Pipeline ADC Built-In Self Test

Guillaume Renaud

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE la Communauté UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Nano-électronique et Nano-technologies**

Arrêté ministériel : 7 Août 2006

Présentée par

Guillaume Renaud

Thèse dirigée par **Salvador Mir**
et co-encadrée par **Manuel J. Barragán**

préparée au sein **TIMA**
et de **l'École Doctorale Électronique, Électrotechnique, Automatique et Traitement du Signal (E.E.A.T.S)**

Auto Test de Convertisseurs de Signal de Type Pipeline

Thèse soutenue publiquement le **29 Novembre 2016**,
devant le jury composé de :

M. Dominique Dallet
Professeur, IMS/Bordeaux INP, Président

Mme. Adoración Rueda
Professeur, IMSE-CNM/Institut de Séville, Rapporteur

M. Serge Bernard
HDR, LIRMM/Université de Montpellier, Rapporteur

M. Hervé Naudet
Ingénieur senior, STMicroelectronics Grenoble, Examinateur

M. Salvador Mir
Directeur de recherche CNRS, TIMA/Université Grenoble-Alpes, Directeur de thèse

M. Manuel J. Barragán
Chargé de recherche CNRS, TIMA/Université Grenoble-Alpes, Co-Encadrant

To my family and my fiancée

Acknowledgments

I would like to first and foremost thank my advisors Salvador Mir and Manuel Barragán for their guidance and support. I learned a lot during my PhD thesis, and it is definitely thanks to them. I would also like to thank Salvador and Emmanuel Simeu for hosting me in the RMS team.

Many thanks also go to the people who helped me during these years. I would like to acknowledge the staff of TIMA Laboratory for their efficient help (Anne-Laure Fourneret Itié, Laurence Bentito, Youness Rajab, Frédéric Chevrot, Ahmed Khalid and Nicolas Garnier). Thanks to Mamadou Diallo for the design of the test chip and the evaluation boards, and thanks to Alexandre Chagaloya for the support during the test chip design phase. Morevover, thanks to Skandar Basrour for letting me access and use the equipment of the test room facilities at CIME. Thanks to Slim Bellil and Robin Rolland for their support on the testing platform, and thanks to the MNS/Hap2U squad (Mickaël, Emilie, François, Adrian, Achraf) for kind of accepting me in the team while I was performing the measurements. Thanks to Hervé Le Gall, Hervé Naudet, and Alexandre Proust for the technical support at STMicroelectronics.

Sincere thanks to the members of the jury for accepting and finding the time in their respective agendas, especially Adoración Rueda, Dominique Dallet, and Serge Bernard for travelling long distances in order to attend my PhD defense.

I would like to thank my (former) colleagues and close friends from the RMS team (Martin, Thanasis, Asma, Rshdee, Hani, Laurent, Brice, Diane) and from other teams (Simon, Alban) for the technical help and the good times we spent together in (and outside) the lab.

Many (if not all) of those who are PhD now know that the PhD thesis is not an easy path to walk, so I would like to specially thank my friends, my family, and my fiancée for their caring love and support during rough times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Context

In today's modern society, electronics are taking a prominent position as they are more and more associated to our daily life. Within the last 50 years, the semiconductor industry have shown a dramatic, exponential growth under the lead of Moore's law [**?**]. Integrated circuits (IC) are now widely used and serve multiple purposes for a large number of applications: consumer goods, avionics, automotive, medical applications, telecommunications, computing, etc.

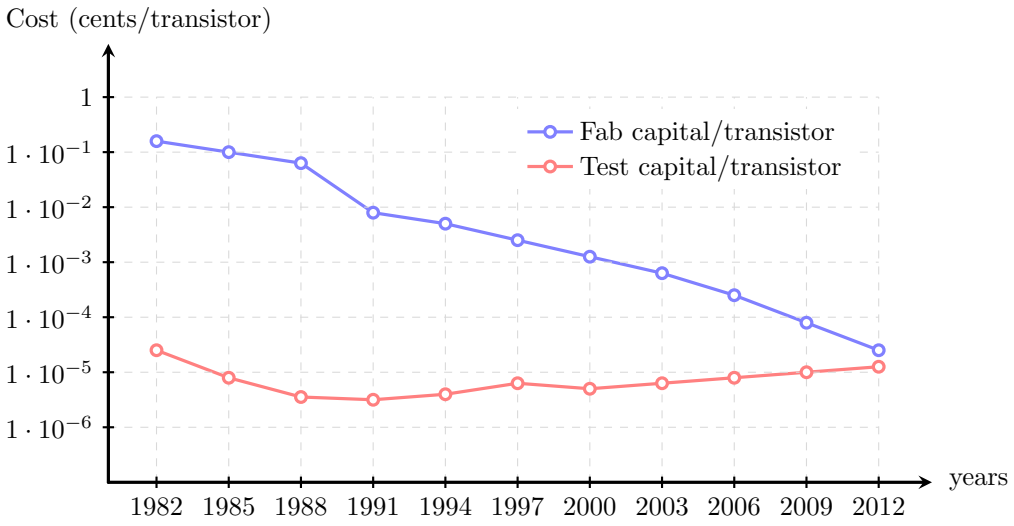


Figure 1.1: Cost of silicon manufacturing and test [**?**]

While the manufacturing cost is steadily dropping (more than 30% cost reduction per year over the last 50 years), test cost tends to rise, or at least remains constant, as seen in Figure 1.1, extracted from the International Technology Roadmap for Semiconductors [**?**]. Indeed, the increasing complexity on a single die makes the test more and more difficult to perform, especially for analog and mixed-signal ICs. The primary role of production testing is to ensure that no defective devices are mistakenly sent to the market.

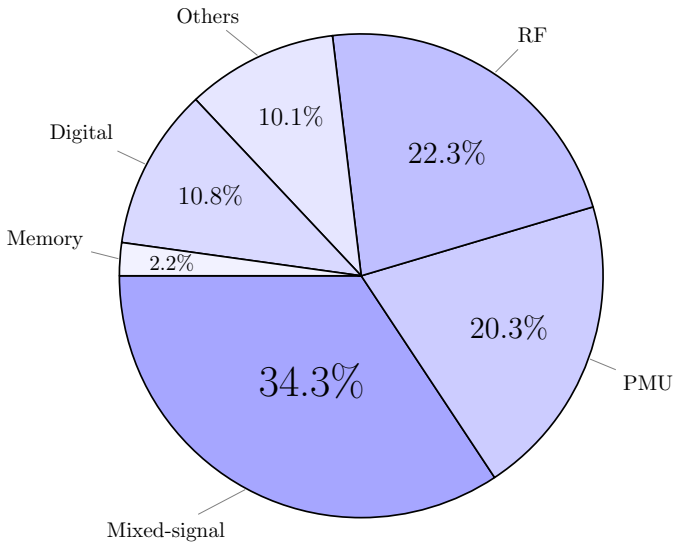


Figure 1.2: Pie chart of the test times per circuitry of a mobile phone SoC [**?**]

As a consequence, more than 50% of the overall production cost can be dedicated to IC testing during this crucial phase [**?**]. We can take the example of system-on-chips (SoC), containing analog, mixed-signal, digital, RF circuitry, even MEMS and optical devices on a single die. They have been developed since a few decades in an effort to reduce fabrication cost by embedding a complex system with a large set of functionalities on a single ship, with a higher level of integration, using new advanced deep submicron technologies. Testing such highly-integrated devices is a challenging task that has a direct impact on the yield and the throughput of the fabrication process. As an example, Figure 1.2 depicts the test time distribution of a mobile phone SoC. It takes into account the test, at wafer and package level, for the different subsections of the SoC (i.e. analog, digital, RF, power, etc.). It can be seen that the test time of the mixed-signal blocks is taking more than one third of the total test time (34.3%). RF and power management unit (PMU) blocks are also taking a significant amount of time compared to digital and memory devices in contrast with the total area occupation, that is dominated by digital circuitry ($\gtrsim$ 90 % of the area of state-of-the-art SoCs).

A lenient test would indeed reduce test time and hence improve the yield and throughput of the fabrication process. However, we cannot forget that the cost of shipping defective parts, what is usually denoted as test escape (TE), is higher (approximately $\times 10$) than the cost associated to yield loss (YL). Even more, it can damage a company public image.

1.2. Industrial production testing

Current industrial test procedures for embedded analog and mixed-signal devices are becoming a major bottleneck in the production of these systems. Manufacturers must check the functionality stated in the data sheet of every IC before shipping them. Actually, the only selling point of a given IC is the set of performance figures listed in its data sheet, hence the importance of guaranteeing these figures to the customer with a proper and comprehensive test program.

Cost effective methods for testing the digital circuitry, based on standardized fault models, are already available. Digital circuits handle a discrete number of signal states, usually at discrete time intervals, and can then be studied from a high level of abstraction. This allows to define efficient structural test techniques aimed at detecting fabrication defects in the circuitry. Digital test is nowadays a (mostly) automated process in which an Automated Test Pattern Generator (ATPG) is used to generate an optimized test sequence that maximizes the defect detection rate for a given circuit based on a standardized set of fault models. Moreover, the silicon implementation of the test is also semi-automatic due to the adoption of standardized test access and BIST circuitry based on scan-chain shift-registers.

On the other hand, testing analog, mixed-signal, and RF circuits still relies on costly functional characterization. In contrasts to their digital counterparts, these circuits handle continuous signals and states affected by many sources of variability, and correct functionality can only be defined in terms of intervals or acceptance regions. Moreover, the relation between signals, states and variation sources is often non-linear and multidimensional. In this scenario, any standardization of a fault model is a challenging task (in spite of interesting recent advances in this line [**?**],[**?**],[**?**]), and the default test strategy relies on the complete functional characterization of the set of circuit performances. Similarly, standardized BIST techniques are also lacking.

In current industrial AMS production test, an ATE generates a very well-controlled test stimulus at the DUT input, and acquires its output for further post-processing and performance assessment [2]. The simplified test bench is illustrated in Figure 1.3, and an example of widely-used ATE is shown in Figure 1.4. This is a time-consuming procedure, that is sensitive to environmental noise, loading conditions, etc. Furthermore, the test equipment, which consists of the ATE itself, the device interface board (DIB) and other daughter boards, is expensive because of the accuracy requirements for testing high-performance mixed-signal devices.

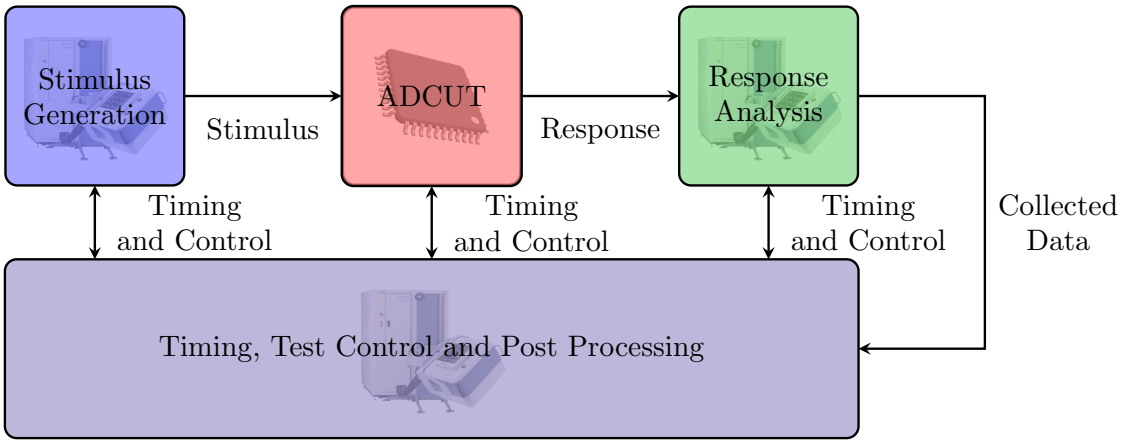


Figure 1.3: Simplified scheme of an industrial testbench

Nowadays, ATE suppliers are proposing a modular tester approach. The tester is equipped with different type of options depending on the DUT requirements: digital options, analog options, RF options, power options, etc. The DUT content (i.e. analog, digital, RF, power, etc.) will then define the tester configuration (tester options) and obviously the tester cost. Test cost can be split into two main components. To a first order, the capital expenditure (CapEx) is defined as the cost of ATE hardware, and the operational expenditure (OpEx) is defined as the labor cost ensuring the ATE is up and running. Other expenses (recurrent or not) are summing to the total cost as test boards, test sockets, or maintenance. However, these can be considered as second order costs. On an industrial point of view, labor cost can be reduced, so the main cost differentiator is the CapEx component. Moreover, when an ATE investment is amortized, the CapEx component disappears from the test cost structure, and the OpEx component is becoming dominant. Nevertheless, due to aging of the test equipment, the maintenance requirement is increasing over time, the CapEx component is still present because of the new spare parts cost.

Furthermore, the analog content of DUT directly have an impact on the CapEx component. The impact of an analog option onto the tester configuration cost is depending on the number of analog channels to address. Using 20% of the total number of analog channels yields about 75% of the tester configuration cost.

1.3. Motivation

Focusing on AMS testing, the ATE must be equipped with analog and digital options. Table 1.1 shows the evolution of the CapEx component for both analog and digital options recorded for three consecutive generations of instruments (roughly equivalent to a

Figure 1.4: Verigy V93000 mixed-signal automated test equipment

15-year period) [**?**]. Over this time, the digital option integration factor has been more effective than the analog one. This could be explained by the fact that digital options take benefit of semiconductor Moore's law (reduced size, increased speed), while analog options performance is not linked to unitary transistor size. In average, the option cost has been decreased by 25% from one generation to another, regardless of the type of option (analog or digital). The cost ratio between a digital tester channel and an analog tester channel is constant along the three generations: an analog tester channel costs about ten times more than a digital tester channel.

In an effort to reduce the test cost, the BIST approach is a promising solution. Built-In-Self-Test consists in integrating some of the ATE functionality on the DUT itself, namely test stimulus generation, test response evaluation and test protocol control. This approach provides several advantages such as wafer-to-system testability, good test quality, at-speed testing, reduced need for ATE, reduced development time, more economical testing, reduced production test time and cost, and reduced time-to-market. Moreover, BIST advantages are not limited to production test. BIST enables in-the-field test during the complete lifetime of the DUT, it may be used for self-repair and adaptive operation, aging monitoring, self-test in inaccessible or rough environment, etc. However, it also comes at the expense of area overhead, risk of performance degradation, and additional design time. Moreover, a lot of challenges, especially for AMS testing, such as area overhead, test quality requirements in state-of-the-art applications, and test time reduction, are still challenging.

In this thesis, we will focus on ADC testing, and particularly on static linearity testing

Table 1.1: Comparison of ATE analog and digital options cost

	Analog option CapEx			Digital option CapEx			Digital/Analog channel cost ratio		
Generation	1	2	3	1	2	3	1	2	3
# of channels per option (integration factor)	Ref.	×2	×4	Ref.	×1.25	×1.25			
Cost per channel evolution	Ref.	-11.3%	-39.8%	Ref.	-31.5%	-16.0%	9.9%	12.8%	9.2%
Cost per channel evolution (average)		-25.5%			-23.7%				

of pipeline ADCs. In this framework, ADCs are key components amongst mixed-signal devices because they can be found in every mixed-signal system along with digital-to-analog converters (DAC). They are the front-end between analog and digital sections in a mixed-signal system. Two types of test are usually performed: a static test targeted at static specifications such as differential and integral nonlinearities, gain and offset, and a dynamic test targeted at dynamic specifications such as equivalent number of bits, signal-to-noise ratio, signal-to-noise and distortion ratio, total harmonic distortion, spurious-free dynamic range, two-tone intermodulation distortion. During production testing of ADCs, Differential Non-Linearity (DNL) and Integral Non-Linearity (INL) are the two main metrics that are estimated for the static test.

In order to evaluate the static parameters of ADCs such as DNL and INL, standard testing techniques employ statistical methods. The main drawbacks appear when dealing with high resolution ADCs. The two main problems are the cost of the test equipment and the test time. Indeed, the input stimulus provided by the ATE has to be very linear, consequently tightening its specifications and increasing its cost. Moreover, the decreasing trend of the LSB amplitude makes the measurements more sensitive to noise. While the DNL (histogram technique) estimation is inherently robust versus Gaussian noise (stochastic behavior), INL estimation is not. A way to reach good INL measurements is to perform noise averaging. This implies a large amount of data to be collected and analyzed. Such a huge amount of data collection makes test time prohibitive.

A solution to the presented issues is to integrate the analog test stimulus generator inside the ADC under test (ADCUT). For example, the test of a 12-bit ADC should require one (single-ended or differential) analog channel for the ADC input, twelve digital channels for the ADC outputs and one more for the ADC clock input. In an embedded configuration, it can be computed that 43% of the CapEx cost is skipped because the analog channel is no longer needed. Another benefit of integrated signal generator is the

noise sensitivity. An internally generated signal routed outside the chip and looped back into the internal ADC might provide ten times more noise to the ADC input, which is problematic. The development of a static BIST method for high-resolution ADCs is then a promising solution for alleviating the cost of static test. Signal generation and manipulations remain internal to the system. The problem of accessing an embedded device is eliminated, and the requirements on the ATE, and hence its cost, are greatly reduced.

1.4. Outline of this thesis

This thesis aims at the exploration of a novel technique for BIST ADC testing. The goal consists of developing a low-cost stimulus generator along with efficient measurement techniques for the evaluation of the static performance of high resolution ADCs, with a focus on pipeline ADCs. A feedback loop solution is implemented using a modified servo-loop technique in order to measure the ADC static metrics. The manuscript contains six chapters organized as follows.

Chapter 2 details the techniques employed for the static test of ADCs during a typical production testing flow. A review of the state-of-the-art techniques in the field of ADC testing and BIST is shown. Several works are detailed on the generation of a high-linearity test stimulus for ADC testing and BIST purposes.

The proposed technique for ADC testing is explained in chapter 3. First, the concept of the technique is analyzed and studies are conducted to prove its efficiency for a BIST approach and show the trade-offs to consider. A comparison is made between the classical approach and our approach in terms of test time and accuracy. Then, a description of the pipeline ADC architecture and operation is provided. Finally, simulation results are presented on the modified technique in combination with a reduced-code linearity test algorithm.

In chapter 4, a high-linearity test stimulus generator designed for the BIST approach is described at system and transistor levels along with design techniques to reduce the inherent design nonidealities.

Chapter 5 details the simulation and experimental results of the proposed generator and BIST technique. First, the generator is characterized at transistor level using electrical simulations, and Monte-Carlo simulations are provided to verify the performance of the proposed generator. Then, the BIST strategy is evaluated using the data collected from the generator model using behavioral simulations. On the second part of the chapter, the results of the physical characterization of the fabricated ICs are presented along with

the test setup. Furthermore, the test chips are used to estimate the static linearity of a commercial ADC on its evaluation PCB, and validate the functionality of the intended stimulus generation.

Finally, chapter 6 concludes this thesis and discusses the directions for our future research on this topic.

Chapter 2

State of the art of ADC testing

2.1. ADC performance testing

The performance of an ADC is usually expressed in terms of two sets of specifications, namely static and dynamic specifications. In the scope of this thesis, we will focus on the description of the static specifications, namely the DNL, INL, gain error and input offset error.

2.1.1. Basic concepts

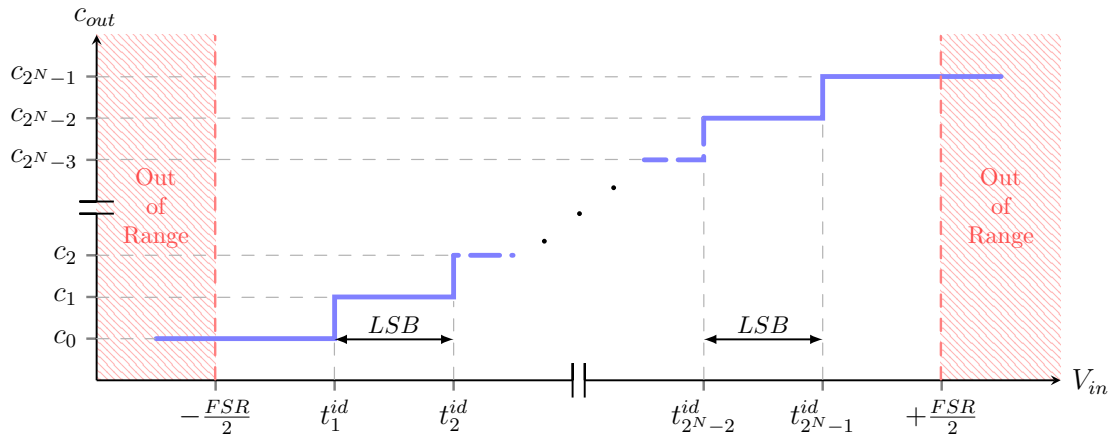
An analog-to-digital converter is a system that transforms samples of an analog signal x_k into a digital representation of these samples. Each analog sample is codified into a n-bit digital word c_k, where n is the ADC resolution and $c_k \in [0; 2^N - 1]$.

In general, an ideal ADC, as depicted in Figure 2.1, associates a different digital code to each of the 2^N intervals in the input analog Full-Scale Range (FSR), $[-FSR/2; FSR/2]$ (if bipolar, $[0; FSR]$ if unipolar), in such a way that, provided an analog input x, the output c of the ADC can be expressed as

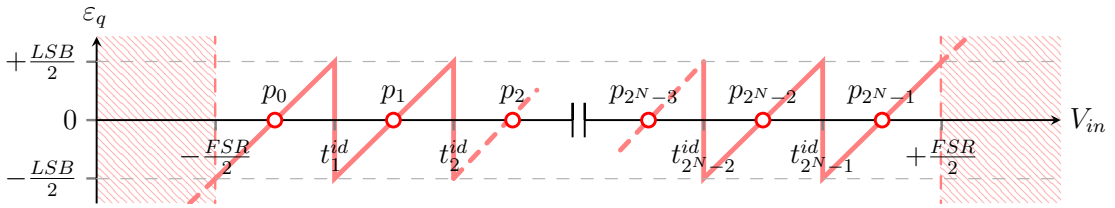
$$c = c_k \iff t_k \leqslant x < t_{k+1} \tag{2.1}$$

where t_k is the code transition threshold that defines the value at which the output of the converter switches from code c_{k-1} to code c_k. In an ideal converter, the distance between two consecutive thresholds is a constant, usually called Least Significant Bit (LSB), or converter quantum q. It can be expressed as

$$LSB = t_{k+1}^{id} - t_k^{id} \tag{2.2}$$



(a) Ideal characteristic (bipolar range with 2's complementary binary output)



(b) Quantization error

Figure 2.1: Transfer function of an ideal N-bit ADC

where t_k^{id} and t_{k+1}^{id} are the ideal transition voltages for codes c_k and c_{k+1}. Equivalently,

$$LSB = \frac{V_{ref}^+ - V_{ref}^-}{2^N} = \frac{FSR}{2^N} \tag{2.3}$$

where V_{ref}^+ and V_{ref}^- represent the maximum and minimum input voltage values, and define the ADC full-scale range FSR. They also correspond to the virtual transitions $t_{2^N}^{id}$ and t_0^{id}, respectively. For a bipolar ADC, $V_{ref}^+ = FSR/2$ and $V_{ref}^- = -FSR/2$. If the ADC is unipolar, $V_{ref}^+ = FSR$ and $V_{ref}^- = 0V$.

Moreover, the ADC can output different data representations. Amongst the most used are the Unipolar Straight Binary representation (USB) and the Binary Two's Complement representation (BTC). The USB representation is the most natural. It can be used for unipolar or bipolar input ranges, and sets $c_0 = 0$ and $c_{2^N-1} = 2^N - 1$. The BTC representation, usually associated to a bipolar analog input range, sets $c_0 = 2^{N-1}$ and $c_{2^N-1} = 2^{N-1} - 1$. The N-bit digital output is given a sign which depends on the value of its MSB.

