to the implementation.

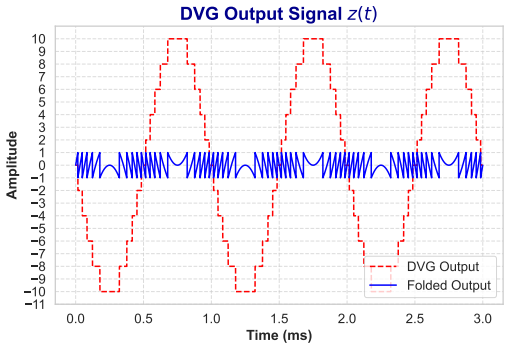
1. **Maximum Input Amplitude**

   As mentioned previously, for different modulo-ADC architectures, including the proposed one, the input dynamic range is theoretically unlimited. In practice however, it is constrained by the supply voltage of the circuit block.
   For instance, if an operational amplifier is used in the summing block, its output voltage cannot exceed the supply rails, otherwise, it will saturate or potentially be damaged.
   Nevertheless, it also affects the design of the DVG in the feedback loop.
   Indeed, the maximum input amplitude determines the maximum voltage that the DVG must be able to generate. A larger input amplitude requires higher multiples of 2λ (in absolute value) at the DVG output.

   Considering again the example of 5 V and 10 V sinusoidal input signals with thresholds fixed at ± 1V, the corresponding DVG outputs are shown in Figure 15.



(a) DVG output for a 5V sinusoidal input signal



(b) DVG output for a 10V sinusoidal input signal

Figure 15: Matlab Simulation: DVG output for different input amplitudes

As shown in Figure 15a, an input signal with amplitude 5V must be folded twice, whereas the 10V signal in Figure 15b must be folded five times in order to be mapped into the interval $[-\lambda, \lambda]$. The folding operation can be written as:

$$y(x) = x - 2\lambda k(x) \tag{22}$$

where $y(x)$ is the folded output and $k(x)$ denotes the integer folding index given by:

$$k(x) = \text{round}\left(\frac{x}{2\lambda}\right) \tag{23}$$

For a threshold value $\lambda = 1$, this yields:

$$k(5) = \text{round}\left(\frac{5}{2}\right) = 2 \tag{24}$$

$$k(10) = \text{round}\left(\frac{10}{2}\right) = 5 \tag{25}$$

Each folding operation introduces a correction of magnitude 2λ. Consequently, the internal DVG output associated with the correction term is:

$$v_{\text{DVG}}(x) = 2\lambda k(x). \tag{26}$$

For a 5V input, the DVG must be capable of producing an output range up to $\pm 4V$. In contrast, for a 10V input, the required output range increases to $\pm 10V$ due to five successive folds. This significantly increases the required voltage range and therefore the supply voltage constraints of the DVG design. Therefore, it is important to define beforehand the maximum input amplitude that the modulo block must handle, in order to select appropriate components and supply voltages.

2. **Feedback Delay**

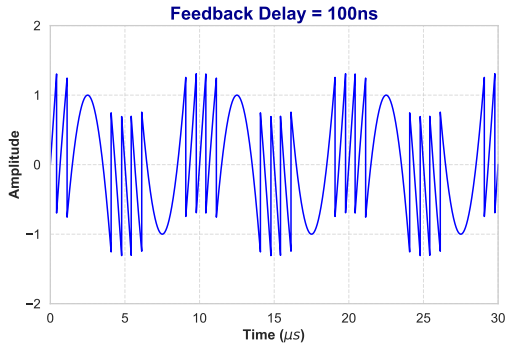
The feedback delay T_d, defined as the time required for the signal to propagate through the comparators and the DVG before producing the correction $z(t)$, is a critical parameter. Indeed, this delay directly limits the maximum frequency of the input signal that can be correctly folded. If the input frequency is too high, the signal varies faster than the correction can be applied, which may cause threshold crossings to be missed or detected too late. Similarly, the maximum allowable amplitude, particularly for a sinusoidal input signal, is also affected, since it is directly related to the frequency through the following expression:

$$x(t) = A\sin(2\pi f t), \qquad \text{slope} := \frac{d}{dt}\big(A\sin(2\pi f t)\big) = 2\pi f A\cos(2\pi f t) \approx 2\pi f A. \tag{27}$$

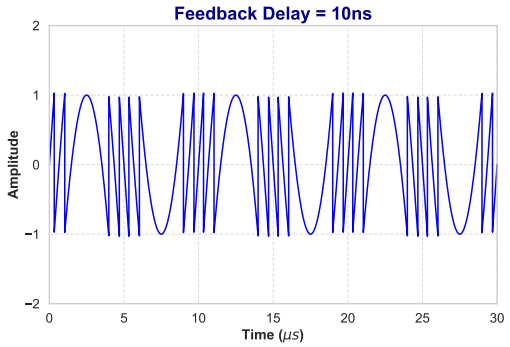
The last approximation follows from the fact that the cosine function is bounded between -1 and 1. From this expression, it is clear that increasing the frequency results in a larger signal slope.

A higher slope increases the likelihood of multiple threshold crossings occurring within a very short time interval, potentially exceeding the system's ability to respond correctly.
Therefore, the maximum input amplitude influences not only the required supply voltage range, but is also intrinsically linked to the maximum allowable input frequency.

For instance, in Figure 16a, with a 100kHz sinuosoïdal input and a feedback delay of 100ns and the thresholds still set to ±1V, the folds become asymmetric due to the feedback delay, and the signal exceeds the range significantly.
In constrast, in Figure 16b, with the same input but a delay reduced to 10ns, the output signal is nearly perfectly contained within the range [-1,1]V.



(a) Output signal for a 100kHz 5V sinusoïdal input signal for T_d=100ns



(b) Output signal for a 100kHz 5V sinusoïdal input signal T_d=10ns

Figure 16: Matlab Simulation: Output signal for different feedback delays

In [11], the issue of feedback delay is addressed. The authors show that, with a finite feedback delay, the overshoot of a smooth signal can be bounded. This holds true if the input signal is *Lipschitz* continuous [14].
Specifically, a signal $y(t)$ is Lipschitz continuous if there exists a positive real number L_y such that, for any $T > 0$,

$$|y(t) - y(t+T)| \leq L_y T. \tag{28}$$

This condition ensures that the signal has a bounded rate of change, i.e., it cannot vary arbitrarily fast. Under this condition, the amplitude of $y(t)$ cannot change by more than $L_y T_d$ (where T_d is the feedback delay) between consecutive folding instants. Therefore, if we choose the dynamic range of the ADC to be

$$[-(\lambda + \Delta.\lambda), (\lambda + \Delta.\lambda)], \quad \text{where } \Delta.\lambda = L_y T_d, \tag{29}$$

where λ is the nominal dynamic range of the ADC (i.e., the expected maximum amplitude), Δ is a dimensionless margin factor accounting for signal variations during the feedback delay, the signal will not be clipped. For example, if $\lambda = 1$V and $\Delta = 0.25$, the ADC range is extended by 0.25λ, resulting in a total range of [-1.25, 1.25]V.

For an input sinusoidal signal $y(t) = A\sin(\omega_c t) = A\sin(2\pi f_0 t)$, the Lipschitz constant is the maximum slope of the signal:

$$L_y = 2\pi A f_0 \tag{30}$$

where A is the amplitude and f_0 the frequency. This allows to determine the maximum overshoot for fixed parameters.

For exemple, for a 10V sinusoïdal signal at 10kHz, the maximum slope is:

$$L_y = 2\pi \cdot 10 \cdot 10^4 \approx 6.28 \cdot 10^6. \tag{31}$$

With a feedback delay of $T_d = 1\mu s$, the maximum overshoot is:

$$\Delta.\lambda = L_y T_d \approx 0.628 \tag{32}$$

which means that the folded signal can exceed the nominal threshold by up to 0.628, resulting in a total range of $[-1.628, 1.628]$V.
The Figure 17, obtained from a Matlab simulation, confirms this theoretical prediction: the maximum and minimum values of the folded signal are 1.6267V and –1.6267V, which closely match the theoretical bounds.

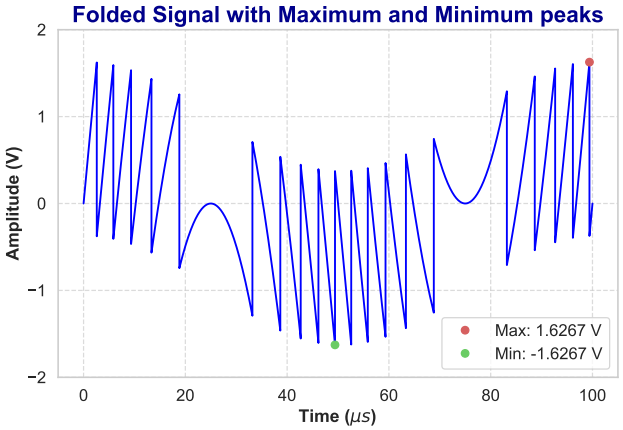


Figure 17: Maximum overshoot of the folded signal

2.4 Conclusion

Throughout this chapter, the theory of US has been presented, together with the modulo mapping used to constrain a signal within the range $[-\lambda, \lambda]$, where λ represents a fixed threshold.
The Modular Decomposition Property has also been explained, as it forms the foundation of the reconstruction algorithm used in this work to recover the original signal using only the modulo samples, without requiring any additional information.

Several architectures proposed in the literature have then been reviewed, highlighting their operating principles and fundamental concepts.
These architectures are generally supported by hardware implementations demonstrating their effectiveness, although key performance metrics such as the maximum input frequency and admissible input amplitude are not always specified. Nevertheless, their main advantages and implementation challenges have been identified and discussed.

The architecture selected for this work, referred to as the Wideband High-Dynamic Range ADC in [11], has been described in detail. A Matlab Simulink behavioral simulation was used to demonstrate its operation and validate the reconstruction principle.
This analysis also highlighted several important challenges, particularly the impact of the input amplitude on the supply voltage requirements of the components, as well as the feedback delay associated with the folding operation, which limits the maximum achievable input frequency.

With the theoretical aspects and the selected architecture now established, the next chapter focuses on the hardware implementation of the system.

3

System Architecture and Design Methodology

The previous chapter introduced the theoretical background of US theory, together with several modulo-ADC architectures proposed in the literature.
Among these architectures, the Wideband High-Dynamic Range ADC presented in [11] was selected as the basis for the hardware implementation developed in this work.
Its mathematical model was analyzed and implemented in a behavioral Matlab Simulink simulation. This study highlighted several practical limitations of the selected architecture, particularly regarding input amplitude constraints and feedback delay.

This chapter presents the design methodology and hardware implementation of the proposed system. It begins with a review of the reference architecture presented in [11], together with its main limitations. Based on this analysis, the system specifications are defined. The architecture developed in this work is then introduced, with particular focus on reducing feedback delay, lowering supply voltage requirements, and improving architectural adaptability.
Each circuit block is subsequently described through its schematic implementation and key characteristics, including component selection, supported by LTspice simulations to validate correct operation.
The gain of each block is then analyzed, accompanied by a feedback delay analysis based on the selected components.
Finally, the hardware implementation is presented, with particular emphasis on the PCB design process.

3.1 Hardware Implementation from the Literature

In [11], a hardware implementation of the Wideband High-Dynamic Range ADC is proposed.
The architecture consists of a front-end folding block that can be inserted directly before an existing ADC and its associated digital post-processing system.

The proposed design includes the following hardware components, as illustrated in Figure 18:

- **Two comparators**: These comparators generate trigger signals when the input exceeds the thresholds $\pm\lambda$. The comparator associated with the upper threshold generates a positive trigger pulse when the input signal exceeds $+\lambda$. Conversely, the comparator associated with the lower threshold generates a negative trigger pulse when the input signal falls below $-\lambda$.
- **UP/DOWN counter**: The counter receives the trigger signals from the comparators. For each positive trigger, the counter increments its state. Conversely, for each negative trigger, it decrements to the previous state.
- **Digital-to-Analog Converter (DAC)**: The DAC converts the digital counter value into an analog voltage. Since the DAC output is often strictly positive, a Multiplexer (MUX) is introduced to handle polarity selection. The MUX uses a sign bit provided by the counter to apply either a positive or negative correction at the input of the multiplier.
- **Multiplier**: This block scales the analog output of the DAC.
- **Adder**: The adder combines the original input signal with the scaled correction signal produced by the multiplier.

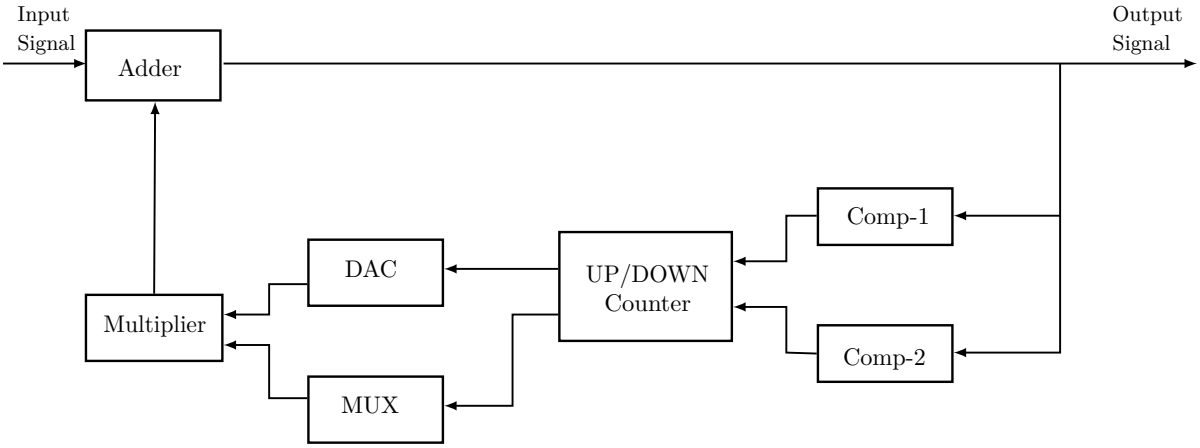


Figure 18: Block Diagram of the architecture from [11]

In [11], the performance of the architecture is evaluated through a hardware PCB implementation. The threshold used in the implementation is fixed at $\lambda = 1.25$V. The results show that the system can process input signals with amplitudes up to $8\lambda = 10$V corresponding to four folds according (23).
It is further demonstrated that, by introducing a margin factor $\Delta = 0.5$, the folded output is allowed to vary within the range $[-(\lambda + \Delta.\lambda), (\lambda + \Delta.\lambda)] = [-1.875, 1.875]$V. Under these conditions, the maximum achievable input signal frequency for a 10V amplitude signal is 10kHz. Although these results represent a significant improvement over previous implementations, such as the sample-and-hold-based approach cited in the paper, which achieves only around 300Hz, the system still presents several limitations:

- **High Feedback Delay**:

  A major drawback of this implementation is the delay introduced by the components in the feedback loop, which results in a relatively large overall feedback delay.
  In particular, the feedback loop includes a counter implemented on a PJRC Teensy 4.1 Development Board [16], a microcontroller platform programmable via the Arduino IDE.
  Although the board operates at a reported clock frequency of 600MHz, the software-based implementation inevitably increases the processing time required to update the counter state and generate the corresponding DAC output, thereby contributing significantly to the total feedback delay. The minimum reported delay of the counter is approximately 1μs. This value does not include additional delays introduced by the DAC and other elements in the feedback loop, which further increase the overall feedback delay. As discussed in Section 2.3.3, this delay limits the maximum input frequency that can be correctly processed by the system, which in this case is approximately 10kHz.

- **Limited input voltage range**:

  It is stated that some of the selected components, such as the multiplexer, multiplier, and amplifiers, can only operate for voltages below 12 V. Consequently, the correction voltage at the output of the multiplier cannot exceed 12 V in absolute value. This limits the maximum input amplitude to $8\lambda = 10$ V, because if the input exceeds 9λ, the correction voltage would need to reach $10\lambda = 12.5$ V according (26), which is beyond the supported operating range.
  Moreover, the adder used requires a supply voltage of 15 V even though the input voltage does not exceed 10 V. This implies the need for multiple supply voltages, including unnecessarily high supply levels that are not fully utilized.
  Increasing the input amplitude would require larger correction signals and therefore the replacement of several components with components supporting higher supply voltages, highlighting the lack of flexibility of the architecture. In addition, increasing the supply voltage of multiple components would significantly increase power consumption.

3.2 Design Specifications and Guidelines

In this context, the proposed implementation is designed according to the following guiding principles:

- **Low-voltage operation:** Minimize the number of supply rails to reduce overall design complexity, and lower the supply voltage of most functional blocks to decrease power consumption.
- **Minimized feedback delay:** Reduce the total delay in the feedback loop to well below $1\,\mu s$, in order to increase the maximum supported input frequency.
- **Design adaptability:** Ensure flexibility to adjust the folding thresholds and the maximum input amplitude according to application requirements.

These guidelines led to the design choices summarized in Table 1, which were followed throughout the development of the implementation proposed in this work.

Parameter	**Specification**
Core Functionality: Modulo Operation	
Output Voltage	$V_{out}(t) = ((V_{in}(t) + \lambda) \bmod 2\lambda) - \lambda$
Threshold Voltage λ	$\pm\lambda = \pm 1$ V
Folding Voltage V_{fold}	$V_{fold} = k \cdot (\pm 2\lambda) = k \cdot (\pm 2\text{ V}), \quad k \in \mathbb{Z}$
Feedback Delay T_d	$T_d < 1\ \mu$s
Reconstruction	Compatible with existing reconstruction algorithms
Input Signal Characteristics	
Input Voltage Range	$\lvert V_{in}\rvert \leq 5V$ with the objective of later increasing the dynamic range
Input Frequency	$f_{in} \geq 10$ kHz
Output Signal Characteristics	
Output Voltage Range	$V_{out} \in [-1.25, 1.25]$V
Electrical Requirements	
Supply Voltage at Input stage	$\lvert V_{supply,in}\rvert \geq \lvert V_{in}\rvert$
Other Blocks Supply Voltage	$1.25V \leq \lvert V_{supply}\rvert \leq 3.3V$
PCB Design Constraints	
Component Selection	Components must have available SPICE models
Component Packaging	Hand-solderable packages only (no BGA)
Component Availability	Available from standard distributors (e.g., Digi-Key, Mouser)
Power Supply Design	All supply rails generated externally
Testability	Test points available at each functional block
Implementation	Fully hardware-based (no software-dependent blocks)
Clock Generation	On-board clock with optional external clock input

Table 1: Specification of the Modulo-ADC Hardware Implementation

As listed in the specifications, the maximum input amplitude is limited to ± 5 V for this first implementation. This limitation is a direct consequence of the requirement that all selected analog components must have an available SPICE model.
Indeed, this work corresponds to a first prototype developed in the context of a Master's thesis, therefore, it is essential to validate the design through simulation using LTspice. This requirement significantly constrained the choice of components available from distributors such as Digi-Key, especially for op-amps, since not all devices are provided with compatible SPICE models. As a result, most op-amps with available SPICE models support supply voltages limited to approximately ± 5 V, which consequently

limits the maximum input amplitude of the entire system.
In Section 4.5.1, alternative component choices are proposed in order to improve the achievable dynamic range. For digital components such as counters, accurate behavioral simulation is generally not possible beforehand. Indeed, LTspice is primarily intended for analog circuit simulation, and modeling detailed digital behavior is either unsupported or very complex to implement. Consequently, the requirement for available SPICE models does not apply to digital components.

Regarding the output voltage range, as shown in Section 4.3.2, it is important for the ADC to support a slightly larger dynamic range than $[-\lambda, \lambda]$. Therefore, even though the threshold λ is set to $\pm 1\,\mathrm{V}$, the accepted output range is chosen as [-1.25,1.25]V.

Finally, the supply voltage of the input stage must naturally support the maximum input range, which in this case is limited to $\pm 5\,\mathrm{V}$. However, for the other blocks, lower supply voltages are preferred in order to minimize the number of distinct supply rails and reduce power consumption. These supply voltages cannot be lower than the required output range to avoid clipping and preferably should not exceed $3.3\,\mathrm{V}$, corresponding to the standard supply voltage commonly used for Low-Voltage Complementary Metal-Oxide-Semiconductor (LVCMOS) digital logic.

3.3 Proposed Hardware Implementation

Based on the specifications and guidelines listed in the previous section, the architecture illustrated in Figure 19 is proposed:

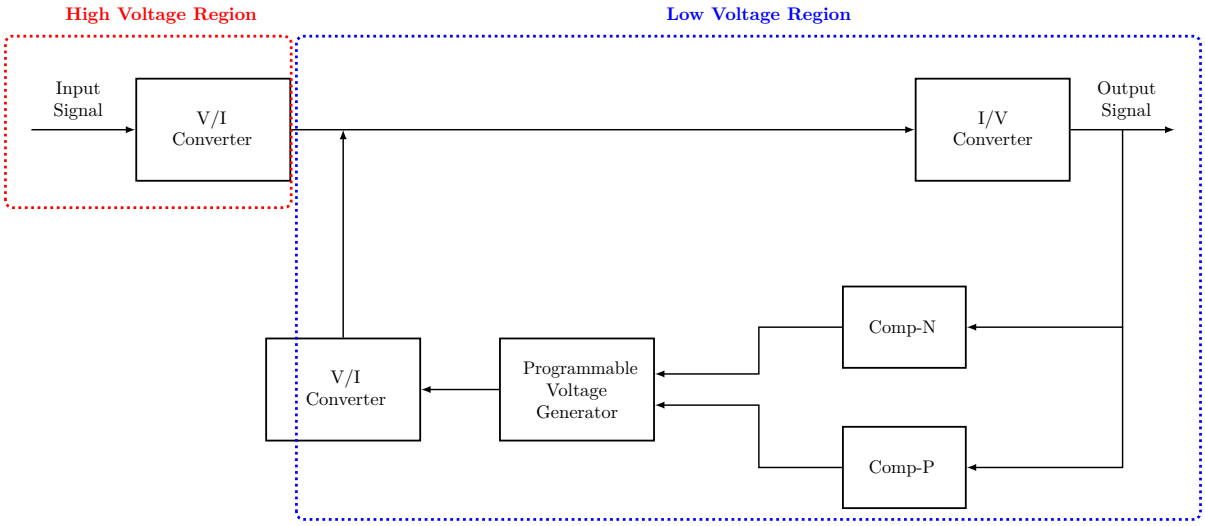


Figure 19: Block Diagram of the proposed architecture

It is consituted of five main blocks:

- **Input Voltage to Current (V/I) Converter**: The first input block converts the input signal from voltage to current.
- **Output Current to Voltage (I/V) Converter**: The output block converts the current back to voltage, providing an output voltage signal for the comparators and the ADC.
- **Comparators (Comp-P and Comp-N)**: Two comparators monitor the output signal and generate trigger pulses when the signal exceeds the upper or lower thresholds.
- **Programmable Voltage Generator (PVG)**: The PVG is responsible for generating the appropriate correction voltage based on the trigger signals produced by the comparators. The block implements a counter, which keeps track of the system state by counting the trigger events, and two DACs, which generate the corresponding analog correction voltage.
- **Feedback Voltage to Current (V/I) Converter**: This block converts the voltage output of the PVG into a current, which is then summed with the input current at the node formed by the interconnection of the outputs of the two V/I converters.

The main idea behind this architecture is to perform the folding operation in the current domain rather than in the voltage domain.
In the approach presented in [11], the folding operation is implemented in the voltage domain. As a result, both the input adder stage and the multiplier that generates the correction signal require high supply voltages to accommodate large input amplitudes. In the proposed architecture, the input voltage is first converted into a current at the front-end stage. Consequently, only this initial conversion stage must handle the full input voltage range and therefore requires a higher supply voltage. Once in the current domain, the folding operation is performed by simply adding or subtracting currents from the input current. As a result, even the PVG can operate at low supply voltage levels, since it only needs to generate a correction current of the same order of magnitude as the input signal.

3.4 Circuit-Level Simulation and Validation

In this section, the different circuit building blocks are presented together with the selected components and validated through LTspice simulations in order to demonstrate their functionality.

3.4.1 Voltage to Current (V/I) Converter

A common way to have a V/I converter that can produce both positive and negative currents, is to use an operational amplifier in a transconductance configuration.
A transconductance circuit converts input voltage into an output current following the relation:

$$I_{out} = g_m \cdot V_{in} \tag{33}$$

where g_m is the transconductance gain of the circuit, V_{in} is the input voltage, and I_{out} is the output current. The transconductance g_m [A/V] is the gain factor that determines how much current is produced for a given input voltage and is typically determined by the circuit design and the choice of values of the resistors.

The operational transconductance amplifier chosen is called Improved Howland Current pump [18], represented in Figure 20.

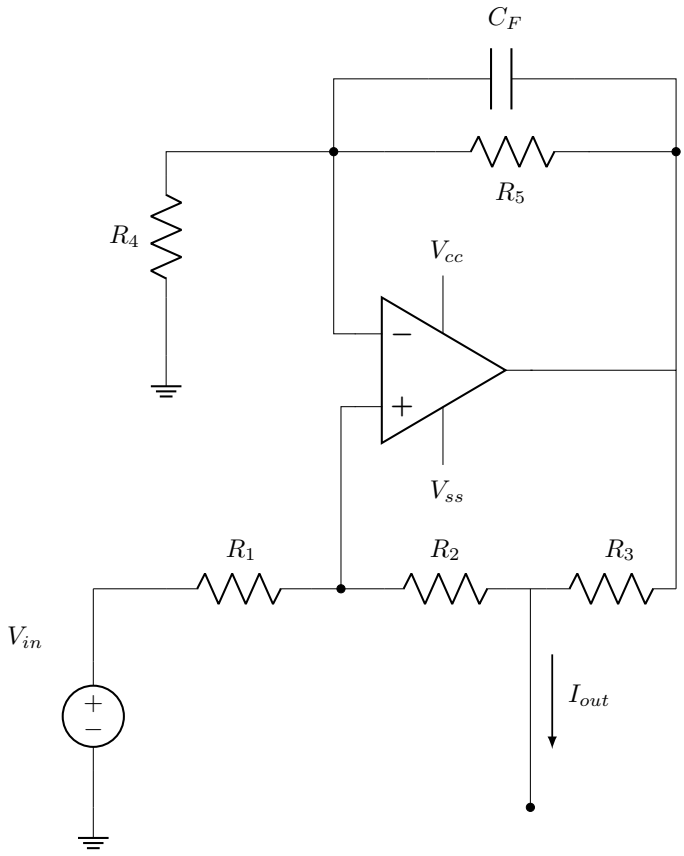


Figure 20: Improved Howland Current Pump Schematic

By applying Kirchhoff's Current Law (KCL), assuming an ideal operational amplifier, and imposing the following resistor matching conditions:

$$\frac{R_1}{R_2 + R_3} = \frac{R_4}{R_5} \tag{34}$$

and

$$R_4 = R_5 \tag{35}$$

The output current can be expressed as:

$$I_{out} = \frac{V_{in}}{R_3} \tag{36}$$

Thus, the transconductance gain is:

$$g_m[\text{A/V}] = \frac{1}{R_3} \tag{37}$$

The complete derivation is provided in Annex A.
This circuit allows an easy design of the transconductance gain by simply choosing the value of R_3 and determining the other resistors according to (34) and (35).
However, if a precise and stable transconductance gain is required, it is important to use high-precision resistors with tight tolerance (e.g., 0.1% or lower).

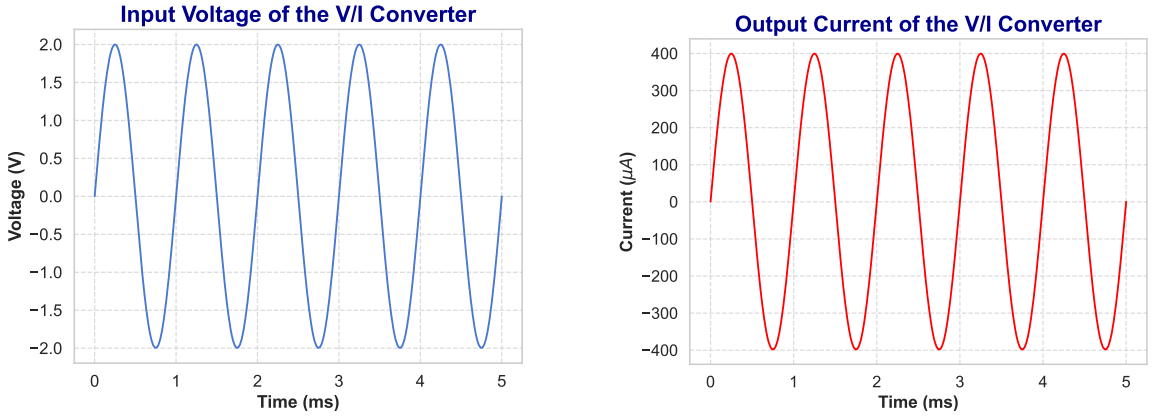
It is recommended to add a capacitor C_F in parallel with R_5. The full development is given in Appendix E of [19], where it is shown that the finite gain-bandwidth product (GBW) of the operational amplifier introduces a virtual output capacitance C_x at the node between R_2 and R_3. This parasitic capacitance can reach several hundreds of picofarads for an operational amplifier with a gain-bandwidth product of 1 MHz, thereby limiting bandwidth and potentially causing instability with reactive loads. Adding C_F in parallel with R_5 introduces a zero in the feedback path that cancels the effect of this parasitic pole. At high frequencies, C_F bypasses R_5 and restores near-unity feedback to the inverting input. A recommended value of the feedback capacitance is 3 pF to 10 pF, small enough to leave the low-frequency transconductance gain $g_m = 1/R_3$ essentially unaffected, while still providing sufficient compensation to mitigate the effects of the parasitic capacitance at higher frequencies where it would otherwise degrade performance.

In [19], a list of suitable operational amplifiers for use in an Improved Howland Current Pump configuration is provided. Based on these recommendations, and considering the requirements for an available SPICE model as well as commercial availability through distributors such as Digi-Key and Mouser, the LT1813 operational amplifier is selected for the V/I converter circuit. This amplifier is said to offer low power consumption while maintaining high-speed performance, with a gain-bandwidth product of 100MHz and a settling time of 30ns. The settling time corresponds to the time required for the output to reach and remain within its final stable value. The absolute maximum total supply voltage ($V^+ - V^-$) for this device is 12.6 V, beyond which the amplifier may not operate correctly and could be permanently damaged. The electrical characteristics provided in the datasheet [22] are specified for a typical supply voltage of $V_s = \pm 5$ V, which is therefore used in this design. With this supply voltage, the input signal amplitude cannot exceed approximately -5 V to $+5$ V. However, as mentioned previously, the objective of this prototype is primarily to validate the architecture rather than to support very large input amplitudes. Possible design improvements to extend this input range are discussed in Section 4.5.1. Figure 21 presents the results of a behavioural simulation of the Improved Howland Current Pump carried out in LTspice, using the selected operational amplifier and the parameters listed in Table 2.