If the ideal ADC transfer function depicted in Figure 2.1 (a) is subtracted to a straight line between $-FSR/2$ and $+FSR/2$, the residue is called the quantization error ε_q. Points

where ε_q is null are called potencies p_k which ideal expression is

$$p_k = t_k^{id} + 0.5LSB, \quad k = 0,1,...2^N - 1. \tag{2.4}$$

As seen in Figure 2.1 (b), finite amplitude resolution introduces a quantization error between the analog input voltage and the reconstructed output voltage. If it is assumed that the quantization error is a white noise having an uniform probability density function (PDF) over the code width from $-LSB/2$ to $+LSB/2$ defined as

$$PDF_{\varepsilon_q} = \left\{ \begin{array}{ll} 1/LSB & if \ \ |\varepsilon_q| < LSB/2 \\ 1 & otherwise \end{array} \right\}, \tag{2.5}$$

then the expected value of the quantization error power can be expressed as

$$P_{\varepsilon_q} = E[\varepsilon_q^2] = \frac{1}{LSB} \int_{-\frac{LSB}{2}}^{+\frac{LSB}{2}} \varepsilon_q^2 \cdot PDF_{\varepsilon_q} \ d\varepsilon_q = \frac{LSB^2}{12}. \tag{2.6}$$

This means that any input signal with a lower power than this power value is smeared into the ADC quantization noise.

2.1.2. Differential Non Linearity

The Differential Non Linearity of an ADC is defined as difference between a specified code width and the ideal code width being the LSB of the ADC, divided by the LSB. It is a consequence of static errors in the components of the ADC under test. For example, the offsets in each comparator of a flash ADC are responsible for this local nonlinearities. Using equation (2.2), the DNL for code c_i, before gain and offset compensation, can be analytically expressed as

$$DNL_k = \frac{W_k - LSB}{LSB}, \quad k = 1,2,...2^N - 2 \tag{2.7}$$

where $W_k = t_{k+1} - t_k$ is the width of code c_k. With equation (2.7), it can be noted that neither DNL_0 nor DNL_{2^N-1} are computed because the code widths W_0 and W_{2^N-1} are not clearly defined.

In Figure 2.2, the DNL values of codes c_{k-1}, c_k, and c_{k+1} are represented. For the given transitions, DNL_{k-1} and DNL_k have a positive value because W_{k-1} and W_k are above the LSB, whereas DNL_{k+1} has a negative value because W_{k+1} is below the LSB.

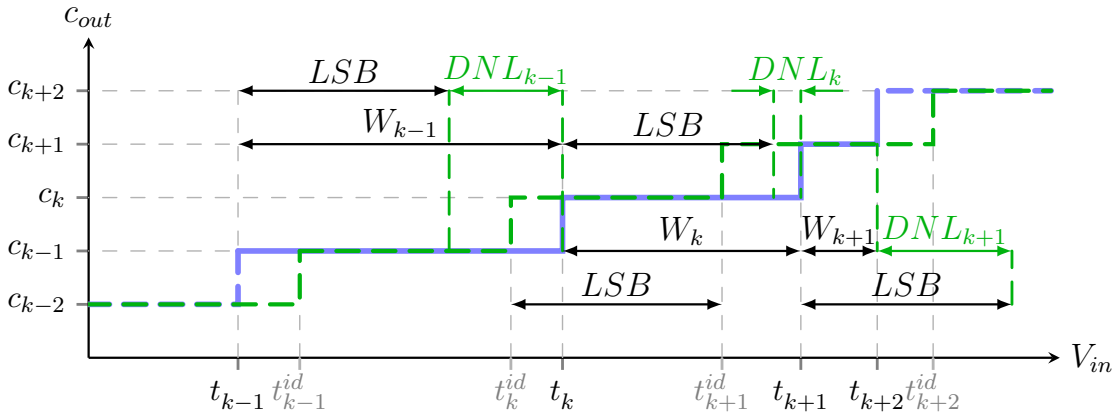


Figure 2.2: Representation of the DNL of an N-bit ADC

If the DNL of a code k is inferior to -1 LSB or superior to 1 LSB, then this code is a missing code. Those cases are shown in Figure 2.3. In Figure 2.3 (a), code c_k is missing because transition t_{k+1} is inferior to transition t_k, causing its DNL to be inferior to -1 LSB. Conversely, in Figure 2.3 (b), code c_k is missing because transition t_k is superior to transition t_{k+1}, causing its DNL to be superior to 1 LSB. In any case, if the DNL of each code is contained between -1 LSB and 1 LSB, the characteristic is assured to be monotonic.

Generally, the DNL of an ADC is expressed as the maximum of $|DNL_k|$ for all k.

2.1.3. Integral Non Linearity

The Integral Non Linearity of an ADC is defined as the difference of its transfer function with respect to the ideal transfer function line. It is seen as the cumulative sum of the DNL, so it can also take positive and negative values. The INL can be expressed before gain and offset compensation as

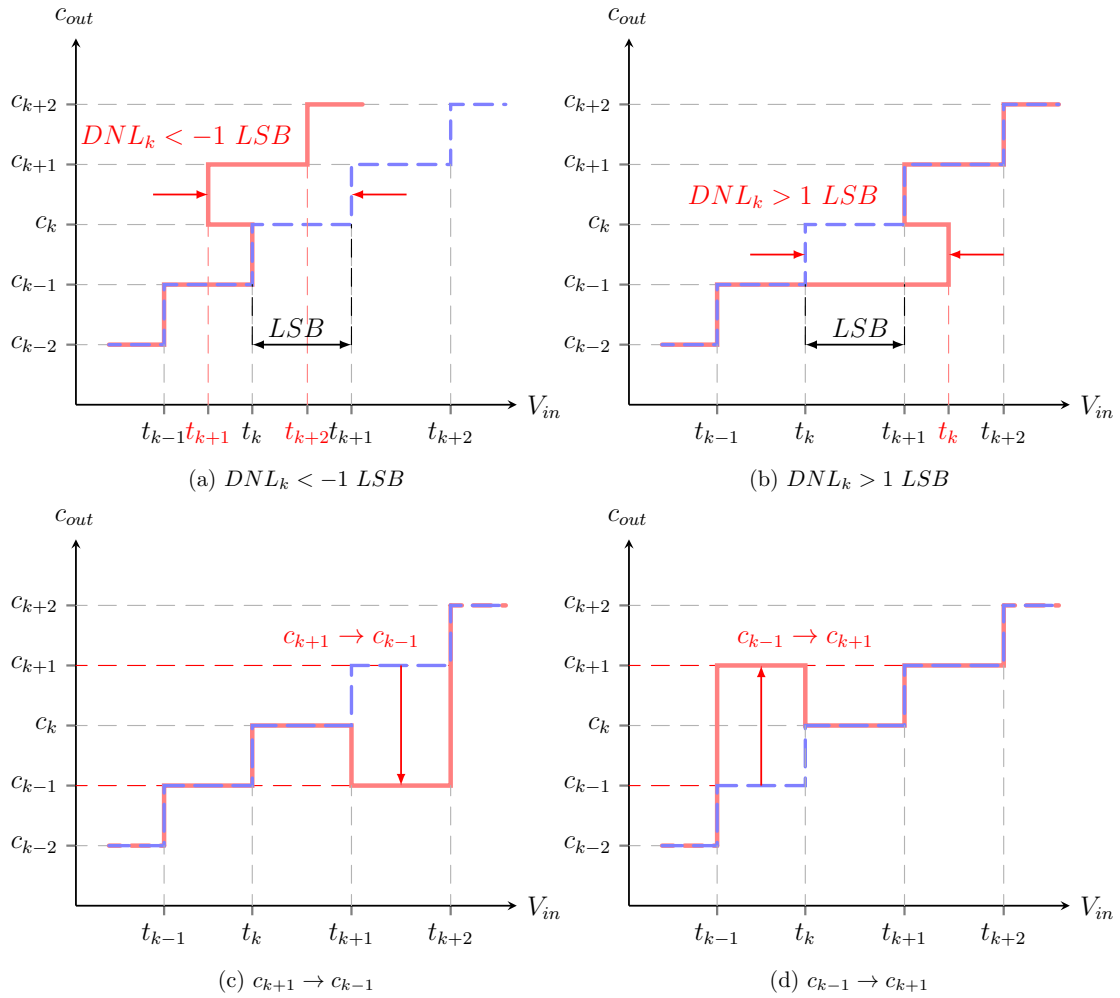
$$INL_k = \frac{t_k - t_k^{id}}{LSB}, \quad k = 1,2,...2^N - 2. \tag{2.8}$$

As an example, in Figure 2.4 are represented the INL values of codes c_{k-1}, c_k, c_{k+1} and c_{k+2}.

Generally, the INL of an ADC is expressed as the maximum of $|INL_k|$ for all k.

The INL can be computed from the DNL as

$$INL_k = \sum_{l=0}^{k} DNL_l, \quad k = 0,1,...2^N - 1 \tag{2.9}$$



(a) $DNL_k < -1\ LSB$ (b) $DNL_k > 1\ LSB$

(c) $c_{k+1} \rightarrow c_{k-1}$ (d) $c_{k-1} \rightarrow c_{k+1}$

Figure 2.3: Representation of missing codes for an N-bit ADC

or vice versa as

$$DNL_k = INL_k - INL_{k-1}, \quad k = 0, 1, ...2^N - 1. \tag{2.10}$$

2.1.4. Gain error

The gain error of an ADC is defined as the difference of the slope of the actual output values and the ideal output values. The actual gain of the ADC can differ from the ideal best-fit straight line and modify its real LSB value. In that case, with only the gain error taken into account, the transfer function for a given analog input x can be modeled as

$$c_k = \frac{x}{LSB_e} - \varepsilon_{q_e}, \quad t_k \leqslant x < t_{k+1} \tag{2.11}$$

with $LSB_e = LSB/G_a$ the effective LSB value being the real value divided by the actual gain G_a of the ADC and $\varepsilon_{q_e} = \varepsilon_q/G_a$ the effective quantization error of the ADC. After

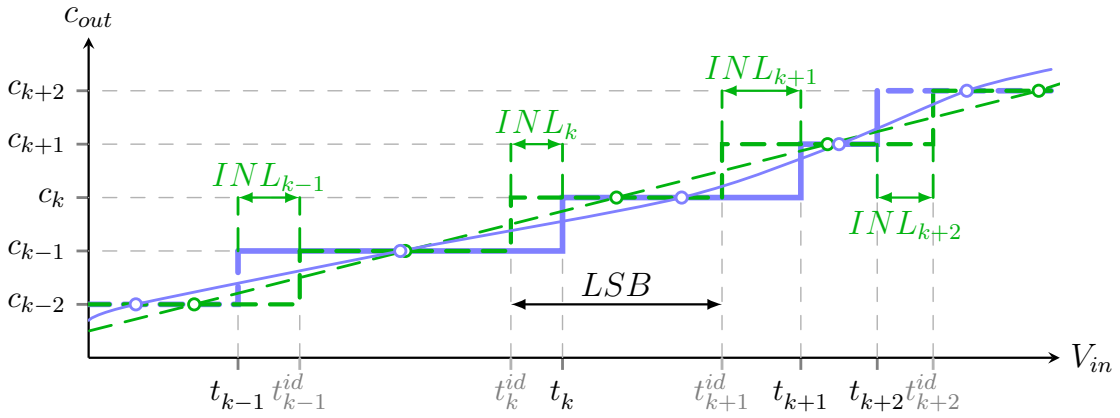


Figure 2.4: Representation of the INL of an N-bit ADC

performing offset calibration, one can then define the resulting gain error as the ratio between the actual slope and the ideal one, giving the expression

$$\varepsilon_G = \frac{G_a - G_{id}}{G_a} \times 100\% \tag{2.12}$$

where G_{id} is the ideal value of the ADC gain and G_a is expressed as

$$\begin{aligned} G_a &= \frac{dc_{out}}{dV_{in}} \\ &= \frac{c_{2^N-1} - c_1}{t^a_{2^N-1} - t^a_1} \\ &= \frac{2^N - 2}{t^a_{2^N-1} - t^a_1} \end{aligned} \tag{2.13}$$

with $t^a_{2^N-1}$ and t^a_1 being the actual transition voltages induced by the gain error.

2.1.5. Input offset error

The input offset error of an ADC is defined as a constant difference, over its FSR, between the actual output value and its ideal output value. It can be found when no input signal is fed to the ADC as the difference between the first actual transition and the first ideal transition of the ADC, namely

$$\varepsilon_{os} = V_{os_a} - V_{os_{id}}. \tag{2.14}$$

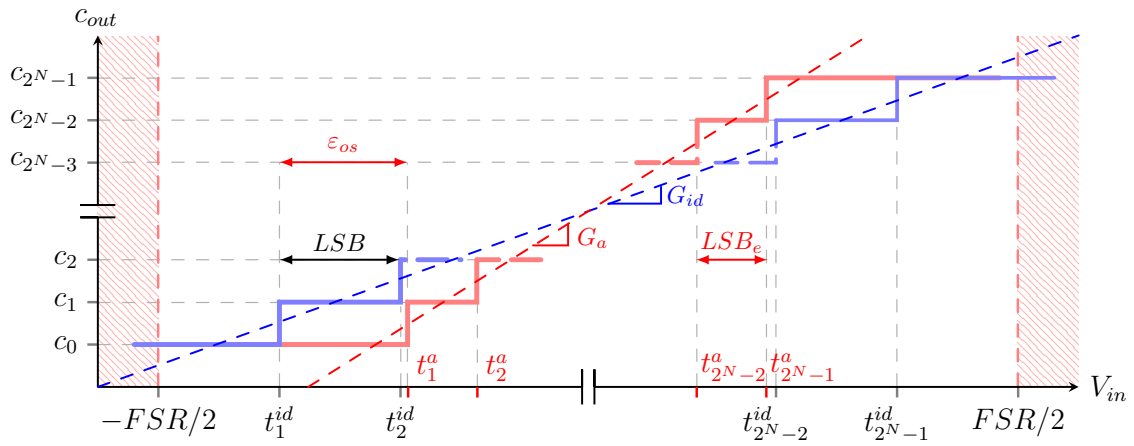


Figure 2.5: Representation of the gain and offset errors of an N-bit ADC

2.1.6. Correction of gain and offset

The gain error and the offset voltage can be corrected with a simple best-fit algorithm [1], which is used in this thesis work to also correct accumulated errors that leads in bad INL estimation.

First, four intermediate variables are computed from the estimated ADC nonlinearity as

$$\left\{ \begin{array}{l} k_1^0 = 0,\ k_2^0 = 0,\ k_3^0 = 0,\ k_4^0 = 0 \\ k_1 = k_1 + i \\ k_2 = k_2 + \widehat{INL}(i) \\ k_3 = k_3 + i^2 \\ k_4 = k_4 + i \cdot \widehat{INL}(i) \end{array} \right\}, \quad i = 1,2,...,2^N - 1. \tag{2.15}$$

Then, the gain error and offset error can be estimated from equation (2.15) as

$$\left\{ \begin{array}{l} \varepsilon_G = ((2^N-1)\cdot k_4 - k_1 \cdot k_2)/((2^N-1)\cdot k_3 - k_1^2) \\ V_{os} = k_2/(2^N-1) - \varepsilon_G \cdot k_1/(2^N-1) \end{array} \right\}, \quad i = 1,2,...,2^N - 1. \tag{2.16}$$

Finally, the best-fit line BF is computed from equation (2.16) and the corrected INL estimation $\widehat{INL_{BF}}$ is evaluated as

$$\left\{ \begin{array}{c} BF(i) = \varepsilon_G \cdot i + V_{os} \\ \widehat{INL_{BF}}(i) = \widehat{INL}(i) - BF(i) \end{array} \right\}, \quad i = 1,2,...,2^N - 1. \tag{2.17}$$

2.2. ADC testing techniques

2.2.1. Standard static linearity test techniques

This subsection first deals with the classical standardized testing techniques widely used in the industry. The two mainly employed techniques are the histogram-based linearity test and the servo-loop-based test. The histogram-based test is an open-loop technique, while the servo-loop-based test is a closed-loop one.

2.2.1.1. Histogram-based testing

In the case of the histogram-based test technique, the ADC is excited by a very linear, quasi-static stimulus at its input. It can either be a ramp stimulus or a sinewave one.

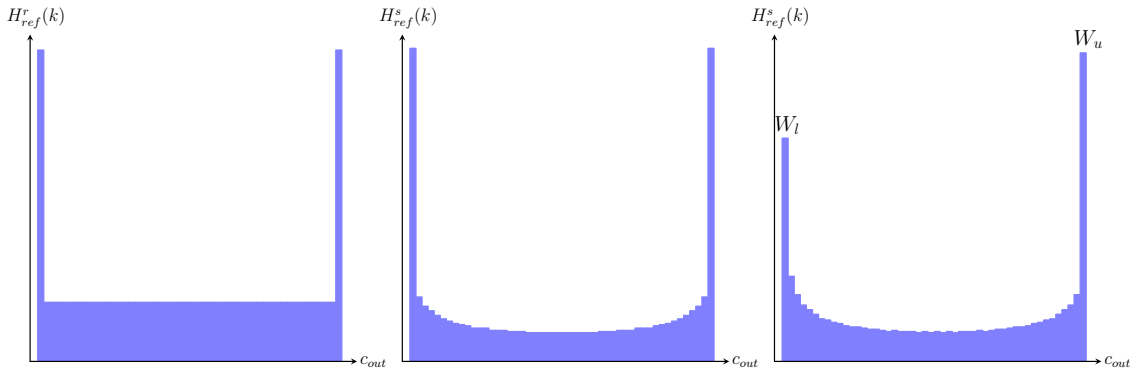
The input is slow enough compared to the ADCUT sampling rate, so a very high amount of samples is collected for each code. Those samples are processed into a histogram where the number of occurrences for each code (hits per code) is represented. If the ADC is ideal, and an ideal ramp stimulus is applied to its input, the same amount of occurrences is expected to appear for each code, except for the first and last code because of the need to overdrive the ADC input. For an actual ADC, the nonlinear behavior is translated to variations of these occurrences. For a given code, the higher the number of hits per code, the higher is the width of the code and its associated DNL.

These differences are revealed by subtracting the actual ADC histogram to the ideal ADC histogram. The DNL is then computed for the whole transfer function of the ADC. Note however that in a practical implementation, the input stimulus might not be perfectly linear and might degrade the DNL estimation as the histogram now contains the nonlinearities of both the ADC and the stimulus. As a standard practice in the industry, the stimulus is chosen with a linearity 2 to 3 bits higher than the ADCUT in order to provide a significant measurement.

For a ramp stimulus, if H_k^r is the number of hits per code for code c_k, the ideal number of hits per code H_{ref}^r can be expressed as

$$H_{ref}^r(k) = n_T \cdot \frac{FSR}{2^N \cdot A_r}, \quad k = 1,2,...2^N - 2 \tag{2.18}$$

where n_T is the total number of samples collected on the ADC FSR and A_r the overdriving amplitude of the ideal ramp. From equation (2.18), the width of code c_k can be computed



(a) Ideal ramp stimulus (b) Ideal sinewave stimulus (c) Sinewave stimulus with gain error and offset voltage

Figure 2.6: Histograms of input stimuli at ideal ADC output

as

$$W_k = \frac{H^r_k}{H^r_{ref}}, \quad k = 1,2,...2^N - 2. \tag{2.19}$$

The DNL and INL of code c_k are then calculated as

$$DNL_k = W_k - 1\ LSB, \quad k = 1,2,...2^N - 2 \tag{2.20}$$

and

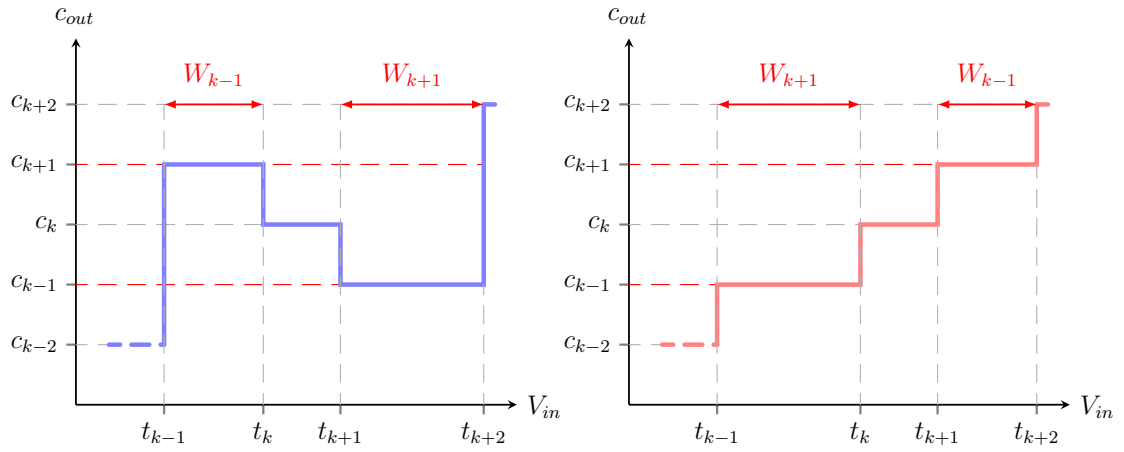
$$INL_k = \sum_{k=1}^{k} DNL_k, \quad k = 1,2,...2^N - 2. \tag{2.21}$$

Histogram-based testing can be also performed using a sinewave stimulus. However, the distribution of the sinewave stimulus voltages is not uniform. Assuming they are acquired at a constant sampling rate, it implies gathering more samples at the extremes of the sinewave because of the higher sample density at its peaks. Then data collection has to be extended in order to have a sufficient number of hits at the center of the sinewave. Knowing the sinusoidal distribution, a compensation is carried out involving trigonometric calculations in order to compute DNL and INL.

The corrected reference histogram H^s_{ref} is computed as

$$H^s_{ref}(k) = \frac{n_T}{\pi} \cdot \left(\arcsin\left[\frac{k - (2^{N-1} - 1) - V^s_{os}}{A_s}\right] - \arcsin\left[\frac{k - 2^{N-1} - V^s_{os}}{A_s}\right] \right), \quad k = 1,2,...2^N - 2 \tag{2.22}$$

where V^s_{os} and A_s are respectively the offset and the gain of the sinusoidal stimulus. Considering W_l and W_u being respectively the occurrences of the first and last ADC codes



(a) Real characteristic (b) Estimated characteristic

Figure 2.7: Errors due to occurence probability of histogram-based test

which are correlated to the lowest and highest peaks of the sinewave stimulus, we can calculate the offset V_{os}^s and the amplitude A_s of the sinusoidal stimulus as

$$V_{os}^s = (2^{N-1}-1)\cdot\left(\frac{\cos(\pi\cdot W_l/n_T)-\cos(\pi\cdot W_u/n_T)}{\cos(\pi\cdot W_l/n_T)+\cos(\pi\cdot W_u/n_T)}\right) \tag{2.23}$$

and

$$A_s = \frac{(2^{N-1}-1)-V_{os}^s}{\cos(\pi\cdot W_u/n_T)}. \tag{2.24}$$

The same reasoning as in the ramp stimulus case is applied for the DNL and INL calculation using equations (2.19), (2.20) and (2.21).

Notice however that this testing method has a major flaw. If the ADC has missing codes, its non-monotonic behavior might not be accurately measured. As an example, in Figure 2.7 (b), due to the accumulative function of the histogram-based method, the occurrences of missing codes c_{k-1} are incorrectly attributed to the occurrences of code c_{k+1} and vice-versa. This leads to two problems: first, codes c_{k-1} and c_{k+1} are not detected as missing, but are given an incorrect amount of occurrences, and secondly, the following transitions (t_k and t_{k+1} in the example) are estimated with offset errors.

2.2.1.2. Servo-loop-based testing

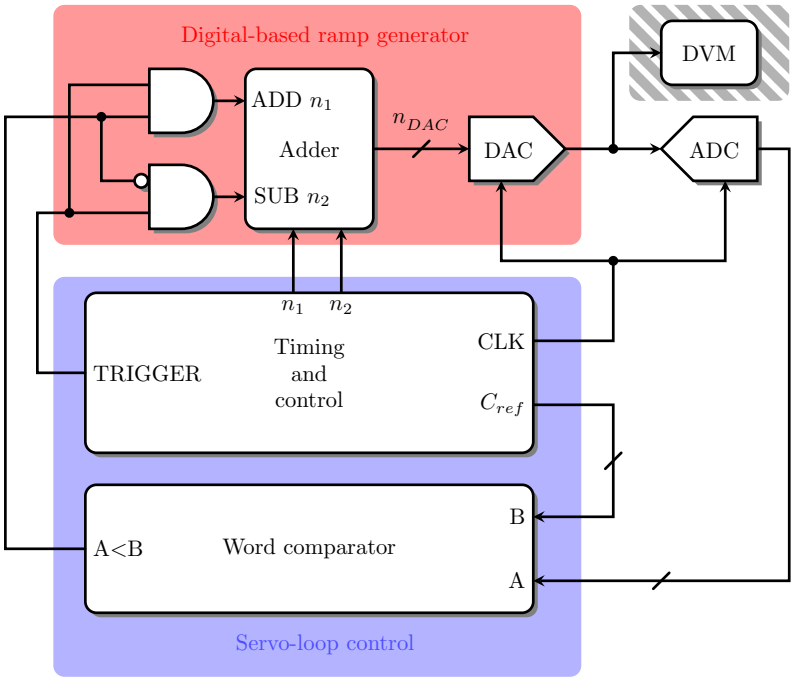
The servo-loop-based techniques are closed-loop approaches. The principle is shown in Figure 2.8. An input is applied to the ADC under test and the ADC outputs are compared to a reference code $C_{ref}(k)$, which specifies the code transition level t_k that needs to be determined.