Parameter	**Value**	**Unit**	**Notes**
Input	2	V	1 kHz sinusoidal signal
Gain resistor R_3	5	$k\Omega$	Sets the gain
Resistor R_1	6	$k\Omega$	
Resistors R_2, R_4, R_5	1	$k\Omega$	
Feedback capacitor C_F	3	pF	
Operational amplifier	LT1813		Datasheet [22]
Negative supply voltage V_{ss}	-5	V	
Positive supply voltage V_{cc}	5	V	

Table 2: LTspice simulation configuration for the V/I converter

The input voltage is a sinusoidal signal with an amplitude of 2V and a frequency of 1kHz. With the gain resistor value set to $5\,k\Omega$, the transconductance gain g_m is equal to 5×10^{-3} A/V. The output current waveform shown in Figure 21b therefore corresponds to the input signal scaled by a factor of 5×10^{-3}. Moreover, when the input voltage becomes negative, the output current also becomes negative, as expected. No instability issues were observed during simulation, regardless of whether the feedback capacitor is included. Nevertheless, the feedback capacitor was retained in the final design, since LTspice behavioural simulations may not fully capture all non-ideal effects and parasitic phenomena associated with the practical implementation of the circuit.



(a) Input voltage

(b) Output current

Figure 21: LTspice Simulation: Input voltage and output current of the Improved Howland Current Pump

3.4.2 Current to Voltage (I/V) Converter

Similarly to the V/I converter, a I/V converter can be achieved using an operational amplifier. In this configuration, the circuit is referred to as a *transimpedance amplifier* [21], as illustrated in Figure 22.

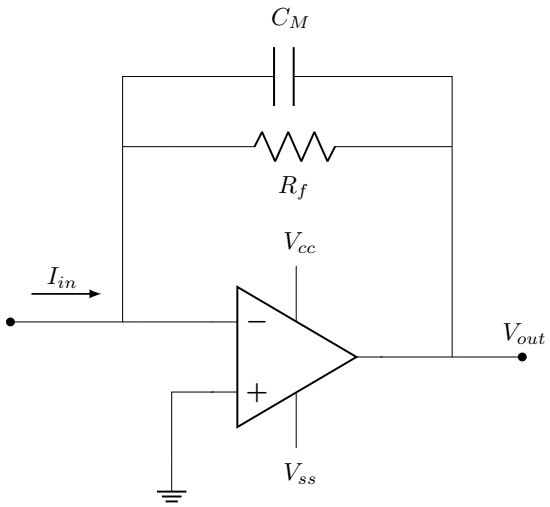


Figure 22: Transimpedance Amplifier Schematic

Applying Kirchhoff's Current Law (KCL) at the inverting input node V_- of the op-amp gives:

$$I_{in} = \frac{V_- - V_{out}}{R_f} \tag{38}$$

Assuming an ideal operational amplifier, there is a virtual short between the two inputs of the op-amp, which gives:

$$V_- = V_+ = 0 \tag{39}$$

Therefore, a transimpedance amplifier converts an input current into an output voltage following the relation:

$$V_{out} = -I_{in} \cdot R_f \tag{40}$$

This configuration provides the same function as a resistor, but with the advantage of a much higher input impedance and a lower output impedance.
The negative sign in the equation indicates that the output voltage is inverted with respect to the input current. This inversion doesn't affect the functionality of the circuit since the output voltage can be easily inverted back if needed by adding an additional inverting stage or in the post-processing stage of the signal. In this work, the latter solution is adopted.
Therefore, the transimpedance gain is determined by the value of the feedback resistor R_f. In this case also, since the gain depends mostly on the resistor, it is better to use resistors with low tolerance to ensure a precise gain.

Again, a feedback capacitor is added in parallel with the feedback resistor to ensure the stability of the circuit. However, in a transimpedance amplifier, the capacitor value is carefully calculated by taking into account the capacitances present at the inverting input node [21].
In this configuration, the current source is directly connected to the inverting input of the operational amplifier, while the non-inverting input is grounded. Due to the high open-loop gain of the op-amp, the inverting node is maintained at a virtual ground. As a result, although the voltage at this node is approximately zero, its impedance is high, making the effect of parasitic capacitances significant.

The total capacitance at the inverting node can be expressed as:

$$C_{tot} = C_n + C_M \tag{41}$$

where C_n represents the intrinsic input capacitance of the op-amp together with any parasitic capacitances, and C_M is the feedback capacitor.
This capacitance forms a pole with the feedback resistor:

$$f_p = \frac{1}{2\pi R_f (C_n + C_M)} \tag{42}$$

Since this pole lies within the feedback loop, it introduces additional phase shift, which can reduce the phase margin and potentially lead to instability.
To compensate for this effect, the feedback capacitor C_M is chosen such that it introduces a zero in the loop-gain that improves the phase margin. By analyzing the loop gain and applying the stability condition near the unity-gain frequency, the following approximation can be derived [21]:

$$C_M \approx \sqrt{\frac{C_n + C_M}{2\pi R_f f_t}} \tag{43}$$

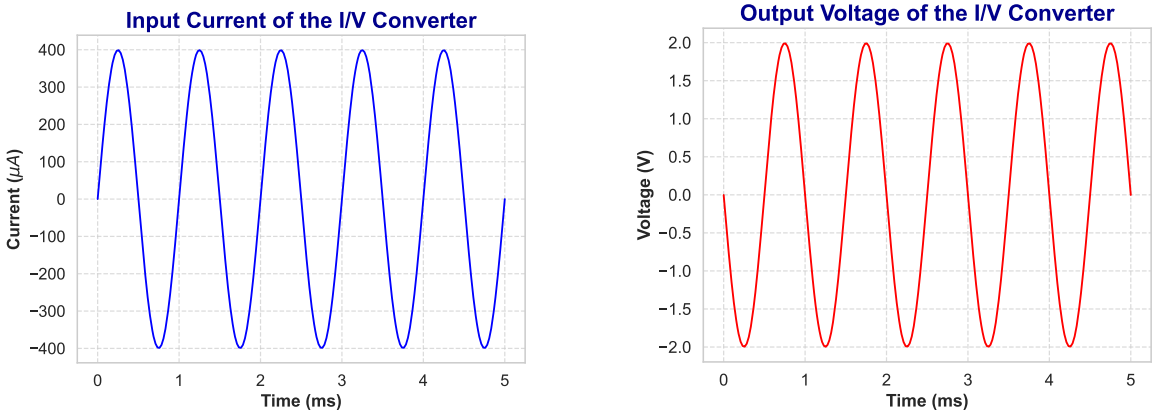
This expression shows that the required feedback capacitance depends on the total input capacitance at the inverting node, the feedback resistor, and the unity-gain frequency f_t of the operational amplifier. Both C_n and f_t are typically provided in the op-amp datasheet.

Since each LT1813 package contains two independant amplifier circuits, in order to use all available amplifier and simplify the implementation by using only one type of op-amp, the LT1813 is also chosen for the I/V converter.
Figure 23 presents the results of a behavioural simulation of the transimpedance amplifier performed in LTspice, using the parameters listed in Table 3.

Parameter	Value	Unit	Notes
Input	400	μA	1 kHz sinusoidal current
Gain resistor R	5	$k\Omega$	Sets the gain
Feedback capacitor C_M	13	pF	Calculated from 43
Operational amplifier	LT1813		Datasheet [22]
Negative supply voltage V_{ss}	-5	V	
Positive supply voltage V_{cc}	5	V	

Table 3: Simulation configuration for the I/V converter

The input current is a sinusoidal signal with an amplitude of $400\mu A$ and a frequency of 1 kHz. The simulation results show that the output voltage is proportional to the input current with a gain factor of $R = 10^3\Omega$. The output signal is inverted with respect to the input current due to the negative gain of the transimpedance amplifier.



(a) Input voltage

(b) Output current

Figure 23: LTspice Simulation: Input voltage and output current of the I/V converter

Moreover, the presence of this feedback capacitor in parallel with the feedback resistor introduces a first-order low-pass filter. The cutoff frequency is therefore determined by the feedback capacitor C_M and the feedback resistor of the I/V converter.
Considering a feedback resistor of $5k\Omega$ and a feedback capacitor of 13 pF, the cutoff frequency is given by:

$$f_c = \frac{1}{2\pi R C_M} = \frac{1}{2\pi \times 5000 \times 13 \times 10^{-12}} \approx 2.45 MHz \tag{44}$$

To verify the usable frequency range of the circuit, corresponding to the passband, a gain Bode diagram is used.
A gain Bode diagram represents the variation of the output amplitude of a system as a function of the input frequency. The horizontal axis shows the frequency on a logarithmic scale, while the vertical axis represents the gain magnitude in decibels (dB), defined as:

$$Gain(dB) = 20 \log_{10} \left(\frac{|V_{out}(j\omega)|}{|V_{in}(j\omega)|} \right) \tag{45}$$

where j is the imaginiary unit and ω the angular frequency given by $\omega = 2\pi f$ [rad/s]. When $\frac{|V_{\text{out}}|}{|V_{\text{in}}|} = 1$, the gain is 0 dB, meaning that the output amplitude is equal to the input and no attenuation occurs.
The cutoff frequency is defined as the frequency at which the gain decreases by 3 dB relative to the passband gain. Beyond this point, the condition $\frac{|V_{\text{out}}|}{|V_{\text{in}}|} < 1$ holds, corresponding to a negative gain in dB and thus attenuation of the output signal.
The resulting Bode diagram is shown on Figure 24.
The measured cutoff frequency is approximately 2.4MHz, which is consistent with the theoretical value calculated in (44). Beyond this frequency, the gain decreases with a slope of approximately -20dB/decade, as expected for a first-order low-pass filter.

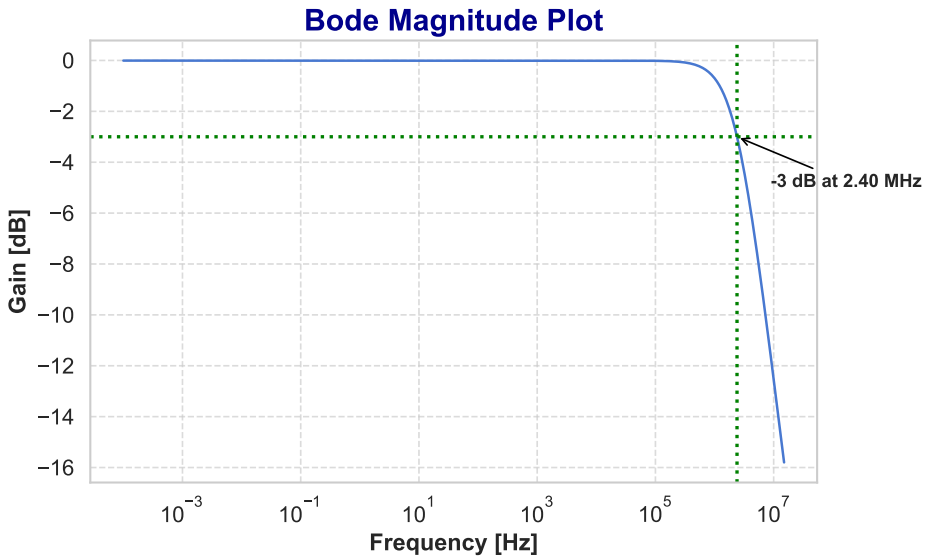


Figure 24: LTspice Simulation: Frequency response of I/V converter

3.4.3 Comparators -N and -P

Two comparators are required: one to trigger when the signal is below the threshold $-\lambda$ and one to trigger when the signal is above the threshold λ. The term trigger means that the output signal of the comparator changes its state when the input crosses the threshold.
More specifically, the output signal of comparator-N (Comp-N, where N stands for negative threshold) becomes high when the signal falls below $-\lambda$, and returns to low when the signal rises above $-\lambda$. Conversely, the output of comparator-P (Comp-P, where P stands for positive threshold) becomes high when the input signal exceeds λ, and returns to low when the signal drops below λ. Therefore, the outputs of the two comparators can never trigger at the same time, since the input signal cannot be simultaneously higher than λ and lower than $-\lambda$.

An operational amplifier associated with a resistive voltage divider, used to fix the threshold voltage, implements such a comparator, as shown in Figure 25.

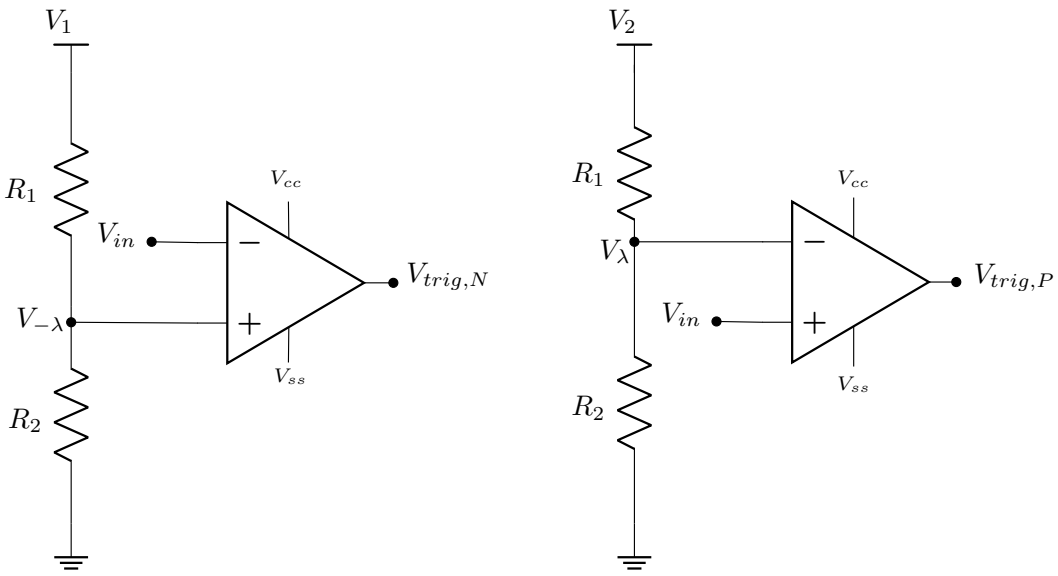


Figure 25: Schematic of Comparator-N

For Comp-N, given that a negative threshold voltage $-\lambda$ is required, the voltage divider must be supplied by a negative voltage source V_1.
The resulting negative threshold voltage is given by:

$$V_{-\lambda} = V_1 \times \frac{R_2}{R_1 + R_2} \tag{46}$$

The use of a voltage divider provides flexibility in the design, as the threshold voltage can be easily adjusted by selecting appropriate resistor values.
In Comp-N, the reference voltage $V_{-\lambda}$ is applied to the non-inverting input of the operational amplifier. The output behavior is therefore:

$$\begin{cases} \text{If } V_{in} < V_{-\lambda} \text{ then } V_{out} = V_{cc} \\ \text{If } V_{in} > V_{-\lambda} \text{ then } V_{out} = V_{ss} \end{cases} \tag{47}$$

Similarly, for Comp-P, a positive threshold voltage λ is obtained by supplying the voltage divider with a positive source V_2 and selecting appropriate resistor values.
The positive threshold voltage is:

$$V_{\lambda} = V_2 \times \frac{R_2}{R_1 + R_2} \tag{48}$$

In this case, the desired behavior is the opposite: the output should be high when the input voltage exceeds the threshold λ. This leads to:

$$\begin{cases} \text{If } V_{in} > V_{\lambda} \text{ then } V_{out} = V_{cc} \\ \text{If } V_{in} < V_{\lambda} \text{ then } V_{out} = V_{ss} \end{cases} \tag{49}$$

To achieve this behavior, the reference voltage generated by the divider is connected to the inverting input of the operational amplifier, while the input signal is applied to the non-inverting input.

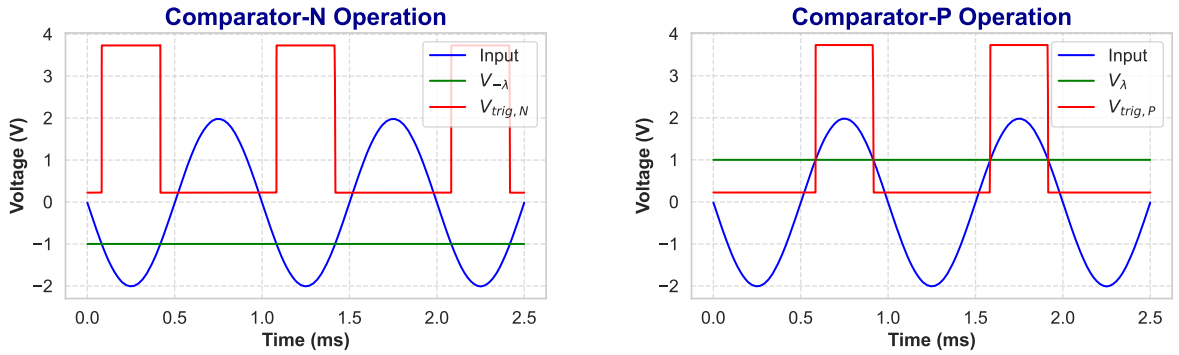
For both Comp-N and Comp-P, the trigger outputs $V_{trig,N}$ and $V_{trig,P}$ must remain within the range 0–3.3 V to ensure compatibility with the LVCMOS digital components used in the following stages, as presented in the next section.
Using the LT1813 operational amplifier in a comparator configuration is not appropriate, since its output would swing from -5 V to $+5$ V. Instead, the LT1016 comparator is selected.
The LT1016 includes an output stage specifically designed for interfacing with digital logic. Although it is supplied with ± 5 V, its output high level is typically around 3.4V, while the output low level is close to 0.3V [28].

Figure 26 shows the simulated behavior of Comp-N and Comp-P using the parameters listed in Table 4.

Parameter	**Value**	**Unit**	**Notes**
Input	2	V	1 kHz sinusoidal voltage
Negative voltage divider supply V_1	-3.3	V	
Positive voltage divider supply V_2	3.3	V	
Voltage divider resistor R_1	4	kΩ	
Voltage divider resistor R_2	1	kΩ	
Comparator	LT1016		Datasheet [28]
Negative supply voltage V_{ss}	-5	V	
Positive supply voltage V_{cc}	5	V	

Table 4: Simulation parameters for the comparators

The input signal is a sine wave with an amplitude of 2 V at a frequency of 1 kHz, while the reference voltages are set to $V_{-\lambda} = -1$V and $V_{\lambda} = 1$V.
The outputs switch correctly when the corresponding threshold voltages are reached. The output low level is approximately 0.3V, as expected. However, the output high level exceeds 3.3V and reaches approximately 3.8V.



(a) Input, Reference voltage $V_{-\lambda}$ and Output $V_{trig,N}$ (b) Input, Reference voltage V_{λ} and Output $V_{trig,P}$

Figure 26: LTspice Simulation: Input, Reference, and Output voltage of the comparators

To protect the following digital stages from voltage levels above 3.3V, a Zener diode is added at the output of each comparator, as shown in Figure 27. The resistor R, around 100Ω, limits the current flowing through the diode during clamping.
A Zener diode allows current to flow in the reverse direction when the reverse voltage exceeds the Zener voltage. Here, a diode with a Zener voltage of 3.3V is chosen, which clamps the comparator output to approximately 3.3V.

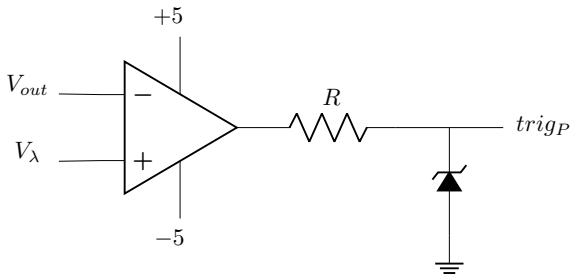
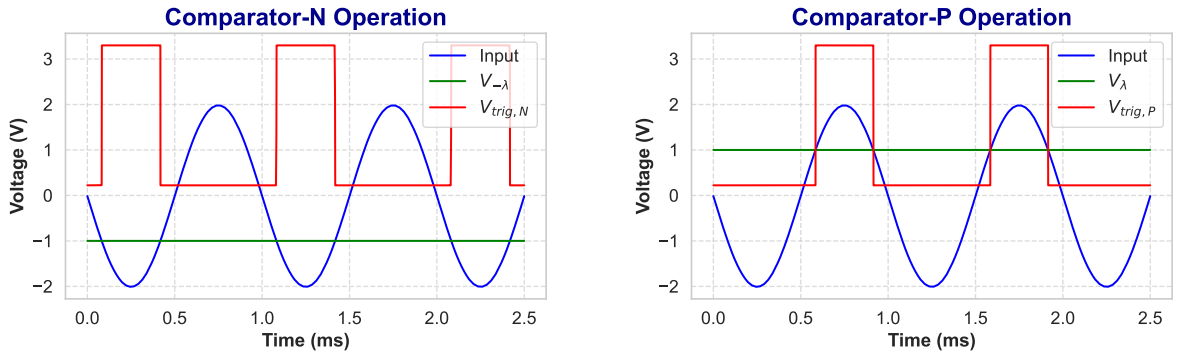


Figure 27: Schematic of comparator-P with Zener diode protection

Figure 28 shows the simulation results obtained with the Zener diode protection. The comparators output high level are now limited to approximately 3.3V.



(a) Input, Reference voltage $V_{-\lambda}$ and Output $V_{trig,N}$ (b) Input, Reference voltage V_{λ} and Output $V_{trig,P}$

Figure 28: LTspice Simulation: Input, Reference, and Output voltage of the comparators with Zener diode

3.4.4 Programmable Voltage Generator

The Programmable Voltage Generator (PVG) receives the trigger signals generated by the two comparators and produces the correction voltage that will subsequently be converted into a current.
The output voltage must be able to take both positive and negative values in order to generate a positive or negative current at the output of the V/I converter, depending on the sign of the fold.

The proposed architecture is shown in Figure 29.

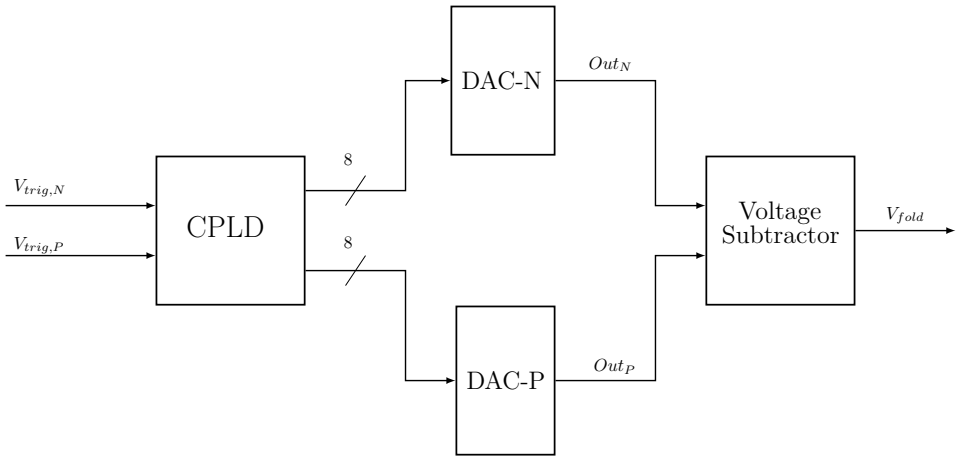


Figure 29: Schematic of the PVG

The PVG consists of three main blocks:

- **Complex Programmable Logic Device (CPLD)**:

  A CPLD is a type of digital integrated circuit that can be programmed by the user to implement custom logic functions. It belongs to the family of programmable logic devices (PLDs), which allow hardware behavior to be defined after manufacturing.
  Internally, a CPLD consists of several programmable logic blocks interconnected through a predictable routing structure. Each block can implement combinational logic (e.g., AND/OR functions) as well as simple sequential elements such as flip-flops. Compared to a Field Programmable Gate Array (FPGA), a CPLD typically contains fewer logic.
  Rather than designing circuits using discrete components such as logic gates or multiplexers, the desired functionality is described using a hardware description language (HDL) such as Verilog or VHDL. This description is then synthesized and mapped onto the internal structure of the CPLD.
  CPLDs are non-volatile, meaning that they retain their configuration even when power is removed. Although a clock is not strictly required, it is often used to synchronize the input and output signals of the device. This makes it possible to accurately predict timing behavior based on the clock period and the implemented logic.

  In the proposed design, as shown in Figure 29, the CPLD receives the two trigger signals $V_{trig,N}$ and $V_{trig,P}$ generated by the comparators. The CPLD produces two 8-bit outputs corresponding to the digital representation of the correction voltage.
  By implementing a Finite State Machine (FSM), which can be viewed as a counter within the CPLD, the digital codes sent to the DACs are updated according to the trigger signals in order to generate the required correction voltage.
  The implementation details of the FSM and the configuration are described in the following section.

  The criteria for selecting the CPLD are availability from distributor websites, a sufficient number of I/O pins to accommodate at least two inputs ($V_{trig,N}$ and $V_{trig,P}$) and two 8-bit outputs for the DACs, enough logic elements (LEs) to implement the FSM, and suitability for high-speed operation, including support for a high-frequency clock and low propagation delays.

It is also preferable to use a CPLD with 3.3V I/O compatibility in order to maintain compatibility with the LVCMOS components used in the design.
Based on these requirements, the 5M80Z from Altera is selected. The device provides 80 Logic Elements (LEs), which is sufficient for the implementation of the FSM, as discussed below, and 79 I/O pins, which largely satisfies the interface requirements. The device also exhibits a very low pin-to-pin delay of 7.5 ns in the worst-case scenario, which corresponds to a full diagonal path across the device with combinational logic implemented using a single Lookup Table (LUT) and an adjacent Logic Array Block (LAB). Although it does not represent the exact delay of the implemented design, since the actual delay depends on the synthesized logic, it provides an indication of the device speed performance. Additionally, it supports an external clock up to 152 MHz, as described in the datasheet [23].
The 5M80Z requires two supply voltages: 1.8 V for the core and 3.3 V for the I/O banks. The core supply powers the internal logic and configuration of the CPLD, while the I/O supply defines the voltage levels of the external input and output pins.
A reset signal is implemented to ensure the system starts in a predictable state. The reset forces internal registers and flip-flops to a known condition, typically initializing the state machine.
In this design, the reset is generated using a voltage supervisor (MAX6444) combined with a push-button. The MAX6444, also powered at 3.3 V, guarantees a clean, single reset pulse of minimum duration even if the push-button is pressed quickly, eliminating noise caused by mechanical switch bouncing.

- **DACs**:

The CPLD drives two DACs, each receiving an 8-bit digital input.
A DAC converts a digital code into a corresponding analog voltage. In other words, it performs the inverse operation of an ADC.
The output voltage V_{out} of an ideal DAC can be expressed as

$$V_{out} = V_{FS} \times \frac{D}{(2^N - 1)} \tag{50}$$

where V_{FS} is the full-scale voltage (i.e., the maximum analog output voltage of the DAC), D is the decimal value of the digital input code with the maximum digital value $D_{max} = 2^N - 1$, N is the number of bits of the DAC, and 2^N represents the total number of possible digital codes.
Therefore, depending on the digital codes produced by the CPLD, the DACs generate the voltages Out_N and Out_P required to produce the appropriate correction signal.

DACs capable of directly generating negative voltages are less common and typically require both positive and negative supply rails, which increases the complexity of the power supply design. To avoid this constraint, this architecture uses two DACs combined with a subtractor stage, as described below.
The CPLD ensures that the two DACs never output non-zero values simultaneously. In other words, only one DAC produces a non-zero output at a given time, depending on the state of the FSM implemented in the CPLD. The voltage substractor stage ensures that the output of the PVG has the correct polarity.

The DACs must satisfy several criteria: availability from distributors, sufficient resolution to allow a large number of correction steps if required, and suitability for high-speed operation. The latter constraint significantly reduces the number of available DACs, especially when requiring a settling time below 100ns.
Considering these requirements, the MAX5190 DAC is selected.
The MAX5190 is an 8-bit voltage-output DAC powered with a 3.3 V supply. It is designed for high-performance signal reconstruction with low distortion and low power consumption, as described in the datasheet [24].
The DAC requires a clock signal at its input to update the output value. The maximum clock frequency is 40 MHz, which corresponds to a minimum clock period of 25 ns. Consequently, the output settling time is said to be at least one clock period.
The MAX5190 provides a fully differential output, which can be converted into a single-ended

voltage using a low-distortion operational amplifier configured as a subtractor, as described in the datasheet. The operational amplifier originally recommended in the datasheet is no longer commercially available. Therefore, an amplifier with similar characteristics, the ADA4857, is selected instead [25].
In addition, the DAC output is unbuffered, meaning that its voltage can be affected by the load presented by the following circuit stage. For this reason, a buffer amplifier implemented with the LT1813 is added at the output.
The full-scale output voltage of the DAC is 400 mV. Consequently, the DAC output voltage can vary from 0 to 400 mV over $2^8 = 256$ discrete levels, although such a fine resolution is not necessarily required for the intended application.
After conversion of the differential output into a single-ended signal and the addition of the buffer stage, the resulting output is a buffered single-ended voltage ranging from 0 to 400 mV.

Both the DACs and the CPLD require an external clock. The DACs use the clock to update their output values, while the CPLD uses it to synchronize the input trigger signals. Synchronizing the input trigger signals with the clock means that the CPLD does not react immediately to asynchronous input changes. Instead, input signals are first sampled on a specific clock edge, ensuring that state transitions in the FSM only occur at discrete, well-defined time steps. This avoids timing ambiguity and ensures deterministic and stable digital behavior.
The maximum clock frequency accepted by the DACs is 40 MHz. Therefore, an oscillator producing a 40 MHz clock is selected.
Although the CPLD can operate at higher clock frequencies, for this first prototype it is preferable to use a single clock for simplicity and to reduce design complexity.
The *SiT9005* component is chosen to generate the clock.
It is a Micro-Electromechanical Systems (MEMS)-based oscillator that produces a 40 MHz square-wave output and operates from a 3.3 V supply. As a result, the clock signal swings approximately from 0 V to 3.3 V [26], making it fully compatible with both the CPLD and the DACs.
The use of a MEMS-based oscillator simplifies the design because the complete clock generation system is already integrated inside a single component. In comparison, a discrete quartz crystal solution would require additional external components such as the crystal itself, load capacitors and an oscillator circuit to generate a stable clock signal.
The SiT9005 can therefore directly provide a stable clock signal without requiring additional tuning or careful design of the crystal network. In addition, its integrated output stage is capable of driving multiple CMOS inputs, which reduces the risk of loading issues when the same clock is distributed to several components. MEMS oscillators are also less sensitive to PCB layout and parasitic effects than discrete quartz crystal circuits, which improves startup reliability and simplifies PCB integration [27].

- **Voltage Subtractor**

To generate both positive and negative correction voltages, the outputs of the two DACs are subtracted according to:

$$V_{fold} = A \times (Out_P - Out_N) \tag{51}$$

where A is the gain of the subtractor stage used to scale the output voltage, since DACs typically provide relatively low output amplitudes.
If a negative fold correction is required, the current produced by the V/I converter must be negative. Consequently, the input voltage of this converter (i.e., V_{fold}) must also be negative. In this case, only the digital code sent to DAC-N is non-zero, while the output of DAC-P is zero. According to (51), this results in a negative value of V_{fold}.
Conversely, if a positive correction current is required, the CPLD drives DAC-P while DAC-N outputs zero, resulting in a positive value of V_{fold}.
The voltage subtractor is implemented using the basic differential amplifier shown in Figure 30.