By setting a given target code transition t_k, if the ADC output code is below the targeted reference code $C_{ref}(k)$, the ADC input is raised until the corresponding transition t_k is met. If the output code is above or equal to the reference code, the ADC input is lowered until the target transition is crossed. Once the code transition level t_k has been reached, the input signal is forced to oscillate across this transition. The measurement of the transition voltage is now a triangle-wave signal oscillating around the real transition value. The signal is filtered in order to eliminate the oscillations and obtain a fixed estimated value, which can be compared later to the ideal voltage value in order to estimate the INL of the measured code $C(k)$.

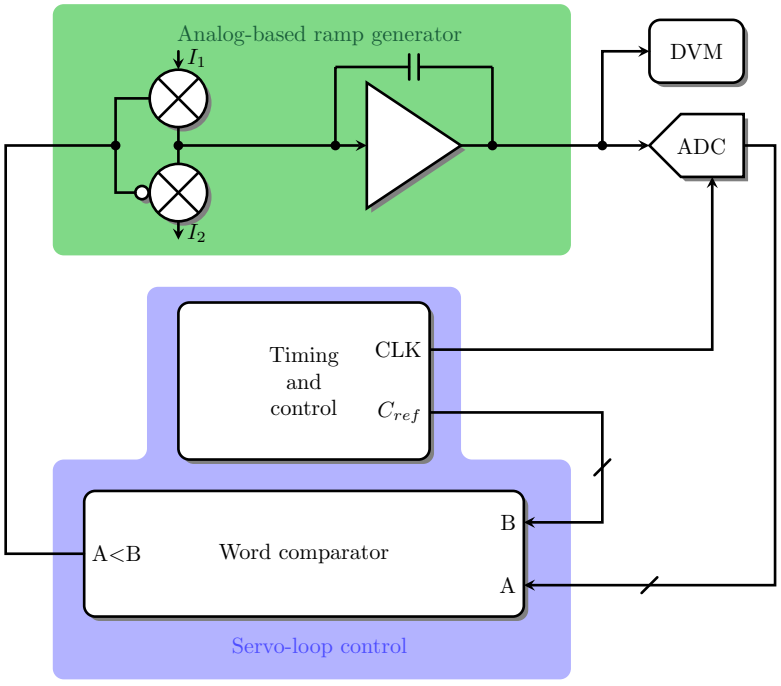
A schematic of the digital-based approach is given in Figure 2.8 (a) where an n_{DAC}-bit DAC generates the input stimulus. In this test, n_1 and n_2 are initially equal and given the value n_0. The DAC input value is set by adding or subtracting n_0 after each conversion cycle according to the result of the comparison between the ADC output code and the targeted reference code $C_{ref}(k)$. This means that transition level t_k is known only to an accuracy of n_0. However, the DAC output step size can be as small as desired down to the DAC resolution (n_0 equal to one DAC input code), so that iterating the test with smaller values of n_0 enhances the measurement accuracy of this transition voltage. The ADC input level is calculated from the known transfer function of the DAC, or is measured by an optional high-precision voltmeter.

A schematic of the analog-based approach is given in Figure 2.8 (b). In this case, the input stimulus is generated by a current-driven integrator, delivering a continuous ramp stimulus. In the same manner, the charge pump composed of current sources I_1 and I_2 delivers a current into the capacitor which integrates it into a voltage ramp. If the ADC output code is below the targeted reference code $C_{ref}(k)$, the current source I_1 injects current into the integrating capacitor, raising the integrator output with a positive slope ramp. If the ADC output code is above or equal to the reference code, the current source I_2 pumps out current from the integrating capacitor and the integrator output is lowered with a negative slope ramp. With a fixed capacitor size, the positive and the negative slopes are determined by the magnitude of the currents sources I_1 and I_2.

In any case, the resolution of the generator, analog or digitally driven, should be higher than the resolution of the ADC under test in order to obtain a desired precision for the measurements. Moreover, due to noise, averaging is necessary and the magnitude of the voltage increment should be chosen accordingly. The interested reader can find more information in [3], [4], [5] and [6] and this topic is further discussed in chapter 3.



(a) Digital approach (optional voltmeter)



(b) Analog approach (required voltmeter)

Figure 2.8: Two implementations of the servo-loop test [2]

2.2.2. State-of-the-art static linearity test techniques

As the complexity and resolution of the ADCUT rise, more effort must be put into producing a test stimulus with greater quality. Conventional testing techniques will eventually

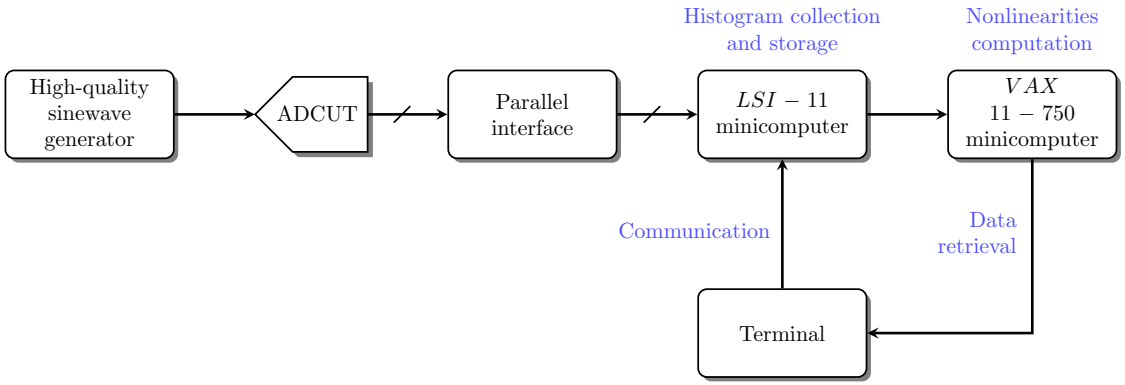


Figure 2.9: Experimental setup for testing an ADC with code density test or FFT test [7]

comply with the requirements, but at a very high expense in terms of test equipment and ever-increasing test time, as stated in section 1.1. To that purpose, alternative test techniques have been investigated over the last years, dampening the need for costly ATE while keeping a relatively high level of accuracy. In the first part of this subsection, we will talk about present advanced DfT strategies for ADC testing while in the second part we will focus on BIST solutions.

2.2.2.1. DfT techniques

DfT is defined as any technique implemented at design level in order to facilitate the test of a particular DUT. In this subsection, we present relevant DfT techniques for ADC static test.

The work in [7] proposes an open-loop computer-aided ADC characterization technique which relies on the code density test and spectral analysis using the Fast Fourier transform (FFT). The code density test produces a histogram of the ADCUT output which samples a known input sinewave stimulus. The DNL and INL can be extracted from the code density as well as gain error, offset error, and internal noise. Three different successive approximation (SAR) ADCS are tested: an 8-bit resistor-string ADC, a 12-bit R-2R ladder ADC, and a 15-bit self-calibrating ADC. For the first ADC, a precision of 0.1 LSB on the INL estimation error is met with 268000 samples. The paper compares the proposed method to the classical servo-loop test method, showing an improvement in the test time by a factor 10, but still very slow. FFT tests are also performed on the 12-bit ADC to measure its INL, distortion, and signal-to-noise ratio. The maximum INL is directly estimated by measuring the spurious free dynamic range (SFDR) out of the FFT curve. The experimental test setup is shown in Figure 2.9. The technique is effective but involves several minutes of test time and complex computations for FFT operation.

The works in [8], [9], and [10] propose to use a noisy signal as input for ADC linearity test. The method in [8] presents a statistical method for characterizing ADCs using Gaussian noise as the test stimulus for the histogram test. Nonetheless, the measurement of maximum INL is prone to large errors and a high number of samples is needed for the test procedure to work. Furthermore, a high-speed DAC is required in order to characterize the ADC.

The work in [9] employs the classical histogram method to evaluate the ADC nonlinearity with 2^{22} collected samples. The noise-based approach is compared to the standard sinewave-based approach. Results show that the two INL curves have the same trend, but the INL estimation error is about 4 LSB at 12 bits. Although the methodology allows an automated and extensive characterization of the ADCUT static parameters, it still requires a high-linearity DAC and the computations are very complex, thus not suited for a BIST implementation.

The work in [10] is based on the use of a simple set of DC voltages on which a white Gaussian noise signal is applied. The test method relies on a set of repeated measurements. Although they are not needed to be equally spaced, the set of DC input signals must be known and are dithered by a small amount of additive white Gaussian noise. The superimposed noise is composed of the input noise of the ADCUT but also of a noise signal generated from a dedicated noise generator. The transition levels of the ADC are then estimated from a statistical analysis. Simulation and experimental results demonstrate the robustness of the theoretical approach. The test time is a few minutes long. For comparison, the classical histogram test is performed with the acquisition of 800000 points of a sinusoidal waveform. The difference between the two resulting INL curves is small and shows the effectiveness of the method. However, this method is applied on a slow, low-resolution oscilloscope. The authors do not discuss whether this method can be applied to high-speed, high-resolution integrated ADCs.

The techniques proposed in [11], [12] and [13] explore the use of a sinewave input stimulus for ADC linearity test. Results concerning the extraction of the ADCUT DNL and INL using a sinewave histogram-based technique are proposed in [11]. Intensive calculations are performed as functions of the noise level, the desired test accuracy and desired confidence level in order to determine the required amount of samples as well as the required stimulus magnitude to overdrive the ADC. An analysis of the error induced by the stimulus harmonic distortion on the results accuracy is carried out as well as simulations on a 6-bit ADC model. It is shown that the number of samples required to obtain any desired tolerance and confidence level using a combination of coherent and random sampling is smaller than that required with pure random sampling.

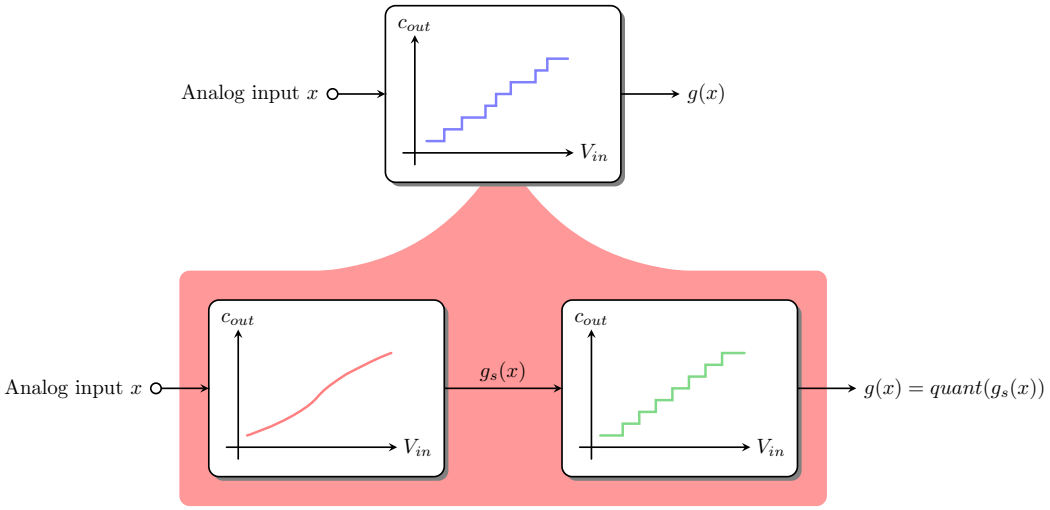


Figure 2.10: Decomposition of the static ADC tranfer function $g(x)$ in the cascade of a smooth nonlinear function $g_s(x)$ and an ideal quantizer $quant(x)$ [12]

The authors in [12] proposed a novel test technique consisting in an approximation of the ADC transfer function by a linear combination of Chebyshev polynomials, its coefficients being the spurious harmonics of the ADC output spectrum in response to as sinewave input stimulus. The concept of the technique is depicted in Figure 2.10. The method reduces the test time significantly and works for any ADC resolution, which appears to be very interesting for the test of high-resolution ADCs. It is sufficient and useful for many practical applications, such as the evaluation of harmonic distortion effects on multi-tone input signals. However it is inaccurate for the measurement of the maximum and minimum INL as it only yields an indirect information being the harmonic distortion generated by the nonlinearity.

On a similar approach, the methodology in [13] is based on spectral analysis. The proposed test flow, based on a statistical analysis, allows evaluating the static linearity parameters of an ADCUT based on dynamic measurements. Complementary spectral tests (such as stimulus magnitude modulation, DC and fundamental components evaluation) are performed after the classical FFT analysis, as shown in Figure 2.11. The first test separates the fault-free ADCs from the faulty ones based only on dynamic measurements, then the second test distinguishes fault-free and faulty devices with respect to static specifications. The test flow is first applied to a population of computer-generated ADCs and then to on-the-shelf ADCs with real specifications. Despite of the method being highly specific to the performances of a given type of ADC, results show a very high test efficiency for both generated and real data. Moreover, the absence of static test and dynamic test analysis greatly reduce the test time. Intensive calculations are still required for the

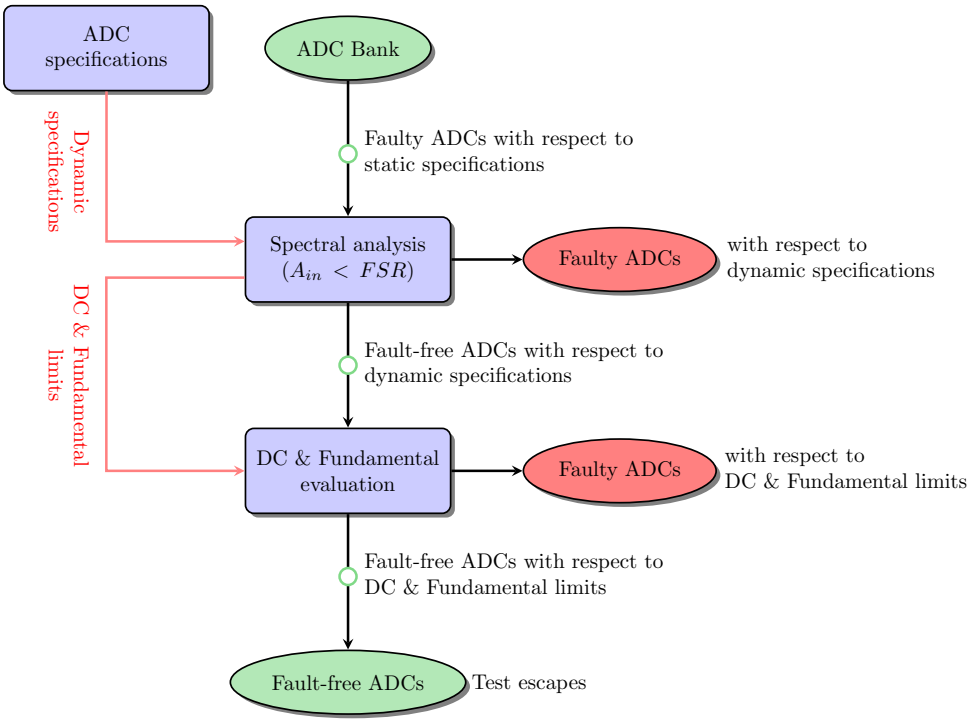


Figure 2.11: (FFT + DC + Fundamental) test procedure [13]

spectral analysis and parameters extraction, which limit the interest of the technique as a BIST solution.

The works in [14] and [15] present a histogram-based technique using small-amplitude triangular waves. A small offset-stepping ramp is added to the original triangle-wave pattern. The difference between two offsets is small enough in order to be assured that the amplitude ranges of adjacent triangular waves are crossed and the whole ADC transfer function is covered. Linearity constraints on the triangular generator are relaxed by using a signal of amplitude much lower than the ADC FSR. The diagram flow of the procedure appears in Figure 2.12. At the same sampling frequency and input stimulus frequency as it would be in the standard histogram procedure, each generated triangular signal is sampled with the same amount of samples and stored in a cumulative histogram. Then calculations are performed in order to gather the data and estimate the ADC static parameters. Closed-form expressions for designing the test as well as for its characterization in terms of efficiency and uncertainty are provided. However, this approach is more complex than a simple classical histogram code hit counting as more calculations are involved. Furthermore, the offset ramping must be accurate and no solutions are reported in the paper regarding the design of a generator for this operation.

The works in [16], [17], [18], and [19] propose a model-based approach for ADC linearity test. The work in [16] introduces a model-based testing technique in order to

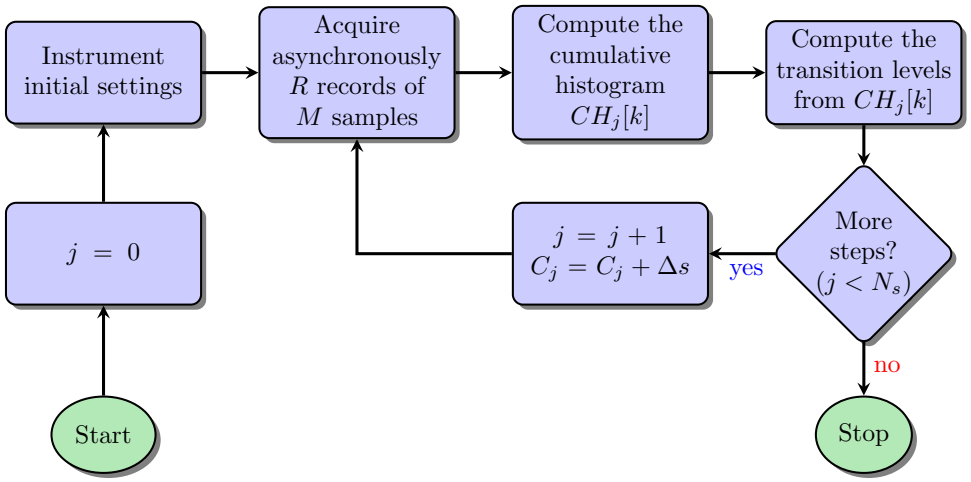


Figure 2.12: Small triangle-wave test procedure [15]

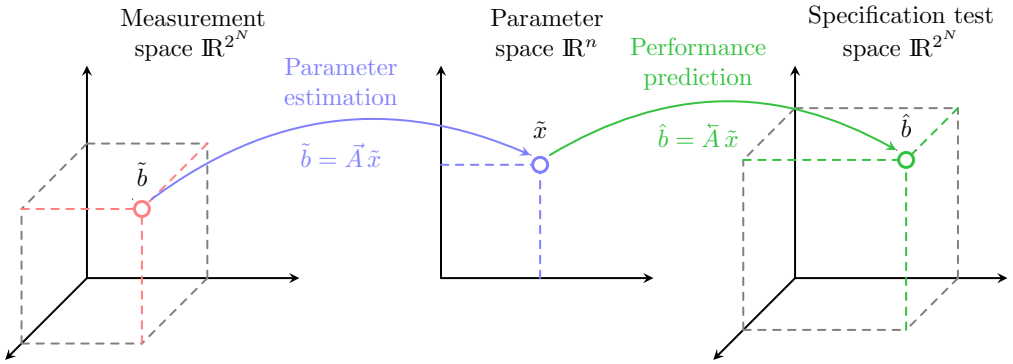


Figure 2.13: Linear model-based testing of an N-bit converter: from a measurement $\tilde{b}$, to the least-squares estimate $\tilde{x}$ of the model parameter vector, and to an approximation $\hat{b}$ of the unknown noise-free device characteristic [16]

estimate the ADC INL with a reduced set of samples and the linear modeling that predicts the performances of the ADCUT, as seen in Figure 2.13. The test uncertainty and the noise contribution in each measured sample are reduced by the means of the ADC linear model, leading to a reduction of the required number of samples. However, the computational aspect of this method (46 parameters for the ADC model) makes it a bad candidate for a BIST implementation.

The work in [17] proposes another model-based test technique. Here, the INL is modeled as a superposition of a low-frequency INL and a high-frequency INL. On one side, the low-frequency INL is evaluated by the means of a spectral analysis, revealing the smooth, global variations of total INL curve. On the other side, the high-frequency INL is identified by measuring the DNL of a set of specific codes with the histogram algorithm using a input triangular stimulus. Those codes are chosen as the ones with dominant DNL error, and selected not only according to the multi-periodicity model in the middle, quarters, and 1/8 of the transfer function, but also from the data of the low-frequency

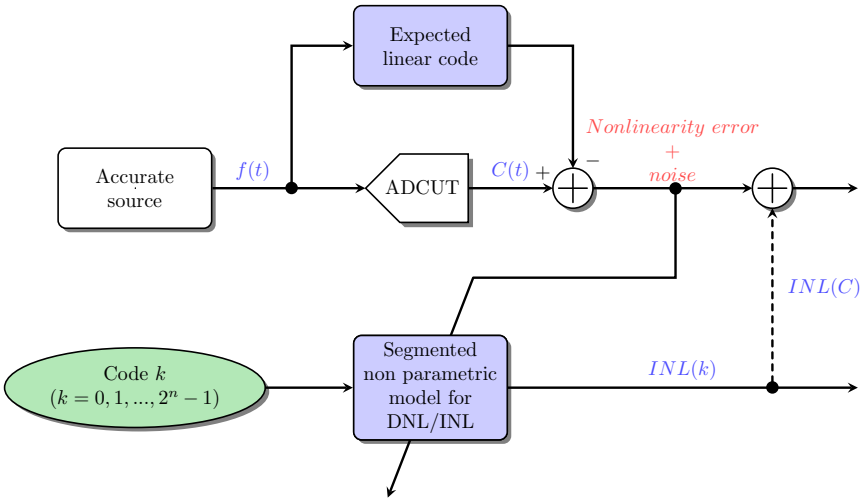


Figure 2.14: Block diagram of the algorithm for dramatically more efficient ADC linearity test [19]

INL estimation. The requirement on the distortion of the triangular voltage is reduced as only small portions of the ADCUT transfer function are measured and implies that the variations with respect to a linear ramp stimulus are very small. Results show that the approximative estimated INL curve match closely with the real measured INL trend, even with a small number of collected samples. However, the calculations for both INL components are very complex to handle, and a significant error in the INL estimation remains.

A model-based technique for pipeline ADCs testing is detailed in [18]. The expression of the input voltage of each pipeline stage is modeled by a Taylor series expression and is represented as a combination of the input voltages of the further stages, weighted by the inverse of their respective interstage gains. The expression includes nonlinear errors such as gain errors and offset voltages of each stage, which are the parameters to be identified by the method. For the identification of the parameters, a collection of M data samples is performed. On a nonlinear statistical approach, starting from initial parameters values, the M error terms are evaluated using an iterative optimization algorithm and the optimal parameters are solved with the Newton-Ralphson method, minimizing the errors in the estimated values. Finally, the computed ADC model is used to compute the transition voltages and estimate the static parameters of the device.

The authors in [19] propose another model-based technique for ADC linearity test which relies on the modeling of the ADC INL with a segmented, non-parametric model, as depicted in Figure 2.14. This model cuts the INL in several segments of constant value being an average value of the real INL: the MSB segments where the MSB code do not change, the ISB segments where the code of intermediate bits do not change, and

finally the remaining LSB segments. The ADCUT must be sufficiently linear in order to begin the test. This is verified by performing a FFT of the ADC response to a very linear sinewave stimulus and checking the total power of the error signal after subtracting the expected linear code from the actual ADC output code. If the error power is reasonably low, then the test can continue with the INL estimation. However, no theoretical demonstration on the validity of the algorithm has been carried out. Moreover, the purity of the input sinewave signal must be very high and the algorithm requires complex calculations to obtain the estimated INL and DNL, which bounds this method to ATE-based test.

The authors in [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], and [31] propose different techniques for relaxing the linearity requirements on the test stimulus for ADC linearity test.

The work in [20], [21], and [22] introduce the so-called Stimulus Identification and Error Removal technique (SEIR). As the ADC resolution goes higher it becomes more and more difficult to generate an input stimulus that complies with the required resolution (2 to 3 bits more than the ADCUT resolution). The SEIR method allows the use of input stimulus that are much less linear than the ADCUT. The proposed test method is shown in Figure 2.15. First, a low resolution stimulus is applied to the ADCUT. Then, the histogram $C_k^{(1)}$ collects the ADC output data. A second low-resolution stimulus is applied to the ADCUT, identical to the first one, but adding an offset voltage V_α. The ADC output data is stored in a second histogram $C_k^{(2)}$. The SEIR algorithm then uses the redundant information from $C_k^{(1)}$ and $C_k^{(2)}$ in order to identify the nonlinearity of the original input stimulus. Finally, the identified errors from the stimulus are removed from the ADC output data, allowing to precisely compute the ADC DNL and INL. The approach was experimentally validated in production test on an industrial 16-bit SAR ADC by using 7-bit linear input signals. Nevertheless, the test time attributed to SEIR algorithm can be insignificant only if the resolution of the ADCUT is sufficiently high. Furthermore, the required offset accuracy is very demanding, as the voltage drift must not be larger than 0.1 LSB in order to obtain a good accuracy in the results. Additionally, from the point of view of an on-chip implementation of the technique, the fact that it relies on least-square error estimations may lead to complex implementations.

The technique in [23] capitalizes on the idea proposed in [21], [22] for testing an ADC with the SEIR method. Instead of using a least-squares error algorithm, another hardware-optimized calculation algorithm for the SEIR method is developed in order to resolve the input stimulus nonlinearity. It involves direct calculations instead of an optimization method, proving to be simpler and more efficient without loss of accuracy in the measurements. From the mean and difference of the histograms obtained in response to the

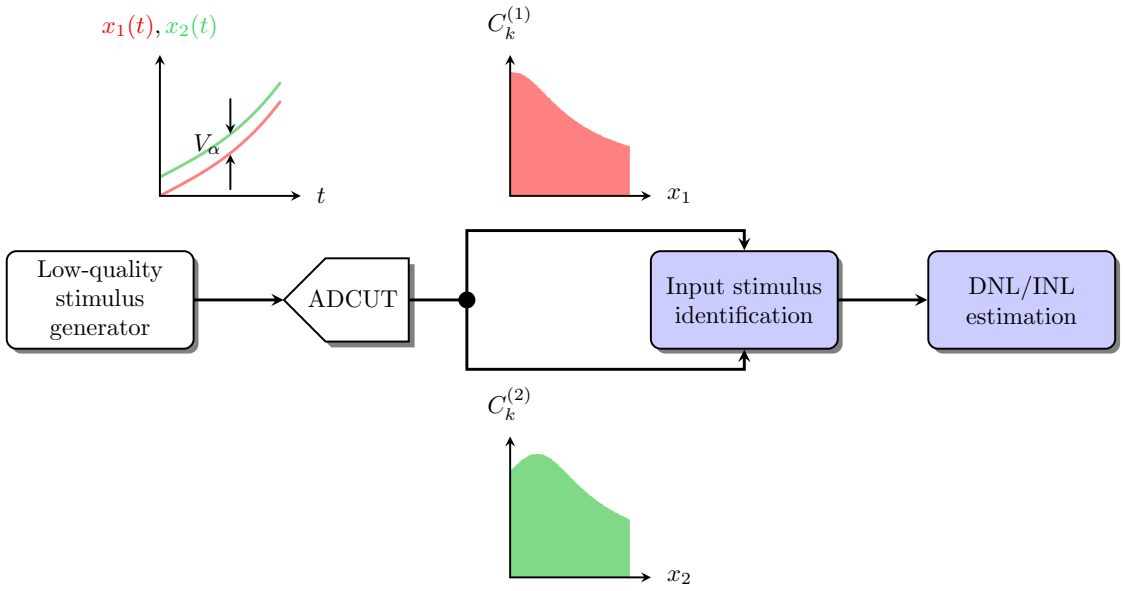


Figure 2.15: ADC test method using SEIR algorithm [21]

two related input stimuli, the numerical differentiation calculations (using forward- and backward-Euler method) can evaluate the offset voltage between both stimuli and then provide an estimation of the code width for each code. The simulation validation is run on 14-bit ADC models and the INL measurement accuracy is proven to be about 1 LSB, but the area requirements are about the same as for the standard histogram method.

In [24], the technique is enhanced on different points. For example, a more sophisticated numerical differentiation method (central-difference method) is exploited to correctly detect the higher derivatives, and the number of stimuli are doubled, in order to make the method even more accurate when the test stimulus has pronounced nonlinearities.

In [25], the technique is described for ADC testing using a low-resolution sinewave test stimulus, but this time, as a preprocessing part, the method is combined with the classical histogram method in order to remove most of the higher derivatives that introduce errors in the stimulus identification algorithm. This allows to estimate the ADC INL with only two related low-resolution sinewave test stimuli(in comparison to the four-stimuli approach in [24]). The authors propose a way to combine the two methods in order to attain a better accuracy in the INL estimation. Simulations involving a 16-bit ADC model show that, when both methods are combined, the maximum INL estimation error is about 1 LSB in presence of noise. Moreover, it is shown that the INL estimation error is independent of the stimuli offset value. A custom circuit on PCB is used to generate the required offset, as seen in Figure 2.16. Results show an maximum INL estimation error of about 0.5 LSB when both methods are combined.

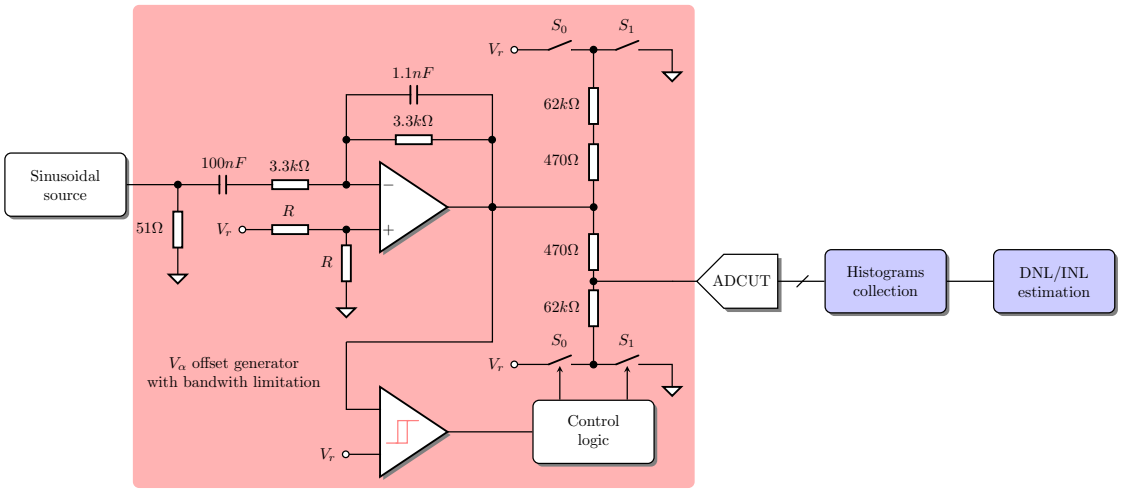


Figure 2.16: SEIR method with signal conditioning and resistive adder for the V_α offset generation [25]

The work in [26] and [27] details the modified SEIR method developed in [23], [24], now targeting a closed-loop ADC testing technique. It uses a loop-back configuration involving a DAC/ADC pair to be tested. In [26], the DNL and INL of the DAC are computed using an inversed histogram formula combined with the stimulus identification algorithm. Simulation results show that on 500 randomly generated 12-bit DAC/ADC pairs, the estimation error of the DNL and INL for both converters is less than 0.4 LSB in 99.2% of the test cases. The experimental results confirm the results from the simulation data. Figure 2.17 shows the proposed test setup. The minimum test time is about one second for a 1 MHz sampling frequency with 256 hits per code.

Finally, [27] proposes an enhanced version of the technique for the static linearity testing of DAC/ADC pairs, where the issue of memory space optimization is addressed. The direct implementation of the developed algorithm requires that the whole histograms are kept in memory until the algorithm can be executed. In order to reduce the memory requirements, the test routine is divided into small segments and the algorithm (modified accordingly) is performed for each one by one. This drastically reduces the required space to keep the histograms data. Furthermore, the algorithm speed is optimized by using fixed-point arithmetics instead of floating-point arithmetics. Simulation results on 100 randomly generated 12-bit DAC/ADC pairs show that the accuracy of the DNL and INL is not significantly changed. In the experimental case, the same test setup is used as in [25] with a pair of commercial 12-bit DAC and ADC. The use of fixed-point arithmetics does not significantly decrease the accuracy of INL estimation.

The same kind of segmentation-based approach is applied in [28]. The technique is based on approximating a non-linear ramp as a combination of linear segments. Measure-

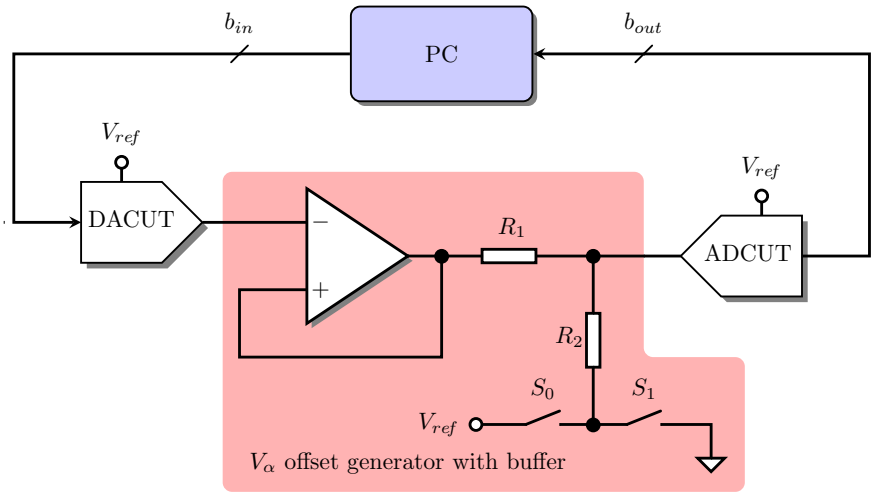


Figure 2.17: Test setup for loop-back DAC/ADC SEIR method and V_α offset generation [26]

ments on an 8-bit ADC as well as a 10-bit ADC have been performed in order to validate the proposed method. Consequently, the requirement on the linearity of the input ramp stimulus can be relaxed, but the technique has not been investigated for high-resolution ADCs.

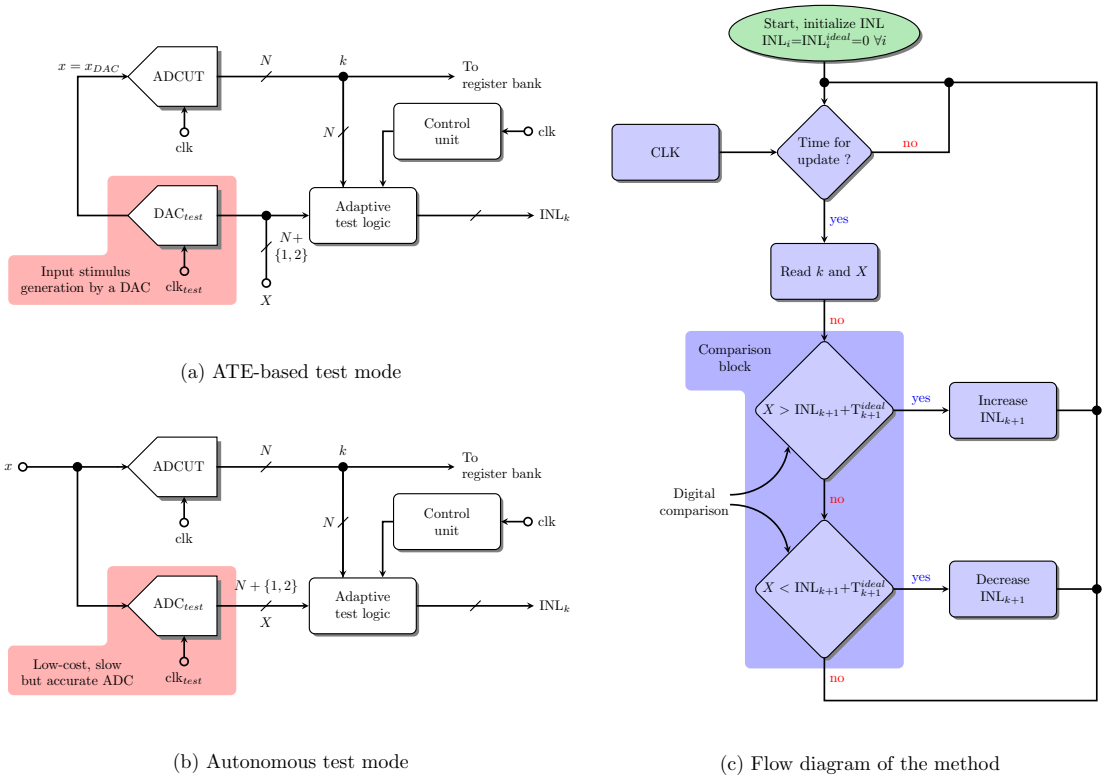
The works in [29], [30], [31], and [32] propose a double-histogram test technique aimed at relaxing the linearity of the test stimulus for ADC linearity test. In [29] and [30], the authors propose a simple and general algorithm to estimate the non-linearity of high-resolution ADCs using the double-histogram method. In this technique, the nonlinearity of the source is considered as an element of the ADCUT. Two sets of the effective ADC transition levels are then estimated using the offsetting technique, and each code width is computed using a very simple relationship without the use of specific basis function or least-squares method. The algorithm works with any type of input waveform, without the necessity of correction terms due to the inclusion of higher derivatives, and also works if the offset injection block is non-stationary in time.

The works in [31] and [32] deal with the hardware implementation of the double-histogram method for BIST applications. The authors propose a novel interfacing solution, based on an efficient resistive buffer design, which has been conveniently modified for the application of the double-histogram method. The high-linearity fully-differential buffer is placed between the stimulus generator and the ADCUT in order to prevent kick-back noise from the ADCUT. Summing branches are added to each input branch of the buffer for high-accuracy offset injection. The double-histogram method is then applied and the nonlinearities from the generator and the ADCUT are evaluated separately.

The method in [33] proposes a novel solution for INL measurements based on a blind

digital estimation of the ADCUT transition voltages. The proposed test strategy is based on comparing the output of the ADCUT with a high-resolution digital version of the ADCUT input signal. In the initial state, the INL of each ADCUT code is set to zero, meaning that the initial guess of ADCUT transfer function is the ideal case. Considering a sample $x[n]$ of the input stimulus x at discrete time n, the algorithm constantly verifies if the digital representation of this sample, $X[n]$, lies between the digital estimations of transition voltages $t_k[n]$ and $t_{k+1}[n]$, $T_k[n]$ and $T_{k+1}[n]$, which corresponds to the ADC output code $k[n]$. X is assumed to have a resolution much higher than the output code $k[n]$. At the pace of the adaptive step $\mu = 2^{-m}$ (m being an integer), the digital estimations get updated towards the real transition values for this particular code. The process is repeated on the whole transfer function, and the INL is directly computed. Two modes of operation are proposed and appear in Figures 2.18 (a) and 2.18 (b). In the first mode, called ATE-based mode, a high-resolution DAC is fed with the digital representation X of the input stimulus x, and needs a multiplexer to switch the ADCUT input from its primary path to the DAC output. In the second mode, called autonomous mode, the estimation of the digital representation X of input stimulus x is provided by a second, slower but much more accurate ADC. Simulations are performed on two 13-bit ADC models with respectively soft and hard errors using the autonomous-mode approach. The extra ADC is a low-speed algorithmic 15-bit ADC. Simulation results show a good INL estimation in both cases in comparison to the histogram-based INL references. Then, the method is experimentally validated on an custom 11-bit pipeline ADC and a commercial 14-bit pipeline ADC. A good estimation of the INL is noted in comparison to the histogram method.

The work in [34], [35], [36], [37], [38], [39], [40], [41], and [42] propose a reduced code testing algorithm for specific ADC linearity test. The work in [34] proposes a reduced code linearity test strategy for SAR ADCs. Taking into account the repetitive functioning of SAR ADCs, it relies on measuring a reduced set of codes to provide an estimation of the static characteristic of the ADC. The conversion algorithm of a SAR ADC carries out a series of successive approximations of the input with respect to a set of reference voltages. Those references are generally generated by an switched-capacitor-based DAC. When a comparison is performed, each of the activated switched capacitors that generate the reference voltage contribute to the ADC DNL due to their respective deviation with respect to their nominal value. It means that when a unique bit is flipped to '1' and the others are flipped to '0', the DNL associated to the related ADC output code can be measured and applied to all the codes where this particular bit was activated. In others terms, it means that for an N-bit SAR ADC, the simple measurement of the width of N selected codes is enough to generate its entire DNL and INL, as detailed in Figure 2.19. To this end, a piecewise linear ramp input signal, being a N-level input stimulus, is generated in



(a) ATE-based test mode

(b) Autonomous test mode

(c) Flow diagram of the method

Figure 2.18: Blind-adaptive INL estimation method [33]

order to target and measure the selected codes of the ADCUT. The method is validated on a 12-bit SAR ADC on both simulation and experimental fields. The reduction in the numbers of samples for the histogram test allows a test time reduction of 76%.

The works in [35], [36], [37], [38], [39], [40], and [41] propose reduced code linearity test techniques for the linearity test of pipelines ADCS. Similarly to SAR ADCs, pipeline ADCs are also based on a repetitive conversion algorithm.

The work in [35] proposes to map the ADC final output transitions to the corresponding comparator transitions of each stage by simply locating the transitions of these comparators on the ADC transfer function. The validity of this mapping method only holds if the aforementioned comparators do not have any offset or gain error. This means that the related transitions are ideal. However, in a real implementation, this is never the case as each comparator bears an offset due to process variations, implying that each transition is shifted from its ideal value by the related offset or gain error amount. Revised versions of this method are found in [36] and [37], where the mapping technique is not affected by the presence of comparator offsets. In order to map the located ADC output transitions to the corresponding comparators, it is suggested to consider the expected, ideal distribution of comparators transitions. Based on this assumption, a fully digital circuit is inserted before the ADC correction block, that forces the LSB of the target stage to '0', revealing

DAC input code	MSB	Bit #3	Bit #2	LSB
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Capacitor #1 active

Capacitor #2 active

Capacitor #3 active

Capacitor #4 active

Figure 2.19: Example of code selection for a 4-bit SAR ADC reduced code linearity test (only one transition per bit to select for complete ADC test) [34]

jumps in the transfer function when a monotonic input stimulus is applied and allowing to generate the mapping of the stage transitions. The method is validated on a 12-bit pipeline ADC in simulations as well as in [37], where a BIST circuit is implemented in a 12-bit pipeline ADC. The number of required samples is about 7% of the total number required for a classical histogram test, which induces a great test time speed-up. Nevertheless, this is only valid if the comparators of each stages are exercised in the expected order, which is rarely the case. Gain errors and noise are more obstacles to this assumption.

The works in [38]−[41] propose an enhanced reduced code linearity technique to overcome the limitations of the previous works in this particular field. The method implies the monitoring of each stage digital output when a monotonic input stimulus is fed to the pipeline ADC. The related algorithm is insensitive to noise and allows a precise mapping of the ADC output transitions. From this mapping, a selected set of codes are measured and the whole transfer function of the ADC can be reconstructed. The method is also validated on simulated and real pipeline ADC, where the internal sub-ADCs outputs are available to the tester. The test employs about 6% of the total number of codes, which implies a great reduction in test time.

Finally, the authors in [42] propose another approach for reduced code testing of pipeline ADCs. The nonlinearity of the ADCUT is tested with a reduced set of input amplitude levels. The method selects a particular set of input amplitude levels, or test points, which consists in applying simple DC stimuli at the selected test points and then capture the related output codes. The INL is then computed using the standard formula. Relying on the fact that the pipeline stages responsible for the MSB generation give the

highest INL discontinuities in the total INL curve, the method consists in only measuring the MSB-related ADC transitions. The second part of the method gathers the estimated transitions corresponding to each selected MSB stages into a set, sorted in an ascending order. From this set, the test points are then computed as the mid value between two consecutive transitions. At last, each of these test points are stimulated with a high-quality input signal, such as DC values with a small dither noise, or a small-amplitude triangle wave and the corresponding transition voltage is measured. The INL is then evaluated for all test points and the total INL is computed from the best-fit line that connects all those points or a terminal-based best-fit line. Simulations results show a maximum $\pm 3\sigma$ INL estimation error on all the test points of ± 0.5 LSB for an uncalibrated ADC, and ± 0.25 LSB for a calibrated ADC. The technique is useful to give a fast estimation of the trend of the INL curve, but lacks of accuracy for INL estimation at LSB level.

2.2.2.2. BIST techniques

A further step to reduce the complexity and cost of ADC static test is to develop a full BIST test strategy in which stimulus generation and response analysis are integrated together with the DUT. In this subsection, we present a comprehensive state-of-the-art on ADC BIST solutions.

The authors in [43] and [37] propose a fully digital BIST approach for ADC linearity test. The method in [43] is aimed at ADCs having a DAC in their structure, such as pipeline ADCs, and eliminate the necessity of calibration. In the case of a pipeline ADC, the method relies on an digital BIST circuit integrated in the first or last stage, and measures its DNL, INL, gain and offset errors. The BIST circuit mainly consists of several registers, a digital multiplexer, a digital counter and some digital subtracters. The method also involves a modification of the involved pipeline stage: the sub-DAC must be modified in order to have a higher resolution than the sub-ADC. The simulation and experimental results on a 3-bit flash ADC and 9-bit pipeline ADC validate the method with an accuracy error on the DNL estimation of about 10%, showing that the proposed BIST method is a compromise between test accuracy, area overhead and test cost.

In relation with the previous works listed in subsection 2.2.2.1, [37] proposes a fully digital BIST scheme with a silicon validation on a custom 12-bit pipeline ADC. An additional capacitor tuning mechanism is used to inject moderate capacitor mismatch in each stage, leading to tunable DNL errors in the stages. The number of required samples is about 9.3% of the total number required for a classical histogram test, which is close to the 7% obtained in the simulation results.

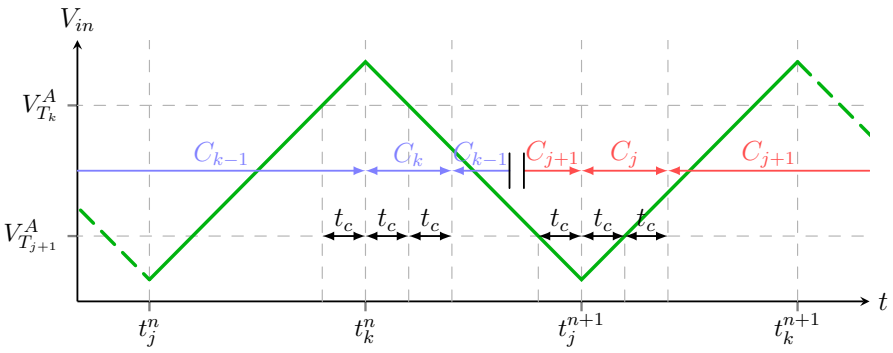


Figure 2.20: Oscillation-based ADC test method [44]

The works in [44] and [45] propose an oscillation-based BIST technique for ADC linearity test. The method in [44] relies on the servo-loop structure defined in subsection 2.2.1.2, but a different algorithm is implemented. After a digital starting pulse, the ADC continuously converts its input voltage. The ADCUT input is forced by the digital BIST control to oscillate between the actual transition voltages $V^A_{T_k-1}$ and $V^A_{T_k}$ of two selected, consecutive codes C_{k-1} and C_k. The proposed test strategy evaluates DNL and INL from the oscillation frequency, which is a function of the conversion time of the ADCUT, the slope of the ramp stimulus, and the difference between the two transition voltages. The slope ramp stimulus being fixed by design, the conversion time can be found by forcing the ADCUT input to oscillate between the two consecutive codes C_{k-1} and C_k, that is, around the actual transition voltage $V^A_{T_k}$ separating the two codes, as seen in Figure 2.20. The voltage difference can then be evaluated as the DNL of the code C_k between the two transitions. The INL is computed as the cumulative sum of the DNL, that means that the INL can also be evaluated directly by forcing the ADC input to oscillate between actual transition voltages $V^A_{T_1}$ and $V^A_{T_k+1}$. Then, the INL error can be evaluated by comparing the actual frequency from the oscillation between the two transition voltages to the nominal frequency value. The method is validated using simulations and experimental data, but no quantitative results on the accuracy of the approach are presented.