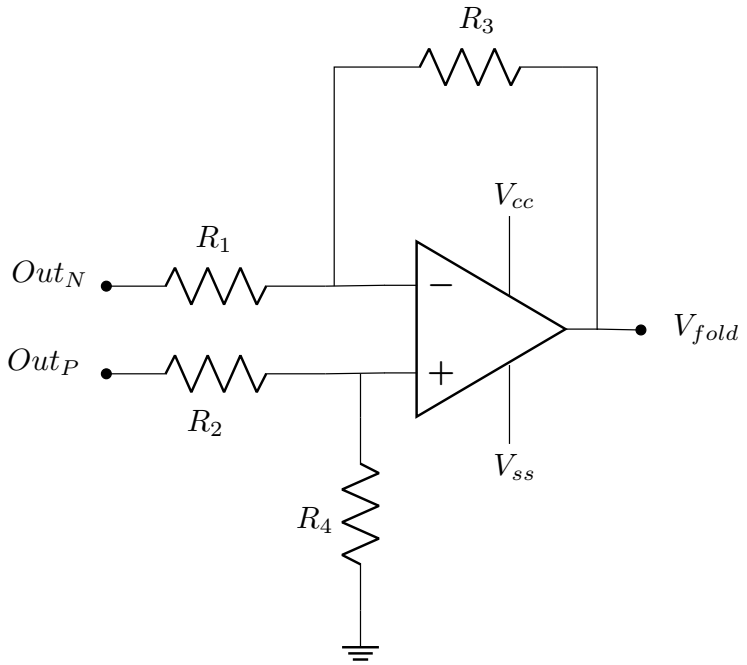


Figure 30: Schematic of Voltage Subtractor

The output of the differential amplifier is given by

$$V_{fold} = \frac{R_3}{R_1} \times (Out_P - Out_N) \tag{52}$$

which is valid provided that $R_1 = R_2$ and $R_3 = R_4$.

The LT1813 operational amplifier is also selected to implement the subtractor stage.
Figure 31 presents the results of a simulation performed to verify the operation of the subtractor stage, using the parameters listed in Table 5.

Parameter	**Value**	**Unit**	**Notes**
Input Out_N	0 to 200	mV	Step waveform with 100mV increments/decrements
Input Out_P	0 to 200	mV	Step waveform with 100mV increments/decrements
Resistor $R_1 = R_2$	400	Ω	Set the Gain
Resistor $R_3 = R_4$	1.6	$k\Omega$	Set the Gain
Operational amplifier	LT1813		Datasheet [22]
Negative supply voltage V_{ss}	−5	V	
Positive supply voltage V_{cc}	5	V	

Table 5: Simulation configuration for the subtractor

The inputs provided by the DACs are designed such that only one of them is non-zero at any given time. Each input takes discrete values of 0, 100 mV, or 200 mV in a stepped waveform.
The resistor values define a gain of 4 for the subtractor stage. As a result, the output of the circuit is also a stepped waveform, which becomes negative when Out_N is active and positive when Out_P is active.

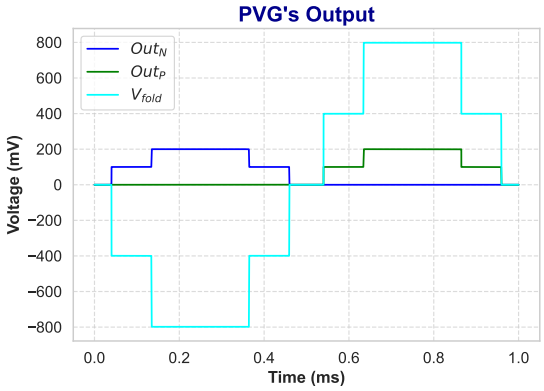


Figure 31: LTspice Simulation: Output of the DACs and output of Voltage Subtractor

Therefore, the PVG is only constituted of hardware blocks, with components such as the CPLD and DACs selected to minimize delay. As no software is involved, the feedback delay can be significantly reduced.

3.5 Sizing and Validation

Since the input range is limited to ±5V by the supply voltage of the components, only two folds are required according to (23). Therefore, with a threshold of $\pm\lambda = 1$V, and for a maximum input amplitude of 5V, the only required correction voltages V_{fold} are ±2V and ±4V, as defined by (26).
Consequently, the gains of the different circuit blocks, the FSM design, and the DACs output values are chosen assuming that only two folds must be handled. Possible approaches to increase this number are presented in Section 4.5.1. Based on these considerations, the FSM implemented in Verilog on the CPLD, whose code is provided in Annex D, is defined as shown in Figure 32.

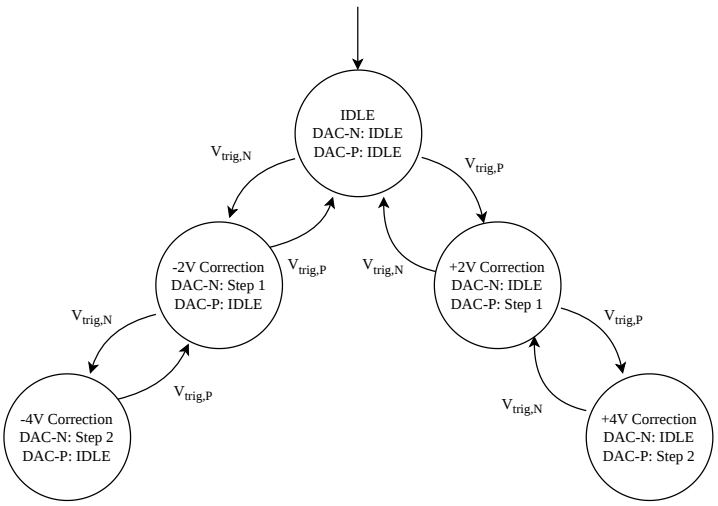


Figure 32: Schematic of the FSM

This FSM is a Moore FSM because its outputs depend only on the current state, meaning that each state has a fixed output value associated with it.
According to the resource estimation provided by Quartus for the Verilog implementation, the FSM requires 42 LEs, which remains below the 80 LEs available on the selected CPLD.
The FSM starts in the initial IDLE state, where no correction is applied and both DACs output are zero. A high $V_{trig,P}$ indicates that the output signal exceeds +1V. Because the I/V converter introduces an inversion, this corresponds to the input signal dropping below −1V. In this case, the correction voltage V_{fold} must be increased, and DAC-P is activated.
Similarly, a high $V_{trig,N}$ indicates that the output signal falls below −1V, which means that the input

signal exceeds +1V. The correction voltage V_{fold} must then be decreased, and DAC-N is activated. The FSM is implemented such that both DACs output are never non-zero simultaneously.
As mentioned previously, the MAX5190 DAC has a full-scale output voltage of 400 mV and an 8-bit input. The input code 00000000 corresponds to 0 V, while 11111111 corresponds to 400 mV at the DAC output. In principle, this provides $2^8 = 256$ discrete output levels, corresponding to 255 voltage steps of approximately 1.57 mV between adjacent levels. However, since the input voltage amplitude limits the system to only two folds, such fine resolution is unnecessary. Therefore, the output steps are set to 100 mV. In this configuration, Step 1 corresponds to a DAC output of 100 mV, while Step 2 corresponds to a DAC output of 200 mV.
This choice leaves open the possibility of increasing the number of folds to four if the input range is slightly extended, without requiring any modification to the rest of the system. Nevertheless, it would also have been possible to choose a 200 mV step size, in which case only two folds would remain possible. Knowing this, the gains of all circuit blocks are determined and are illustrated in Figure 33.

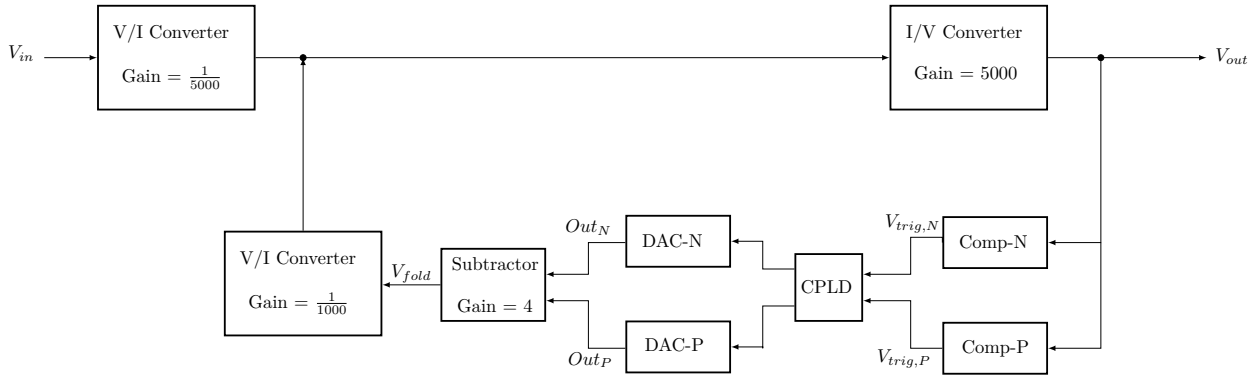


Figure 33: Schematic of the implentation with gain for each block

Figure 34 shows the result of a simulation performed using the final selected components and resistor values. Annexes B and C provide the complete circuit, including all component values, decoupling capacitors, and related elements. Since it is not possible to simulate the behavior of the DACs and the CPLD in this environment, only the analog part of the circuit is simulated. The DACs output are therefore emulated using a predefined voltage table, as illustrated in Figure 31. The input signal is a 4 V sinusoidal signal at 1kHz, therefore it should be folded twice. After two folding operations, the final output signal remains perfectly confined within the range [-1,1]. The signal is inverted with respect to the input due to the action of the output I/V converter.

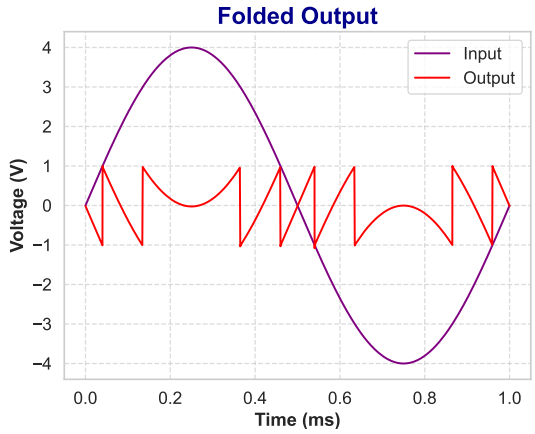


Figure 34: LTspice Simulation: Input and Output for the analog parts sized

3.6 Theoretical Feedback Delay

As discussed in the Section 2.3.2, the feedback delay of the loop has a significant impact on the maximum frequency at which the PCB can correctly fold the signal within the desired range.
The relationship between the maximum frequency and the delay is obtained using a Lipschitz property and can be expressed as:

$$f_{max} = \frac{\Delta.\lambda}{T_d \cdot 2\pi \cdot A} \tag{53}$$

where Δ is the margin factor accounting for the signal variation during the feedback delay, T_d is the feedback delay, and A is the amplitude of the signal.
Thus, the maximum frequency f_{max} is inversely proportional to the feedback delay: the larger the delay, the lower the maximum achievable frequency. Therefore, it is important to estimate the delay in order to determine the maximum operating frequency depending on the amplitude of the input signal.

The feedback delay depends only on the architecture and the components used. It does not depend on the amplitude or the frequency of the input signal, and can therefore be considered as a fixed parameter of the system.
Datasheets typically provide two types of delay: propagation delay and output settling time. In practice, when available, the output settling time should be used as the theoretical delay, since it indicates when the output has reached its stable value. In contrast, the propagation delay only indicates when the output begins to respond and does not guarantee stability.
Considering each component in the feedback loop, from the comparator inputs to the output of the subtractor, the delay contributions can be approximated as follows:

- **Comparators *LT1016***:
  No settling time is specified for the LT1016, which is typical for comparators due to their latch-like switching behavior rather than linear operation. Instead, only the propagation delay is provided. When supplied with ±5 V, the comparator exhibits a typical propagation delay of 10 ns.

- **CPLD *5M80ZT***:
  Similarly, only the propagation delay is specified in the datasheet, with a reference value of 7.5 ns. This value corresponds to the maximum pin-to-pin combinational delay through one logic path inside the device under specific test conditions. Therefore, it does not represent the actual delay of the CPLD in the implemented design but rather indicates the reference speed parameter of the device.
  To estimate the effective delay, the FSM implemented in the CPLD (Annex D) is analyzed.
  The trigger signals $V_{trig,N}$ and $V_{trig,P}$ are asynchronous with respect to the clock. Consequently, they cannot be used directly inside the FSM because this could lead to metastability issues. A synchronization stage is therefore required to safely align these signals with the clock before they are processed by the sequential logic.
  In addition, the FSM must react only once to each trigger event. For this reason, a rising-edge detection stage is implemented after synchronization. Without this mechanism, the FSM could continuously change state while the trigger remains high for several clock cycles.
  These synchronization and edge-detection stages guarantee reliable and predictable operation of the FSM, but they also introduce additional latency between the trigger event and the output transition.

  Figure 35 shows a simplified timing diagram obtained from a ModelSim testbench. In this example, the signal $V_{trig,N}$ is asserted asynchronously, although the behavior is identical for $V_{trig,P}$.
  This timing diagram does not include all internal signals of the Verilog implementation but only the signals relevant to the delay analysis.
  Since $V_{trig,N}$ changes asynchronously, the synchronized signal V_{sync} becomes valid only at the next rising clock edge. The rising edge of the synchronized trigger is then detected using the expression $V_{sync}\&\neg V_{sync,prev}$, where $V_{sync,prev}$ corresponds to the delayed version of V_{sync} by one clock cycle. Once the rising edge is detected, the FSM output is updated at the following clock edge. Therefore, two clock cycles are required between the asynchronous trigger event and the output transition.

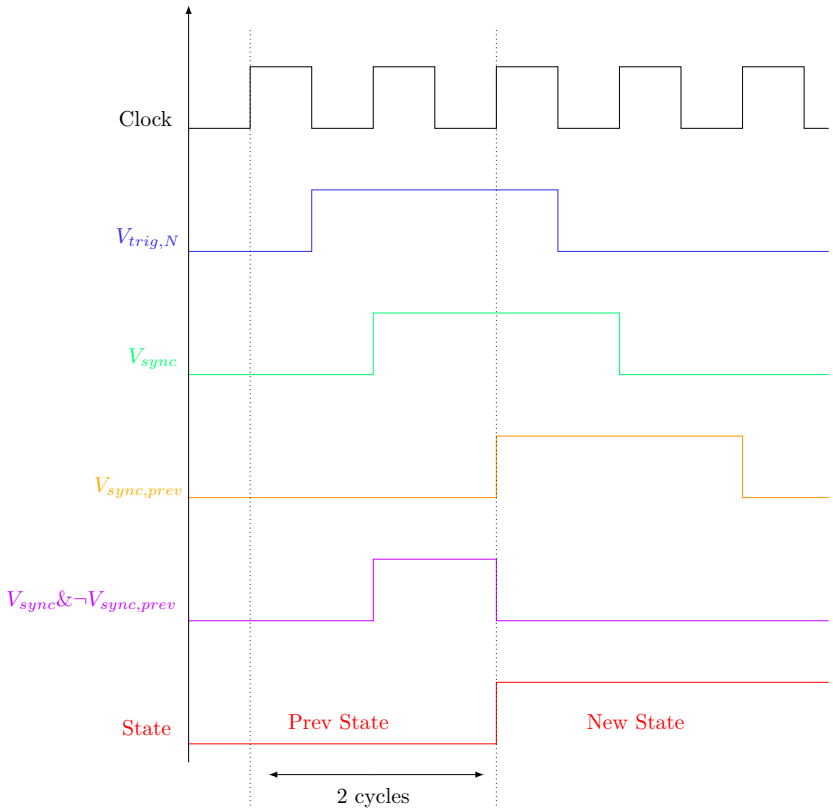


Figure 35: ModelSim Simulation: Simplified timing diagram of the synchronization and edge-detection process implemented in the CPLD.

In addition, an extra delay t_{logic} must be considered to account for the propagation of register values to the output pins.

The total CPLD latency can therefore be approximated by:

$$t_{delay} \approx 2 \times T_{clk} + t_{logic} \tag{54}$$

where T_{clk} is the clock period.

The clock used in this implementation is a 40 MHz oscillator located on the PCB, resulting in a clock period of:

$$T_{clk} = \frac{1}{f_{clk}} = 25 \text{ ns}$$

Assuming the worst-case value $t_{logic} = 7.5$ ns, the estimated CPLD delay becomes:

$$t_{delay} \approx 2 \times 25 + 7.5 = 57.5 \text{ ns}$$

- **DACs *MAX5190***:
  The datasheet specifies an output settling time approximately equal to one clock cycle. With a 40MHz clock, this corresponds to about 25 ns.
  Additionally, an ADA4857 op-amp is used to convert the differential DAC output to a single-ended signal. When supplied with ±5 V, it exhibits a typical settling time of 15 ns.

- **Op-Amps *LT1813***:
  The subtractor contains one LT1813, but two followers stages are added using also the LT1813, one at the output of each DAC and one at the output of the substractor to avoid any loading problem. Each has a typical settling time of 30ns, resulting in a total contribution of 90ns.

Summing all contributions, the estimated feedback delay is:

$$10 + 57.5 + (25 + 15) + (3 \times 30) = 197.5 \text{ ns}$$

Additional delays due to PCB routing and parasitic effects must also be considered. Therefore, the total feedback delay can be reasonably approximated to about 220 ns.
Figure 36 shows the theoretical distribution of the feedback loop delay.

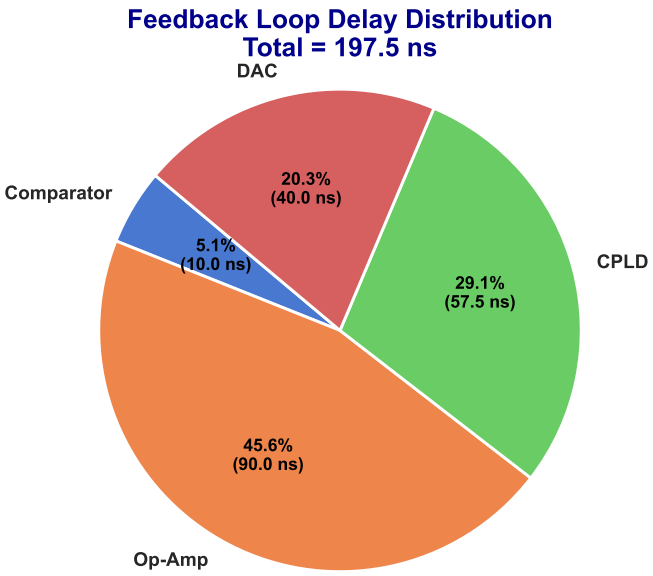


Figure 36: Theoretical Feedback Loop Delay Distribution

The dominant contribution comes from the delay introduced by the LT1813 operational amplifiers. The CPLD and DAC stages also represent significant contributions, mainly due to the clock frequency used in the system.
Consequently, these blocks constitute the primary targets for improvement, as discussed in Section 4.5.2, where several approaches are proposed to increase the maximum frequency.

3.7 PCB Design and Hardware Integration

After validating the architecture and component sizing through simulation, the next step is the hardware implementation in the form of a PCB. The first step consisted in developing the complete schematic of the system including all electronic components and their interconnections. This stage requires careful examination of the datasheets of each component in order to identify their specific requirements, such as decoupling capacitors (used to filter high-frequency noise and stabilize the supply voltage locally), pull-up or pull-down resistors (which define a default logic state on digital inputs when no active driver is present), supply voltage ranges, and recommended operating conditions.
The complete schematic diagrams are provided in Annex B.
During this stage, some modifications are introduced compared to the initial architecture. In particular, an optional external clock input is added to allow the use of alternative clock frequencies if needed. The selection between the internal oscillator and the external clock can be achieved by soldering or removing a series resistor, allowing the oscillator to be disconnected if required.
As a result, the PCB includes seven external connections: the input signal, four power supply inputs (5 V, -5 V, 3.3 V and 1.8 V), the output, and the optional external clock signal. The input and output signals are connected using Bayonet Neill-Concelman (BNC) connectors.
Regarding the power supplies, an alternative approach could consist in providing a single external voltage and generating the required supply voltages locally on the PCB using voltage regulators or DC-DC converters. This solution would reduce the number of external power supplies and cables. However, for a first prototype, this approach is not necessary and it is considered simpler and more reliable to provide the required voltages directly from laboratory power supplies.

Once the schematic completed, the PCB layout is designed.
A two-layer PCB is implemented, consisting of a Top Layer and a Bottom Layer. The stack-up parameters such as copper thickness, dielectric material and board thickness are shown in Figure 37. These parameters correspond to standard values commonly used for two-layer PCBs and are the ones adopted in this work.

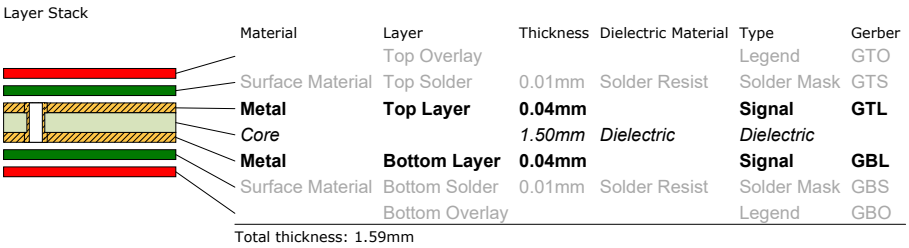


Figure 37: PCB layers stack-up

Two layers are sufficient to route all the signals given the number of components used in the design. Electrical connections between the top and bottom layers are achieved using vias, which are plated holes allowing signals to pass from one copper layer to another.
In order to reduce noise coupling between the analog and digital parts of the system, the Analog Ground (AGND) and Digital Ground (DGND) are initially separated using dedicated ground planes. Ground planes consist of large copper areas connected to the ground potential, which help reduce electromagnetic interference and provide a low-impedance return path for currents. The analog and digital grounds are then connected at a single point using a net connection ensuring a common reference potential for the entire circuit.
The component placement strategy consisted in grouping the components belonging to the same functional block close to each other in order to simplify routing and reduce parasitic effects. At the same time, the routing is optimized to minimize the length of the traces connecting different blocks. Particular attention is paid to high-frequency signals such as the clock signal, since long traces may introduce parasitic capacitance and inductance. Furthermore, all decoupling capacitors are placed as close as possible to the power supply pins of the Integrated Circuits (ICs).

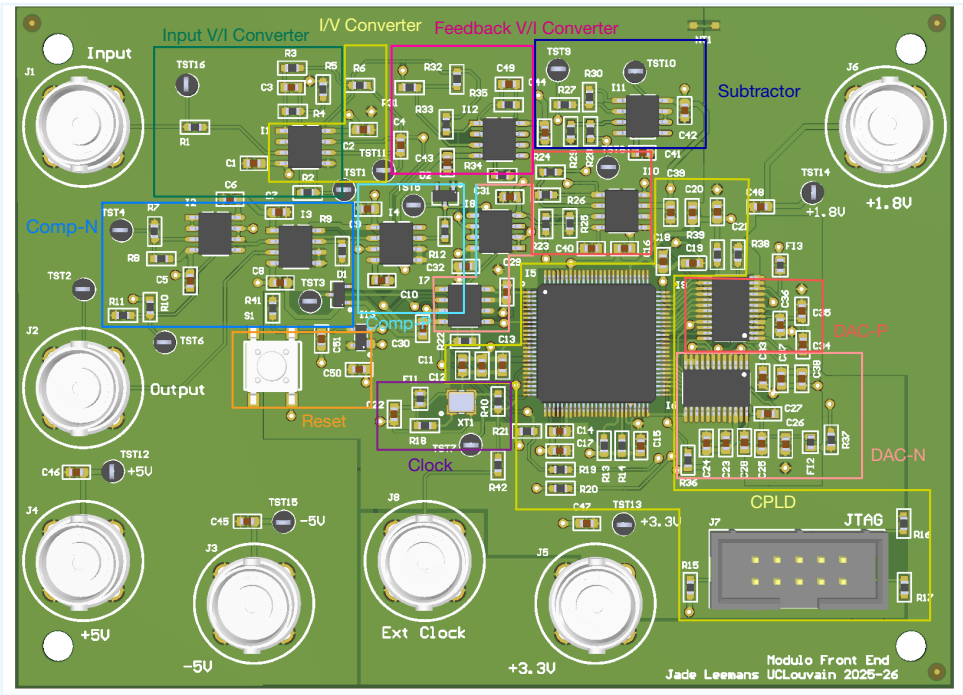


Figure 38: Top view of the PCB layout

For debugging and measurement purposes, several test points are placed at the outputs of the different functional blocks. These test points allow easy probing using an oscilloscope during testing and verification of the system operation.
The placement of the components is also optimized to reduce the overall surface area of the PCB, following common practices used in industrial designs. However, the BNC connectors are intentionally spaced sufficiently apart to allow easy connection of coaxial cables during laboratory measurements.
To improve the mechanical robustness and manufacturability of the PCB, teardrops are added at the junction between traces and pads. Teardrops are small copper extensions that reinforce the connection between a trace and a pad or via, reducing the risk of trace breakage during manufacturing or soldering.
The final PCB layout is shown in Figure 38 with the different blocks highlighted.

3.8 Conclusion

This chapter first presented an existing implementation of the Wideband High-Dynamic Range ADC proposed in [11]. The reported implementation achieved promising performances, supporting input signals with amplitudes up to 10V and frequencies up to 10kHz.
However, these performances remain limited for certain applications, and the implementation also presents several drawbacks, such as the high supply voltages required by some circuit blocks and the relatively large feedback delay.

Based on these observations, the specifications for the implementation proposed in this work were defined. One important requirement was the availability of components compatible with LTspice simulations for all analog blocks, which constrained the achievable input dynamic range to $\pm$5V, limiting the system to a maximum of two folds.
The proposed implementation offers the advantage of reduced supply voltage requirements by operating in both the current and voltage domains. This approach also provides flexibility for increasing the number of folds if required.

Each circuit block of the proposed implementation was designed, characterized using the selected components, and validated through LTspice simulations. The FSM controlling the fold correction was also defined according to the final gain of each block.
Furthermore, in order to maximize the achievable input frequency, the feedback delay was minimized and theoretically estimated, based on datasheet values, to be approximately 197.5ns, or around 220ns when including additional layout-related delays.
Finally, the PCB design of the system was presented.

Now that the proposed implementation and the theoretical background of each building block have been introduced, the next chapter focuses on the experimental verification of the hardware implementation and provides an in-depth analysis of its performance.

4

Experimental Results and Performance Analysis

In the previous chapter, the specifications of the hardware implementation were established. The design was optimized to minimize the supply voltage of the different blocks while maximizing the input range, and to reduce the feedback delay in order to maximize the achievable input frequency.
After introducing the global architecture, each block was analyzed in detail, including the underlying theoretical principles, the selected components, and the corresponding LTspice simulations.
The gain sizing of the different blocks was then determined, followed by an estimation of the theoretical feedback delay and a brief presentation of the PCB design.

Although simulations are essential for validating the theoretical operation of the system, experimental validation is necessary to evaluate the actual performance of the implemented architecture. In practice, simulations cannot fully account for non-ideal effects such as component tolerances, PCB propagation delays, noise, parasitic effects, and other implementation constraints.
The objective of this chapter is therefore to analyze the performance of the fabricated PCB under realistic operating conditions.
The chapter is organized as follows. First, the measurement setup is presented.
The functional behavior of the system is then experimentally validated by applying input signals with increasing amplitudes, corresponding to different numbers of folds.
Next, the reconstruction algorithm based on the US framework is analyzed.
The feedback delay is subsequently measured and compared with the theoretical estimation, followed by an evaluation of the maximum operating frequency and a Bode diagram analysis to complement the frequency-domain characterization.
A noise analysis is then carried out. Finally, a static error analysis, based on conventional data-converter theory, is performed to identify potential hardware-induced errors, followed by an evaluation of the effective resolution of the system.
The chapter concludes with a discussion of possible improvements to further increase the input dynamic range and the maximum operating frequency.

4.1 Measurement Setup

To ensure consistent and comparable results throughout all performance evaluations, the same experimental setup is used for all measurements.
The setup consists of three main measurement instruments connected as shown in Figure 39:

- Power supply: *N6705B DC Power Analyzer* [30], used to generate the four supply voltages: +5 V, -5 V, 3.3 V, and 1.8 V.
- Waveform generator: *Tektronix AFG2021* [31], used to generate the input signal.
- Oscilloscope: *Tektronix 5 Series Mixed Signal Oscilloscope* [32], used to monitor and record all relevant signals.

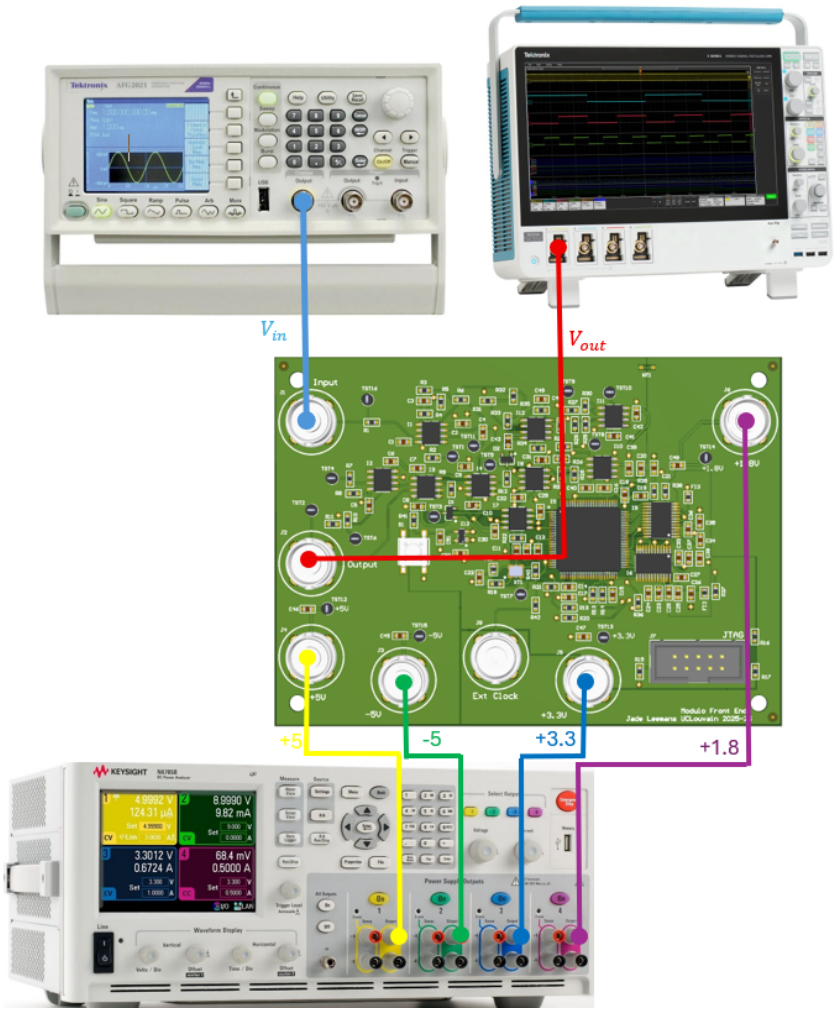


Figure 39: Measurement setup

All external signals, power supply connections, and output signals are connected using BNC cables. When internal or intermediate signals need to be observed, standard passive probes are connected to dedicated PCB test points. Only the clock signal is measured using an active probe, since the capacitance of a passive probe can distort high-speed signals. The measured clock signal is presented in Annex F. As the waveform is clean and the frequency is correctly measured at 40MHz, the BNC connector intended for a potential external clock input is not used.