The approach in [45] is also based on an on-chip implementation of the servo-loop technique. Figure 2.21 shows the proposed BIST approach. The integrator, which is a simple capacitor charged by a current mirror, is accurately controlled by a digital circuit named integration time controller. Two possible modes are defined, being coarse and fine location modes. First, in coarse location mode, the integrator is driven in order to find the location of the target transition voltage. If the integrator output is higher than the target transition voltage, the digital BIST control forces the integrator to oscillate around this value, and the fine location mode is enabled. In this mode, the oscillation period is refined for a defined number of clock cycles until the required accuracy is met. From

this operation, the average integration time interval can be computed. Assuming a linear relationship between the integrator output and the integrating time, an estimation of the target transition can be done. Then, the same process can be applied to the upper transition voltage, and the voltage difference from both estimated values allows to evaluate the width of the related target code, and its DNL/INL. The proposed test strategy is validated by simulations using an 8-bit ADC as DUT. The BIST circuit occupies less than 10% of the total ADC chip area. Provided results show a worst case measurement error of DNL and INL of 0.0369 LSB.

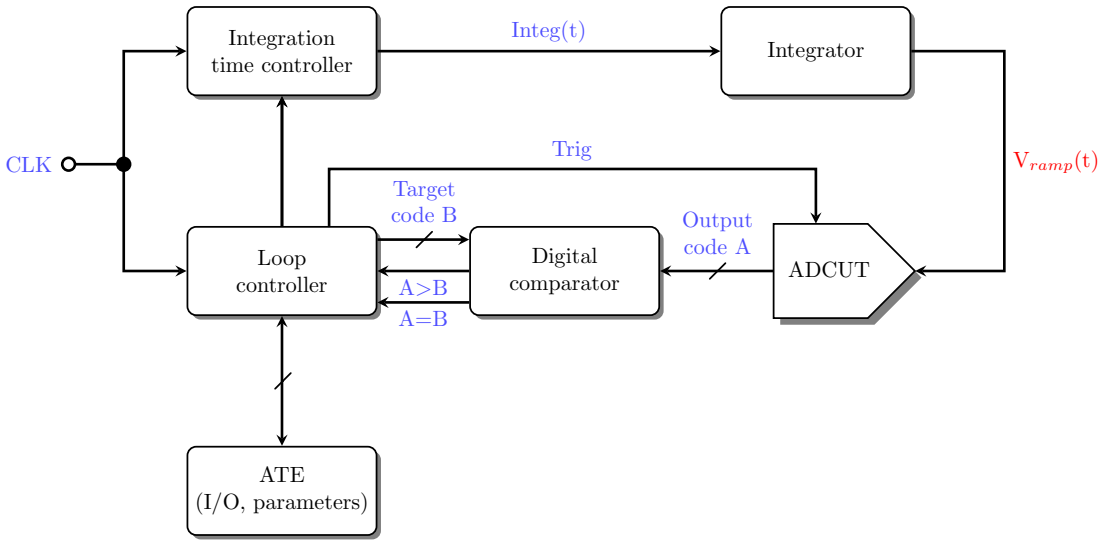


Figure 2.21: Embedded servo-loop ADC test method [45]

The works in [46], [47], and [48] propose a digital counter-based BIST technique for ADC linearity test. In [46], the proposed method consists in reducing the number of ADCUT output bits that need to be monitored externally. This partial monitoring is dependent on the frequency of the applied test signal. In the case of a low-frequency test signal, only the LSB needs to be monitored. The linearity of the ADCUT is evaluated for each code by counting the number of samples between two LSB transitions. A trade-off analysis is conducted between the size of the BIST circuitry and the accuracy of the technique. Figure 2.22 details the diagram flow of the proposed method. The simulations and experiments are performed on 6-bit flash ADCs. Results show that the probability of mistakenly reject a good device is very small, and the accuracy on the estimation of the LSB code width is high, and it is as well enhanced if the resolution of the counter is raised. However, clock jitter and noise are limitations that damper the precision of this technique. Moreover, ADC input noise would definitely reduce the accuracy of the counting algorithm, but the behavior of the method with respect to this matter is not discussed in the paper.

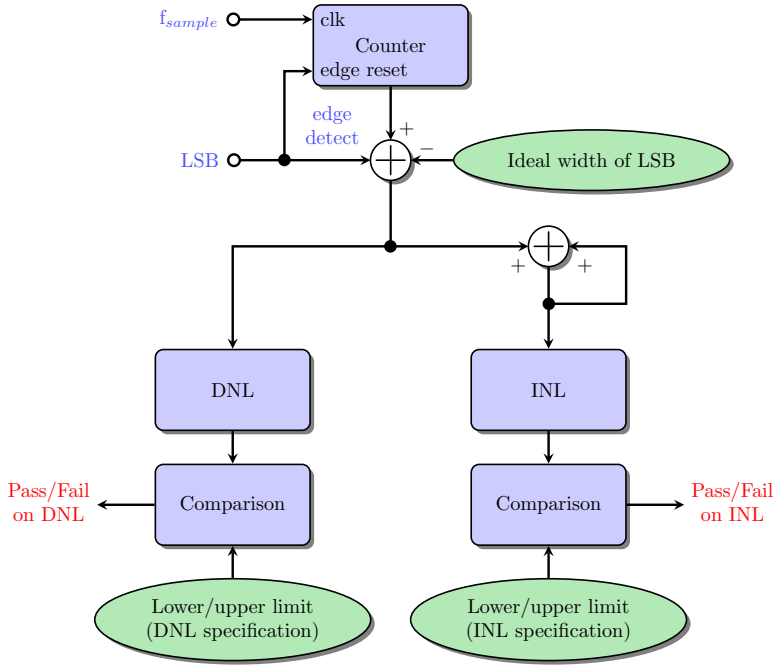
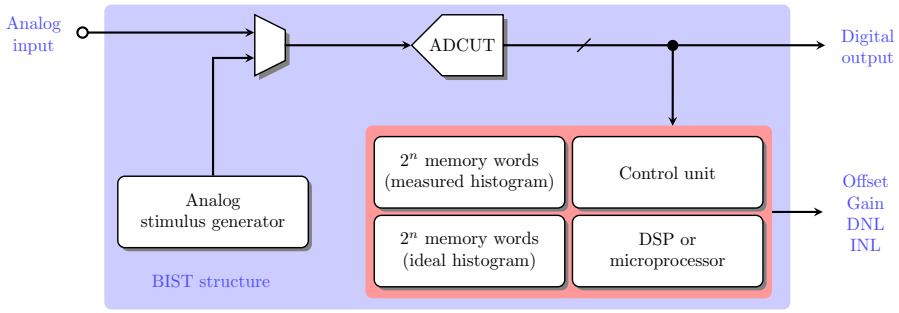


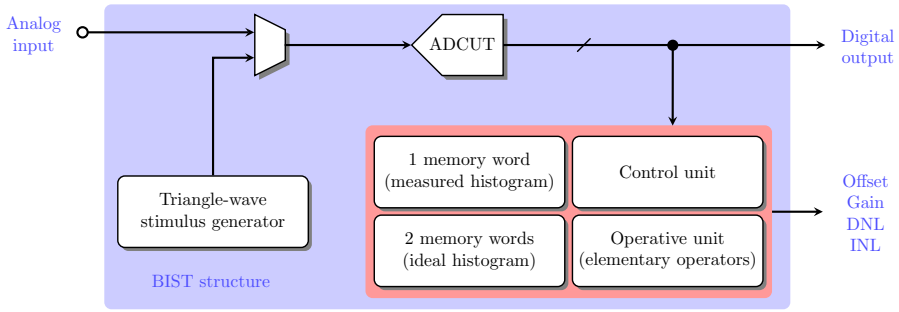
Figure 2.22: Digital counter-based ADC BIST method [46]

A similar idea is presented in [47] and [48]. The difference with the previous method stands in the counting process, because the n-bit ADC output is directly compared to a $(n+m)$-bit counter, where m is the accuracy factor of the proposed method. A high-linearity integrator, calibrated and synchronized with this counter, generates a ramp stimulus. The ramp signal is divided into 2^{n+m} segments, where each segment corresponds to one output combination of the $(n+m)$-bit counter. The static parameters are extracted from the comparison of the ADC output with the $(n+m)$-bit counter. Simulations are performed on a 12-bit ADC model with the use of a 15-bit counter, and results show an accuracy within 0.5% LSB. However, the main drawback of the technique lies in the linearity of the input ramp that limits the measurement accuracy. Furthermore, as in the previous works, no concerns towards the issue of noise have been addressed.

The works in [49], [50], [51], [52], and [53] propose a histogram-based BIST with reduced hardware requirements. In [49] and [50], it is stated that the main problem for a BIST implementation of the histogram method is due to the overdrive of the ramp signal that leads to higher number of occurrences in extreme codes, which increases the required memory resources. Figure 2.23 details the structure of the proposed resource minimization. The method presented here consists in representing the complete ideal histogram with only two values: one corresponding to the ideal count of any non-extreme code, and one corresponding to the ideal count of the extreme codes. This leads to great memory savings, as the memory for the storage of the ideal histogram is reduced from 2^n memory words down to 2 memory words. Moreover, the use of a ramp stimulus in comparison



(a) General BIST scheme



(b) Proposed BIST scheme

Figure 2.23: ADC BIST resource minimization [50]

to a sinewave stimulus helps a further reduction of the hardware resources. With respect to the computational aspect of the BIST structure, the calculation of the ADC static parameters are far easier to compute in the case of a ramp stimulus, in comparison to the sinewave stimulus case, where trigonometric arithmetics are involved. This means that a simpler computational unit can be designed, reducing again the BIST overhead. Finally, the storage of the measured histogram data is limited to a single memory word. Using a time decomposition technique, the algorithm is modified in order to process the data after each code measurement. Simulation results on 6-bit ADC show a DNL accuracy of 0.03 LSB, and a BIST overhead of 6.7% compared to the ADC area. Nevertheless, the reduction of hardware resources comes at the expense of a longer test time, which can be prohibitive for high-resolution ADCs testing.

The same approach is discussed in [51] using a sinewave signal as input stimulus. Taking advantage of the symmetrical property of the ideal histogram, only half of needs to be considered. A piece-wise approximation defining several domains in half the ideal histogram is performed. Simulations on a 8-bit ADC model allow to conclude on a 0.05% difference between the actual and estimated values of the ADC static parameters. The storage of the data is reduced in comparison to the histogram method, but more memory

is needed in order to store the data of the piece-wise histogram compared to the method in [50].

A novel histogram-based method using a sinewave stimulus with the CORDIC algorithm is proposed in [52]. The ADC static parameters are directly obtained from the sinewave histogram test. It is shown that a relation exists between the static parameters of the ADCUT and its signal-to-noise (SNR) value. Then, approximations are derived from the original complex formulas for the sinewave histogram method using a built-in synthesized ADC output analyzer circuit. The COordinate Rotation DIgital Computer (CORDIC) algorithm is implemented in order to solve the required histogram functions. Simulations and experiments implementing the method are performed on a 8-bit ADC. Results show small absolute differences between the reference and estimated SNR degradations. However, the method still requires a high-quality sinewave stimulus in order to obtain decent estimation errors on the measurements using the proposed method.

The method in [53] proposes an histogram-based BIST scheme using a space decomposition technique. For a monotonic ADC, the output codes have an approximate stair-like relationship proportional to the input signal. The space decomposition technique consists in capturing not only the ADC output, but also around the voltage window where target code should appear. By doing so, several more codes can be treated in a less time-consuming operation. Simulations are performed on a 6-bit ADC model. The maximum error on DNL and INL is of 0.03 LSB. By utilizing this technique, ADC static parameters can be estimated in shorter testing time with low hardware overhead, although 4 times larger than the time decomposition technique overhead in [50].

The authors in [54] propose a BIST method consisting in delaying the phase of the input stimulus over each period in order to provide coherent sampling. Because of incoherent sampling, a large majority of codes are never exercised and the rest, on the contrary, are always exercised, which limits the interest for building an histogram. In order to overcome this issue, a 8-bit digital to time converter (DTC), capable of producing several delays, is implemented in order to delay the starting time of each generated ramp. This allows to achieve a coherent sampling of the input stimuli. Figure 2.24 shows the closed-loop implementation of the technique. Simulation results show that the BIST scheme can test a 16-bit ADC model with a maximum INL value of 0.67 LSB. Errors in delay do not cause significant errors in the INL estimation. The linearity of the ramp stimulus is a limitation to the method, but the use of techniques such as the SEIR technique helps minimizing this issue.

The work in [55] proposes a DSP-based BIST method in order to adress the test time issue. In a mixed-signal SoC, the integrated DSP can be used in order to test the integrated

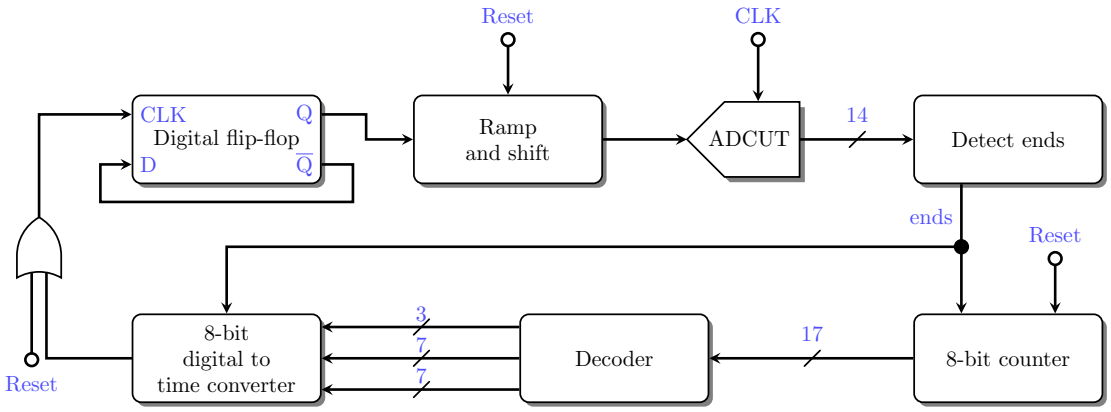


Figure 2.24: Phase-controlled-stimulus ADC BIST technique [54]

ADCs. In this method, a ramp stimulus is applied to the FSR of the ADCUT. The sampled ADC output data can be separated in two sections with respect to the center of the FSR. The two groups of data are accumulated in two sums, from which the ramp stimulus parameters are derived using the integrated DSP. Knowing the reconstructed test stimulus characteristic, the width of each measured code is obtained. Simulations are run on a 10-bit flash ADC with a simple capacitor charged by a current source as ramp stimulus generator. Results show a maximum absolute INL estimation error of about 0.3 LSB, although no discussion about the reduction of noise contribution are discussed.

2.3. On-chip stimulus generation

It is clear that one of the most challenging issues to enable full static linearity BIST is the on-chip integration of an appropriate test stimulus generator. As it is usually accepted by the industry, the test stimulus has to be 2 to 3 bits more linear than the ADCUT. Given that the ADC resolution is steadily increasing, this requirement is becoming a major challenge for the successful integration of static linearity BIST solutions.

The on-chip generation of highly linear ramp stimuli for ADC static linearity test applications remains an open research topic and several works in this line have been presented in the last years. Roughly speaking, two types of stimulus generation techniques can be found in the literature: integration-based generation and DAC-based generation.

2.3.1. Integration-based stimulus generator

Generally speaking, integration-based ramp stimulus generation consists in charging a capacitor with a constant current source in order to generate a voltage ramp stimulus.

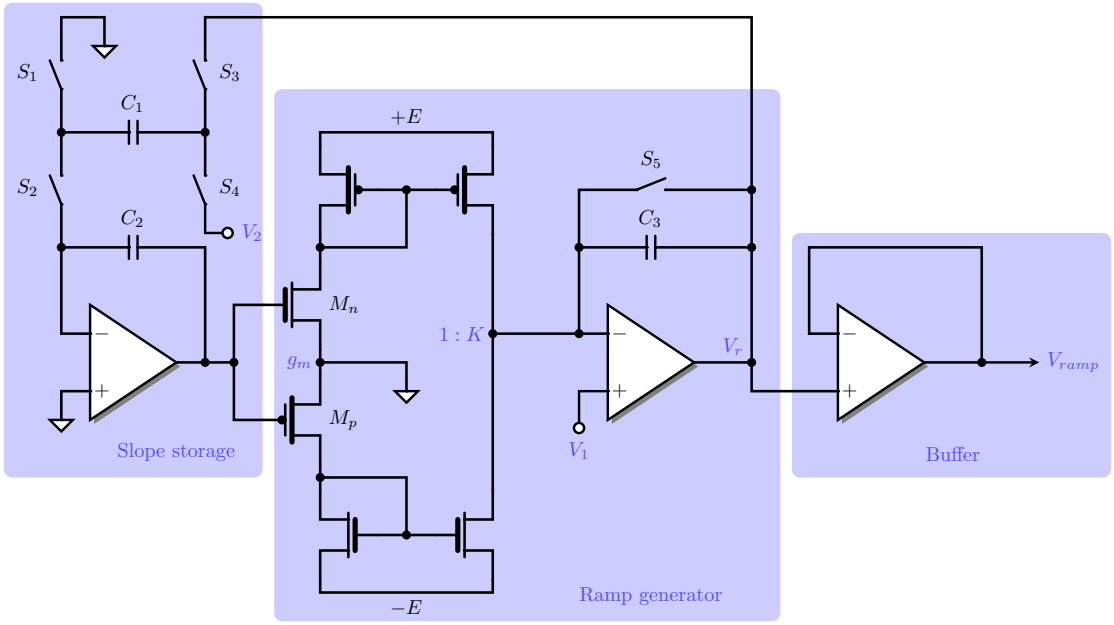
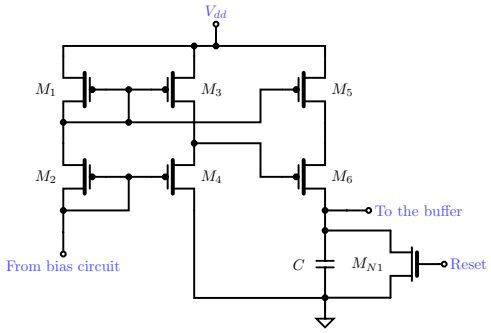


Figure 2.25: Linear ramp generator with automatic slope adjustment [56]

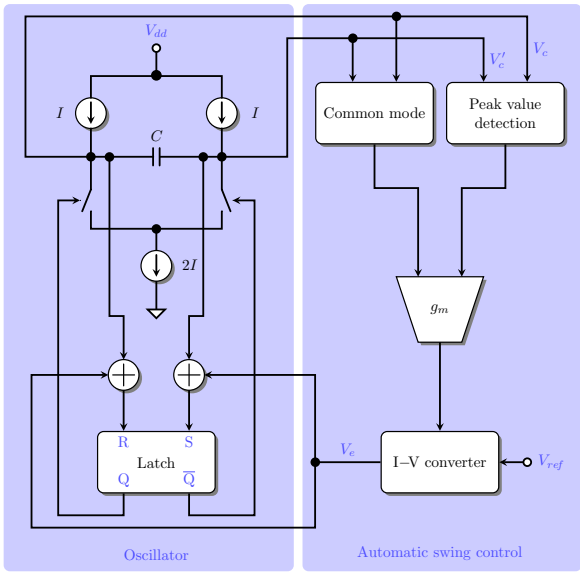
In [56], a linear ramp is generated from a closed-loop ramp generator with an eight-phase clock. The schematic of the proposed generator is shown in Figure 2.25. It is composed of an integrator that senses the amplitude of the output ramp and generates a feedback voltage that is proportional to its slope. This signal is then fed to a voltage-to-current converter whose transconductance controls the current at the input of the ramp integrator. The frequency of operation can be set between 100 Hz and 100 kHz for a ramp between -2 V and 2 V on a 1 μs to 1 s time range, but the generator is not compensated for the operational amplifier offset and the amplitude feedback only guarantees the starting and ending voltages to be correct, and not the linearity of the ramp during the sweep.

The work in [57] presents two different architectures for the implementation of on-chip ramp generators, shown in Figure 2.26. The first circuit is a slow-slope current source-C ramp generator which charges a large capacitor with a small current generated by a cascode current source. A voltage buffer is added to the output in order to prevent current leakage. The second circuit is a relaxation triangle-wave generator. It consists of a control circuit controlling two switches to charge or discharge a capacitor. It is also intended to work at a higher frequency to explore the dynamic behavior of the ADC. Simulation results show a maximum DNL of 0.03 LSB (of an ideal 10-bit ADC) on a 0-9 V peak-to-peak range for the first generator designed in a CMOS 2.0 μm technology and 0.15 LSB on a 0.2-1.4V peak-to-peak range for the second generator designed in a CMOS 1.2 μm technology, both with a 3 V supply voltage.

The authors in [58] and [59] propose an improved on-chip ramp generator with a cor-



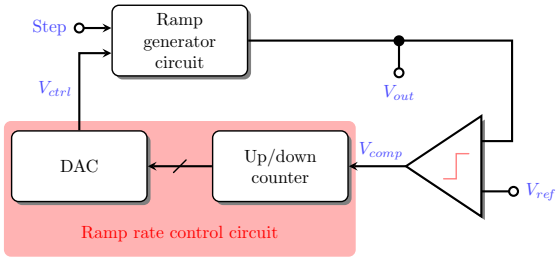
(a) Slow-slope ramp generator



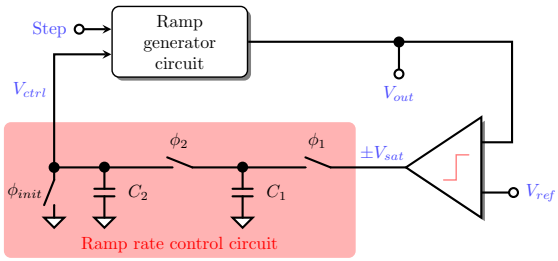
(b) Relaxation oscillator architecture

Figure 2.26: Two types of linear generators [57]

rective adaptive scheme that takes into account the effects of process variations. The basic ramp generator consists of a cascode current mirror where a current source of 0.3 μA is controlled by a feedback voltage V_{ctrl} and charges a 10 pF capacitor. Complementary CMOS switches are used to stop the charging of the capacitor when they are off. Moreover, a switch connected to the terminals of the capacitor allows to initialize the output voltage to a voltage V_{init}. The output ramp sweeps symmetrically on an amplitude of 3 V with $V_{init} = -1.5$ V and a frequency of 10 kHz. The maximum INL of the ramp is 80 μV, which translates to a 0.03% slope error and a linearity of 15 bits. Two corrective adaptive schemes are proposed for regulation, as depicted in Figure 2.27. Figure 2.27 (a) shows a digital adaptive scheme that employs a DAC for the slope control of the ramp generator. The idea is to adjust the current source value of the ramp generator to a proper value which forces the ramp output to reach a reference voltage V_{ref} within the given period. First, a digital scheme is implemented and uses a digital counter to control a simple cyclic DAC which produces the voltage V_{ctrl}. The technique is effective, and the accuracy is about the LSB of the DAC, which means that a high-resolution counter must be employed to minimize the ramp output voltage ripple. Due to silicon constraints in a BIST environment, an analog scheme is implemented. Figure 2.27 (b) shows a capacitive structure similar to the cyclic DAC for the slope control of the ramp generator. The capacitive structure is directly controlled by the output $\pm V_{sat}$ generated by the comparison between the ramp output and V_{ref}. This solution is far more compact than the digital approach. The generator is designed in a CMOS 0.6 μm technology. The transistor-level simulation results



(a) Digital adaptive scheme



(b) Analog adaptive scheme

Figure 2.27: Digital and analog adaptive ramp generators [58]

show a maximum INL of 91 μV which still translates to a 15-bit linearity, an average slope of 0.4% within a calibration time of about 10 cycles.

The work in [60] also employs the charging capacitor concept using a discrete-time adaptive scheme to calibrate the output ramp. The process variations in the capacitor and current source are adaptively compensated by a feedback circuit that controls the ramp generator, as shown in Figure 2.28. This adaptive control integrates the difference between the reference voltage V_{ref} and its output V_{outf} and verifies that this difference satisfies a Least Mean Square (LMS) condition. Three circuits with increasing complexity within the LMS block are proposed. In the first design, an operational transconductance amplifier-capacitor (OTA-C) integrator is put in place of the LMS block. The difference is integrated once the ramp is over. The leakage of the integrating capacitor C_{integ} is contained with switches that are open when the discrete-time integration is performed. But the control voltage is not precise if the ramp is long, such as in an histogram-based ADC test. In order to eliminate that issue, the second design implements two single-ended OTA-C integrators in parallel, reducing the error due to charge injection. There is still a discharge of the capacitor which stores the control voltage V_{ctrl} due to leakage current in the switches during the ramp operation, but the voltage drop is contained to 670 μV, which is smaller than the voltage errors of the first design. Finally, the third ramp generator is particularly aimed at an ADC BIST solution with an histogram-based test. The circuit is fully-differential, and is aimed at improving the ramp stability and

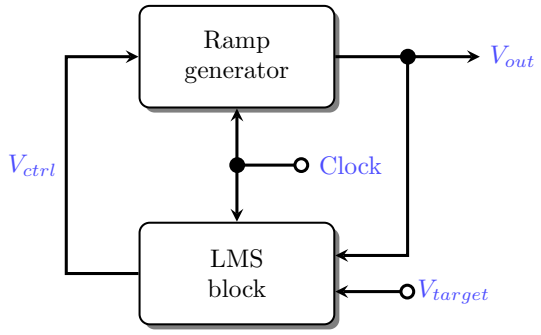


Figure 2.28: Block diagram of the adaptive LMS ramp generator [60]

linearity. The ramp is better stabilized by employing a self-biased Widlar current source instead of the classical current source. The ramp linearity is improved by connecting an operational amplifier (op-amp) at the terminals of the 10 pF charging capacitor which reduces the effect of the generator output resistance and leads to a smaller INL of the ramp. Experimental results are shown for the three generators fabricated in an ORBIT 2.0 μm technology from MOSIS. At an operating frequency of 10 kHz, the first generator has a slope error of 0.6%, with a final offset error of 20 mV on the ramp output which is set in 9 iterations. The second generator has a linearity of 10 bits (close to the 11 bits from simulations), a slope error of 1.5% and a final offset error of 14 mV on the ramp output which is set in 9 iterations as well. Finally, the third generator has a linearity of 11 bits (15 bits in simulation), a slope error of 0.6% and a final offset error of 12 mV on the ramp output which is set in 12 iterations.

2.3.2. DAC-based stimulus generator

Another common strategy for providing an accurate stimulus for static linearity BIST application is based on the use of a DAC.

The works in [61] and [62] propose a test stimulus generator based on a ΣΔ modulator. Authors in [61] propose a BIST scheme to test an ADC and a DAC on the same chip. An histogram is performed on an 8-bit ADCUT by using a 1-bit DAC converting a stored 1-bit ΣΔ stream into a sawtooth ramp. The maximum simulated INL estimation error is about 0.049 LSB for a test of 12781 clock cycles. However, extensive resources are needed to filter the shaped quantization noise in the ΣΔ sequence.

Similarly, the method in [62] proposes a linear ramp waveform generator based on a digital ΣΔ modulator for ADC histogram-based BIST scheme. Figure 2.29 depicts the proposed stimulus generation circuit. The generator is composed of a n-bit digital counter which produces the linear digital input pattern, a digital ΣΔ modulator which converts the

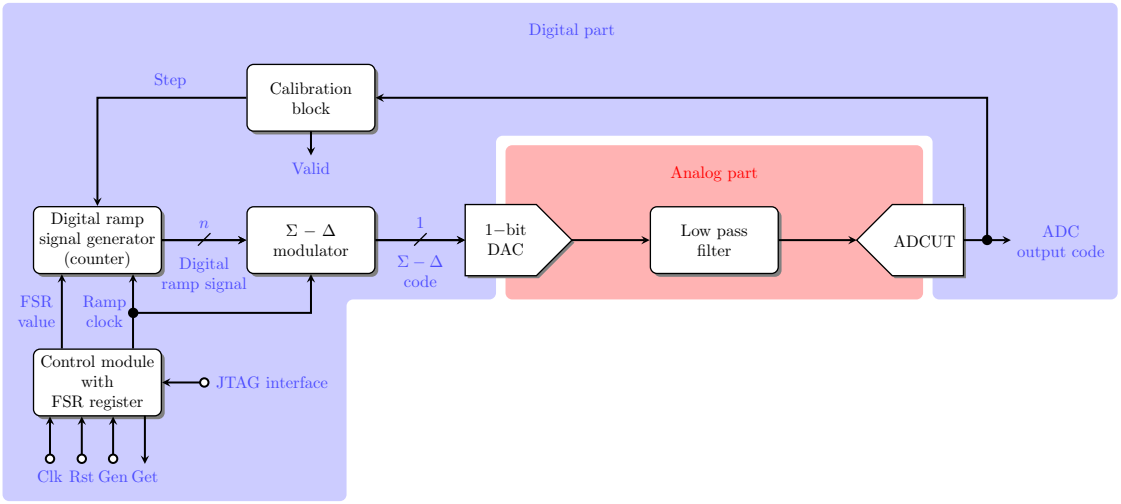


Figure 2.29: Structure of the ΣΔ-based on-chip analog ramp generator [62]

n-bit digital input pattern into a 1-bit digital stream, a control module (itself controlled by JTAG) which control the digital counter, generates the clocks, and controls the output swing of the ramp, a 1-bit DAC that converts the 1-bit digital stream into an analog ramp stimulus, and a low-pass filter to filter out-of-band modulation noise. Finally, a calibration block monitors the slope of the ramp. It counts the number of hits for all the ADC output codes in response to a given ramp, and feedbacks a control voltage to the counter in order to regulate the slope, which is validated when the flag signal *Valid* is raised. The circuit is implemented using Verilog language for the digital part, and a 0.18 μm technology from HJTC for the analog part of the generator. Simulation results show that the ramp is linear on the 0.4-1.4 V range with a 1.8 V supply voltage. Its maximum INL equals $\pm$190 μV which corresponds to a linearity of 12 bits.

The works in [63], [64] and [65] are aimed at relaxing the linearity requirements of the generator instead of targeting the generation of a highly pure stimulus. The proposed techniques use a low-resolution DAC with redundant, poorly matched elements together with a Deterministic Dynamic Element Matching (DDEM) strategy in order to estimate the static characteristics of a high-resolution ADC. The authors in [63] propose to use a DDEM DAC with a higher (or equal) resolution than the ADCUT, but the real linearity requirements of the DAC is much lower than that of the tested device. The DDEM DAC is a current-steering thermometer-coded DAC, which consists of a bank of digitally-controlled identical current sources, as depicted in Figure 2.30. The output is a sum of the currents which is function of the digital input pattern controlled by the DDEM technique. This total current is then fed to an active current-to-voltage converter. By choosing appropriate deterministic switching sequences, the performance of the cyclic DDEM algorithm is much more effective and simple to physically implement than that of a random DEM

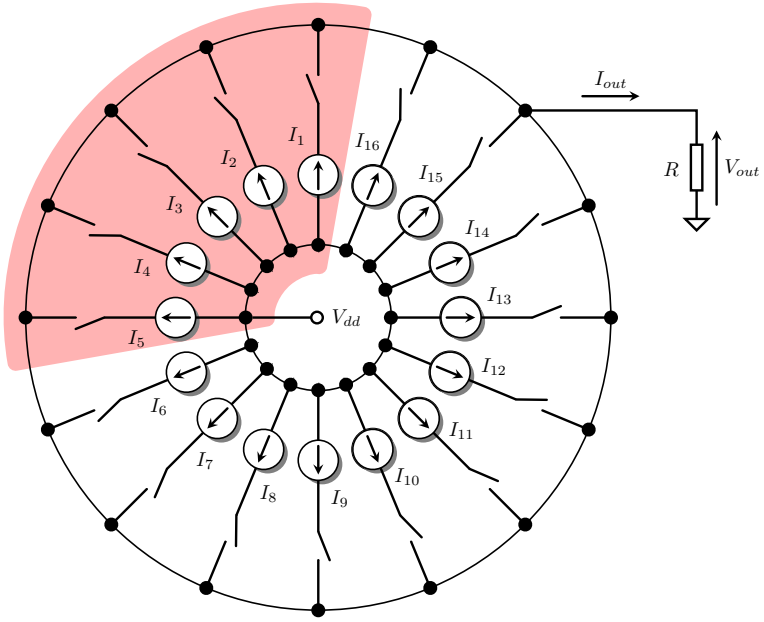


Figure 2.30: Four-bit DDEM DAC with $P = 4$ and $k = 5$, where sources I_1 to I_5 are active (red section) [63]

algorithm. One thousand randomly generated 16-bit ADC/DAC pairs are simulated with 128 samples per code using the DDEM method. The simulation results show an absolute maximum INL error estimation mostly less than 0.5 LSB, where the elements of the DAC have at best a linearity of 12 bits.

In [64], an 8-bit DDEM DAC is designed and fabricated in a 0.50 μm standard CMOS process. The 8-bit coarse DAC is only 5-bit linear, but it can provide test performance equivalent to that of a 13-bit linear DAC. Without DDEM, the original 8-bit DAC has an INL error of 10.3 LSB, which translates linearity of less than 4 bits. The DAC output is measured and stored in order to be used as input stimulus for 11-bit and 12-bit simulated ADCs with 5 to 10 LSB INL. When using the full DDEM implementation, the 8-bit DAC can test the 11-bit and 12-bit ADCs with error of $\pm$0.5 LSB and $\pm$1 LSB respectively. The enhanced DAC linearity is equivalent to 13 to 14 bits.

The work in [65] proposes another DDEM DAC for ADC linearity test based on the previous works. Figure 2.31 shows the proposed DDEM-based generator. The generator consists of two blocks, a segmented $(n_M + n_L)$-bit DDEM DAC block and a n_d-bit dither DAC block, where n_M is the number of MSB bits on which the DDEM algorithm is performed, and n_L is the number of LSB bits. The n_d-bit dither DAC block enhances the DDEM DAC output by n_d bits. The DDEM DAC is part of a closed-loop structure along with the ADCUT, using a specific BIST strategy. The DDEM DAC is simulated with a 7-bit MSB array and a 6-bit LSB array, while the dither DAC has a 4-bit resolution,

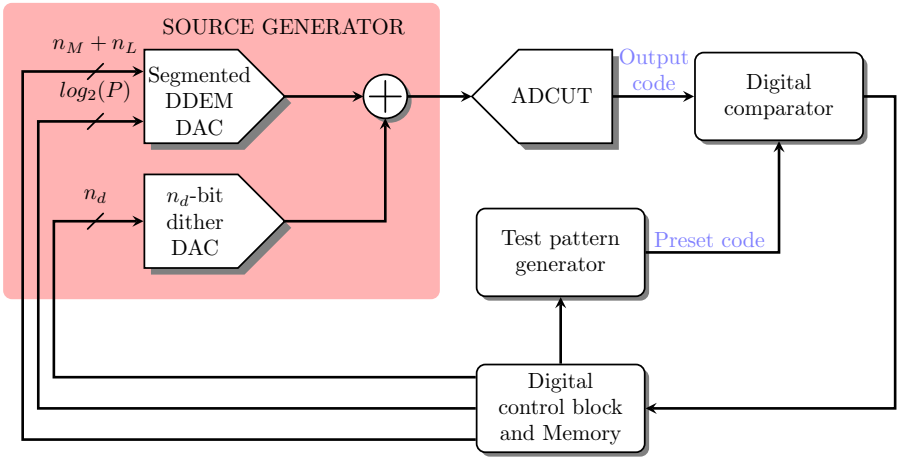


Figure 2.31: Segmented DDEM DAC with n_d-bit dither DAC BIST structure [65]

summing a 15-bit effective resolution. The test is performed on a 12-bit ADC model with 4 LSB INL error. Results show that the INL estimation error is bound to ± 0.15 LSB. Then, 100 ADC/DAC pairs are generated. The generated ADCs have an INL between 1.5 LSB and 5.5 LSB and the generated DACs are set with different mismatch errors. The INL estimation errors are contained between -0.118 LSB and 0.139 LSB, which proves the method to be robust. The DDEM DAC is then fabricated in a CMOS 0.5 μm technology and is used to test a 12-bit simulated ADC with 16 samples per code. Results show that the absolute INL estimation error is below 0.18 LSB.

2.3.3. Discussion

This section has detailed the different contributions over the last years in the field of ramp generation for analog/mixed-signal BIST, an especially for ADC BIST. Table 2.1 shows the different ramp generation techniques presented in section 2.3. Two distinct groups for ramp generation can be observed: current-charged, capacitor-based ramp generators and DAC-based ramp generators.

Table 2.1: Comparison of ramp generation techniques

Reference	Technique	Complexity	Linearity (bits)		Slope error (%)
			Simulation	Measurement	
[56]	Automatic slope adjustment ramp generation	medium	N/A	8	N/A
[57]	Current source-C ramp generation	low	14	N/A	N/A
	Relaxation oscillator architecture	medium	12	N/A	N/A
[59]	Adaptive triangle-wave generation	low	14	N/A	0.4
[60]	Adaptive LMS integrators	low	N/A	N/A	0.6
		low	11	10	1.5
		medium	15	11	0.6
[61]	On-chip DAC and ADC testing	high	14	N/A	N/A
[62]	ΣΔ-based ramp generation	high	12	N/A	N/A
[64]	Dynamic Element Matching DAC	high	14	N/A	N/A

Although a quite consistent list of works about on-chip generators have been proposed in the last years, very few of them show experimental data. Moreover, the large differences in linearity between simulation and experimental results imply that on-chip stimulus generation is not trivial.

Chapter 3

Servo-loop algorithm

The proposed BIST technique is focused on the use of a modified version of the servo-loop algorithm. In this section, a study of the classical servo-loop and the proposed servo-loop algorithms is performed, and a comparison of the two techniques in terms of accuracy and test time is discussed. Then, the pipeline ADC structure is presented and the reduced code linearity testing technique (RCLT) for pipeline ADCs from [38]−[41] is explained. Finally, a study of the modified servo-loop technique combined with RCLT for pipeline ADCs is conducted.

3.1. Description

Before any comparison between the two techniques, the state-of-the-art of the classical servo-loop is presented, then the proposed servo-loop is described.

3.1.1. Classical servo-loop technique

The measurement of high-speed ADC static linearity using a closed-loop technique is not new and has been used in the industry since several decades. The classical servo-loop technique has been detailed in section 2.2, illustrated in Figure 2.8, and is fully described in [2].

A few works have been done over the past years in an effort to give a mathematical background to this technique. The work in [3] compares three methods used to measure the linearity of Nyquist-rate ADCs, namely the tally and weight method, the histogram method (open-loop configurations), and the classical servo-loop method (closed-loop configuration). The authors propose a linear modeling of the servo-loop technique in order to improve its efficiency and accuracy. The RMS estimation error for transition levels is compared to the maximum theoretical performances described by the Cramér-Rao bounds as a function of the number of conversion cycles and ADC input noise magnitude.

It is shown that the classical servo-loop can achieve maximum efficiency through the maximum-likelihood estimation of the ADC transition levels. For ADCs with low input-referred noise, the maximum likelihood method does not provide a significant accuracy improvement, whereas in the case of high input-referred noise, the maximum likelihood estimates remain close to the theoretical bound. However, this approach is not suitable for a BIST approach as it requires a significant amount of computational resources.

The work in [5] and [6] propose a study of the servo-loop performance. In both contributions, the technique is described and analyzed under the variation of several test parameters. In [5], the study is conducted towards the influence of input-referred noise and step size on the digital-based version of the servo-loop. Several points are discussed, such as the required number of points for measurement averaging, the optimal step size of the DAC output with respect to specified measurement accuracy and elapsed test time, and optimum test time with respect to the ADC input-referred noise and specified measurement accuracy. However, the study does not take into account the output-referred DAC noise which renders the study incomplete.

In [6], the servo-loop performance is assessed using a mathematical model based on a Markov chain. The probability that the integrator output voltage being equal to ADC transition voltage t_k at normalized time $n = t/T_c$ is indicated by a steady-state probability vector $P(n)$ filled with transition probabilities P_k ($\sum_{k=1}^{N-1} P_k = 1$, N being the number of ADC levels). Depending on the magnitude of the noise and the distance between the integrator output to the next transition, each P vector have a limited amount of non-zero elements since the variations are close to the targeted code. All P vectors are compiled in a transition matrix T, and solving the system $P = T \cdot P$ allows to predict the future evolution of the servo-loop, which is typical of a Markov chain. The servo-loop model uses a continuous-time integrator in order to generate the stimulus. The integrator output is sampled by the clock signal having a period T_c. The equivalent step size is set to a fixed amount $\delta = G \cdot V_{ref} \cdot T_c$, where G is the gain of the integrator and V_{ref} is a reference voltage. The input-referred ADC noise and the output-referred integrator noise are taken into account and summed in a total input-referred ADC noise contribution with standard deviation σ_n. The servo-loop performance is assessed in terms of the number of samples for measurement averaging versus the ratio of the standard deviation σ_{t_k} of a noisy transition over the step size δ, with respect to the ratio σ_n/δ. From these results, the test parameters can be evaluated for any given ADC. The authors state that, when σ_n is not negligible, the ADC input drifts from the targeted transition voltage t_k by an amount equal to the step size δ, but it should be considered that the output-referred integrator noise makes the step size δ vary in time, independently of the value of the input-referred ADC noise.

3.1.2. Proposed servo-loop technique for BIST application

As defined in [2], the classical servo-loop technique relies on a high-resolution voltmeter for the accurate measurement of each target transition voltage. In a BIST implementation, the available integrated circuitry must remain small while providing a sufficient test accuracy. Such a complex system would include at least a high-linearity DAC, a comparator and several logic blocks, thus requiring a non-negligible area which would add to the stimulus generator area. This system would be comparable in size to the ADCUT core area, which is not acceptable for a BIST application. A different measurement technique must then be developed for a small yet effective BIST circuitry. In this section, we propose a new servo-loop technique where the measurement of each code width is performed without the need for a voltmeter, but only using digital blocks.

The proposed servo-loop configuration is a modified version of the classical servo-loop [2], [44]. As depicted in Figure 3.1, the continuous-time ramp stimulus generator used in the standard servo-loop is replaced by the proposed discrete-time ramp stimulus generator. The operation is quite similar to the classical servo-loop, but instead of measuring the voltage level of the ADC transitions, we will measure directly the code width of each code. The code width W_k of a code c_k ($k = 1,2,...,2^N - 1$) is measured by counting the number of ramp steps with a digital counter as we traverse the input range between the two transitions t_k and t_{k+1} that corresponds to the target code. Specifically, starting at the transition t_k at the left of a target code c_k, if the ADC output code is below or equal to the target code, the ADC input voltage is raised in steps and the counter starts incrementing until the transition t_{k+1} at the right of the code is crossed, in which case the ADC output code changes to upper code c_{k+1}. Then, the ADC input is lowered in steps, and the counter continues to increment until transition t_k is crossed, in which case the ADC output code becomes lower code c_{k-1}. The process is repeated r times oscillating the ADC input around the input range that corresponds to target code c_k in order to average the combined ramp generator and ADC noises. The number of steps per code for code c_k is expressed as

$$\widehat{m_k} = \frac{\sum_{i=1}^{r} n_i}{r}, \quad k = 1,2,...,2^N - 1 \tag{3.1}$$

where n_i the number of steps counted during excursion i. Both upward and downward excursions counts are summed, and the expression is divided by the number of excursions, r, in order to provide a mean value estimate of the number of steps for code width W_k. The number of steps per code $\widehat{m_k}$ is then multiplied by the estimated step size $\widehat{s}$ in order to evaluate code width W_k. Finally, DNL and INL figures can be then easily extracted from the static ADC characteristic using their standard definitions. Their expressions are then

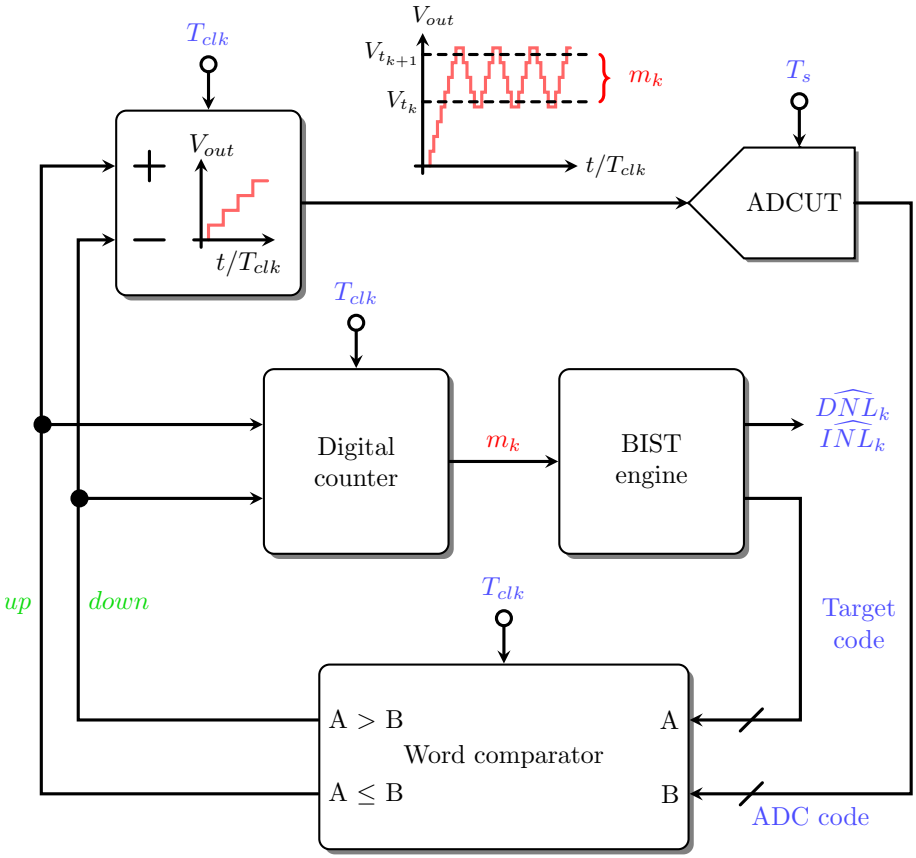


Figure 3.1: Schematic of the modified servo-loop circuit

defined as

$$\left\{ \begin{array}{ll} \widehat{DNL_k} & = \widehat{m_k} \cdot \widehat{s} - 1\ LSB \\ \widehat{INL_k} & = \sum_{i=1}^{k} \widehat{DNL_k} \end{array} \right\}, \quad k = 1, 2, \ldots, 2^N - 1 \tag{3.2}$$

where

$$\widehat{s} = \frac{\sum_{k=1}^{2^N-1} \widehat{m_k}}{2^N - 1}. \tag{3.3}$$

The approximation on the measurement and the step size of the generator implies an static error in the DNL estimation of each measured code. This error is cumulatively summed in the INL estimation, but can be removed using the best-fit algorithm [1] presented in subsection 2.1.6, that can be easily adapted in a BIST approach.

Let us write the equation of the generator output during an upward excursion. After an initialization routine, the generator output is set at an initial voltage $V_{out}(n = 0)$ which is just below transition voltage V_{t_k}. The generator output increments until transition voltage

$V_{t_{k+1}}$ is crossed.

$$\begin{aligned} V_{out}(1) &= V_{out}(0) + s \\ V_{out}(2) &= V_{out}(1) + s = V_{out}(0) + 2 \cdot s \\ V_{out}(3) &= V_{out}(2) + s = V_{out}(0) + 3 \cdot s \\ &\dots \end{aligned} \tag{3.4}$$

where s is the step size of the generator. When the upper transition t_{k+1} is crossed, we get

$$V_{out}(n_u) = V_{out}(0) + n_u \cdot s \tag{3.5}$$

where n_u is the number of steps needed for the generator output to go from initial voltage $V_{out}(0)$ to the voltage just after transition t_{k+1} is crossed.

The BIST only allows to monitor the ADCUT output code, so no information can be gathered from the generator output voltage while it traverses the voltage range between the two adjacent transitions. Consequently, while the digital counter is incremented, the BIST engine waits for a reversal of the state of the word comparator, which means that the transition t_{k+1} is crossed. Provided the generator output started just below transition voltage V_{t_k}, the correct number of steps for the best estimation of the code width W_k is found when the generator output finishes its upward excursion below transition t_{k+1}. In order to do so, the BIST engine forces the generator output to go down. The number of steps done during the upward excursion n_u is reduced by a number n_d which corresponds to the number of steps needed for the generator output to go from its actual voltage above transition voltage $V_{t_{k+1}}$ to the voltage once $V_{t_{k+1}}$ is crossed. Equation (3.5) changes to

$$V_{out}(n_1) = V_{out}(0) + n_1 \cdot s \tag{3.6}$$

where $n_1 = n_u - n_d$ is the number of counted during this first upward excursion.