Oscilloscope data are captured in Python through a LAN connection, allowing automated signal processing and generation of the plots presented in the following sections.
For each measurement, the main configuration parameters, such as the input signal characteristics and oscilloscope settings, are summarized in a dedicated table. In general, a sampling frequency of 25MHz is used in order to satisfy the sampling frequency condition which ensures accurate reconstruction of the original signal from its modulo samples (5).

The oscilloscope acquisition resolution is set to 16 bits in order to maximize the precision of the measurements. In this configuration, the oscilloscope acts similarly to a high-resolution ADC, since a 16-bit resolution means that the oscilloscope can represent 2^{16} different amplitude values. Using a lower resolution, or emulating a lower-resolution ADC in Python before processing the data and generating the graphs, reduces the number of possible output values. As a consequence, the measured curves contain fewer amplitude levels and the resulting graphs appear less precise.
Moreover, the vertical scale of the oscilloscope is adjusted such that the measured signal occupies most of the screen height without clipping. This allows the full resolution range of the oscilloscope to be used, thereby maximizing the measurement precision.

For some tests, the arbitrary waveform generation mode of the waveform generator is used. This mode allows custom signals to be generated from uploaded waveform files. These files are created using the *ArbExpress Application* [33], which converts CSV files generated in Python into a format compatible with the waveform generator. This approach enables the generation of more complex waveforms, such as multi-tone sine signals or staircase signals, as presented in the following sections.

4.2 Functional Validation

This section aims to verify the correct functional operation of the circuit by analyzing the different signals throughout the system, including those of the intermediate building blocks and the resulting folded output signal.
The analysis begins with the case where the input signal already lies within the range $[-1, 1]$V, followed by cases requiring one fold and two folds.
Finally, the accuracy of the reconstruction of the original signal from the modulo samples at the output is evaluated.

4.2.1 Operation Without Folding

A first verification step consists in testing the system when the input signal is already contained within the interval $[-1, 1]$V. In this situation, no folding operation occurs, and the output signal simply corresponds to the inverted version of the input signal.
Therefore, Comp-P and Comp-N never trigger, since the input signal does not exceed the defined thresholds. Consequently, the CPLD and the DACs remain inactive, and their outputs remain at zero.

Parameter	Value	Unit	Notes
Input amplitude	0.5	V	Sinusoidal signal at 1 kHz
Channel	CH1	–	Input channel
Channel	CH2	–	Output channel
Vertical scale	100	mV/div	Same setting for CH1 and CH2
Sample rate	25	MS/s	Instrument sampling rate
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting
Trigger mode	Positive Edge	–	Trigger configuration
Trigger level	0	V	Trigger threshold on CH1

Table 6: Measurement configuration used for Figure 40

Figure 40 shows the measured input and output signals obtained using the configuration listed in Table 6. As expected, the output signal follows the same waveform as the input signal, remains within the range $[-1, 1]$V, and is simply inverted.

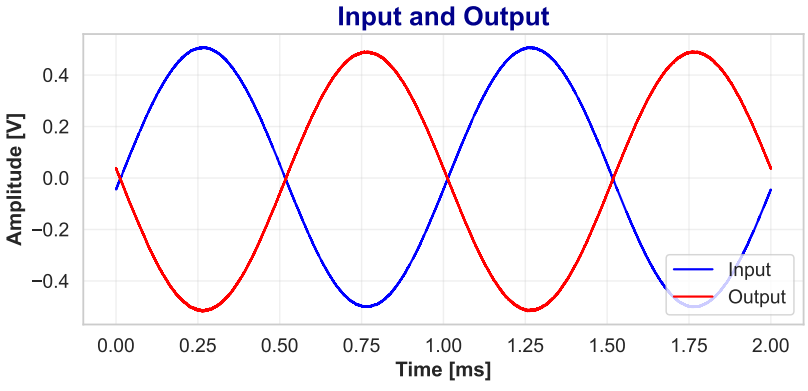


Figure 40: Measurements: Input and output signals for a 0.5 V amplitude, 1 kHz sinusoidal input

4.2.2 Operation with one fold

After verifying that the architecture operates correctly when no fold is required, the next step is to test its behavior when a single fold occurs.
For this test, a sinusoidal input signal with an amplitude of 2V and a frequency of 1kHz is applied. Since the signal amplitude is only twice the threshold value, only one fold occurs, corresponding to correction states of +2V and -2V.

Parameter	Value	Unit	Notes
Input amplitude	2	V	Sinusoidal signal at 1 kHz
Channel	CH1	–	Input channel
Channel	CH2	–	Output channel
Channel	CH3	–	$V_{trig,N}$
Channel	CH4	–	Dac-N Output
Channel	CH5	–	Dac-P Output
Channel	CH6	–	PVG Output
Vertical scale	440	mV/div	Setting for CH1
Vertical scale	240	mV/div	Setting for CH2
Vertical scale	700	mV/div	Setting for CH3
Vertical scale	25	mV/div	Setting for CH4
Vertical scale	25	mV/div	Setting for CH5
Vertical scale	85	mV/div	Setting for CH6
Sample rate	25	MS/s	Instrument sampling rate
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting
Trigger mode	Positive Edge	–	Trigger configuration
Trigger level	0	V	Trigger threshold on CH1

Table 7: Measurement configuration used for Figure 41

The measured input and folded output signals are shown in Figure 41, with the settings listed in Table 7.

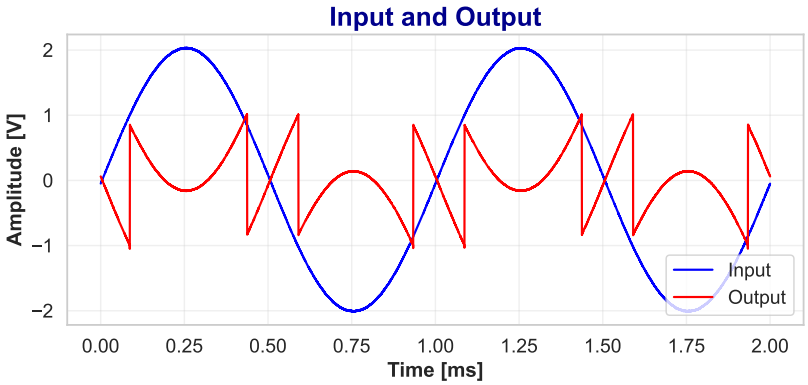
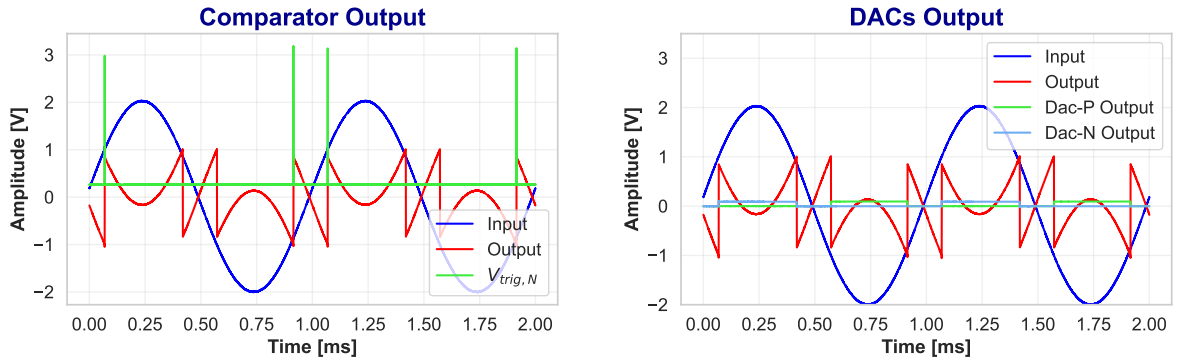


Figure 41: Measurements: Input and output signals for a 2 V amplitude, 1 kHz sinusoidal input

The output signal is correctly folded and inverted. However, two main observations can be made. First, the output slightly overshoots, reaching approximately +1.01 V and -1.01 V. Second, when a fold occurs, the signal is not brought exactly back to ±1 V, but instead closer to ±0.9 V. These two effects can be attributed to the delay present in the feedback loop.
As discussed previously, a finite amount of time is required for the correction signal to be generated after a threshold crossing is detected. During this delay, the input signal continues to evolve before the correction is applied.
Since the signal slightly exceeds ±1 V before the correction is applied, adding a correction voltage of ±2 V does not return the signal exactly to ±1 V, introducing a slight asymmetry between positive and negative

folds. As discussed in the following section on the reconstruction algorithm, this asymmetry does not affect the final signal reconstruction.

The intermediate signals of the other circuit blocks can also be measured using the same configuration listed in Table 7.
The output of the V/I converter cannot be directly measured, since it converts the signal into current and therefore corresponds to very low voltage levels.
Figure 42a shows the output of Comp-N, denoted $V_{trig,N}$. It can be observed that whenever the output reaches -1V, $V_{trig,N}$ goes high and reaches approximately 3.3V.
The behavior of Comp-P is similar, except that it goes high when the output exceeds 1V.
Figure 42b shows the outputs of both DACs.
When only one fold is needed, the output of the DACs take the form of a step-like waveform alternating between 0V and 100mV.
Both DACs are never simultaneously non-zero, but they can both be zero at the same time when the input signal remains within the valid range and does not require correction.
Within one signal period, $V_{trig,N}$ goes high twice. For instance, in the interval from 0.75ms to 1.25ms, Comp-N triggers transitions first from the +2V correction state (where DAC-P outputs 100mV) to the no-correction (IDLE) state (both DACs at zero), and then from the IDLE state to the -2V correction state (where only DAC-N outputs 100mV).



(a) Input, Output and $V_{trig,N}$

(b) Input, Output, and DACs Output

Figure 42: Measurements: 2 V amplitude, 1 kHz sinusoidal input

Figure 43 shows the output of the PVG, specifically the output of the subtractor that combines the outputs of the DACs. Since only one fold is required, the PVG output takes values of either 0V or ±400mV in a step-like waveform.
Consequently, the FSM transitions between only three states: IDLE, +2 V correction, and -2 V correction.

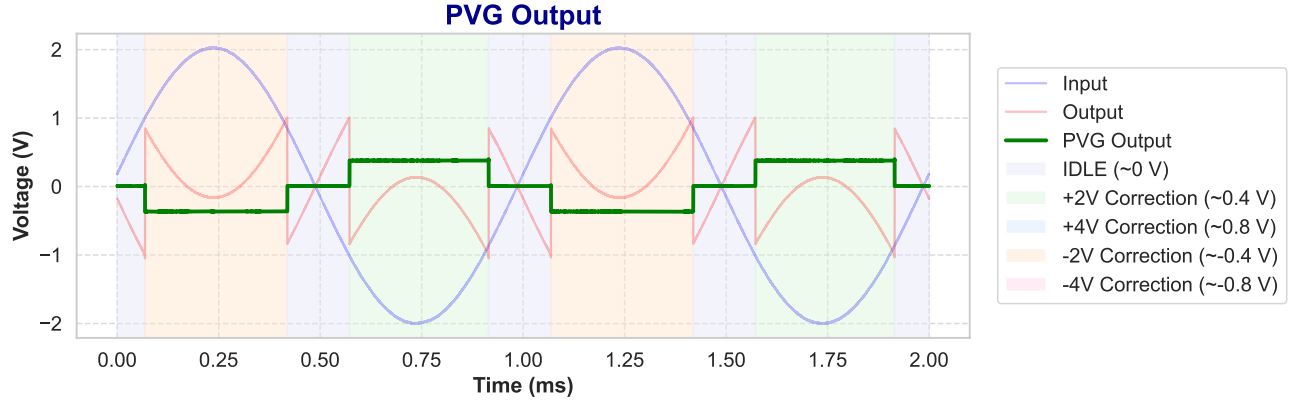


Figure 43: Measurements: 2V amplitude, 1kHz sinusoidal input

4.2.3 Operation with two folds

To evaluate the system behavior when two folds occur, the amplitude of the input signal must be increased. This condition can be achieved, for example, with an input amplitude of 4 V. Under these conditions, the system goes through the full set of FSM states.

Parameter	Value	Unit	Notes
Input amplitude	4	V	Sinusoidal signal at 1 kHz
Channel	CH1	–	Input channel
Channel	CH2	–	Output channel
Channel	CH3	–	$V_{trig,N}$
Channel	CH4	–	Dac-N Output
Channel	CH5	–	Dac-P Output
Channel	CH6	–	PVG Output
Vertical scale	850	mV/div	Setting for CH1
Vertical scale	240	mV/div	Setting for CH2
Vertical scale	700	mV/div	Setting for CH3
Vertical scale	45	mV/div	Setting for CH4
Vertical scale	45	mV/div	Setting for CH5
Vertical scale	165	mV/div	Setting for CH6
Sample rate	25	MS/s	Instrument sampling rate
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting
Trigger mode	Positive Edge	–	Trigger configuration
Trigger level	0	V	Trigger threshold on CH1

Table 8: Measurement configuration used for Figure 44

The measured input and output signals are shown in Figure 44, using the configuration parameters listed in Table 8.

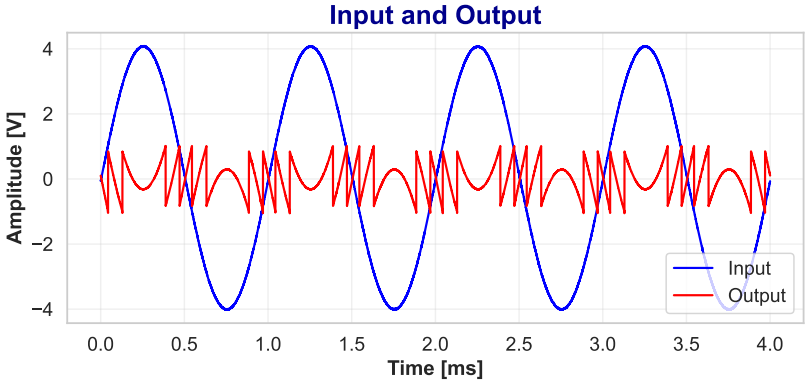
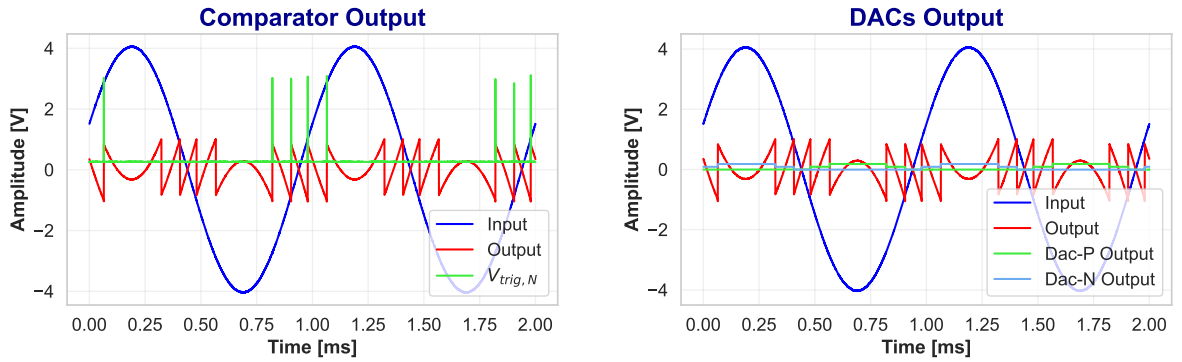


Figure 44: Measurements: Input and output signals for a 4 V amplitude, 1 kHz sinusoidal input

As in the 2 V amplitude case, the feedback delay slightly affects the exact folding thresholds. Nevertheless, the output signal remains correctly folded and largely confined within the desired range.
Figure 45a shows the output of Comp-N $V_{trig,N}$.
The DAC outputs take values of 0 V, 100 mV, or 200 mV, forming a step-like waveform, as shown in Figure 45b. In this case, the comparator is triggered four times per signal period. Considering the interval from 0.75ms to 1.25ms, the trigger causes successive transitions from the +4 V correction state (DAC-P outputs 200mV) to the +2 V correction state (DAC-P outputs 100mV), then to the IDLE state (both DACs at zero), followed by the −2 V (DAC-N outputs 100mV) and finally the −4 V correction state (DAC-N outputs 200mV).
The reverse sequence of FSM transitions is initiated by Comp-P.



(a) Input, Output and $V_{trig,N}$

(b) Input, Output, and DACs Output

Figure 45: Measurements: 4 V amplitude, 1 kHz sinusoidal input

Since two folds are required, the subtractor output is expected to take values of 0 V, ±400 mV, or ±800 mV. This results from the gain factor of 4 between the DACs output and the subtractor output, combined with the fact that only one DAC output is non-zero at a time, allowing both positive and negative correction levels depending on the FSM state.
This behavior is observed in Figure 46, confirming that the FSM transitions through all possible states.

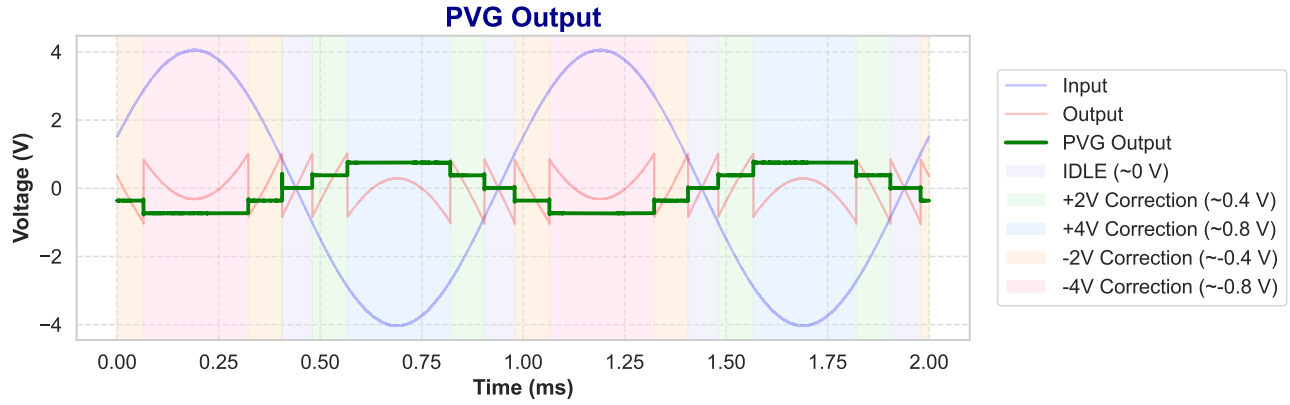


Figure 46: Measurements: 4 V amplitude, 1 kHz sinusoidal input

It is important to note that the maximum usable input amplitude is approximately ±4.6 V rather than ±5 V. Although the operational amplifier used in the input V/I converter is supplied with ±5 V, measurements show that when a sinusoidal input signal with an amplitude exceeding approximately 4.6 V is applied, the output signal is no longer correctly folded.
This behavior can be explained by the limited output swing and nonlinear operation of the operational amplifier close to the supply rails. As the input amplitude approaches the supply voltage, the amplifier progressively leaves its linear operating region, introducing distortion and preventing the V/I converter from generating the correct output. Consequently, reliable operation is only ensured up to approximately ±4.6 V.

4.2.4 Recovery Algorithm

Once the folded signal is captured at the output of the PCB, a recovery algorithm can be applied to reconstruct the original signal.
The reconstruction algorithm is implemented in Python (see Annex E) and is based on the method proposed in [4]. It relies on the Modular Decomposition Property and finite differences, as described in Chapter 2. According to this property, the original signal can be decomposed as follows:

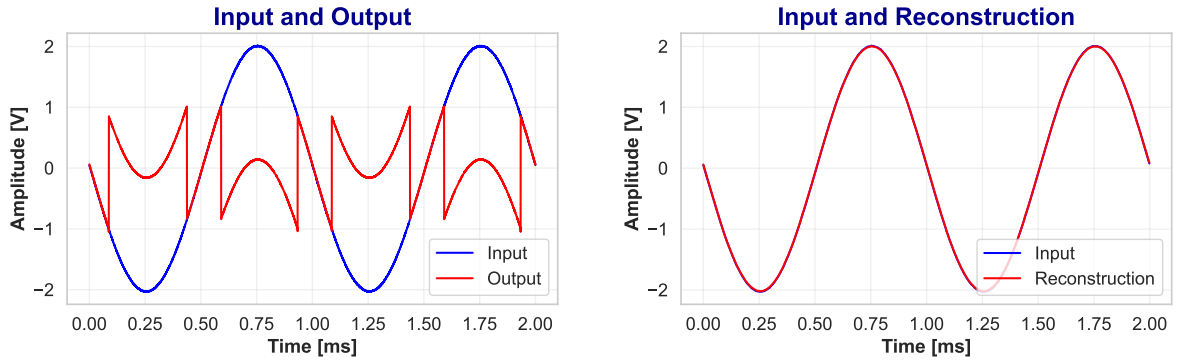
$$g(t) = y(t) + \epsilon_g(t) \tag{55}$$

where $y(t) = M_\lambda(g(t))$ is the folded signal, and the residual function $\epsilon_g(t)$ is a discontinuous function taking values in $2\lambda\mathbb{Z}$.
By applying the N-th order finite difference, it can be shown [4] that:

$$\Delta^N g = M_\lambda(\Delta^N y) \tag{56}$$

The original signal $g(t)$ can then be recovered by applying a cumulative sum.

Using this approach, the original signal can be reconstructed from the modulo output of the PCB. Figure 47 shows the reconstructed signal for a 2 V amplitude, 1 kHz sinusoidal input, using the same parameters as listed in Table 7. For visualization purposes, the input signal is inverted to compensate for the inversion introduced by the output I/V converter.



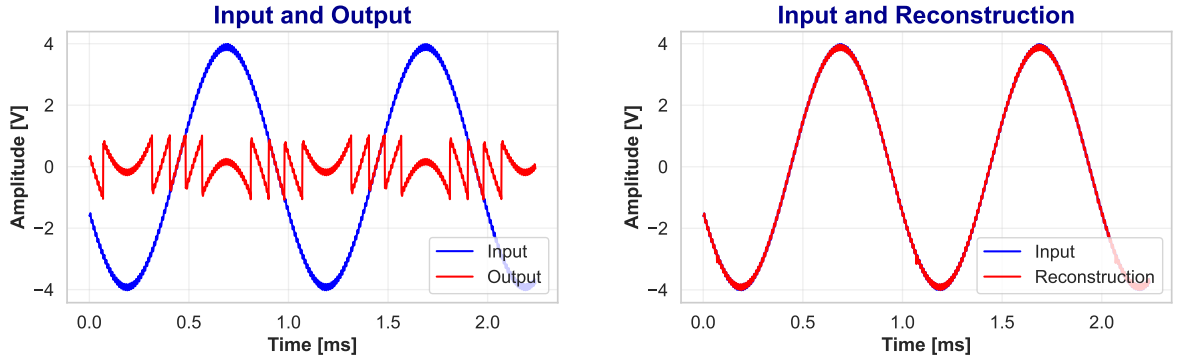
(a) Input and Output Signals (b) Input and Reconstructed Signals

Figure 47: Measurements: Reconstruction for a 2 V amplitude, 1 kHz sinusoidal input

This method is also applicable to more complex signals, such as sums of sinusoids with different frequencies and amplitudes. For instance, the following input signal can be successfully reconstructed:

$$V_{\text{in}}(t) = 4\sin(2\pi \cdot 10^3 t) + 0.1\sin(2\pi \cdot 10^5 t) \tag{57}$$

The corresponding reconstruction is shown in Figure 48, using the same parameters as those listed in Table 8, except for the input signal itself.



(a) (b)

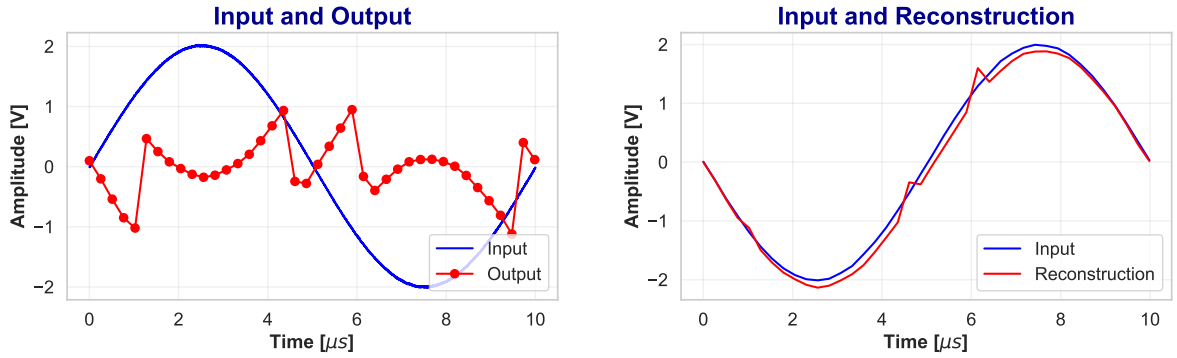
Figure 48: Measurements: Reconstruction of a multi-tone input signal

In practice, the direct application of this algorithm to measured signals may fail due to hardware non-idealities. The US Framework assumes that the modulo operation is applied instantaneously. However, in a real implementation, the feedback loop introduces a finite delay and the electronic components exhibit a limited slew rate. As a result, folding transitions occur over a finite time interval rather than instantaneously.

During these transition intervals, samples may be acquired that do not satisfy the theoretical assumptions. In particular, the residual function $\epsilon_g(t)$ may temporarily take values outside $2\lambda\mathbb{Z}$. Consequently, the condition $M_\lambda(\Delta^N \epsilon_g) = 0$ is no longer guaranteed, and (56) becomes:

$$\Delta^N g = M_\lambda(\Delta^N y) + M_\lambda(\Delta^N \epsilon_g) \tag{58}$$

which introduces reconstruction errors.
An example of this effect is shown in Figure 49 for a 2V, 100kHz sine-wave input, where some samples are acquired during a folding transition, leading to visible distortions in the reconstructed signal. A similar artifact can also be observed in Figure 48b, where a spike appears around 1ms.

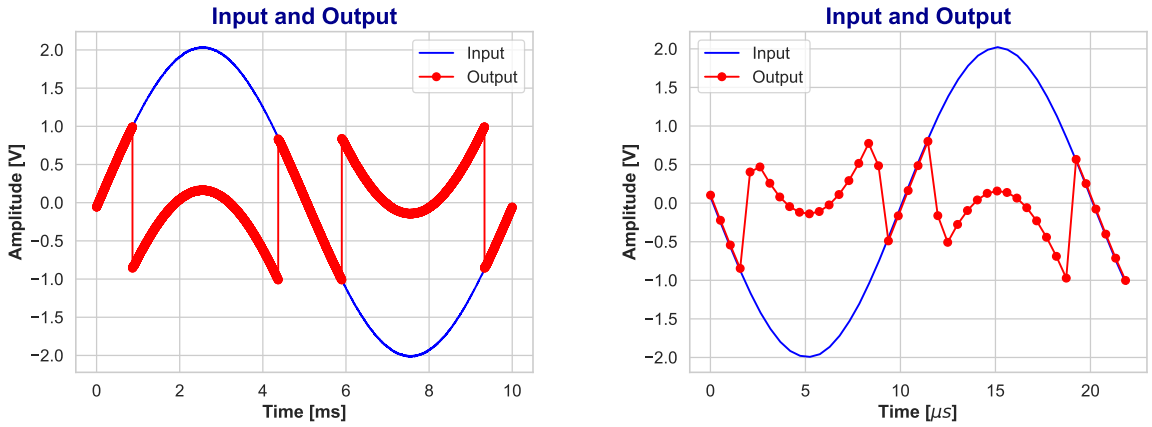


(a) 2V 100kHz Input and Output signals

(b) 2V 100kHz Input and Reconstructed signals

Figure 49: Measurements: Reconstruction of an imperfect modulo signal

To mitigate this issue, the signal can be downsampled prior to reconstruction so that no samples are taken during folding transitions.
This effect is illustrated in Figure 50a. For a 2 V, 100 Hz sinusoidal input sampled at 25 MHz and downsampled by a factor of 25, the signal evolves slowly and the folding transitions represent a negligible portion of the period. Therefore, it is unlikely that any sample falls within a transition interval.
In contrast, for a 2 V, 50 kHz input under the same sampling conditions, the folding transitions represent a much larger fraction of the signal period. Consequently, downsampling by the same factor still results in samples being taken during transitions, which leads to reconstruction artifacts, as shown in Figure 50b.



(a) 2V 100Hz Input signal and Output signal downsampled by 25

(b) 2V 50kHz Input signal and Output signal downsampled by 25

Figure 50: Measurements: Comparison of downsampled output signals by a factor of 25 for low- and high-frequency inputs

To avoid sampling during folding transitions, the downsampling factor can be increased. However, this reduces the number of available samples and leads to a less smooth reconstruction. Therefore, a

trade-off must be established between avoiding transition samples and maintaining sufficient temporal resolution. This trade-off can be addressed by selecting the downsampling factor and the sampling offset (i.e., the index at which samples start to be retained) such that the reconstruction accuracy is maximized. To quantify this accuracy, the Signal-to-Reconstruction Error Ratio (SRER) [6] is used as an evaluation metric. The SRER quantifies the quality of a reconstructed signal with respect to the original one. It is defined as the ratio between the power of the original signal and the power of the reconstruction error. It is expressed in dB as:

$$\mathrm{SRER} = 10 \log_{10} \left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{error}}} \right) \tag{59}$$

where $P_{\mathrm{signal}} = \sigma^2_{\mathrm{signal}}$ is the variance of the original signal and $P_{\mathrm{error}} = \sigma^2_{\mathrm{error}}$ is the variance of the reconstruction error, defined as:

$$e[n] = x[n] - \hat{x}[n] \tag{60}$$

where $e[n]$ denotes the reconstruction error, $x[n]$ the original signal, and $\hat{x}[n]$ the reconstructed signal.

In [6], a SRER of at least 60 dB is considered to indicate very high accuracy for simulated signals. However, such simulations typically do not include noise or hardware non-idealities. Therefore, in practice, when working with real measured signals, the SRER is expected to be lower.
In the example of Figure 47, the resulting SRER is 51.12 dB, which can still be considered a highly accurate reconstruction for real measurements.

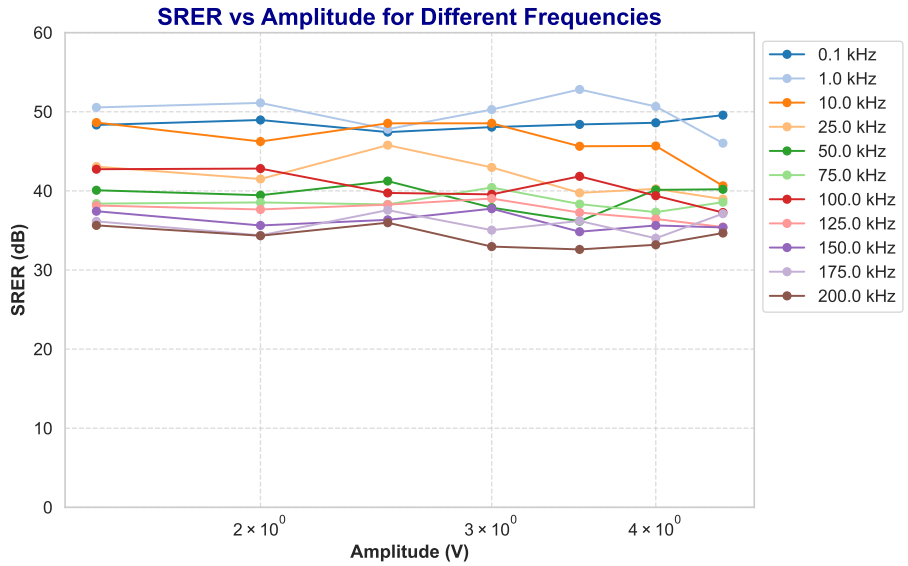


Figure 51: Measurements: SRER as a function of amplitude for sinusoidal input signals

Figure 51 shows the evolution of the SRER as a function of signal amplitude for different input frequencies. The SRER remains approximately constant with respect to amplitude, as expected. However, it decreases as the input frequency increases.
For instance, for a 2 V sinusoidal input, the SRER decreases from 48.96 dB at 100 Hz to 42.82 dB at 1 kHz, with a more pronounced degradation at higher frequencies. This behavior is not mainly caused by intrinsic PCB limitations, but rather by the increasing difficulty of avoiding samples acquired during folding transitions as the signal frequency increases, even when the downsampling factor and the selected sampling index are optimized to maximize the SRER.
More advanced reconstruction algorithms have been proposed to address non-ideal modulo behavior with finite transition times [2]. However, since the objective of this work is to evaluate the hardware implementation rather than optimize reconstruction algorithms, the method presented here is sufficient.

4.3 Delay and Frequency Analysis

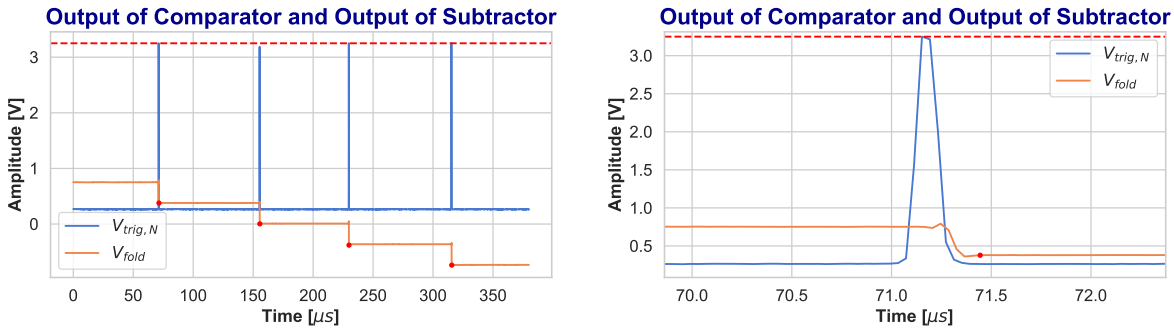
Now that the functional operation of the hardware implementation has been verified, the feedback delay can be measured and compared with the theoretical value obtained in Chapter 3.
Based on these results, the maximum operating frequency for which the output remains within the range $[-1.25, 1.25]$ V can be determined.
In addition, a gain Bode diagram is obtained to characterize the passband of the system.

4.3.1 Measured Feedback Delay

After estimating the theoretical delay in Section 3.6, the actual feedback delay is evaluated by measuring signals directly on the PCB and comparing the results with the theoretical value of 197.5ns.
Since accurately measuring all signal transitions is difficult, especially for the CPLD digital outputs, the total feedback-loop delay is estimated by measuring the time difference between:

- the moment when the comparator output switches and reaches its high level (3.3V), and
- the moment when the subtractor output settles to a stable value corresponding to a new FSM state.

Figure 52 shows the Comp-N output $V_{trig,N}$ and the subtractor output V_{fold} for a 4V input signal at 1kHz, using the same parameters as those listed in Table 8. Figure 52b presents a zoom on one of these impulses.



(a) Multiple trigger events over half a period

(b) Zoom on a single trigger event

Figure 52: Measurements: Delay between the comparator trigger and the resulting change at the subtractor output.

The measured delay ranges from 230ns to 250ns, which remains reasonably close to the estimated theoretical delay of 197.5ns when accounting for the additional propagation delays introduced by the PCB layout.

4.3.2 Maximum Frequency

Once the delay has been determined, the maximum operating frequency can be estimated using:

$$f_{max} = \frac{\Delta.\lambda}{T_d \cdot 2\pi \cdot A} \tag{61}$$

In this expression, two parameters are not fixed: the input amplitude A and the margin factor Δ.
Assuming a delay of $T_d = 197.5\,\text{ns}$, the theoretical maximum frequencies for different input amplitudes and values of Δ are given in the Table 9.

Amplitude	$\Delta = 0.01$	$\Delta = 0.1$	$\Delta = 0.25$	$\Delta = 0.5$
1.5 V	5.3 kHz	53.7 kHz	134 kHz	268.6 kHz
2 V	4 kHz	40.3 kHz	100.7 kHz	201.5 kHz
2.5 V	3.2 kHz	32.2 kHz	80.6 kHz	161.2 kHz
3 V	2.6 kHz	26 kHz	67.2 kHz	134.3 kHz
3.5 V	2.3 kHz	23 kHz	57.6 kHz	115.1 kHz
4 V	2 kHz	20.1 kHz	50.4 kHz	100.7 kHz
4.5 V	1.8 kHz	18 kHz	44.8 kHz	89.5 kHz

Table 9: Theoretical maximum frequency for different amplitudes and overshoot margins.

Both (61) and Table 9 highlight that keeping the signal strictly within $[-\lambda, \lambda]$ is either impractical or only possible at very low frequencies.
For example, when $\Delta = 0.01$, the allowed overshoot is limited to $\Delta.\lambda = 0.01$V, meaning that the output must remain within $[-1.01, 1.01]V$. Under this constraint, the maximum frequency remains very limited, not exceeding 5.3kHz for a 1.5V input amplitude.
For $\Delta.\lambda = 0.1$V, the achievable frequencies already reach several tens of kilohertz. This becomes even more pronounced for $\Delta.\lambda = 0.25$V and $\Delta.\lambda = 0.5$V, where frequencies in the hundreds of kilohertz range are achievable.
These observations highlight the importance of choosing threshold values smaller than the actual ADC limits to guarantee that the signal remains within the measurable range.
In this work, the thresholds are set to $\pm\lambda = \pm 1$, while the ADC input range is $[-1.25, 1.25]$V (see Table 1).
From Table 9, the corresponding theoretical maximum frequency lies approximately between 44.8 kHz (for $A = 4.5$ V) and 134 kHz (for $A = 1.5$ V).

The overshoot at the output is measured using the method illustrated in Figure 53.
This figure shows the output signal for a 2 V, 100 kHz sinusoidal input. The overshoot $\Delta.\lambda$ is defined as the absolute difference between the maximum value of the folded signal and the threshold. In this example, $\Delta.\lambda$ is equal to 0.156V.

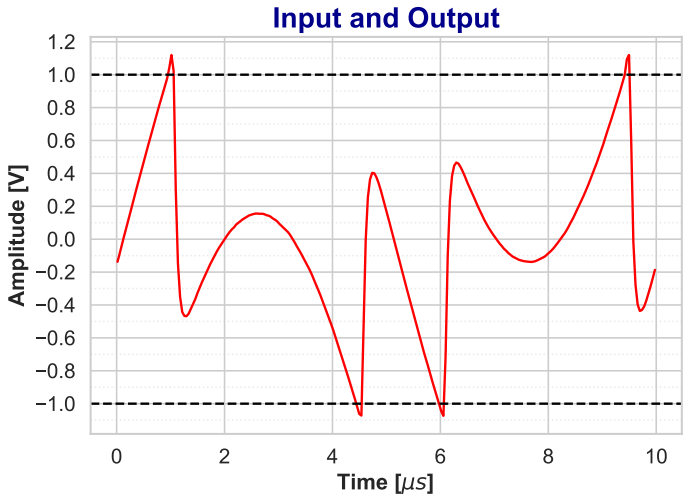


Figure 53: Measurements: Overshoot margin for a 2 V, 100 kHz sinusoidal input

Figure 54 presents the measured maximum overshoot at the output as a function of the input amplitude for sinusoidal signals at different frequencies. The results show that the overshoot increases with both the input amplitude and the signal frequency, with a more pronounced increase at higher amplitudes and frequencies.

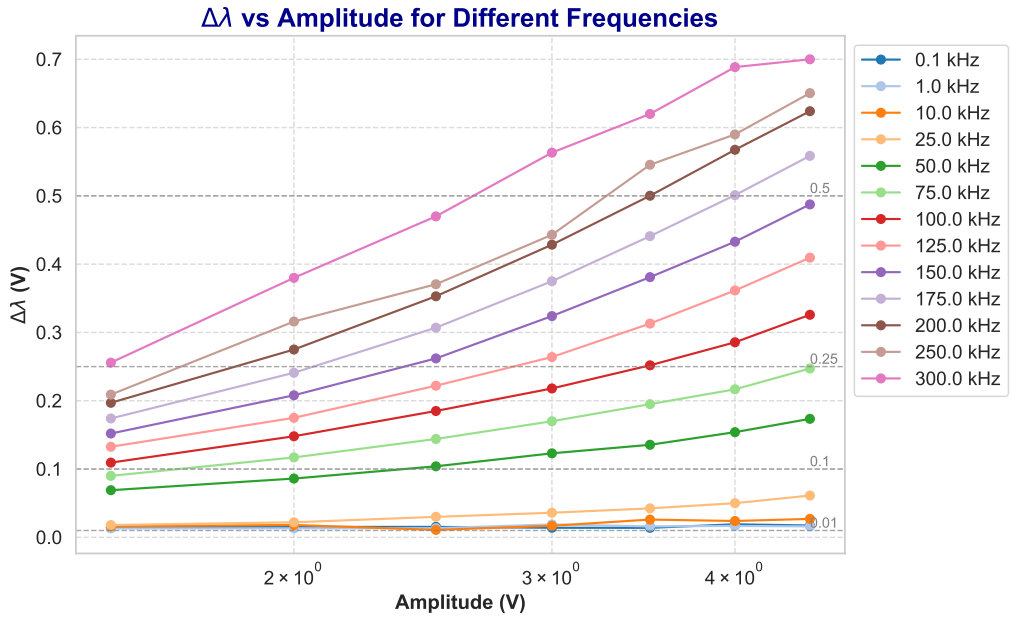


Figure 54: Measurements: Overshoot margin as a function of amplitude for sinusoidal input signals of different frequencies

Based on Figure 54, the table below provides the maximum input frequency for each amplitude such that the output remains within $[-1.25, 1.25]$V.

Amplitude	Maximum Frequency
1.5 V	270kHz
2 V	180 kHz
2.5 V	140 kHz
3 V	115 kHz
3.5 V	100 kHz
4 V	85 kHz
4.5 V	75 kHz

The experimental results show higher maximum frequencies than predicted by the theoretical model. This can be explained by the fact that (61) provides a theoretical lower bound based on the maximum slope achievable by a sinusoidal signal, as discussed in Section 2.3.3 on the maximum achievable frequency [11]. Consequently, practical performance can exceed this estimate.

4.3.3 Bode diagram

To determine the usable frequency range of the circuit a Bode diagram is used.
The Bode diagram is obtained by measuring the input and output signals for a sinusoidal input with an amplitude ranging from -0.5 V to 0.5 V over a range of frequencies. This amplitude is chosen to avoid any folding, which eliminates the need for a reconstruction algorithm. As discussed previously, reconstruction becomes more challenging at higher frequencies due to samples acquired during folding transitions, which can introduce distortions in the reconstructed signal.
The resulting gain Bode diagram is shown in Figure 55.

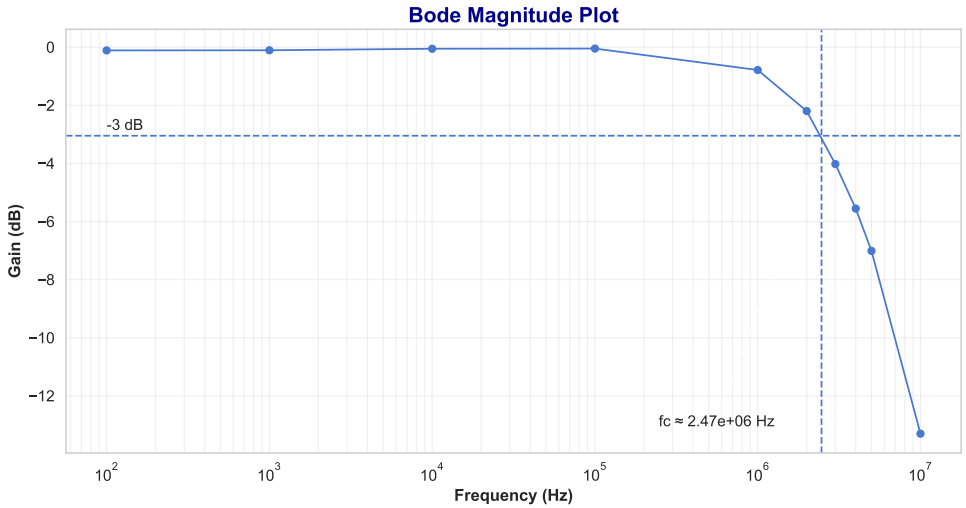


Figure 55: Measurements: Gain Bode diagram for a 0.5 V sine input

From the Bode diagram, the passband extends from DC up to approximately 2.5 MHz, where the gain remains close to 0 dB. The corresponding cutoff frequency is therefore approximately 2.5 MHz.
Beyond this frequency, the circuit exhibits a low-pass behavior, resulting in a progressive attenuation of the output voltage.
This cutoff frequency is not limited by the operational amplifiers used in the circuit. In particular, the LT1813 has a gain-bandwidth product of approximately 100 MHz, which is significantly higher than the measured cutoff frequency.
The observed limitation instead arises from the circuit itself, more specifically from the I/V converter at the output described in Section 3.4.2. As previously explained, this stage includes a feedback capacitor placed in parallel with a resistor, thereby introducing a low-pass filter behavior.
The simulated Bode magnitude response at the output of the I/V converter, shown in Figure 24, predicts a cutoff frequency of approximately 2.45 MHz, which is consistent with the value measured at the PCB output.

4.4 Noise Analysis

Noise analysis is an essential step in the evaluation of the hardware, since noise directly limits the signal fidelity and reconstruction accuracy and most importantly achievable resolution.

The Power Spectral Density (PSD) describes how the noise power of a signal is distributed over frequency. It provides insight into both the amplitude of the noise and the frequency ranges in which it is most significant.
Mathematically, the PSD is related to the signal variance (or total noise power) as:

$$\sigma_X^2 \triangleq \int_0^{+\infty} S_X(f)\, df \tag{62}$$

where σ_X^2 is the variance (expected noise power) and $S_X(f)$ is the PSD, expressed in $[V^2/Hz]$.

To measure the intrinsic noise of the circuit, the input is short-circuited to ground. This ensures that no external signal is present, and that the measured output corresponds only to internally generated noise [34]. Nevertheless, external disturbances such as Electromagnetic Interference (EMI) and measurement equipment can still introduce parasitic contributions to the measured PSD.
An alternative measurement method using a 0.5 V DC bias supplied by a DC power supply is presented in Annex F. However, this configuration did not provide better results, as the measured PSD appeared to be more affected by external disturbances, resulting in a larger number of spectral spikes.

The resulting noise power spectral density (PSD) is shown in Figure 56, and is measured using the configuration in Table 10.

Parameter	Value	Unit	Notes
Input	0	V	Grounded (short-circuit)
Channel	CH2	–	Output channel
Sample rate	8	MS/s	Instrument sampling rate
Acquisition time	10	ms	Signal measurement duration
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting

Table 10: Measurement configuration used for noise power spectral density (PSD) acquisition.

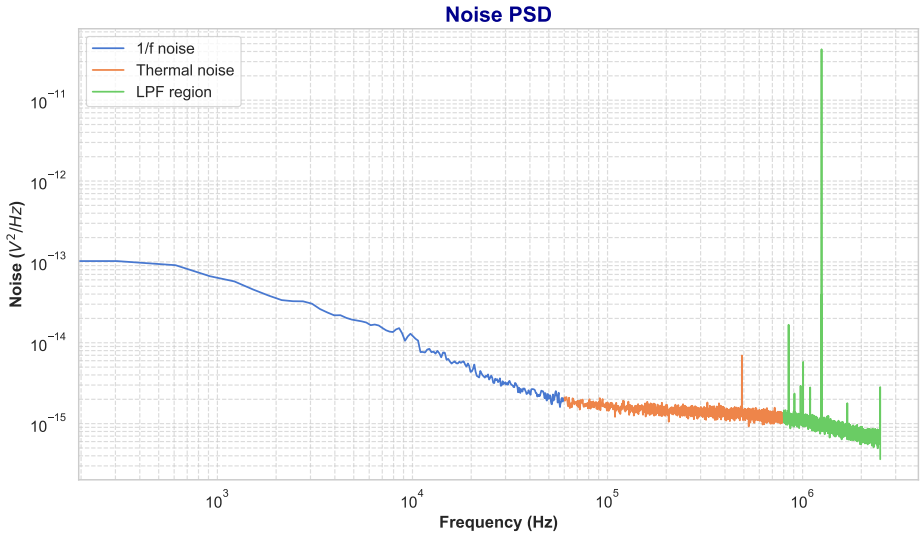


Figure 56: Measurements: Output Noise PSD

Three distinct regions can be identified in the obtained PSD:

1. **Flicker noise (1/f noise):**
   From DC up to approximately 7kHz, the PSD exhibits a decreasing behavior that appears as a straight line with a slope close to -1 in a log-log representation. This corresponds to flicker noise, also known as $1/f$ noise, for which:
   $$S_X(f) \propto \frac{1}{f} \tag{63}$$
   This noise originates from microscopic effects in semiconductor devices, such as charge trapping and detrapping at defects near the oxide-semiconductor interface, or fluctuations in carrier mobility. Flicker noise is dominant at low frequencies and can significantly impact precision applications. In operational amplifiers, it is mainly determined by the input stage and sets the low-frequency noise floor. In ADCs, it can degrade the effective resolution for low-frequency signals [35].

2. **Thermal noise:**
   Above approximately 7 kHz, the PSD reaches a flat plateau, indicating that the noise is independent of frequency. This region corresponds to thermal noise, which originates from the random thermal motion of charge carriers in resistive elements. The thermal noise level is proportional to both the absolute temperature and the resistance of the system, and represents the fundamental noise floor of the circuit.

3. **Low-pass filtering region:**
   At higher frequencies, around 2MHz, the PSD decreases, indicating attenuation of the noise. This behavior is not due to noise generation but rather to the limited bandwidth of the system, which

acts as a low-pass filter, as identified in the Bode diagram analysis section.
Additionally, narrow spectral peaks are observed in this frequency range. These are likely caused by external electromagnetic interference from laboratory equipment (e.g., switching power supplies or digital electronics) and are not intrinsic to the circuit itself. To minimize environmental perturbations, the PCB should be isolated and shielded, using an insulating box for example.

To better understand the origin of these contributions, information can be gathered from the datasheets of the components used in the system. However, for most of the selected components, only limited noise characterization is available. For example, the operational amplifiers used in the design (the LT1813 and ADA4857 for the differential-to-single-ended conversion associated with the DAC stages) provide input-referred noise spectral density information. Both devices exhibit comparable trends: relatively higher noise at low frequencies, followed by a gradual decrease with increasing frequency until approximately 1kHz, beyond which the noise reaches a nearly constant level associated with thermal noise. This behavior is consistent with the measured output noise PSD.

A more comprehensive evaluation of the noise origin would require a detailed analysis across the full circuit chain, including the contribution of each stage and component. However, such an assessment is limited by the incomplete noise characterization available for several of the selected components, as these devices are not specifically designed or documented as low-noise components. Therefore they do not provide extensive noise-related information in their datasheets.

To reduce the overall noise, several approaches can be considered:

- Selecting low-noise components with more complete noise characterization available in the datasheets,
- Minimizing thermal noise by reducing resistances and, when possible, operating at lower temperatures,
- Improving PCB design through proper shielding, grounding, and layout techniques to limit external interference.

4.5 Data Converter Characteristics

An ideal ADC is often described as a perfectly precise bridge between the continuous analog world and the discrete digital domain. It uniquely maps every analog input value within a defined range to a corresponding digital output code. The number of discrete output codes depends on the converter resolution N and is given by 2^N.
This means that the full-scale range (V_{FS}), which represents the total input range of the ADC, is divided into 2^N steps. The width of each step is identified as the Least Significant Bit (LSB):

$$1 \text{ LSB} = \frac{V_{FS}}{2^N} \tag{64}$$

As the number of bits increases, the step size decreases, and the transfer characteristic approaches a straight line. For infinite resolution, the ADC would behave as a perfectly linear system.

An example of the transfer characteristic of an ideal ADC is shown in Figure 57 for $N = 3$, with an input range from 0 to 8 V. The ideal transfer characteristic can be represented by a straight line passing through the midpoint of each quantization step.

For a DAC, the concept is similar, except that the x-axis represents the digital code and the y-axis represents the analog output value. In this case, the LSB corresponds to the height of each output step.

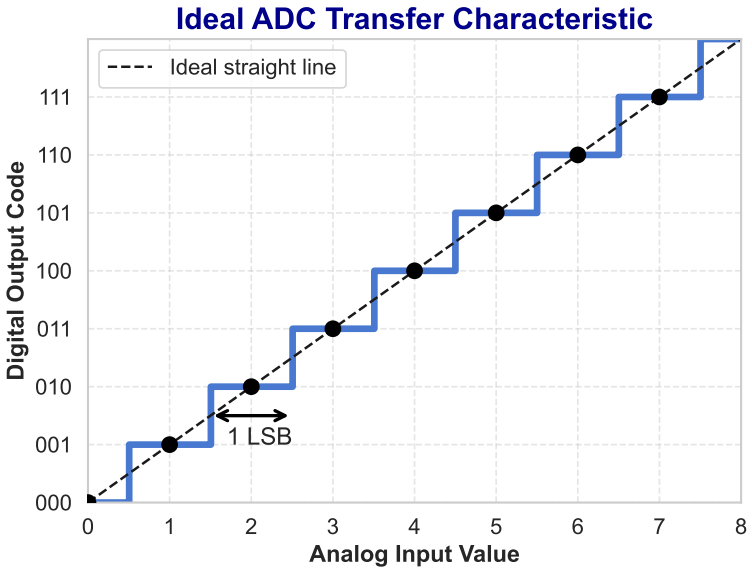


Figure 57: Python Simulation: Ideal ADC

In the ideal case, the conversion is instantaneous, noiseless, and perfectly linear: each increase in input corresponds to a proportional increase in output, and all quantization steps are uniform.
However, like all practical electronic components, ADCs and DACs deviate from this ideal behavior due to a variety of non-idealities [36]. One important class of imperfections is static errors, which describe deviations in the transfer characteristic. These errors appear even for constant input signals and are independent of time or frequency. They include offset error, gain error, Differential nonlinearity error (DNL), and Integral nonlinearity error (INL) [37].
The offset error is defined as the difference between the ideal transfer characteristic and the actual mid-step transition value when the digital output code is zero. For a DAC, it corresponds to the deviation of the analog output from the ideal value at zero digital input. Offset error shifts the entire transfer function by a constant amount.
The gain error represents the deviation between the ideal and actual transfer characteristics at full scale, after offset correction has been applied. In an ADC, it is measured at the full-scale output code, while in a DAC it corresponds to the analog output error at the maximum digital input value. Gain error results in a slope difference between the actual and ideal transfer functions.
DNL is the deviation of an actual code width (for an ADC) or step height (for a DAC) from the ideal value of 1 LSB.
INL is the deviation of the actual transfer function from the ideal straight line. INL can be interpreted as the cumulative effect of DNL errors across the converter range.
In practice, offset and gain errors are often corrected before evaluating DNL and INL, in order to isolate the purely non-linear behavior of the converter.

While static errors describe the shape of the transfer characteristic, they do not fully capture ADC performance under real operating conditions. In practice, additional imperfections such as noise, finite bandwidth, and other dynamic effects degrade the accuracy of the conversion.
To account for these effects, the Effective Number Of Bits (ENOB) is introduced. ENOB quantifies the actual usable resolution of the converter by taking into account noise and distortion. As a result, even if an ADC has a high nominal resolution, its ENOB may be significantly lower due to noise and dynamic non-idealities [38].

4.5.1 Static Errors

A common method for ADC characterization consists of applying a known reference signal generated by a high-performance DAC [39]. In this approach, a digital code is sent to a DAC with well-characterized performance, producing a precise analog voltage. This voltage is then applied to the ADC under test, and the resulting digital output code is compared to the expected value.
By comparing the expected and measured codes, it is possible to evaluate several static performance

parameters, such as the offset error, gain error, DNL, and INL. Since the exact input-output relationship is known, deviations from the ideal transfer characteristic can be quantified.
In this work, however, the objective is not to characterize a standalone ADC, but rather to evaluate how the PCB and the modulo operation could affect the signal before it reaches a future ADC stage. Therefore, instead of using a DAC-generated reference, a known analog signal is applied directly to the PCB input.

The classical characterization chain [39]:

$$\text{Digital Code} \rightarrow \text{DAC} \rightarrow \text{Analog Signal} \rightarrow \text{ADC}$$

is replaced by:

$$\text{Known Analog Staircase} \rightarrow \text{PCB} \rightarrow \text{Measured Output}$$

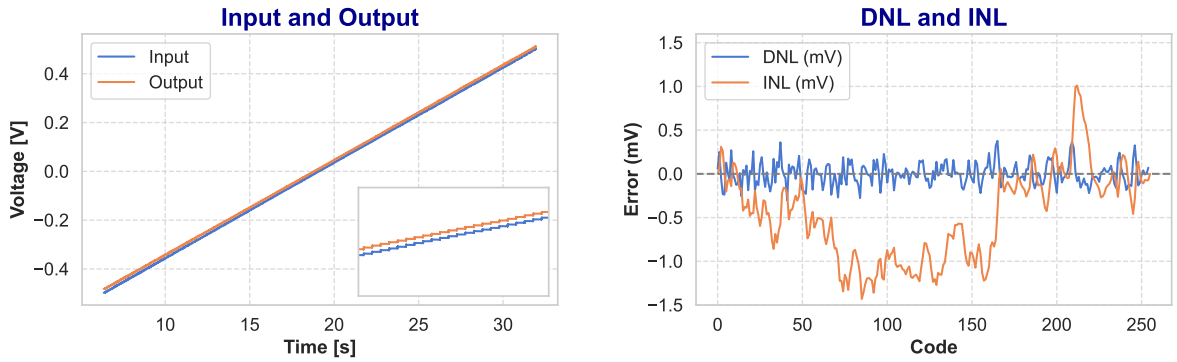
A staircase input signal is particularly suitable because it resembles the quantized behavior of an ADC transfer function. Each step represents a quantization level, while each plateau corresponds to a stable code region.
By applying a staircase waveform to the PCB input and comparing the measured output with the expected staircase response, it becomes possible to evaluate the offset, gain, DNL, and INL.

Since, the real resolution given by the ENOB is not yet determined, it is first chosen to work with 2^8 quantization levels, and will be adapted once the ENOB characterized.
The first test case considers an input signal already contained within the range $[-1, 1]$V, thereby avoiding any additional effects caused by the modulo operation or the reconstruction process.
As these measurements are typically performed under quasi-static conditions, the step duration is set to 100 ms in order to approximate such operating conditions.
The resulting staircase waveform is shown in Figure 58a with the configuration parameters described in Table 11.

Parameter	**Value**	**Unit**	**Notes**
Input Signal	-0.5 to 0.5	V	Staircase signal ranging from -0.5V to 0.5V with 2^8 quantization steps
Step Duration	0.1	s	Duration of each staircase level
Channel	CH1	–	Input channel
Channel	CH2	–	Output channel
Vertical scale	100	mV/div	Setting for CH1
Vertical scale	100	mV/div	Setting for CH2
Sample rate	32.5	kS/s	Instrument sampling rate
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting
Trigger mode	Positive Edge	–	Trigger configuration
Trigger level	0	V	Trigger threshold

Table 11: Measurement configuration used for Figure 58a.

A low-pass window filter is applied to each step to reduce the influence of noise and isolate static nonlinearity effects.
From the waveform close-up, an offset can already be observed and is measured to be 0.01527V.
After offset correction, the gain error can be calculated.
The measured gain error is -0.54%, meaning that the output slope increases slightly less rapidly than the input, producing a compressed transfer characteristic.



(a) Input and Output

(b) DNL and INL

Figure 58: Measurements: Staircase 8 bit from -0.5 to 0.5

After correcting the offset and gain, DNL and INL can be analyzed.
The DNL remains centered around zero, indicating stable converter behavior. The step height is approximately 3mV according to (64). Since the DNL does not reach this value, no missing codes are observed.
In [41] [42], a polynomial decomposition of the ADC transfer characteristic is introduced to model the INL. In this approach, the INL can be approximated using a polynomial expansion of the form:

$$INL(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots \tag{65}$$

This decomposition allows the dominant non-linearity behavior to be identified. In this context, the order of the INL refers to the highest significant polynomial term contributing to the shape of the INL curve. Different polynomial orders produce characteristic INL shapes: a first-order (linear-dominant) response produces a linearly sloped INL. A second-order term produces a parabolic INL curve, while a third-order term results in an S-shaped curve. Higher-order terms introduce additional inflection points, leading to more complex behavior.
In Figure 58b, a second-order polynomial fit is observed, although it appears to be mixed with higher-order contributions, which become more visible at higher codes.
Some spikes, for example near code 200, may also be attributed to residual measurement noise. Even after filtering, small local variations in the measured code transition levels can affect the estimated average plateau levels of the quantization steps.