Finally, the downward excursion is performed and finishes when the generator output crosses transition voltage V_{t_k} again. Equation (3.6) updates to

$$V_{out}(n_1 + n_2) = V_{out}(0) + (n_1 - n_2) \cdot s \tag{3.7}$$

where n_2 is the number of counted steps during this first downward excursion, and is equal to n_1 in the ideal case.

However, the BIST technique is performed under the influence of the nonidealities from both the generator and the ADCUT, especially noise. Noise at the generator output makes

the step size vary for each increment or decrement, while noise at the ADCUT input have a influence on each transition voltage. After the upward excursion is performed, we can re-write equation (3.5) as

$$V_{out}(n_1) = V_{out}(0) + n_1 \cdot s + \sum_{k=0}^{n_1} V_n(k). \tag{3.8}$$

where $V_n(k)$ is the generator output noise voltage at time k. The generator output after the downward phase is then expressed as

$$V_{out}(n_1+n_2) = V_{out}(0) + (n_1-n_2) \cdot s + \sum_{n=0}^{n_1+n_2} V_n(n). \tag{3.9}$$

Equation (3.9) shows that the values of n_1 and n_2 might not be equal and might greatly differ from the ideal number of required steps for code width W_k because of the noisy step size and transition voltages. For example, considering an upward excursion towards $V_{t_{k+1}}$, large values of the step size combined to lower values of the transition voltage might greatly reduce the value of n_1, while smaller values of the step size combined to higher values of the transition voltage might give to a much bigger count for n_1, leading to a large inaccuracy on the code width estimation. In order to reduce this undesired noise effect, upward and downward excursions are iterated for a given number of times, which is dependent on the noise magnitude and the generator step size with respect to the measured code width. The procedure is iterated r times, and for each excursion i ($i = 1, ..., r$), each n_i value is kept in memory. Then, using equation (3.1), the number of steps per code $\widehat{m_k}$ for code c_k is estimated as the mean value over the r iterations. The r estimations of code width W_k are expressed as

$$\begin{aligned}
\widehat{W_{k,1}} &= n_1 \cdot s = V_{out}(n_1) - V_{out}(0) + \sum_{n=0}^{n_1} V_n(n) \\
\widehat{W_{k,2}} &= n_2 \cdot s = V_{out}(n_1) - V_{out}(n_1+n_2) + \sum_{n_1+1}^{n_2} V_n(n) \\
&\dots \\
\widehat{W_{k,r}} &= n_r \cdot s = V_{out}(n_1+\dots+n_{r-1}) - V_{out}(n_1+\dots+n_r) + \sum_{n_{r-1}+1}^{n_r} V_n(n).
\end{aligned} \tag{3.10}$$

By summing all $\widehat{W_{k,i}}$ and dividing by $r \cdot s$, we can compute the mean value over the r

iterations of the estimated number of steps for code width W_k as

$$\sum_{i=1}^{r} \frac{\widehat{W_{k,i}}}{r \cdot s} = \widehat{m_k} \tag{3.11}$$

which gives

$$\widehat{m_k} = \frac{\sum_{k=1}^{r} (-1)^{k+1} \cdot (V_{out}(n_1 + \ldots + n_k) - V_{out}(n_1 + \ldots + n_{k-1}))}{r \cdot s} + \frac{\sum_{n=0}^{n_1+\ldots+n_r} V_n(n)}{r \cdot s}. \tag{3.12}$$

In equation (3.12), if we consider a noiseless ADC, the first component is the sum over r iterations of the same voltage difference. The equation can be simply rewritten as

$$\widehat{m_k} = \widehat{m_k}^{id} + \frac{\sum_{n=0}^{n_1+\ldots+n_r} V_n(n)}{r} \tag{3.13}$$

where we can clearly see two components. The first component $\widehat{m_k}^{id}$ is exactly $\widehat{W_k}/s$, which corresponds to the ideal equivalent number of steps required to cover W_k using a step size of s, and the second component is a noise component. The noise component is effectively averaged, provided the noise is a white Gaussian noise.

The mean value and standard deviation of the set of n_i values can be easily computed in the BIST implementation for each code width measurement. Furthermore, they can be computed on-line using a simple algorithm, requiring a small amount of computational resources and memory [**?**].

- Initialization: $\widehat{m_k} = 0$, $\widehat{m_{k,p}} = 0$, $S = 0$ ▷ $\widehat{m_{k,p}}$ previous value of $\widehat{m_k}$

for $i = 1 \rightarrow r$ **do** ▷ for each new measurement of code width W_k

$\widehat{m_{k,p}} = \widehat{m_k}$

$\widehat{m_k} = \widehat{m_k} + (n_i - \widehat{m_k})/i$ ▷ running mean estimation

$S = S + (n_i - \widehat{m_k}) \cdot (n_i - \widehat{m_{k,p}})$

$\widehat{\sigma_k} = \sqrt{S/i}$ ▷ running standard deviation estimation

end for

3.2. Characterization and comparison of the two techniques

In this section, the classical servo-loop technique and the discrete-time servo-loop technique proposed in the previous section are characterized and compared. The comparison covers the following points:

- number of samples to average,
- generator step size,
- accuracy-test time trade-off,
- INL estimation accuracy.

3.2.1. Number of averaged samples

In the classical servo-loop technique, the measurement of a transition begins when the ADCUT input is settled, meaning the generator output voltage has to cross the targeted transition voltage. Then, an averaging over n points is performed in order to reduce the noise contribution during the measurement. For the proposed servo-loop technique, the algorithm waits for the generator output voltage to cross the targeted transition voltage. Then the generator output is lowered until its voltage is just below the targeted transition voltage.

Both classical and proposed servo-loop techniques are compared by means of behavioral simulations implemented in Matlab. The developed behavioral model takes into account the main nonidealities affecting the accuracy of the servo-loop measurements. Thus, the model includes white noise contributions from the ramp generator and the ADCUT which are added to the ADCUT input, with respective standard deviations $\sigma_{ramp} = 0.5$ LSB and $\sigma_{ADC} = 0.15$ LSB, corresponding to absolute maximum noise contributions of 1.5 LSB and 0.45 LSB (at $\pm 3\ \sigma$). The value of σ_{ramp} is chosen in order to simulate the technique under a large noise contribution, while the value of σ_{ADC} is chosen as a typical RMS noise value for a commercial ADC. Moreover, the starting voltage of the ramp generator is set to 0 V and the voltage of transition t_k is set to 5 times the step size which is fixed to 0.2 LSB, meaning $V_{t_k} = 1$ LSB.

In this set of simulations, we will study the effect of the total number of averaged samples, n, and for that we will sweep n from 1 to 2^{15}. For each value of n, the simulation is repeated 2000 times.

3.2.1.1. Classical servo-loop technique

Table 3.1 lists the mean value and standard deviation of the voltage estimation error ε_k, defined as the difference between the estimated transition voltage $\widehat{t_k}$ and the actual transition voltage t_k.

Table 3.1: Mean value and standard deviation in LSB of the estimation error for the measurement of a transition voltage V_{t_k} using the classical servo-loop technique ($V_{t_k} = 1$ LSB, $s = 0.2$ LSB step, $\sigma_{ramp} = 0.5$ LSB, $\sigma_{ADC} = 0.15$ LSB)

n	$\mu_{\varepsilon_k}(LSB)$	$\sigma_{\varepsilon_k}(LSB)$	n	$\mu_{\varepsilon_k}(LSB)$	$\sigma_{\varepsilon_k}(LSB)$
1	-0.24	0.59	2^8	-0.0067	0.22
2	-0.17	0.602	2^9	0.0021	0.16
2^2	-0.11	0.58	2^{10}	0.0016	0.11
2^3	-0.07	0.59	2^{11}	-0.0025	0.08
2^4	-0.05	0.57	2^{12}	$4.11\cdot10^{-4}$	0.056
2^5	-0.035	0.51	2^{13}	$-2.87\cdot10^{-4}$	0.041
2^6	-0.021	0.42	2^{14}	$6.99\cdot10^{-4}$	0.027
2^7	-0.0062	0.299	2^{15}	$8.08\cdot10^{-5}$	0.019

As described in [5], the results show that the accuracy of the measurement determines the required number of points for sufficient averaging. The mean value of the estimation error committed on transition voltage measurement becomes sufficiently low when n is about 2^6 to 2^7. However, for small values of n (1 to 2^4), the standard deviation of the estimation error remains high, then steadily decreases for intermediate values and remains low for large values of n (2^{12} to 2^{15}). This forces to choose a large number of samples in order to ensure a good measurement accuracy for each targeted transition.

In order to show the influence of the number of samples to average with respect to the error committed on the estimated DNL of a code, the simulation is performed for two adjacent transition voltages t_k and t_{k+1} of the ADCUT in the same conditions than for the previous case. Once the voltage of transition t_k is estimated, the ADC input targets transition t_{k+1} which voltage is in turn estimated. The difference between the two estimated transitions $\widehat{W_k} = \widehat{t_{k+1}} - \widehat{t_k}$ is subtracted to the real code width $W_k = t_{k+1} - t_k$, which is equal to the DNL estimation error for code c_k. The simulation is repeated over 2000 runs for a code width $W_k = 1$ LSB. The mean value and the standard deviation of the DNL estimation error ε_{DNL} for code c_k are computed. Table 3.2 lists the results of the simulation.

Table 3.2: Mean value and standard deviation in LSB of the DNL estimation error of code width W_k using the classical servo-loop technique ($W_k = 1$ LSB, $s = 0.2$ LSB step, $\sigma_{ramp} = 0.5$ LSB, $\sigma_{ADC} = 0.15$ LSB)

n	$\mu_{\varepsilon_{DNL}}$	$\sigma_{\varepsilon_{DNL}}$	n	$\mu_{\varepsilon_{DNL}}$	$\sigma_{\varepsilon_{DNL}}$
1	0.0021	0.82	2^8	-0.00104	0.31
2	-0.048	0.85	2^9	0.0034	0.22
2^2	-0.0069	0.86	2^{10}	-0.0057	0.15
2^3	-0.022	0.84	2^{11}	$-5.23{\cdot}10^{-4}$	0.11
2^4	0.0105	0.78	2^{12}	$-5.72{\cdot}10^{-4}$	0.079
2^5	0.0019	0.75	2^{13}	0.0014	0.055
2^6	0.012	0.55	2^{14}	0.0018	0.041
2^7	-0.0029	0.45	2^{15}	$-2.69{\cdot}10^{-4}$	0.026

Results show that the mean value of ε_{DNL} remains very low independently of the number of samples, which is explained by the fact that both transitions are measured with the same accuracy for each value of n. The standard deviation becomes low around large values of n.

3.2.1.2. Proposed servo-loop technique

Table 3.3 lists the mean value and standard deviation of ε_k for $\sigma_{ramp} = 0.5$ LSB, while Figure 3.2 shows the mean value and standard deviation for increasing values of σ_{ramp}.

Table 3.3: Mean value and standard deviation in LSB for the measurement of a code width W_k using the proposed servo-loop technique ($W_k = 1$ LSB, $s = 0.2$ LSB step, $\sigma_{ramp} = 0.5$ LSB, $\sigma_{ADC} = 0.15$ LSB)

n	$\mu_{\varepsilon_{DNL}}$	$\sigma_{\varepsilon_{DNL}}$	n	$\mu_{\varepsilon_{DNL}}$	$\sigma_{\varepsilon_{DNL}}$
1	-0.035	1.021	2^8	-0.037	0.272
2	-0.048	0.982	2^9	-0.047	0.191
2^2	-0.041	1.024	2^{10}	-0.054	0.130
2^3	-0.008	1.029	2^{11}	-0.057	0.092
2^4	0.102	1.032	2^{12}	-0.060	0.063
2^5	0.099	0.939	2^{13}	-0.058	0.045
2^6	0.035	0.616	2^{14}	-0.060	0.031
2^7	-0.008	0.415	2^{15}	-0.059	0.023

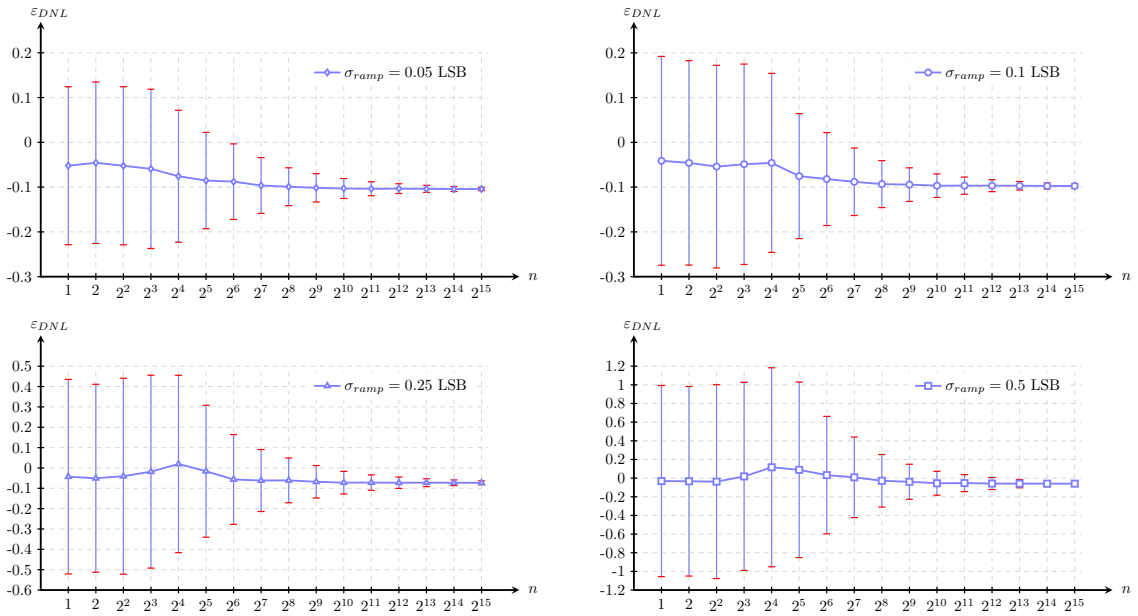


Figure 3.2: Mean and standard deviation of DNL estimation error with respect to the number of samples to average using the proposed servo-loop technique

The mean value stays relatively constant in each case. As n increases, the mean value tends towards a stable value which depends on σ_{ramp}, but shows a non-negligible DNL estimation error for most cases in comparison to the classical technique. For large values of n, the measurement has very little variation, but at the expense of the test time. Moreover, as σ_{ramp} increases, the standard deviation becomes larger and a higher number of samples are required to keep the same measurement accuracy.

3.2.1.3. Comparison

Considering identical test times, and despite the offset estimation error for the proposed technique, the comparison of Table 3.2 and Table 3.3 shows that the classical and the modified servo-loop techniques provide similar performance in terms of measurement variability. The accumulated error has very little influence on the DNL estimation, but implies large errors in the INL estimation. Fortunately, provided a sufficient number of samples per code measurement is taken, the accumulated error is approximately linear, and can be easily corrected using the best-fit algorithm presented in subsection 3.1.2.

3.2.2. Generator step size

The performance of the technique is highly dependent on the step size of the generator. In order to study the influence of the step size on the DNL estimation error ε_{DNL}, two

simulations are performed measuring the width of a code c_k of width $W_k = 1$ LSB over 2000 runs. For each run, the measurement of the DNL is done by collecting 1024 samples for noise averaging.

First, the step size is varied with respect to the ADC input noise magnitude, which is represented by the ratio s/σ_{ADC}. The standard deviation of the ADC input-referred noise is fixed to $\sigma_{ADC} = 0.15$ LSB, which corresponds to a $\pm 3\ \sigma_{ADC}$ variation of ± 0.45 LSB. The step size then varies from a small value ($s/\sigma_{ADC} = 0.25$, meaning $s = 0.0375$ LSB) to a large value ($s/\sigma_{ADC} = 4$, meaning $s = 0.6$ LSB). Moreover, the magnitude of the generator output noise is varied for different values: low ($\sigma_{ramp} = 0.05$ LSB), medium ($\sigma_{ramp} = 0.1$ LSB), large ($\sigma_{ramp} = 0.25$ LSB), and very large magnitudes ($\sigma_{ramp} = 0.5$ LSB). The value of σ_{ramp} is chosen in order to simulate the technique under several noise contributions which can exist in a real case, while the value of σ_{ADC} is chosen as a typical RMS noise value for a commercial ADC.

The second simulation consists in varying the step size with respect to the LSB of the ADCUT. The chosen parameters are: $\sigma_{ADC} = 0.15$ LSB, code width W_k of 1 LSB, step size varying from 0.05 LSB to 2 LSB.

3.2.2.1. Classical servo-loop technique

Figure 3.3 shows the mean value and standard deviation of the DNL estimation error with respect to different values of the ratio s/σ_{ADC}, while Figure 3.4 plots the mean value and standard deviation of the DNL estimation error with respect to different values of the ratio s/W_k.

The first simulation case shows two main results. First, the mean value of the DNL estimation error remains close to zero for any value of the ratio s/σ_{ADC} and for any noise magnitude at the generator output. Secondly, as the noise magnitude at the generator increases, the standard deviation of the DNL estimation error becomes increasingly large when the step size becomes smaller with respect to the ADC input noise.

In the second case study, the mean value of the DNL estimation error is also close to zero. Its standard deviation is significant for large values of ratio s/W_k and small magnitudes of the generator output noise but as σ_{ramp} rises, the trend reverses and its standard deviation becomes large for small values of ratio s/W_k.

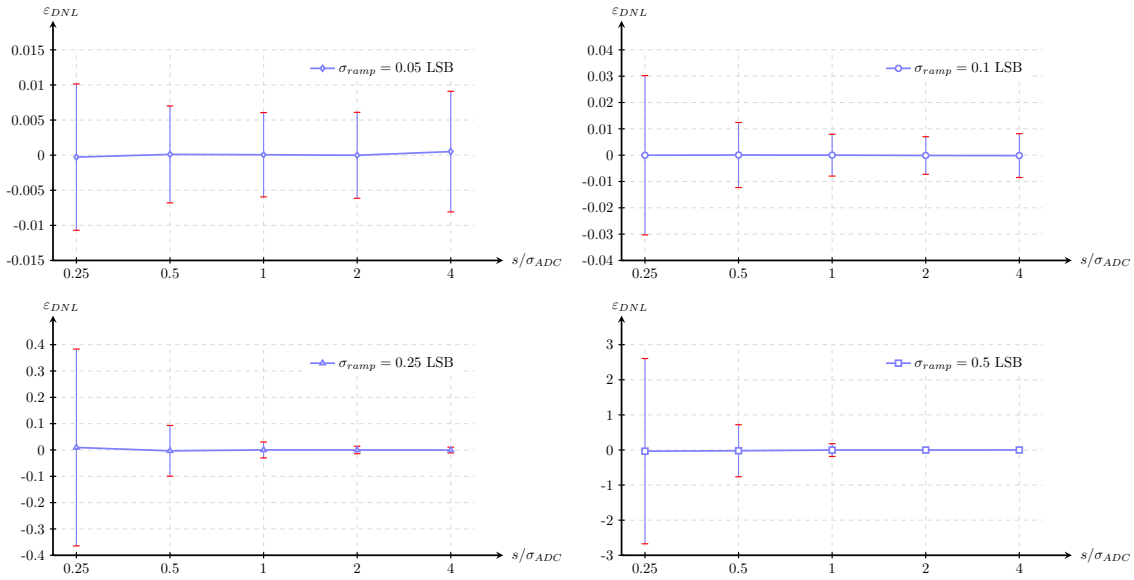


Figure 3.3: Mean and standard deviation of DNL estimation error with respect to the ratio of generator step size over the ADC input-referred noise magnitude using the classical servo-loop technique

3.2.2.2. Proposed servo-loop technique

As in the classical technique study, two simulations are performed for a code c_k of width $W_k = 1$ LSB in order to show the influence of the step size on the accuracy of the modified servo-loop technique.

For various values of σ_{ramp} ($\sigma_{ramp} = 0.05$ LSB to 0.5 LSB), the first simulation parameters are: $\sigma_{ADC} = 0.15$ LSB, step size varying from a small value (0.0375 LSB) to a large value (0.6 LSB). Figure 3.5 shows the mean value and standard deviation of the DNL estimation error with respect to different values of the ratio s/σ_{ADC}.

In the second simulation, the step size varies with respect to the LSB of the ADCUT. The chosen parameters are: $\sigma_{ADC} = 0.15$ LSB, code width W_k of 1 LSB, step size varying from 0.05 LSB to 2 LSB for $W_k = 1$ LSB. Figure 3.6 plots the mean value and standard deviation of the DNL estimation error with respect to different values of the ratio s/W_k.

Results in Figure 3.5 show that while σ_{ramp} is small, the noise at the ADC input becomes dominant, and for a small s/σ_{ADC} ratio, the mean error value is the largest because the step size is very small and a larger number of samples is required in order to reduce the estimation error. However, as the ratio increases, the ADC input-referred noise becomes less dominant with respect to the step size, and the DNL estimation error tends towards the offset value of -0.1 LSB already observed when the number of samples is varied. As the ramp noise increases, the ADC input-referred noise becomes less dominant with

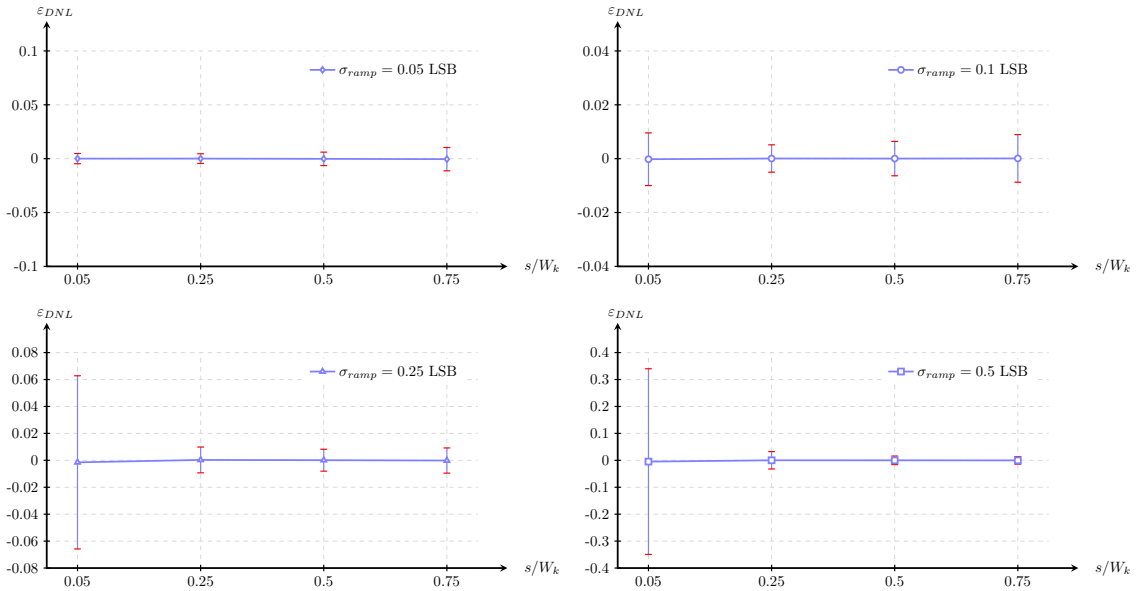


Figure 3.4: Mean and standard deviation of DNL estimation error with respect to the ratio of the generator step size over the LSB of the ADCUT using the classical servo-loop technique

respect to the generator output noise, but the total noise contribution is larger, and leads to large standard deviation for small s/σ_{ADC} values because the number of averaged samples is insufficient for an acceptable DNL estimation error.

For the second simulation case, an analysis of the results in Figure 3.6 shows that the DNL estimation mean value remains constant, around -0.1 LSB, for step values between 0.25 LSB and 0.75 LSB, with very little influence of the generator output noise magnitude. The standard deviation remains low as well for maximum DNL estimation errors between more or less 0 and -0.2 LSB. If σ_{ramp} is low, the standard deviation of the estimation error past $s/W_k = 0.75$ would become unacceptably large. This is because the large step sizes give a small number of steps per code for the measurement of W_k (such as 1, 2, or 3 steps per code), and lead to a large inconsistency of the estimation error. The mean value is then bound to increase as the step size becomes larger. However, as σ_{ramp} increases, the standard deviation would reduce, because the equivalent input noise now acts as a dither noise for large step values. However, it is excessively high for low step sizes, as the related standard deviation is now too large and would require a prohibitive number of averaged samples in order to provide an acceptable code width measurement.