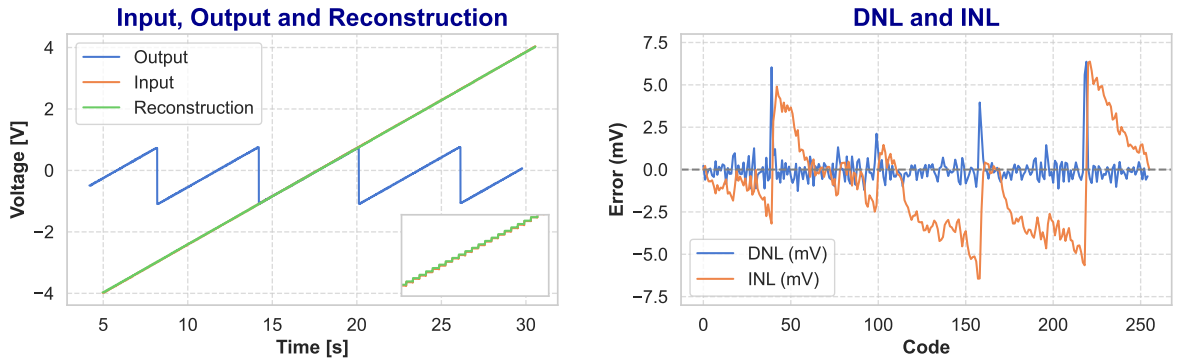
Parameter	Value	Unit	Notes
Input Signal	-4 to 4	V	Staircase signal ranging from -4V to 4V with 2^8 quantization steps
Step Duration	0.1	s	Duration of each staircase level
Channel	CH1	–	Input channel
Channel	CH2	–	Output channel
Vertical scale	850	mV/div	Setting for CH1
Vertical scale	240	mV/div	Setting for CH2
Sample rate	32.5	kS/s	Instrument sampling rate
Vertical resolution	16	bits	Oscilloscope setting
Bandwidth limit	10	MHz	Filter setting
Trigger mode	Positive Edge	–	Trigger configuration
Trigger level	0	V	Trigger threshold

Table 12: Measurement configuration used for Figure 59.

Figure 59 presents the second test case, in which the input signal exceeds the range $[-1, 1]$V and therefore requires folding. The corresponding configuration parameters are listed in Table 12. Due to its amplitude, the signal undergoes two folding operations before reconstruction.
Because the signal span is larger, the step height becomes 31.25mV according to (64), which is nearly ten times larger than in the previous case.
The measured offset is approximately 0.0157 V, similar to the previous measurement.
After offset correction, the measured gain error equals -0.35%. Although still negative, the gain deviation

is a bit smaller than in the previous case, indicating a transfer slope closer to the ideal response.

Following offset and gain correction, DNL and INL are calculated and shown in Figure 59b.



(a) Input and Output

(b) DNL and INL

Figure 59: Measurements: Staircase 8 bit from -4 to 4

The DNL behavior is similar to that of the previous case, with values centered around zero. However, there are spikes reaching approximately 7.5 mV, which are not observed when the signal is not folded. Four distinct spikes can be identified, corresponding to the four folding events. These spikes appear near the codes where folding occurs. Therefore, they can be attributed to the folding and reconstruction process, which could affect the effective step height.
Due to these larger spikes, it is difficult to determine an appropriate polynomial order for the INL characterization. If the spikes were reduced, the INL shape would likely resemble that of the previous case.

Overall, when comparing this folded case with the first test case, the offset and gain are not significantly affected, as they remain close in value. However, DNL and INL are slightly impacted, likely because folding alters the step levels and the reconstruction does not perfectly restore the intended quantization levels, thereby increasing the error.

4.5.2 Signal-to-Noise and Distortion Ratio (SINAD) Analysis

In order to extract the ENOB, it is useful to first introduce the Signal-to-Noise Ratio (SNR).
The SNR measures the ratio between the power of the useful signal and the power of the noise affecting it. In other words, it quantifies how strong a signal is compared to background noise:

- High SNR: the useful signal clearly dominates the noise
- Low SNR: the useful signal is buried in noise

However, in real data converters, the signal is not only affected by noise but also by distortion (e.g., harmonic components).
Therefore, a more relevant metric is the Signal-to-Noise-and-Distortion Ratio (SINAD), which extends the SNR definition by including all spectral components except the fundamental signal.
To estimate the signal quality, the output signal is analyzed in the frequency domain. The Fast Fourier Transform (FFT) is an efficient algorithm used to compute the Discrete Fourier Transform (DFT), which converts a discrete-time signal into its frequency-domain representation.
In the time domain, the signal and noise are mixed together. However, in the frequency domain, the useful signal often appears as peaks at specific frequencies, while the noise is spread across a wide range of frequencies.

For a discrete signal $x[n]$ of length N, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}} \tag{66}$$

where $x[n]$ is the time-domain signal, $X[k]$ is its frequency-domain representation, N is the number of samples, and $k = 0, 1, \ldots, N-1$.

Each index k corresponds to a specific frequency:

$$f_k = \frac{k f_s}{N} \tag{67}$$

where f_s is the sampling frequency. The quantity $\frac{f_s}{N}$ is called the frequency resolution (or bin width), representing the spacing between consecutive frequency bins.
A larger N provides finer frequency resolution, while a higher sampling rate f_s increases the overall frequency range.

Once the FFT $X[k]$ is computed, the power spectrum is obtained as:

$$P[k] = |X[k]|^2 \tag{68}$$

The frequency bins are then separated into two sets:

- $\mathcal{S}$: bins corresponding to the fundamental signal
- $\mathcal{N}$: bins containing all other components (noise and distortion)

The corresponding powers are:

$$P_{\text{signal}} = \sum_{k \in \mathcal{S}} |X[k]|^2 \tag{69}$$

$$P_{\text{noise+distortion}} = \sum_{k \in \mathcal{N}} |X[k]|^2 \tag{70}$$

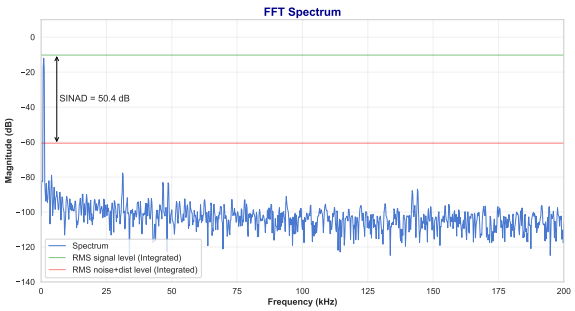
The SINAD is then defined as:

$$SINAD = \frac{P_{\text{signal}}}{P_{\text{noise+distortion}}} \tag{71}$$

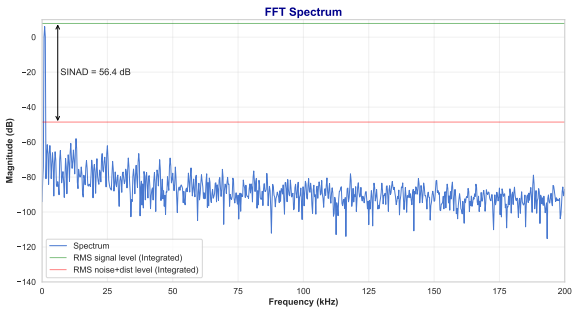
and expressed in decibels:

$$SINAD_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise+distortion}}} \right) \tag{72}$$

Figure 60 shows the FFT of the output signal for two sinusoidal inputs at 1kHz: a 0.5V signal (Figure 60a), which remains within the input range and therefore does not require folding, and a 4V signal (Figure 60b), which undergoes folding followed by reconstruction. The measurement parameters for the 0.5V case are identical to those listed in Table 6, while the parameters for the 4V case correspond to those given in Table 8.



(a) FFT of the output for an input sinusoidal of 0.5V of 1kHz



(b) FFT of the output for an input sinusoidal of 4V of 1kHz

Figure 60: Measurements: FFT of the output signal

For both cases, the highest peak corresponding to the useful signal is correctly located at 1kHz. The difference in peak levels (approximately -10 dB for 0.5 V and 8 dB for 4 V) reflects the difference in signal amplitudes, since power increases with amplitude.
Indeed, the amplitude ratio in decibels is:

$$20 \log_{10} \left(\frac{4}{0.5} \right) = 18.06 \text{ dB} \tag{73}$$

which is consistent with the observed increase in peak level for the 4 V signal.
Thus, a higher peak corresponds to higher signal power and can potentially increase the SINAD.

However, the RMS level of noise and distortion, corresponding to all other frequency components, is also higher in the 4 V case, increasing from approximately -60.64 dB to -48.56 dB. This degradation can be attributed to the folding and reconstruction process. Indeed, folding is a non-linear operation, which introduces additional distortion due to non-idealities in the system (e.g., feedback loop components).
As a result, although the signal power is higher for the 4 V case, the SINAD does not improve proportionally because of the increased noise and additional distortions.
After computation, the SINAD is 50.12 dB for the 0.5 V case and 56.67 dB for the 4 V case. While an ideal amplitude scaling would suggest an improvement of approximately 18 dB, only about 6 dB is observed. This discrepancy is explained by the 12 dB increase in the noise and distortion level, which limits the overall improvement in SINAD.

An additional SINAD analysis is performed to better identify the origin of the measured noise and distortion. For this purpose, the signal generator is directly connected to the oscilloscope, without passing through the PCB, in order to evaluate the contribution of the generator and the measurement setup itself. The corresponding FFT spectrum is shown in Figure 61 for a 0.5 V sinusoidal signal at 1 kHz, using the same measurement parameters as those listed in Table 6.

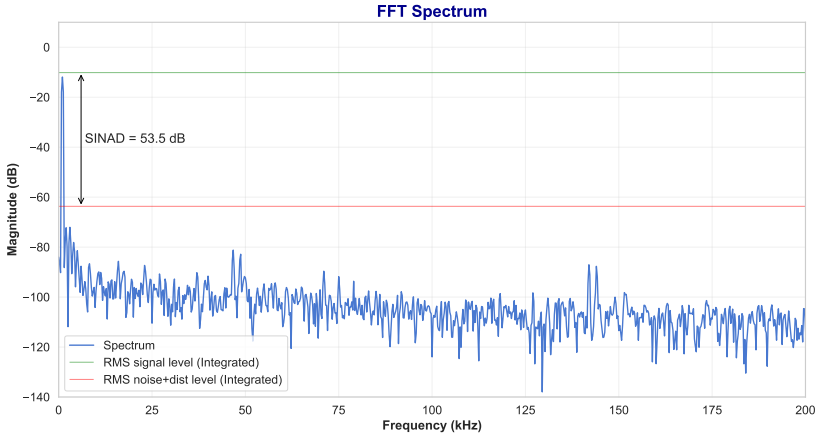


Figure 61: Measurements: FFT spectrum of the generator output directly connected to the oscilloscope for a 0.5 V, 1 kHz sinusoidal input.

By comparing this measurement with Figure 60a, it can be observed that the SINAD measured at the output of the PCB is only approximately 3 dB lower than the SINAD obtained directly from the generator. This indicates that a significant portion of the measured noise and distortion already originates from the generator and the acquisition setup itself.
A degradation of approximately 3 dB corresponds to a doubling of the total noise and distortion power. Consequently, the PCB contributes a level of noise and distortion comparable to that already present in the generator signal. Therefore, the measured SINAD is not only limited by the PCB performance, but also by the intrinsic quality of the signal generator and the measurement instruments.

SINAD is commonly represented for various input amplitudes and frequencies. Figure 62 presents the measured SINAD versus input amplitude for several sinusoidal input frequencies.

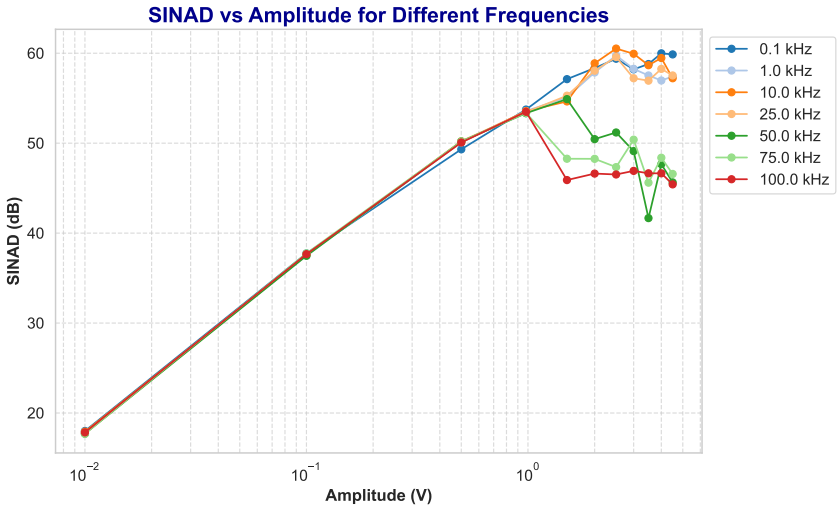


Figure 62: Measurements: SINAD Analysis

The SINAD curves exhibit two distinct regions: an initial linear increase followed by a saturation region.
At low amplitudes (approximately 10 mV to 1 V), SINAD increases almost linearly with input amplitude. In this region, the useful signal power increases while noise and distortion remain nearly constant. For very small amplitudes, the signal level is comparable to the noise floor and distortion components, resulting in poor SINAD performance. As the amplitude increases, the signal becomes progressively dominant over these error sources, leading to an improvement in SINAD.
All frequencies behave similarly, which is expected since they lie within the system passband as discussed in the Bode Diagram section.
When the input amplitude becomes sufficiently large, SINAD approaches a plateau around 60 dB. Beyond this point, increasing the signal amplitude no longer produces a significant improvement in SINAD. This saturation occurs because the noise and distortion components begin to increase together with the signal amplitude.
For example, considering a 1 kHz sine wave, increasing the amplitude from 2 V to 4 V theoretically increases the signal power by 3 dB according to (73). Although the 2 V case is not shown in the figures, its RMS noise-and-distortion level was also evaluated and measured to be approximately -50.73 dB, compared to -48.56 dB for the represented 4 V case. Therefore, although the signal power increases, the error power rises by a similar amount, resulting in little to no improvement in SINAD.
This behavior is typical of practical acquisition systems, particularly ADCs. For an ADC-based system, this limit is directly related to the finite number of quantization levels available. A N-bit converter can only distinguish between 2^N discrete amplitude levels, which imposes a theoretical upper bound on the achievable SNR given by [40]:

$$SNR[dB] = 6.02 \times N + 1.78 \tag{74}$$

For example, an ideal 12-bit ADC has a theoretical maximum SNR of approximately 74 dB. Beyond this limit, increasing the input amplitude cannot significantly improve SNR because the system has already reached its maximum effective resolution.
In practice, this saturation also reflects the finite dynamic range of the analog front-end. As the signal amplitude approaches the limit imposed by the supply voltage, the input V/I converter operates closer to its nonlinear region. Small nonlinearities and clipping effects then introduce additional distortion, further reinforcing the SINAD plateau.

Another observable trend is that the SINAD of folded and then reconstructed signals slightly decreases as the input frequency increases. This behavior is directly related to the reconstruction process described in Section 4.2.4.
In particular, the folded signal must be downsampled prior to reconstruction in order to discard samples acquired during folding transitions, which are not instantaneous in practice.
If samples acquired during these transitions are not fully removed, the reconstruction exhibits distortions.

Moreover, selecting an appropriate downsampling factor becomes increasingly difficult as the input frequency increases, since folding events occur more frequently while their duration remains approximately constant. This effect was illustrated in Figure 51, where the reconstruction accuracy decreases with increasing frequency. These reconstruction errors directly contribute to a reduction in SINAD.
However, this limitation does not significantly affect the estimation of the ENOB, since SINAD measurements at lower input frequencies remain largely unaffected by reconstruction errors. Therefore, ENOB can be reliably determined using low-frequency measurements.

4.5.3 ENOB

Once the SINAD has been extracted, the ENOB can be evaluated by replacing the SNR term in (74) with SINAD and solving for N.
This yields [40]:

$$\text{ENOB} = \frac{\text{SINAD[dB]} - 1.76}{6.02} \tag{75}$$

The resulting ENOB, computed using the average SINAD values over the low-frequency range (100 Hz to 25 kHz), is shown in Figure 63.

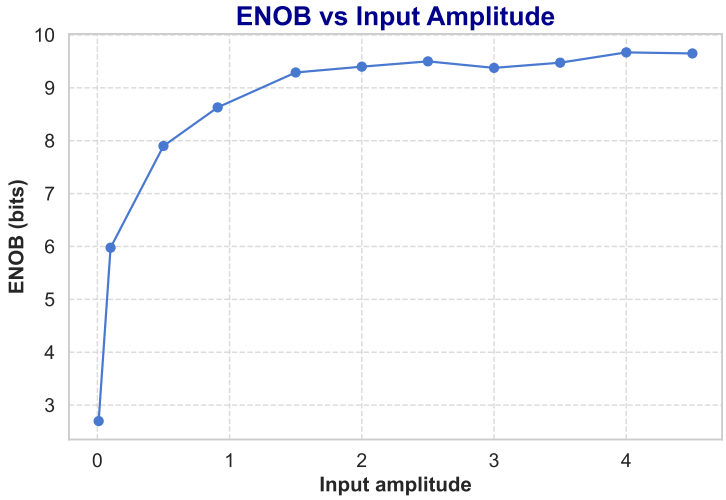


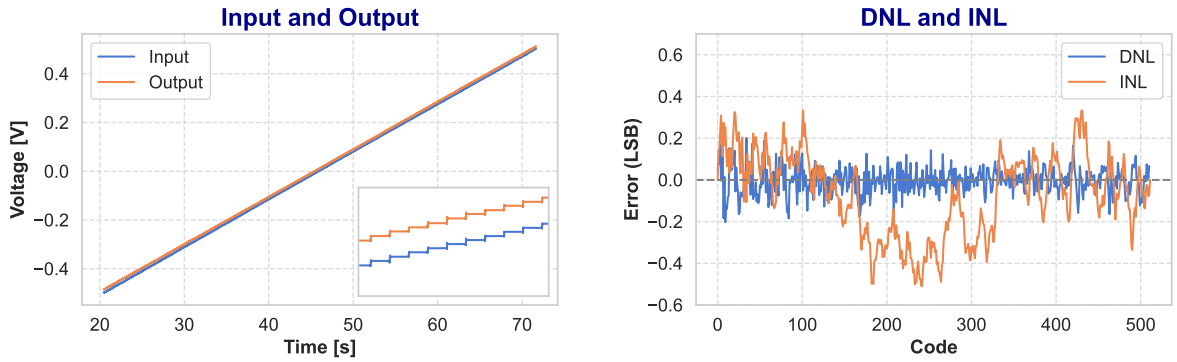
Figure 63: Measured ENOB as a function of input amplitude

Similarly to the SINAD curves, the ENOB is low for small input amplitudes, as the signal magnitude is comparable to the noise and distortion levels, which significantly degrades the effective resolution.
As the input amplitude increases, the ENOB increases approximately linearly, reflecting an improved SINAD. At higher amplitudes, the ENOB reaches a saturation region consistent with the plateau observed in the SINAD curves, indicating that the system has reached its maximum effective resolution.
From this plateau, the maximum achievable resolution of the system can be estimated to be approximately 9 bits.

Knowing this maximum achievable resolution, the static ADC error analysis can be repeated using this updated resolution and compared with the previous results.
First, a staircase signal ranging from -0.5 V to 0.5 V is considered such that no folding occurs.
For this case, the LSB becomes:

$$1\,LSB = \frac{(0.5 - (-0.5))}{2^9} = 1.953 \text{ mV} \tag{76}$$

The resulting graphs are presented in Figure 64, using the same parameters as those listed in Table 11.



(a) Input and Output

(b) DNL and INL

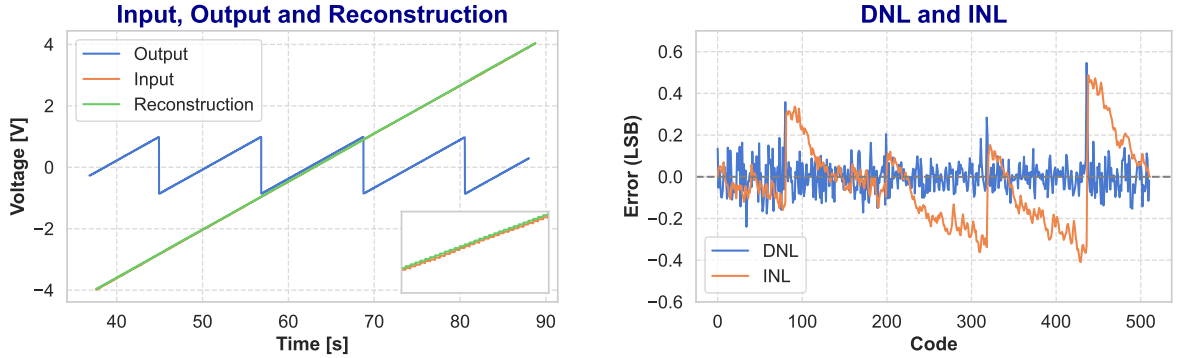
Figure 64: Measurements: Staircase 9 bit from -0.5 to 0.5 V

The offset and gain are very similar to those obtained in the previous 8-bit measurements.
The measured offset is 0.0129 V, corresponding to 6.62 LSB, meaning that the output is shifted upward by nearly seven quantization levels. The measured gain error is -0.41%, very similar to the previous 8-bit case.
Figure 64b shows the resulting DNL and INL expressed in LSB rather than in millivolts.
A DNL larger than $+1$ LSB would indicate a missing code because the code width exceeds one quantization step. Conversely, a DNL smaller than -1 LSB would imply non-monotonic behavior, where the output decreases while the input increases.
Neither condition is observed in the measured results. Most DNL values remain within approximately ± 0.2 LSB, indicating relatively uniform quantization intervals.
The INL curve is again very similar to that obtained in Figure 58b, exhibiting a U-shaped behavior characteristic of a dominant second-order contribution, although higher-order components appear slightly more pronounced.

Figure 65 shows the results for an input range from -4 V to 4 V, again using the same parameters listed in Table 12.
In this case, the LSB becomes:

$$1\,LSB = \frac{(4-(-4))}{2^9} = 15.625 \text{ mV} \tag{77}$$

The offset and gain remain similar. The measured offset is 0.0131 V, corresponding to 1.29 LSB. Since the LSB is larger, the offset expressed in LSB is lower than in the previous case, meaning that the output curve is shifted upward by approximately one quantization step.
The measured gain error is -0.43%.



(a) Input and Output

(b) DNL and INL

Figure 65: Measurements: Staircase 9 bit from -4 to 4 V

In Figure 65b, the DNL and INL curves are expressed in LSB. The DNL does not exceed ±1 LSB, indicating the absence of missing codes and non-monotonic behavior.
The behavior is very similar to that observed in Figure 59b, where larger spikes appear near the codes corresponding to folding and reconstruction events.
These spikes directly affect the INL curve, which therefore also resembles the 8-bit case.

Overall, the results obtained at 9 bits are very similar to those obtained at 8 bits. Once again, the reconstruction process appears to affect the DNL and INL by increasing the error near the codes where folding occurs.

4.6 Improvements

Ideally, the performances obtained should be compared with implementations proposed in the literature. However, many papers only present an architecture without providing a complete implementation. Except for [11], which includes detailed information about the components used, supply voltages, maximum number of folds, and the maximum operating frequency, as described in Section 3.1, most papers only validate their architecture through simulations or provide little information about performance metrics. For this reason, the specifications listed in Table 1 were mainly based on the implementation presented in [11]. In particular, the target was to achieve a feedback delay below $1\mu s$ in order to maximize the operating frequency beyond 10kHz. While supply voltages of $\pm 15V$ are used in [11], the input range is limited to $\pm 10V$. Therefore, the goal here was to reduce the supply voltages while still maximizing the input dynamic range.
For this hardware implementation, the main limitation affecting the input dynamic range was the requirement to use only components with available SPICE models, so that the circuit could be accurately simulated. As a consequence, the achievable input range was reduced to approximately ±4.6 V (9.2V peak-to-peak).
This limitation originates from the operational amplifier used in the input V/I converter, which is supplied with ±5 V. During measurements, the amplifier exhibited nonlinear and unstable behavior when the input voltage exceeded approximately ±4.6 V, preventing reliable operation close to the supply rails. Therefore, the purpose of this section is to investigate methods to increase the input dynamic range while still minimizing the supply voltages of most circuit blocks.
Moreover, although the objective of achieving a feedback delay lower than $1\mu s$ has been reached, this section presents solutions to further reduce this delay and increase the maximum operating frequency while ensuring that the output remains within the ADC input range of [-1.25,1.25]V.

4.6.1 Input Dynamic Range

As explained in the Section 3.3, performing the correction in the current domain instead of the voltage domain significantly improves the achievable input dynamic range without requiring high supply voltages for all circuit blocks.
Only the input stage must withstand the maximum input voltage, while the following stages, such as the PVG, can operate with lower supply voltages. This is possible because the correction is applied through currents rather than by generating large output voltages.
For instance, as shown in Figure 15b, if the correction were performed in the voltage domain, a 10 V input signal with a 1 V threshold would require the PVG to produce an output up to 10 V. Consequently, the PVG would need a supply voltage of at least 10 V, along with components rated for such voltage levels.

The current system supports an input signal range from -4.6V to 4.6V, although this range can be further extended. Several design modifications are nevertheless required to further extend the input dynamic range. First, the input V/I converter requires a higher supply voltage, which implies replacing the current operational amplifier (LT1813), whose maximum supply voltage is limited to 12.6 V peak-to-peak. Second, the CPLD code must be adapted to allow a larger number of DAC steps. In the current implementation, the DAC uses 100 mV steps and is limited to two steps since only two folds are required for the present input range of ±5 V. The maximum DAC output is 400 mV, meaning that, even if the input range is increased without any additional modifications, the DACs would only support four folding levels, corresponding to a maximum input range of approximately ±8 V. Therefore, further increasing the input range requires reducing the DAC step size in order to enable more folding levels.
Finally, the gains of the input and feedback V/I converters, as well as the output I/V converter, must be adjusted according to the selected DAC step size to maintain correction values equal to multiples of

$\pm 2\lambda = \pm 2$ V. This can be achieved by modifying the resistors that define the converter gains.
If the objective is only to increase the input range, no additional components or supply voltages need to be modified, making the design relatively easy to adapt depending on the target application.

However, if the supply voltages of the remaining blocks are to be reduced in order to satisfy the specification 1.25 V $\leq |V_{supply}| \leq$ 3.3 V, additional modifications are required.
In this first prototype, all operational amplifiers (LT1813) and comparators (LT1016) are supplied with ± 5 V. Since the threshold is fixed at ± 1 V, the internal signal levels should never approach ± 5 V. Therefore, such high supply voltages are not strictly necessary and can be reduced.
With the selected components, the supply voltage can, for example, be reduced to ± 3.3 V, which also makes it compatible with the supply requirements of the digital components such as the DACs and CPLD.
It is even possible to further reduce the supply voltage, provided that all components still operate correctly and no signal clipping occurs. For instance, the LT1813 can operate with a minimum supply voltage of ± 2 V. If lower supply voltages are required, alternative components should be considered.
However, the ADA4857 op-amp [25], used at the output of the DACs to convert their differential outputs into a single-ended signal, cannot operate with supply voltages lower than 5 V. Therefore, it would need to be replaced by another op-amp compatible with lower supply voltages.

As an example, if a 40 V input signal is applied at the input. In this case, 20 folding levels would be required to bring the signal back into the valid operating range.
Since the LTspice simulation does not include the digital components or the ADA4857, a simulation is performed with the supply voltage of the input V/I converter increased to ± 40 V while reducing the supply voltage of the other blocks to ± 3.3 V.
To achieve 20 folding levels, the DAC step size can be reduced from 100 mV to 20 mV for example, allowing 20 correction levels over the 400 mV range. The gains of the V/I and I/V converters must then be adjusted accordingly. In addition, the finite state machine (FSM) implemented in the CPLD must be modified to generate the required DAC control values. The FSM shown in Figure 32, which currently contains 5 states, would need to be expanded to a total of 41 states: 20 positive states, 20 negative states, and 1 initial state.

The resulting LTspice simulation for half a period of a 40 V, 1 kHz sinusoidal input signal is shown in Figure 66, while the main parameter changes are summarized in Table 13.

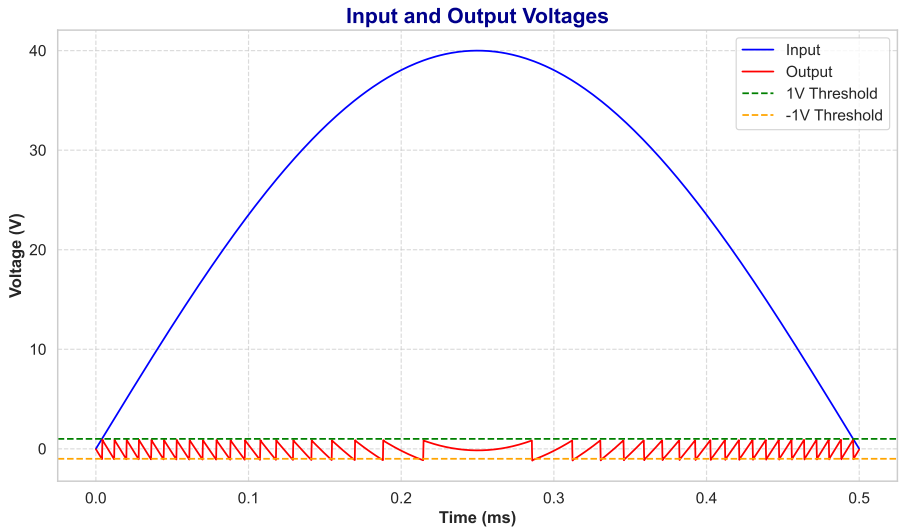


Figure 66: LTspice Simulation: Folding of a 40 V input sinusoidal signal at 1 kHz

Parameter	Value	Unit	Notes
Input	40	V	1kHz sinusoïdal signal voltage
Input Op-Amp Supply Voltage	±40	V	Ideal Op-Amp model
Op-amp supply voltage of all other op-amps	±3.3	V	LT1813
Comparator Supply voltage	±3.3	V	LT1016
Gain Resistor of the input V/I converter	50	$k\Omega$	
Gain Resistor of the output I/V converter	50	$k\Omega$	
Gain Resistor of the feedback V/I converter	2	$k\Omega$	

Table 13: Main configuration changed used for the simulation of Figure 66.

To perform this simulation, the operational amplifier used in the input V/I converter is replaced with an ideal LTspice model capable of supporting arbitrary supply voltages. Although suitable commercial operational amplifiers exist, no compatible LTspice models are available.
The gain of the input V/I converter is reduced from $\frac{1}{5000}$ to $\frac{1}{50000}$. Similarly, the gain of the output I/V converter is increased from 5000 to 50000, while the gain of the feedback V/I converter is changed from $\frac{1}{4}$ to $\frac{1}{2}$.

The input dynamic range could be extended even further, provided that a suitable op-amp and supply voltage are used for the input V/I converter. Since the DAC has an 8-bit resolution, it provides $2^8 = 256$ possible levels. In theory, this would allow 255 voltage steps across 400 mV, corresponding to a step size of approximately 1.57 mV and therefore up to 256 folding levels.
In practice, however, such a small step size would be difficult to exploit reliably because noise could significantly affect these low-level signals, thereby reducing the accuracy of the folding correction. Achieving such resolution would therefore require careful noise optimization.

4.6.2 Maximum Frequency

The maximum frequency is defined as the highest input frequency, for a given input amplitude, such that the folded output signal remains within the range $[-1.25, 1.25]$V.
To increase this maximum frequency, two approaches are considered: one based on reducing the feedback delay, and another based on modifying the operating threshold of the architecture.

1. **Feedback Delay**:

   As shown in Section 2.3.3, the maximum input frequency for which the folded output remains within $[-1.25, 1.25]$V is strongly dependent on the feedback delay.
   Indeed, this maximum frequency is inversely proportional to the total feedback delay, meaning that reducing the delay directly increases the maximum achievable frequency. The objective is therefore to minimize the feedback delay.
   As illustrated in Figure 36, the main contributor to the total feedback delay is the LT1813 operational amplifier, used in the subtractor and in the two follower stages (one at the DAC output and one at the subtractor output). These stages contribute approximately 90 ns, corresponding to 45.6% of the total feedback delay.
   Each LT1813 op-amp has a typical settling time of approximately 30 ns. Consequently, one possible approach to reduce the feedback delay is to replace these amplifiers with faster alternatives.
   For example, the AD8001 has a settling time of approximately 10 ns [43], which would reduce the combined contribution of the three op-amp stages from 90 ns to roughly 30 ns.

   The other main contributors are the CPLD and the DACs, contributing respectively 57.5 ns (29.1%) and 40 ns (20.3%). These delays are mainly determined by the clock period.
   As previously demonstrated, the implemented Verilog code requires approximately two clock cycles to update the FSM state and the corresponding output values, which corresponds to about 50 ns with a 40 MHz clock frequency. In addition, the DAC requires approximately one clock cycle to update its output, corresponding to 25 ns at 40 MHz. The DAC clock frequency cannot be increased beyond 40 MHz [24] without replacing the DAC with a faster alternative. However, the CPLD supports clock frequencies up to 152 MHz [23]. Therefore, a dedicated higher-frequency clock could be added exclusively for the CPLD in order to reduce its execution delay.

For example, using a dedicated 100 MHz oscillator for the CPLD would reduce the code execution delay from approximately 50 ns to about 20 ns.

By combining faster op-amps with an increased CPLD clock frequency, the total feedback delay could theoretically be reduced from approximately 197.5 ns to around 107.5 ns.
Using the behavioral Matlab Simulink model presented in Chapter 2, the impact of reducing the feedback delay from 197.5 ns to 107.5 ns can be evaluated. The corresponding results are shown in Figure 67 for a 4 V, 100 kHz sinusoidal input signal.

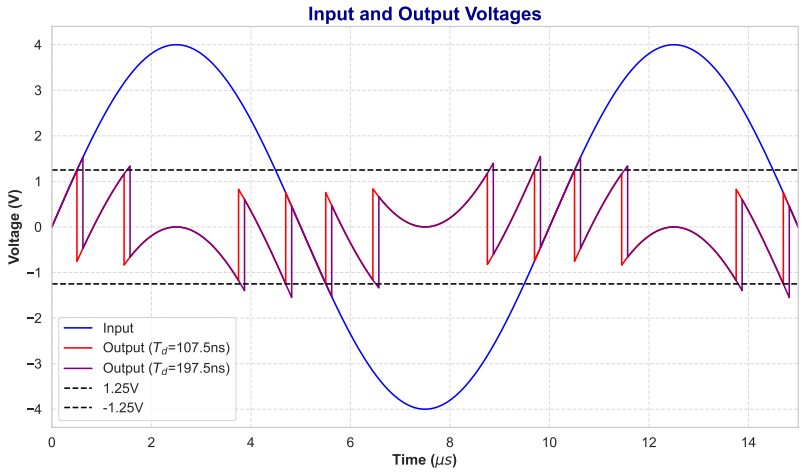


Figure 67: Matlab Simulation: Output signals for different feedback delays

As shown in Figure 67, reducing the feedback delay allows the folded signal to remain properly contained within the range $[-1.25, 1.25]$V.

Based on (53), the corresponding improvement in the maximum achievable frequency is:

$$\frac{f_{max,new}}{f_{max,prev}} = \frac{\frac{\Delta.\lambda}{T_{d,new}\cdot 2\pi\cdot A}}{\frac{\Delta.\lambda}{T_{d,prev}\cdot 2\pi\cdot A}} = \frac{T_{d,prev}}{T_{d,new}} = \frac{197.5}{107.5} \approx 1.837 \tag{78}$$

Therefore, the maximum achievable frequency could theoretically be increased by nearly a factor of two through these modifications.

2. **Threshold**:
Another approach for increasing the maximum frequency, proposed in [11], consists in reducing the threshold reference voltage λ rather than decreasing the feedback delay.
As discussed in Section 4.3.2, it is practically impossible to guarantee that the folded signal always remains strictly within $[-\lambda, \lambda]$. For this reason, a lower value of λ is generally selected to ensure that the output remains within the ADC input range. In the present implementation, $\lambda = 1$ is used for an ADC range of $[-1.25, 1.25]$V.
A possible improvement is therefore to reduce the threshold value λ. Although this does not reduce the feedback delay itself, it allows the comparators to trigger earlier. As a consequence, the correction begins sooner and can complete before the signal exceeds the allowed range.
Using again the behavioral Matlab Simulink model, while keeping the feedback delay fixed at 197.5 ns, the threshold can be reduced from 1 to 0.6V in order to anticipate the folding operation. The resulting behavior is shown in Figure 68 for a 4 V, 100 kHz sinusoidal input signal.

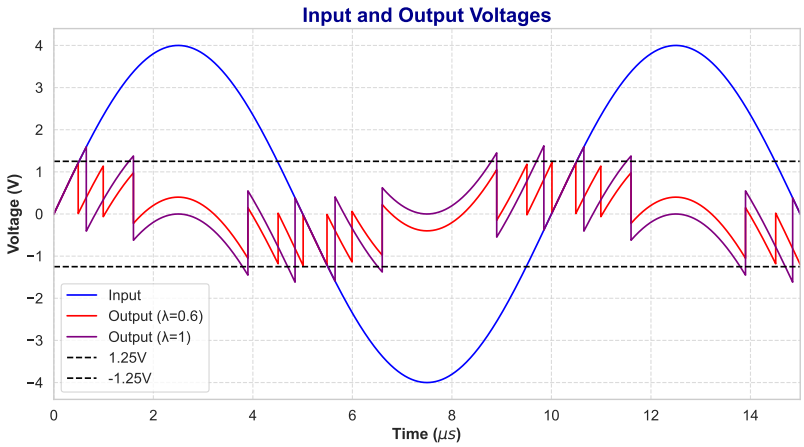


Figure 68: Matlab Simulation: Output signals for different thresholds

As shown in Figure 68, reducing the threshold from 1 to 0.6V increases the number of folding events. However, the folded signal remains properly contained within the range $[-1.25, 1.25]$V.

From (53), the corresponding improvement in the maximum achievable frequency for this threshold reduction is:

$$\frac{f_{max,new}}{f_{max,prev}} = \frac{\frac{\Delta_{new}\cdot\lambda_{new}}{T_d\cdot 2\pi\cdot A}}{\frac{\Delta_{prev}\cdot\lambda_{prev}}{T_d\cdot 2\pi\cdot A}} = \frac{\frac{1.08\times 0.6}{T_d\cdot 2\pi\cdot A}}{\frac{0.25\times 1}{T_d\cdot 2\pi\cdot A}} \approx 2.6 \tag{79}$$

where Δ_{new} is computed such that the admissible overshoot is equal to $1.25 - 0.6 = 0.65$V, leading to:

$$\Delta_{new} = \frac{0.65}{0.6} \tag{80}$$

This indicates that, for this example, the maximum achievable frequency could theoretically be increased by a factor of approximately 2.6.

Modifying the threshold voltage also requires several changes in the architecture.
The threshold voltages are defined by the voltage dividers connected to the inputs of Comp-N and Comp-P. Therefore, the resistor values setting these thresholds must be modified. In addition, since the correction voltage must remain equal to an integer multiple of 2λ, corresponding here to 1.2 V for $\lambda = 0.6$, the gain of the feedback V/I converter must be adjusted from $\frac{1}{1000}$ to $\frac{3}{5000}$, while the other converter gains can remain unchanged.

4.7 Conclusion

This chapter first evaluated the fundamental operation of the system by measuring the output and intermediate signals for cases requiring zero, one, and two folds.
It was observed that the achievable input range is not exactly ±5 V, but rather approximately ±4.6 V. Although the operational amplifier used in the input V/I converter is supplied with ±5 V (10V peak-to-peak), measurements showed that beyond ±4.6 V (9.2V peak-to-peak) the amplifier enters a non-linear operating region, preventing the signal from being folded correctly.
The reconstruction of the signals was then analyzed, revealing a limitation at high frequencies where samples may be acquired during fold transitions, leading to distortions in the reconstructed signal.
The feedback delay was subsequently measured and found to be approximately 230 ns, which is consistent with the theoretical estimation obtained in Chapter 3.
The maximum operating frequency was then evaluated as a function of the input amplitude, with measured values exceeding the theoretical estimates, ranging from 75 kHz for a 4 V input signal up to 270 kHz for a 1.5 V signal. In addition, the Bode diagram analysis showed that the system passband extends up to approximately 2.45 MHz. A noise analysis was also performed in order to identify the different

contributions originating from the intrinsic noise of the selected components.
Finally, a detailed analysis based on conventional data-converter characteristics was carried out to evaluate the static errors introduced by the PCB implementation, including offset, gain error, DNL, and INL. This analysis was performed both for signals without folds and for signals undergoing folds. The results showed that the offset and gain errors remain relatively unchanged, whereas the DNL and INL are affected by the folding and reconstruction processes. A SINAD analysis further allowed the effective resolution of the system to be estimated at approximately 9 bits.
The chapter concludes with several proposed improvements to further increase the input dynamic range. Improvements are also suggested to increase the maximum operating frequency, either by reducing the feedback delay through the replacement of certain components or by lowering the threshold values.

5

Conclusion and Perspectives

This work investigated a new hardware implementation of a modulo-ADC architecture ensuring that the folded output signal remains within the ADC operating range, thereby overcoming the dynamic range limitation of conventional ADC.

Chapter 2 introduced the Unlimited Sampling framework and the principles of modulo sampling. In particular, the modulo mapping operation was presented as a technique allowing signals to be wrapped within a fixed range, thus preventing clipping when the input exceeds the ADC range.
The associated signal recovery theory was also discussed, demonstrating that the original signal can be reconstructed solely from modulo samples.
Existing modulo-ADC architectures from the literature were reviewed, highlighting the differences between architectures employing a front-end modulo stage and those combining modulo processing and quantization simultaneously.
Among these approaches, the Wideband High Dynamic Range ADC architecture proposed in [11] was selected as the basis for the hardware implementation developed in this work.
Behavioral simulations performed in Matlab validated the operating principle of the architecture.
Its operation is based on comparators that continuously check whether the input signal exceeds the threshold λ. When this occurs, a correction voltage is generated accordingly in order to fold the signal back within the admissible range.
These simulations confirmed the correct interaction between the folding mechanism and the reconstruction process, while also highlighting several key implementation challenges.
In particular, increasing the input dynamic range requires larger correction values, potentially leading to higher supply voltage requirements.
In addition, the feedback structure introduces a delay between fold detection and signal correction. It was shown that this delay directly limits the maximum allowable input frequency for a given signal amplitude in order to guarantee that the folded signal remains entirely within the range $[-\lambda, \lambda]$.
Consequently, minimizing the feedback delay emerged as one of the primary design challenges of the proposed implementation.

Chapter 3 focused on the hardware implementation of the architecture.
Starting from the design proposed in [11], the main limitations regarding maximum input frequency and input dynamic range were identified.
The previous implementation from [11] was limited to an input amplitude of 10V at 10kHz, mainly due to a feedback delay of approximately $1\mu s$.
Based on these observations, the design specifications of the proposed implementation were established, targeting a feedback delay below $1\mu s$ while maximizing the achievable input dynamic range.
The threshold value was fixed at $\lambda = 1$V in order to constrain the output within the interval $[-1.25, 1.25]$V.
Choosing a threshold smaller than the accepted ADC limit ensures that small overshoots beyond the threshold remain safely within the ADC input range.
However, due to the requirement of validating the complete implementation in LTspice, the input dynamic range of this first prototype was limited to $\pm$5V, since the available components with compatible SPICE models were supplied by $\pm$5V rails.
Nevertheless, this limitation does not fundamentally constrain the proposed architecture for larger dynamic ranges. Indeed, the correction bringing the signal back within the admissible range is performed in the current domain rather than in the voltage domain. As a result, only the first stage converting the input voltage into current would require a higher supply voltage, while the remaining blocks could continue operating with lower supply voltages even for significantly larger input amplitudes.
The complete architecture was then presented, consisting of V/I and I/V converters, comparators, and a correction current generated by a CPLD associated with DACs.

Each block was presented in details, with the choice of the components and an LTspice simulation.
After the sizing process, the FSM implemented on the CPLD was developed to generate the appropriate correction for a maximum input range of ±5V. For the chosen threshold, this corresponds to a maximum of two folds. A theoretical analysis of the feedback path was then carried out, estimating the total feedback delay to be approximately 197.5ns.
Finally, the chapter concluded with a brief description of the PCB design and implementation.

Chapter 4 presented the experimental results and the performance analysis of the implemented hardware PCB. The measurement setup was first described before verifying the correct operation of the circuit in three operating conditions: when no correction is required, when one fold occurs, and when two folds occur. The performance of the recovery algorithm was then analyzed, revealing an important practical challenge. The theoretical framework assumes that the fold is applied instantaneously, whereas in practice the fold transition requires a finite amount of time. During this transition interval, samples can still be acquired, which introduces distortions in the reconstructed signal.
Downsampling the signal to remove these unwanted samples was shown to be an effective solution. However, at higher frequencies, completely removing all unwanted samples becomes more difficult, resulting in slightly degraded reconstruction accuracy.
A delay and frequency analysis was subsequently performed, showing that the measured feedback delay is approximately 230ns.
The maximum input frequency as a function of the input amplitude was then determined, demonstrating that the maximum operating frequency lies approximately between 75kHz for an amplitude of 4.5V and 270kHz for an amplitude of 1.5V.
A noise PSD analysis was also carried out, highlighting the contributions of flicker noise, thermal noise, and a low-pass behavior consistent with the cutoff frequency identified during the Bode diagram analysis, namely 2.45MHz.
In addition, an analysis inspired by classical data converter characterization techniques was performed to evaluate the non-idealities of the proposed system. Static errors such as offset, gain error, DNL, and INL were computed. This analysis showed that the folding and reconstruction process locally increases the DNL and INL errors around the regions where folds occur and are subsequently reconstructed.
A SINAD analysis was then performed in order to evaluate the ratio between the useful signal power and the combined noise and distortion power. The system reached a maximum SINAD of approximately 60dB. From the measured SINAD, the effective resolution of the system, expressed in terms of ENOB, was determined to reach a maximum value of approximately 9 bits.

Finally, several possible improvements and future research directions were presented.
The first improvement concerns increasing the input dynamic range of the modulo-ADC.
Although the current implementation theoretically targets an input range of ±5V, in practice the maximum input amplitude is limited to approximately ±4.6V because the input operational amplifier no longer operates in its linear region beyond this value, preventing proper folding of the signal.
However, this limitation is not inherent to the proposed hardware architecture itself but rather results from the design constraints imposed by the requirement to simulate the complete circuit in LTspice.
It was therefore shown that replacing the input operational amplifier with a device supporting higher supply voltages would allow the input dynamic range to be significantly increased. At the same time, the supply voltage of the remaining blocks could potentially be reduced below 5V since the folded output remains confined within the interval $[-1.25, 1.25]V$.
The current circuit is already capable of supporting up to four folds, and it was shown that this number could be increased further by reducing the DAC step size.
At present, the DAC step size is 100mV for a maximum output of 400mV. Reducing the step size to 20mV would theoretically allow up to 20 folds, while a step size of 5mV could enable up to 80 folds, provided that careful noise isolation is implemented to preserve accuracy at such low correction levels.
Achieving this improvement would require modifications to the FSM implemented on the CPLD, as well as adjustments to resistor values in order to modify the gains of the V/I and I/V converters.
Although this improvement has already been validated through simulations, it has not yet been experimentally tested. Future work could therefore focus on replacing the input operational amplifier and the other required components directly on the PCB in order to experimentally validate these extended operating conditions.

Another important improvement concerns the increase of the maximum input frequency.
One possible solution consists of further reducing the feedback delay, which would require replacing certain components, particularly the operational amplifiers, with devices exhibiting shorter settling times. In addition, the feedback delay also depends on the clock frequency used within the digital correction loop. Consequently, increasing the clock frequency could further improve the maximum operating frequency, since the CPLD itself supports clock frequencies up to 152MHz. However, the currently used DAC cannot support clock frequencies above 40MHz and would therefore need to be replaced.
Another possibility for increasing the maximum operating frequency consists of lowering the threshold λ, allowing earlier fold detection and therefore earlier correction. It was shown, for example, that reducing the threshold from 1V to 0.6V could theoretically increase the maximum allowable input frequency by approximately a factor of two.

In a longer-term perspective, the proposed hardware implementation could serve as the basis for a fully integrated System-on-Chip (SoC) solution, where the complete modulo-ADC architecture would be implemented within a single integrated circuit.
In such an implementation, the threshold λ could also become a configurable parameter, allowing the system to be adapted to a wide range of applications and operating requirements.

In conclusion, this work constitutes an important step toward the practical implementation of Unlimited Sampling theory in real electronic systems. While several technological challenges still remain, the obtained results demonstrate that modulo-based conversion techniques represent a promising alternative to conventional ADC architectures for high-dynamic-range signal acquisition.
Future research could therefore focus on fully integrated implementations, faster feedback paths, and adaptive reconstruction algorithms in order to ultimately overcome the dynamic range limitations of conventional ADCs in a wide range of scientific and industrial applications.

References

[1] S. Louwsma, E. van Tuijl and B. Nauta, *Time-Interleaved Analog-to-Digital Converters*, Analog Circuits and Signal Processing Series, Springer Netherlands, 1st ed., © 2011, ISBN 978-90-481-9716-3 (eBook), ISBN 978-94-007-9951-6 (print).

[2] D. Florescu, F. Krahmer and A. Bhandari, "The Surprising Benefits of Hysteresis in Unlimited Sampling: Theory, Algorithms and Experiments," IEEE Transactions on Signal Processing, vol. 70, pp. 616–630, Jan. 2022, doi: 10.1109/TSP.2022.3142507.

[3] Y. Kvich, S. Savariego, M. Namer and Y. C. Eldar, "Practical Modulo Sampling: Mitigating High-Frequency Components," Jan. 2025, arXiv: 2501.11330, doi: 10.48550/arXiv.2501.11330.

[4] Y. Zhu and A. Bhandari, "Unleashing Dynamic Range and Resolution in Unlimited Sensing Framework via Novel Hardware", 2024 IEEE SENSORS, Kobe, Japan, 2024, pp. 1-4, doi: 10.1109/SENSORS60989.2024.10784656.

[5] A. Bhandari, F. Krahmer and T. Poskitt, "Unlimited Sampling from Theory to Practice: Fourier-Prony Recovery and Prototype ADC," IEEE Transactions on Signal Processing, vol. 70, pp. 1131–1141, 2022, doi: 10.1109/TSP.2021.3113497.

[6] A. Krishna, S. Rudresh, V. Shaw, H. R. Sabbella, C. S. Seelamantula and C. S. Thakur, "Unlimited Dynamic Range Analog-to-Digital Conversion", Nov. 2019, arXiv: 1911.09371.

[7] Analog Devices, "MT-090: Sample-and-Hold Amplifiers," Tutorial MT-090, Rev. 0, Oct. 2008. [Online]. Available: `https://www.analog.com/media/cn/training-seminars/tutorials/MT-090.pdf`. Accessed: May 16, 2026.

[8] Atmel Corporation, "Atmel 7810: Automotive Microcontrollers - ATmega328P Datasheet," Rev. 7810D-AVR-01/15. [Online]. Available: `https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf`. Accessed: May 16, 2026.

[9] A. Bhandari, F. Krahmer and R. Raskar, "On Unlimited Sampling and Reconstruction," May 2019, arXiv: 1905.03901v2, doi: 10.48550/arXiv.1905.03901.

[10] J. Weemaels, *Hardware Implementation of Unlimited Sampling for the Analog-to-Digital Conversion of High Dynamic Range Signals*, Ecole polytechnique de Louvain, Université catholique de Louvain, 2025. Promoters: D. Flandre and T. Feuillen. Available: `http://hdl.handle.net/2078.1/thesis:49607`

[11] S. Mulleti, E. Reznitskiy, S. Savariego, M. Namer, N. Glazer and Y. C. Eldar, "A Hardware Prototype of Wideband High-Dynamic Range ADC", Jan. 2023, arXiv: 2301.09609v2.

[12] Y. Wenyi, G. Lu, H. Shaoqing and L. Hongqing, "Towards Optimized Multi-Channel Modulo-ADCs: Modulo Selection Strategies and Bit Depth Analysis", 2024 IEEE Speech and Signal Processing, pp. 1-5, doi: 10.1109/ICASSP48485.2024.10448478.

[13] O. Ordentlich, G. Tabak, P. K. Hanumolu, A. C. Singer and G. W. Wornell, "A Modulo-Based Architecture for Analog-to-Digital Conversion", in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 5, pp. 825-840, Oct. 2018, doi: 10.1109/JSTSP.2018.2863139.

[14] L. Brasco, *Handbook of Calculus of Variations for Absolute Beginners*, UNITEXT, vol. 163, Springer Cham, 1st ed., ISBN 978-3-031-87163-4 (print), ISBN 978-3-031-87164-1 (eBook), May 2025. doi: 10.1007/978-3-031-87164-1.

[15] A. Papoulis, "Limits on Bandlimited Signals" Proceedings of the IEEE, vol. 55, no. 10, pp. 1677–1686, Oct. 1967, doi: 10.1109/PROC.1967.5962.

[16] PJRC, "Teensy 4.0 Development Board," PJRC Store, 2026. [Online]. Available: `https://www.pjrc.com/store/teensy40.html#timing`. Accessed: Feb. 28, 2026.

[17] Analog Devices, "ADG1608/ADG1609 Data Sheet: 4.5 Ω R_{ON}, 4-/8-Channel Multiplexers," Analog Devices, Oct. 9, 2015. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADG1608_1609.pdf. Accessed: Feb. 28, 2026.

[18] Texas Instruments, "Analysis of Improved Howland Current Pump Configurations" Application Note. [Online]. Available: https://www.ti.com/lit/an/sboa437a/sboa437a.pdf. Accessed: May. 10, 2026.

[19] Texas Instruments, "AN-1515 A Comprehensive Study of the Howland Current Pump." [Online]. Available: https://www.ti.com/lit/an/snoa474a/snoa474a.pdf. Accessed: Mar. 04, 2026.

[20] Analog Devices, "LT1818/LT1819: 400MHz, 2500V/μs, 9mA Single/Dual Operational Amplifiers Data Sheet." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/18189fb.pdf. Accessed: Mar. 05, 2026.

[21] All About Circuits, "Transimpedance Amplifier Stability." [Online]. Available: https://www.allaboutcircuits.com/technical-articles/transimpedance-amplifier-stability/. Accessed: Mar. 04, 2026.

[22] Analog Devices, "LT1813/LT1814: Dual/Quad 3mA, 100MHz, 750V/μs Operational Amplifiers Data Sheet." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/18134fa.pdf. Accessed: Mar. 09, 2026.

[23] Altera Corporation, "MAX V Device Handbook," CPLD Device Family Documentation. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/6165/mv5v1_01.pdf. Accessed: Mar. 09, 2026.

[24] Maxim Integrated, "MAX5187/MAX5190: 8-Bit, 40MHz, Current/Voltage-Output DACs Data Sheet." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX5187-MAX5190.pdf. Accessed: Mar. 09, 2026.

[25] Analog Devices, "ADA4857-1/ADA4857-2: Ultralow Distortion, Low Power, Low Noise, High Speed Op Amp Data Sheet," Rev. D, Mar. 2008. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4857-1_4857-2.pdf. Accessed: May 12, 2026.

[26] SiTime, "SiT9005AI-21-33ED40.000000 MEMS Oscillator (SiT9005 Series) Data Sheet." [Online]. Available: https://www.digikey.be/fr/products/detail/sitime/SIT9005AI-21-33ED40-000000/11441369. Accessed: Mar. 09, 2026.

[27] SiTime, "MEMS vs Quartz Crystal Oscillators: Benefits, Reliability & Applications." [Online]. Available: https://www.sitime.com/company/newsroom/blog/mems-vs-quartz-crystal-oscillators-benefits-reliability-applications. Accessed: May 07, 2026.

[28] Analog Devices, "LT1016: Precision Fast Operational Amplifier Data Sheet." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/lt1016.pdf. Accessed: Mar. 09, 2026.

[29] W. Yan, Z. Li, L. Gan, H. Liu, and G. Li, "Difference-Based Recovery for Modulo Sampling: Tightened Bounds and Robustness Guarantees" arXiv preprint arXiv:2509.12971, 2025.

[30] Keysight Technologies, "N6705B DC Power Analyzer Mainframes Data Sheet." [Online]. Available: https://www.testequipmenthq.com/datasheets/Keysight-N6705B-Datasheet.pdf. Accessed: May 18, 2026.

[31] Tektronix, "Arbitrary/Function Generators Data Sheet." [Online]. Available: https://www.tek.com/en/datasheet/arbitrary-function-generator-0. Accessed: May 18, 2026.

[32] Tektronix, "5 Series B MSO Mixed Signal Oscilloscope." [Online]. Available: https://www.tek.com/en/products/oscilloscopes/5-series-mso. Accessed: May 18, 2026.

[33] Tektronix, "ArbExpress Signal Generator Software," 2026. [Online]. Available: https://www.tek.com/en/products/software/arbexpress-signal-generator. Accessed: Apr. 24, 2026.

[34] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*. Oxford University Press, 7th ed., 2015.

[35] Liquid Instruments, "Understanding 1/f Noise (Flicker Noise)." [Online]. Available: https://liquidinstruments.com/blog/understanding-1-f-noise-flicker-noise/. Accessed: Apr. 18, 2026.

[36] Analog Devices, "Data Conversion Handbook," Chapter 5: Testing Data Converters, Analog Devices Inc., 2005. [Online]. Available: https://www.analog.com/media/en/training-seminars/design-handbooks/Data-Conversion-Handbook/Chapter5.pdf. Accessed: Apr. 24, 2026.

[37] D. Flandre, *Analog Electronic Systems*, course slides, École Polytechnique de Louvain (EPL), UCLouvain, Louvain-la-Neuve, Belgium, 2025.

[38] Texas Instruments, "Impact of Voltage Reference Noise on ADC ENOB and Noise-Free Resolution," Application Report SLYT854. [Online]. Available: https://www.ti.com/lit/an/slyt854/slyt854.pdf. Accessed: May. 2, 2026.

[39] Texas Instruments, "Understanding Data Converters," Application Report SLAA013, 1995. [Online]. Available: https://www.ti.com/lit/an/slaa013/slaa013.pdf. Accessed: Apr. 24, 2026.

[40] W. Kester, "Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor," Tutorial MT-003, Analog Devices, Rev. A, Oct. 2008. [Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf. Accessed: May. 3, 2026.

[41] S. Medawar, P. Handel, N. Björsell, M. Jansson, "ADC Characterization by Dynamic Integral Nonlinearity" [Online]. Available: https://www.imeko.org/publications/iwadc-2008/IMEKO-IWADC-2008-023.pdf. Accessed: May 13, 2026.

[42] Analog Devices, "High Performance Source for ADC," Analog Dialogue / Technical Articles. [Online]. Available: https://www.analog.com/en/resources/technical-articles/high-performance-source-for-adc.html. Accessed: May 13, 2026.

[43] Analog Devices, "AD8001: 800 MHz, 50 mW Current Feedback Amplifier Data Sheet." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8001.pdf. Accessed: Mar. 09, 2026.

A

Improved Howland Current Pump: Output Derivation

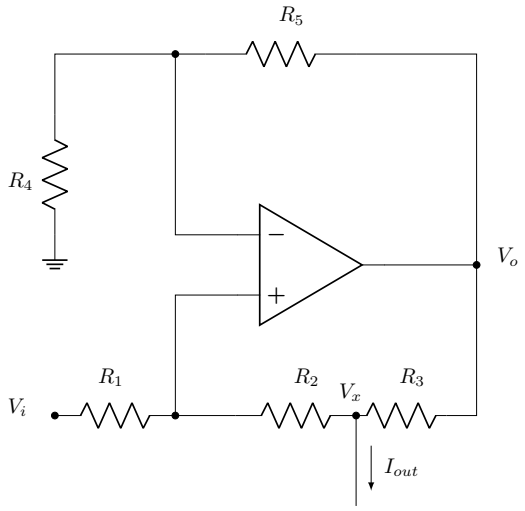


Figure 69: Improved Howland Current Pump Schematic

Applying Kirchhoff's Current Law (KCL) at the non-inverting and inverting input nodes gives:

$$\frac{V_i - V_+}{R_1} = \frac{V_+ - V_x}{R_2} \tag{81}$$

and:

$$\frac{-V_-}{R_4} = \frac{V_- - V_o}{R_5} \tag{82}$$

From (81), V_+ can be isolated as:

$$V_+ = \frac{R_2 V_i + R_1 V_x}{R_1 + R_2} \tag{83}$$

Similarly, from (82), V_- is obtained as:

$$V_- = \frac{R_4}{R_4 + R_5} V_o \tag{84}$$

Assuming an ideal operational amplifier with infinite open-loop gain, the virtual short condition applies:

$$V_+ = V_- \tag{85}$$

Equating the two previous expressions and solving for V_o yields:

$$V_o = \frac{V_i R_2 (R_4 + R_5) + V_x R_1 (R_4 + R_5)}{R_4 (R_1 + R_2)} \tag{86}$$

The output current is defined as:

$$I_{out} = \frac{V_+ - V_x}{R_2} + \frac{V_o - V_x}{R_3} \tag{87}$$

Substituting the expressions of V_+ and V_o into the previous equation and simplifying gives:

$$I_{out} = \frac{V_i(R_3R_4 + R_2R_4 + R_2R_5) + V_x(R_1R_5 - R_2R_4 - R_3R_4)}{R_3R_4(R_1 + R_2)} \tag{88}$$

In order to make the output current independent of V_x, the coefficient multiplying V_x must be equal to zero:

$$R_1R_5 - R_2R_4 - R_3R_4 = 0 \tag{89}$$

This leads to the matching condition:

$$\frac{R_1}{R_2 + R_3} = \frac{R_4}{R_5} \tag{90}$$

Applying this condition to the expression of I_{out} results in:

$$I_{out} = \frac{R_1R_5 + R_2R_5}{R_3R_4(R_1 + R_2)}V_i \tag{91}$$

Finally, if the additional condition:

$$R_4 = R_5 \tag{92}$$

is satisfied, the expression simplifies to:

$$I_{out} = \frac{V_i}{R_3} \tag{93}$$

B

Complete Circuit Schematic

B.1 Top Sheet: Block Schematic

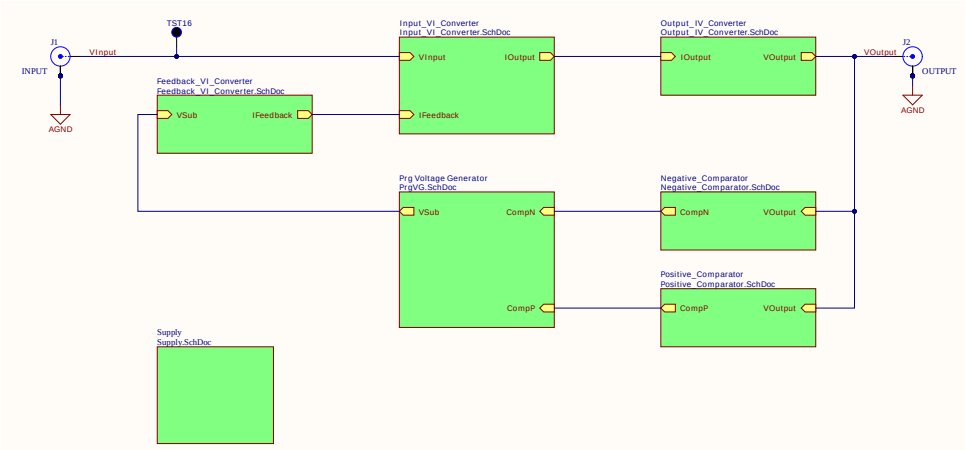


Figure 70: Altium Schematic: Top Sheet

B.2 Input Voltage to Current Converter

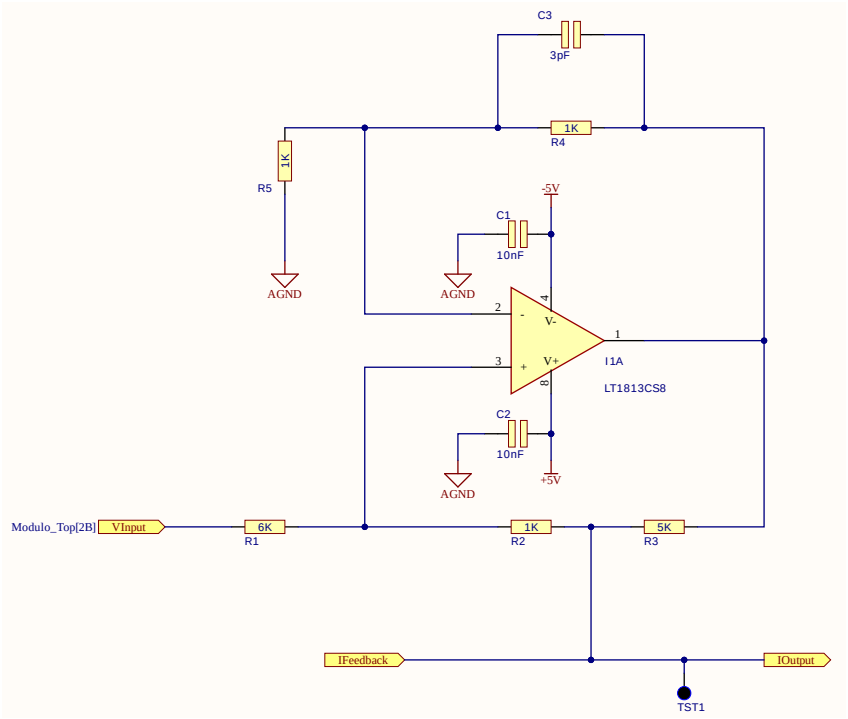


Figure 71: Altium Schematic: Input Voltage to Current Converter

B.3 Output Curren to Voltage Converter

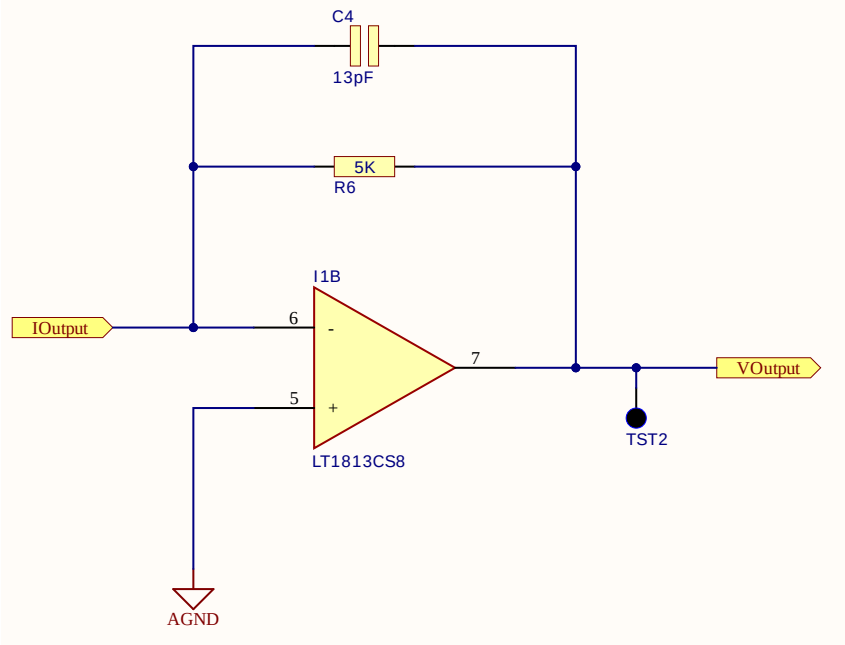


Figure 72: Altium Schematic: Output Current to Voltage Converter

B.4 Comparator comp-N

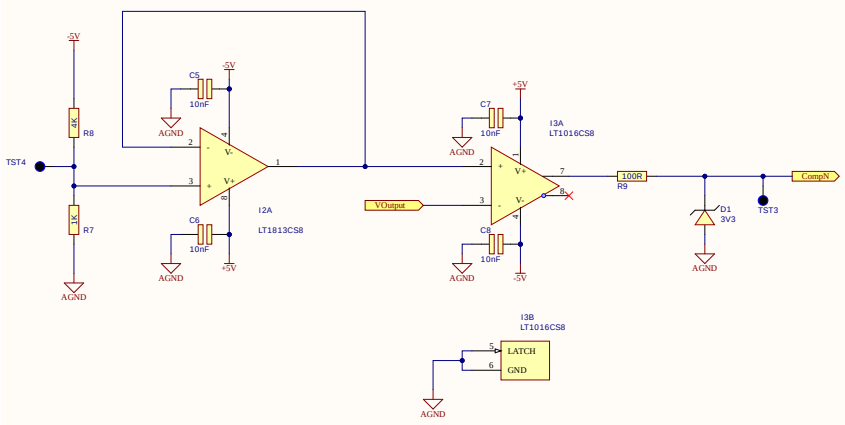


Figure 73: Altium Schematic: Comparator - N

B.5 Comparator comp-P

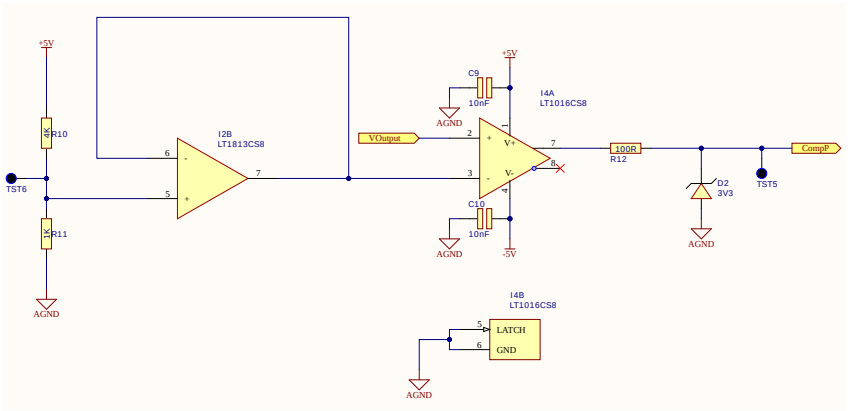


Figure 74: Altium Schematic: Comparator - P

B.6 Programmable Voltage Generator (PVG)

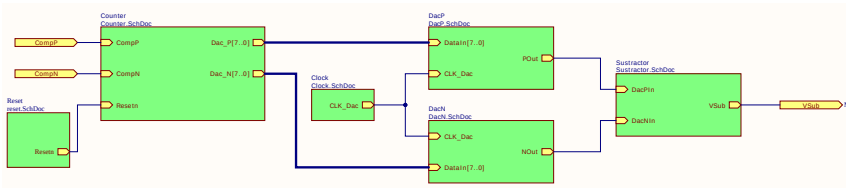


Figure 75: Altium Schematic: PVG

B.7 Counter: CPLD

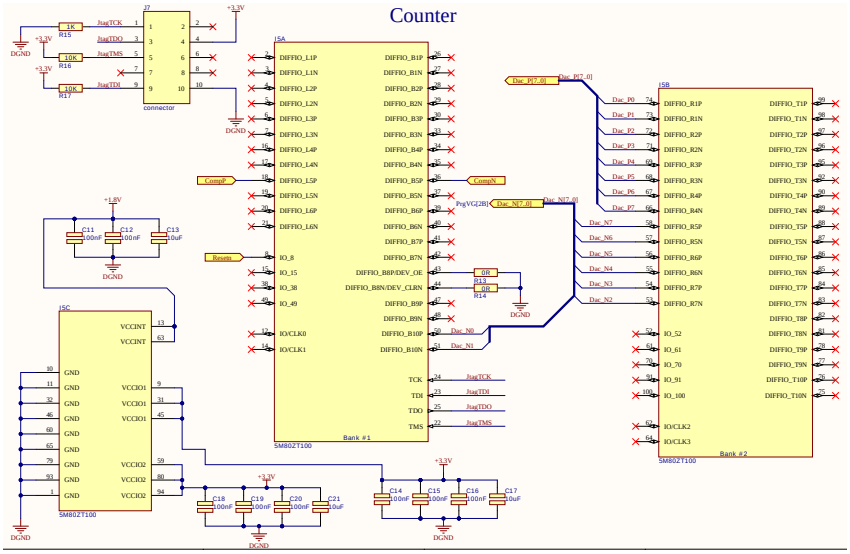


Figure 76: Altium Schematic: CPLD

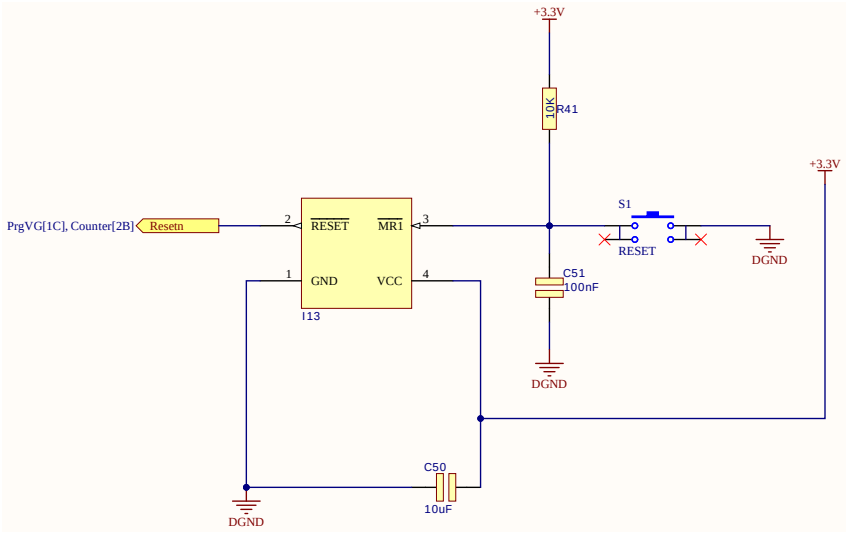


Figure 77: Altium Schematic: Reset CPLD

B.8 Clock

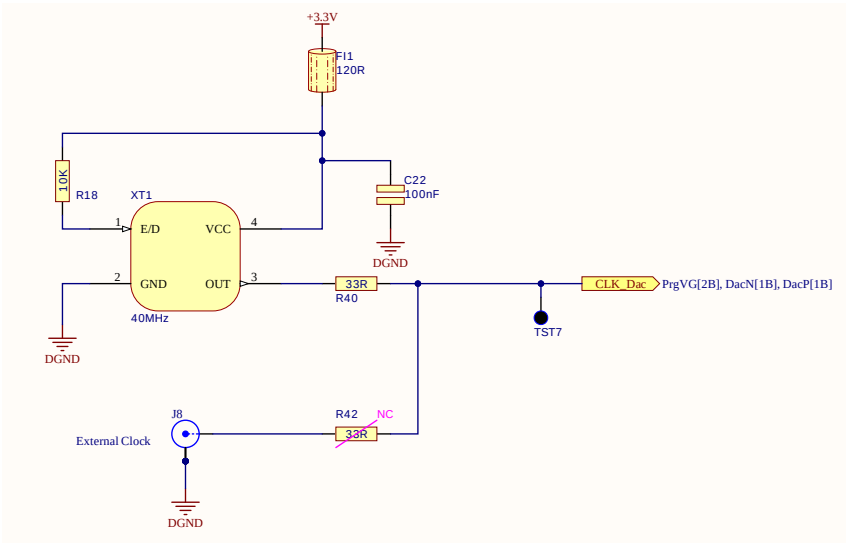


Figure 78: Altium Schematic: Clock

B.9 DAC-N

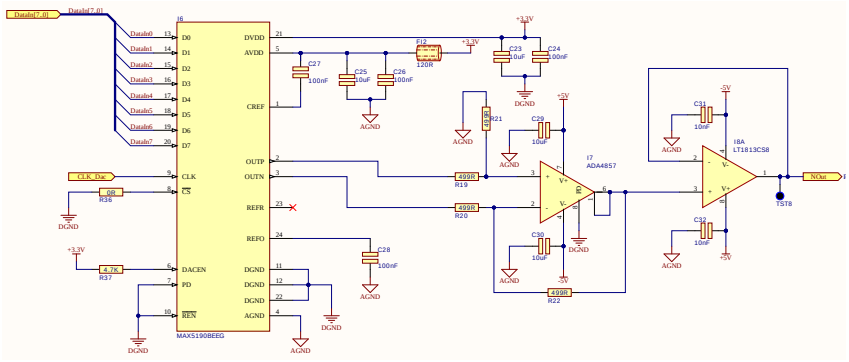


Figure 79: Altium Schematic: DAC-N

B.10 DAC-P

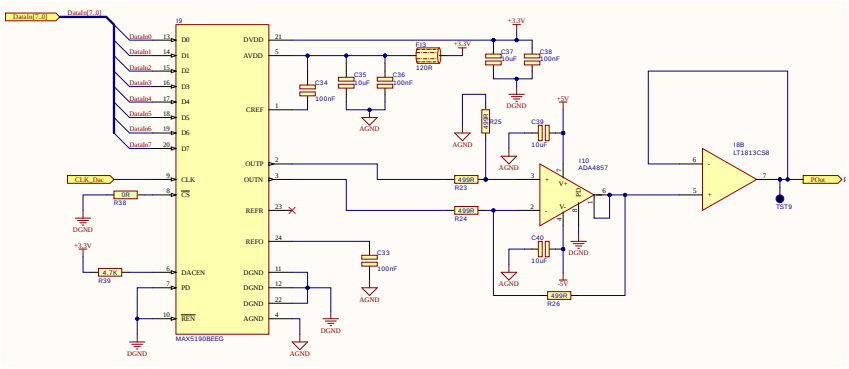


Figure 80: Altium Schematic: DAC-P

B.11 Subtractor

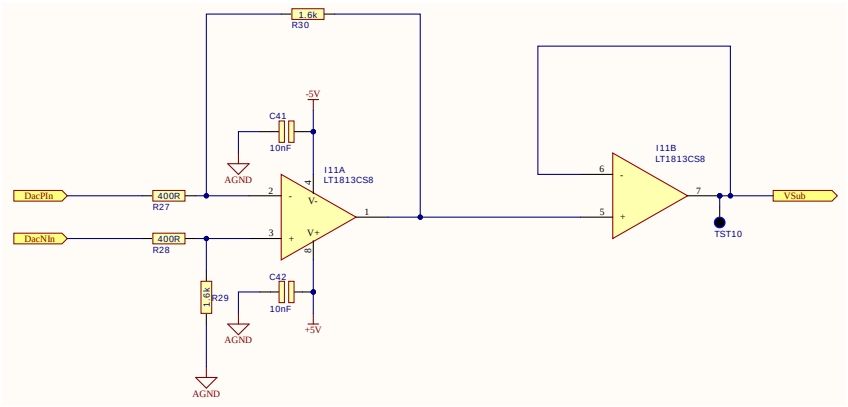


Figure 81: Altium Schematic: Subtractor

B.12 Feedback Voltage to Current Converter

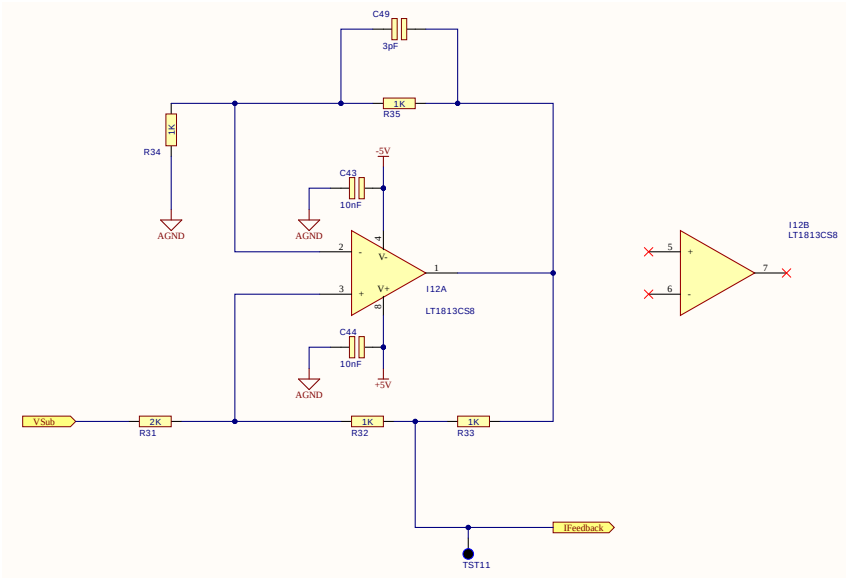


Figure 82: Altium Schematic: Feedback Voltage to Current Converter

B.13 Supply

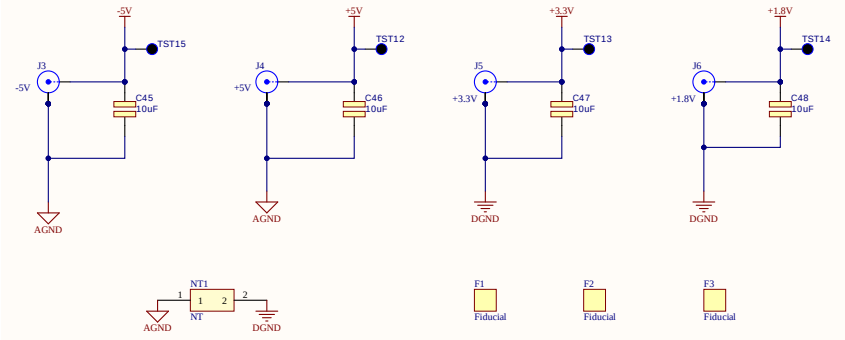


Figure 83: Altium Schematic: Supply

C

List of Components

Component	Designator(s)	Quantity	Description
10nF Capacitor	C1, C2, C5–C10, C31, C32, C41–C44	14	COG 25V 0603 10%
3pF Capacitor	C3, C49	2	COG 50V 0603
13pF Capacitor	C4	1	COG 50V 0603 5%
100nF Capacitor	C11, C12, C14–C20, C22, C24, C26–C28, C33, C34, C36, C38, C51	18	COG 50V 0603 5%
10uF Capacitor	C13, C17, C21, C23, C25, C29, C30, C35, C37, C39, C40, C45–C50	16	COG 25V 0603 10%
3.3V Zener Diode	D1, D2	2	BZX84 3V3 SOT23
Ferrite Bead 120R	FI1–FI3	3	2A 0603 Ferrite Bead
LT1813CS8	I1, I2, I8, I11, I12	5	Dual 100MHz Operational Amplifier
LT1016CS8	I3, I4	2	Ultra-fast Precision Comparator
5M80ZT100	I5	1	MAX V CPLD TQFP100
MAX5190BEEG	I6, I9	2	8-bit 40MHz DAC
ADA4857	I7, I10	2	Operational Amplifier
MAX6444	I13	1	Voltage Supervisor
Input Connector	J1	1	BNC Vertical Connector
Output Connector	J2	1	BNC Vertical Connector
−5V Connector	J3	1	BNC Vertical Connector
+5V Connector	J4	1	BNC Vertical Connector
+3.3V Connector	J5	1	BNC Vertical Connector
+1.8V Connector	J6	1	BNC Vertical Connector
Programming Connector	J7	1	Vertical 2×5 Connector
External Clock Connector	J8	1	BNC Vertical Connector
6kΩ Resistor	R1	1	0603 0.1%
1kΩ Resistor	R2, R4, R5, R7, R11, R15, R32–R35	10	0603 0.1%
5kΩ Resistor	R3, R6	2	0603 0.1%
100Ω Resistor	R9, R12	2	0603 1%
0Ω Resistor	R13, R14, R36, R38	4	0603 Jumper
10kΩ Resistor	R16, R17, R18, R41	4	0603 0.1%
499Ω Resistor	R19–R26	8	0603 1%
1.6kΩ Resistor	R29, R30	2	0603 0.5%
2kΩ Resistor	R31	1	0603 0.1%
4.7kΩ Resistor	R37, R39	2	0603 1%
4kΩ Resistor	R8, R10	2	0603 1%
400Ω Resistor	R27, R28	2	0603 1%
33Ω Resistor	R40, R42	2	0603 1%
Reset Push Button	S1	1	SPST SMD Push Button
Test Points	TST1–TST16	16	Keystone 5001 Test Point
SiT9005	XT1	1	40MHz MEMS Oscillator

Table 14: Bill of materials of the PCB prototype.

D

FSM Implementation: Verilog Code

```
module modulo_cpld (
    input  N_trig,
    input  P_trig,
    input  rst,
    input clk,

    output reg [7:0] DAC_1,
    output reg [7:0] DAC_2,
    output [2:0] state_out
);

// ====================================================
// State Encoding
// ====================================================
localparam S_IDLE   = 3'b000,
           N_Step_1 = 3'b001,
           N_Step_2 = 3'b010,
           P_Step_1 = 3'b011,
           P_Step_2 = 3'b100;

reg [2:0] state;
assign state_out = state;

// ====================================================
// Synchronizers for external triggers
// ====================================================
reg N_sync_0, N_sync_1;
reg P_sync_0, P_sync_1;

always @(posedge clk) begin
    N_sync_0 <= N_trig;
    N_sync_1 <= N_sync_0;

    P_sync_0 <= P_trig;
    P_sync_1 <= P_sync_0;
end

// Rising edge detection
wire N_rise = N_sync_0 & ~N_sync_1;
wire P_rise = P_sync_0 & ~P_sync_1;

// ====================================================
// Fully synchronous FSM
// ====================================================
always @(posedge clk or negedge rst) begin

    if (!rst) begin
        state <= S_IDLE;
        DAC_1 <= 8'h00;
        DAC_2 <= 8'h00;
    end
    else begin

        case (state)

        // ================= IDLE =================
        S_IDLE: begin
            if (N_rise) begin
                state <= N_Step_1;
                DAC_2 <= 8'b0011_0011;
                DAC_1 <= 8'h00;
```

```
            end
            else if (P_rise) begin
                state <= P_Step_1;
                DAC_2 <= 8'h00;
                DAC_1 <= 8'b0011_0011;
            end
        end

        // ================= N PATH =================
        N_Step_1: begin
            if (N_rise) begin
                state <= N_Step_2;
                DAC_2 <= 8'b0110_0110;
                DAC_1 <= 8'h00;
            end
            else if (P_rise) begin
                state <= S_IDLE;
                DAC_2 <= 8'h00;
                DAC_1 <= 8'h00;
            end
        end

        N_Step_2: begin
            if (P_rise) begin
                state <= N_Step_1;
                DAC_2 <= 8'b0011_0011;
                DAC_1 <= 8'h00;
            end
        end

        // ================= P PATH =================
        P_Step_1: begin
            if (P_rise) begin
                state <= P_Step_2;
                DAC_2 <= 8'h00;
                DAC_1 <= 8'b0110_0110;
            end
            else if (N_rise) begin
                state <= S_IDLE;
                DAC_2 <= 8'h00;
                DAC_1 <= 8'h00;
            end
        end

        P_Step_2: begin
            if (N_rise) begin
                state <= P_Step_1;
                DAC_2 <= 8'h00;
                DAC_1 <= 8'b0011_0011;
            end
        end

        default: begin
            state <= S_IDLE;
            DAC_1 <= 8'h00;
            DAC_2 <= 8'h00;
        end

        endcase
    end
end

endmodule
```

Listing 1: Verilog implementation of the CPLD finite-state machine.

E

Recovery Algorithm Implementation

```python
import numpy as np


# Round Function
def RD(x, L):
    """
    x: Original Signal
    L: Threhsold
    """
    return np.round(x / (2 * L)) * 2 * L

# Origin Centered Modulo Function
def M(x, T):
"""
    x: Original Signal
    L: Threhsold
    """
    return np.mod(x + L, 2 * L) - L

def Bg(y, L):  # the one to use
    return np.ceil(np.max(abs(y)) / 2 / L) * 2 * L

# Difference Order
def DO_US(T, L, Bg):
    """
    T: Period
    L: Threhsold
    Bg: Computed Parameter
    """
    return int(np.ceil((np.log(L) - np.log(Bg)) / np.log(T * np.exp(1))))

def US_algo(y, Bg, DO, L):
    DnE = M(np.diff(y, DO), L) - np.diff(y, DO)
    sn = np.array(DnE)
    sn1 = 0
    for i in range(DO - 1):
        sn1 = S(sn)
        sn1 = RD(sn1, L)
        sn1 += 2 * L * kn(sn, L, Bg)
        sn = sn1
    rec = y + S(sn)
    return rec
```

Listing 2: Python Implementation of the Reconstruction Algorithm

F

Complementary Graphs

F.1 Noise PSD: Measurement Under DC Bias

Figure 84 shows the output noise PSD measured with a 0.5 V DC bias applied at the input using a DC power supply, instead of short-circuiting the input to ground.

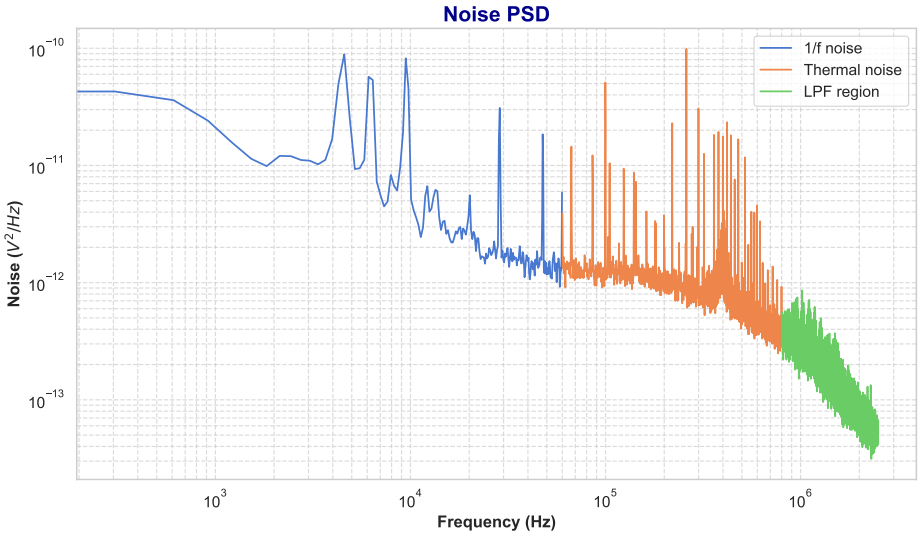


Figure 84: Measurements: Output Noise PSD with a DC signal

F.2 Clock Signal Measurement

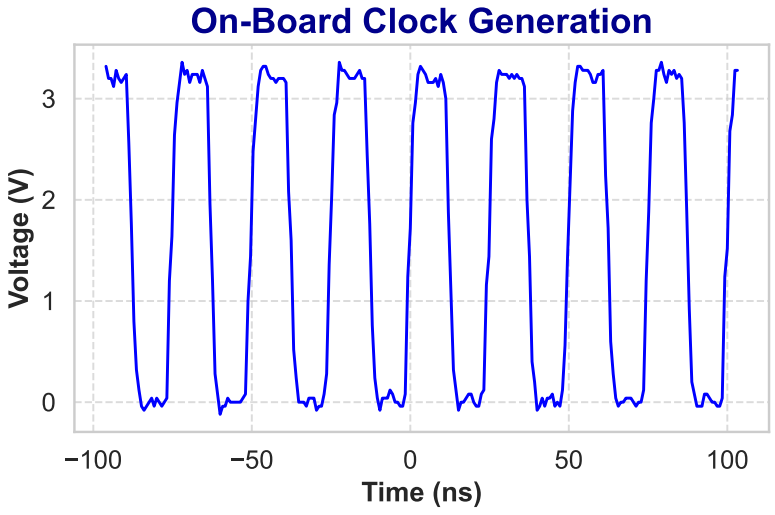


Figure 85: Measurements: Measured 40 MHz clock signal generated on the PCB

G

Final Performance Summary

Parameter	Final Performance
Core Functionality	
Transfer Function	$V_{out}(t) = ((V_{in}(t) + \lambda) \bmod 2\lambda) - \lambda$
Threshold Voltage λ	$\pm\lambda = \pm 1$ V
Output Voltage Range	$V_{out} \in [-1.25\text{ V}, 1.25\text{ V}]$
Minimum Number of Folds	0
Maximum Number of Folds	2
Input Signal Characteristics	
Minimum Input Voltage	$V_{in,\min} = -4.6$ V
Maximum Input Voltage	$V_{in,\max} = 4.6$ V
Input Dynamic Range	9.2 V_{pp}
Minimum Input Frequency	DC
Maximum Input Frequency	75 kHz for 9 V_{pp} input 270 kHz for 3 V_{pp} input
Cutoff Frequency	2.45 MHz
Performance Metrics	
Effective Number of Bits (ENOB)	9 bits
Electrical Characteristics	
Input Stage Supply Voltage	$\lvert V_{\text{supply,in}}\rvert = 5$ V
Other Analog Blocks Supply Voltage	$\lvert V_{\text{supply}}\rvert = 5$ V
Other Digital Blocks Supply Voltage	$V_{\text{supply}} = 3.3$ V
CPLD Core Supply Voltage	$V_{\text{core}} = 1.8$ V

Table 15: Final Performance Summary of the Modulo-ADC PCB Prototype

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**