3.2.2.3. Comparison

In this study of the step size variation with respect to σ_{ADC} and W_k, the performance of the proposed servo-loop technique is slightly below the performance of the classical servo-

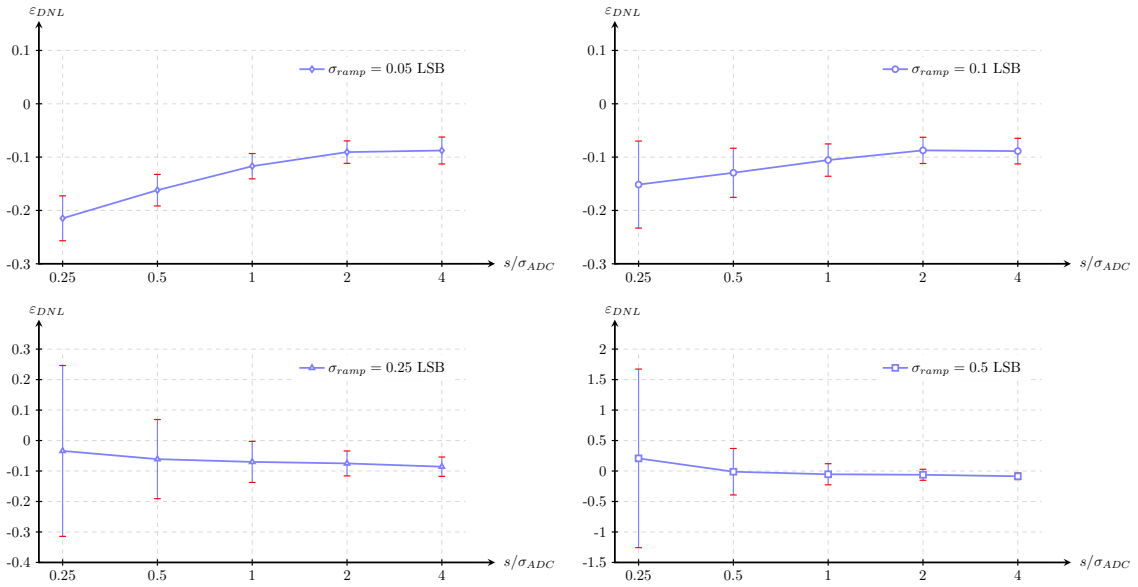


Figure 3.5: Mean and standard deviation of DNL estimation error with respect to the ratio of generator step size over the ADC input-referred noise magnitude using the proposed servo-loop technique

loop technique. Nevertheless, trade-offs are to be considered for acceptable DNL and INL estimation error. As shown in the results, a very small step size is clearly not advised for an accurate DNL estimation, and a larger step size must be considered (moderate step sizes) with respect to the code width, which corresponds to values such as 1 to 4 times the noise contribution of the ADCUT in our case study. This allows to relax the generator requirements with respect to its resolution. Another solution would be to increase the number of samples for noise averaging, but this could lead to impractical test times.

3.2.3. Accuracy-test time trade-off

In this subsection, the accuracy with respect to the test time for the measurement of code width W_k is assessed with the step size of the generator being varied with respect to the code width. The parameters fixed for this simulation are: $\sigma_{ADC} = 0.15$ LSB, code width W_k of 1 LSB, step size varying from 0.05 LSB to 0.75 LSB, 2000 runs per step size variation. Each measurement is performed for a number of samples that is dependent on the total noise contribution for a targeted accuracy. In order to assess the maximum limit of the accuracy for each test case, an iterative simulation is implemented. The initial accuracy on the DNL estimation error is fixed to 0.01 LSB. A limit of 1024 samples per code measurement is given for the simulation to reach the required accuracy. At fixed intervals, the simulation checks if the limit of the number of samples is crossed. If this condition is fulfilled, it means that the technique failed to reach the targeted accuracy.

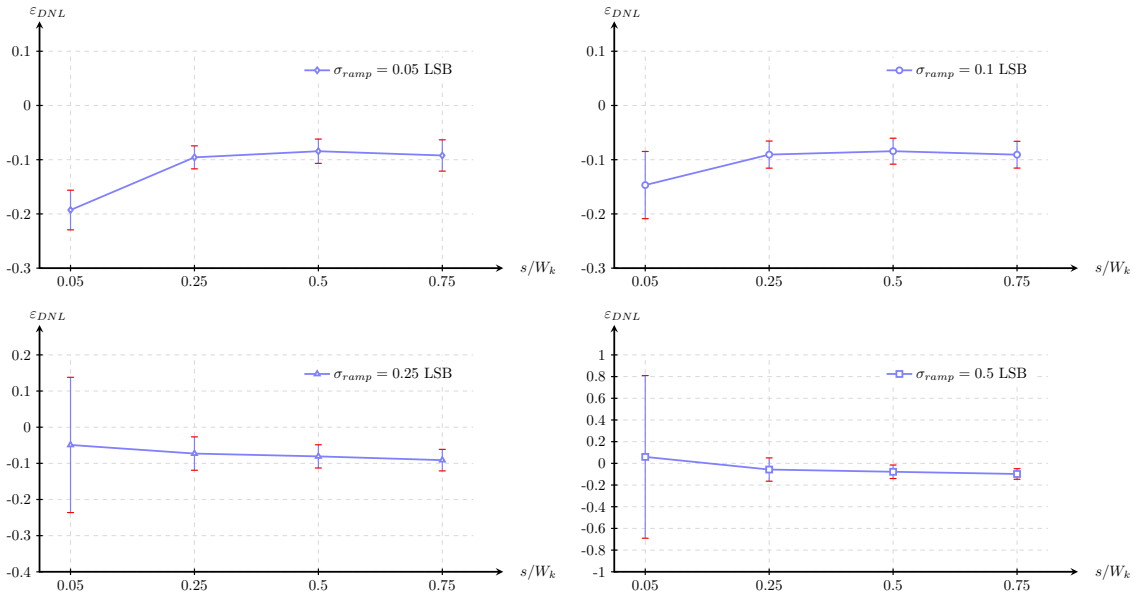


Figure 3.6: Mean and standard deviation of DNL estimation error with respect to the ratio of the generator step size over the LSB of the ADCUT using the proposed servo-loop technique

Consequently, the accuracy parameter is incremented by 0.01 LSB, and the simulation is run again and repeated until the simulation succeeds, meaning that the targeted accuracy is met within the 1024 samples allowed for the measurement.

3.2.3.1. Classical servo-loop technique

Assuming the simulation starts when the generator output is just below the transition voltage V_{t_k}, the counting of the number of samples is divided in three elements: the number of samples for the estimation of voltage V_{t_k}, the number of samples in order to attain voltage $V_{t_{k+1}}$, and the number of samples for the estimation of voltage $V_{t_{k+1}}$. The three counts are then summed in order to give the total number of samples for this measurement.

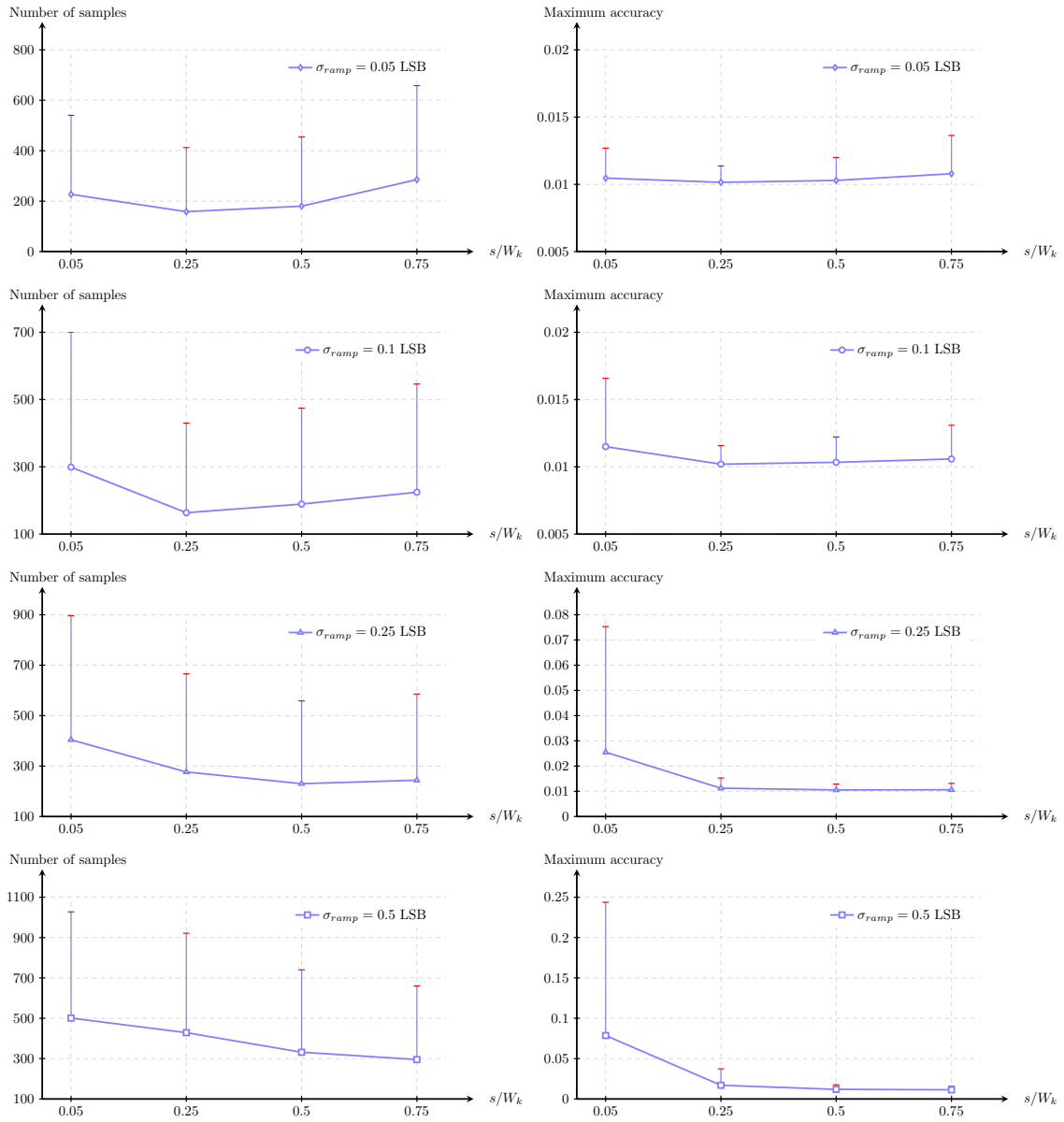


Figure 3.7: Mean value and standard deviation of the number of samples with respect to the ratio of the generator step size over the LSB of the ADCUT using the classical servo-loop technique

Figure 3.7 shows two columns of graphs. The left column represents the mean value and the standard deviation of the number of samples required for the minimum DNL estimation error with respect to ratio s/W_k, which corresponds to the maximum available test accuracy for this measurement. The right column shows the mean value and the standard deviation of the corresponding maximum achievable accuracy for the DNL estimation with respect to the ratio s/W_k. For instance, if we have a look at the two plots on the top row (i.e. $\sigma_{ramp} = 0.05$ LSB), the first data point at $s/W_k = 0.05$ means that a maximum accuracy of $\sim$0.01 LSB (data point in right side plot) can be achieved in the measure-

ment of a code width of $W_k = 1$ LSB using ∼200 samples for the measurement (data point in left side plot). For each figure, only the positive part of the standard deviation is shown because negative values are irrelevant for the number of samples and the maximum achievable accuracy.

Results in the left column of Figure 3.7 show that for low to moderate noise magnitudes, the number of samples remains low for small values of the step size. As the generator output noise becomes large, the trend is reversed and the lowest number of samples is achieved for large values of the step size. The standard deviation for small step sizes varies from ∼200 samples in the best case to more than 500 samples in the worst case. This is explainable by the fact that as the generator output noise remains small, a small step size is better in order to quickly estimate a transition voltage, whereas a large step size implies a less precise test with the same, and a longer test time is required in order to obtain the required accuracy. On the contrary, the effects are reversed for a large generator output noise because the noise magnitude becomes much higher than the small step size, requiring additional test time to meet the targeted accuracy, while a dither effect appears for larger step sizes, which reduces the test time for the transition voltage measurement.

Results in the right column of Figure 3.7 follow the same trend as in the left column. The lowest accuracy values are related to a low number of samples. For low values of σ_{ramp}, the accuracy is best if a small to moderate step size is used, with a maximum accuracy equal to ∼0.02 LSB. For large values of σ_{ramp}, the accuracy is best for moderate to large step sizes, and becomes prohibitively large when the test is performed using a small step size. For example, the maximum accuracy can be equal to 0.75 LSB for $s/W_k = 0.05$ ($s = 0.05$ LSB) and $\sigma_{ramp} = 0.5$ LSB. The accuracy remains good for moderate step size values.

3.2.3.2. Proposed servo-loop technique

Assuming the simulation starts when the generator output is just below the transition voltage V_{t_k}, the counting of the number of samples is divided in two elements: the number of samples for the measurement of code width W_k, and the number of samples in order to attain voltage $V_{t_{k+1}}$ for the measurement of code width W_{k+1}. The two counts are then summed in order to give the total number of samples for this measurement.

Figure 3.8 shows two columns of graphs. The left column shows the mean value and the standard deviation of the number of samples required for the minimum DNL estimation error with respect to ratio s/W_k, which corresponds to the maximum available test accuracy for this measurement. The right column of Figure 3.8 shows the mean value

and the standard deviation of the corresponding maximum achievable accuracy for the DNL estimation with respect to the ratio s/W_k. For each figure, only the positive part of the standard deviation is shown because negative values are irrelevant for the number of samples and the maximum achievable accuracy.

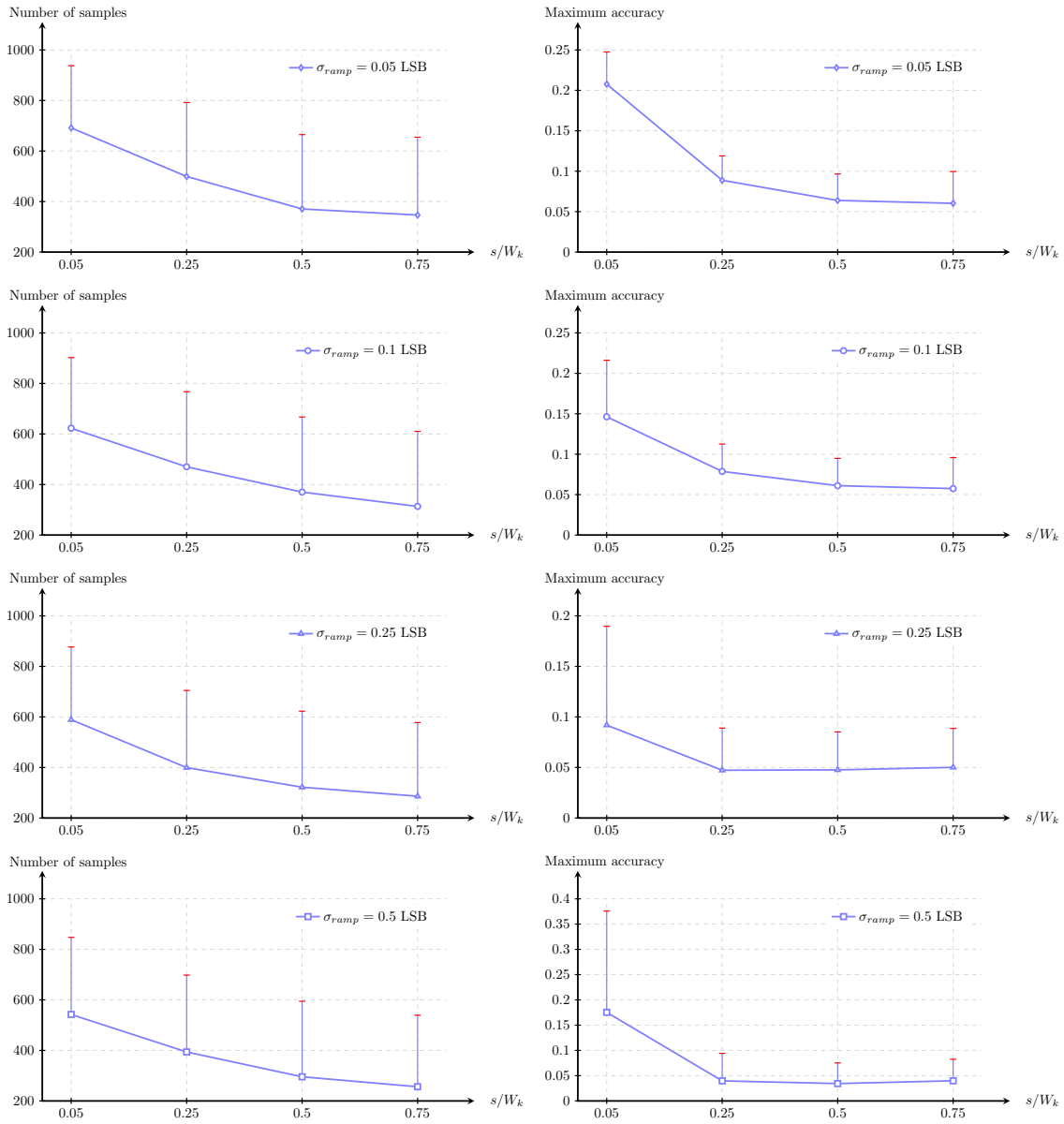


Figure 3.8: Mean value and standard deviation of the number of samples with respect to the ratio of the generator step size over the LSB of the ADCUT using the proposed servo-loop technique

Results in the left column of Figure 3.8 show a decreasing trend for each σ_{ramp} value. The mean value of the number of samples goes from $580-700$ for small step size values to $\sim$200 for the ratio $s/W_k = 0.75$. The standard deviation remains constant with $\sim$250 samples for all ratio values. This shows that the level of noise at the generator output

has a small impact on the test time variability. This is explainable by the fact that as the generator output noise remains small, a small step size is better in order to quickly estimate a transition voltage, whereas a large step size implies a less precise test with the same noise level, and a longer test time is required in order to obtain the required accuracy. On the contrary, the effects are reversed for a large generator output noise because the noise magnitude becomes much higher than the small step size, requiring additional test time to meet the targeted accuracy, while a dither effect appears for larger step sizes, which reduces the test time for the transition voltage measurement.

In the right column of Figure 3.8, for any σ_{ramp} value, the maximum achievable accuracy remains more or less constant for moderate step sizes (0.25 LSB to 0.75 LSB), with a mean value around 0.05 LSB with a standard deviation of $\sim$0.05 LSB as well, which correspond to a maximum accuracy of 0.2 LSB at $\pm 3\ \sigma$ level, that is the ideal DNL estimation error. As shown in the left column of Figure 3.8, low step size values lead to the highest test time, and this corresponds to the worst maximum achievable accuracy in the right column. This is explained by the fact that the requirement to achieve a given accuracy for the fixed number of samples used for this simulation is too high and a small step size requires a larger number of samples for a better DNL estimation.

3.2.3.3. Comparison

The proposed servo-loop technique is able to estimate a code width in an equal to lower test time in comparison to the classical servo-loop. The maximum achievable accuracy is worse than the one in the classical case, but is still very acceptable for moderate step sizes ($\pm 3\ \sigma$ error of 0.2 LSB). For best accuracy, it is then advised to select a moderate step size to relax the requirements of the ramp generator, to increase the number of averaged samples, or to design the generator with extra care on reducing the output noise contribution.

3.2.4. INL estimation

The modified technique is assessed with respect to the INL estimation of the ADCUT in this subsection. For this purpose, 1000 11-bit, 2 V FSR generic ADCs are generated using a parameterized model. The standard deviation of the ADC input-referred noise is fixed to $\sigma_{ADC} = 0.15$ LSB, which corresponds to a $\pm 3\ \sigma_{ADC}$ variation of ± 0.45 LSB. The maximum swing of the generator output noise is chosen to be ± 3 times its step size. Fixing the step size of the generator to $s = 0.1$ LSB, the standard deviation of the generator

output noise is $\sigma_{ramp} = 0.1$ LSB. For each of the 1000 runs, 1024 samples are collected for each ADC code measurement. The parameters of the ADC model are randomly varied giving an INL ranging from 0.2 LSB to 4.3 LSB. The INL of the 1000 ADCs used in the simulation cover a large range of values, and most would not pass acceptance criteria for marketable 11-bit ADCs. However, this simulation is useful to assess the ability of the proposed technique to estimate the INL of good and bad ADCs.

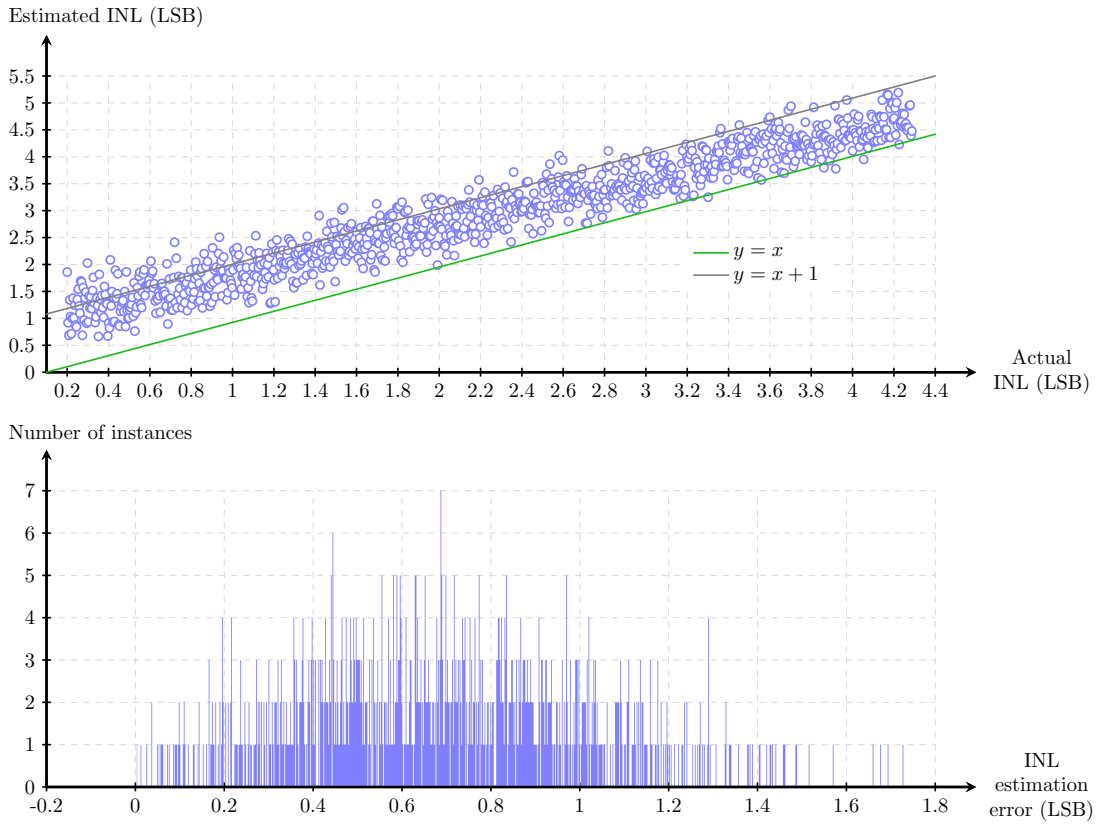


Figure 3.9: Estimated INL versus actual INL and respective histograms of the INL estimation error for 1000 ADC/ramp generator pairs using the proposed servo-loop technique without RCLT

Figure 3.9 shows two graphs. The top graph plots the estimated INL value of the tested ADCs versus their actual INL value. The green line represents the ideal $y = x$ line, and the grey line represents the 1 LSB INL estimation error line. The bottom graph is an histogram of the INL estimation error defined as the maximum absolute difference between the estimated INL and the actual INL. Each estimated INL curve is corrected using the gain and offset correction algorithm from [1].

Results show that while the actual INL ramps from 0.2 LSB to 4.3 LSB, the estimated INL varies from 0.55 LSB to 5.4 LSB, and the INL estimation error varies from 0 LSB to 1.73 LSB. The average INL estimation error is 0.68 LSB. 165 of the 1000 ADC/ramp

generator pairs have been measured with an INL estimation error superior to 1 LSB, and 5 over 1000 have been measured with an INL estimation error below 0 LSB.

3.3. Reduced-code testing techniques for pipeline ADCs

This section first details the architecture and functioning of a pipeline ADC, then develops on the reduced-code linearity techniques specifically developed for this type of ADC.

3.3.1. Pipeline ADC overview

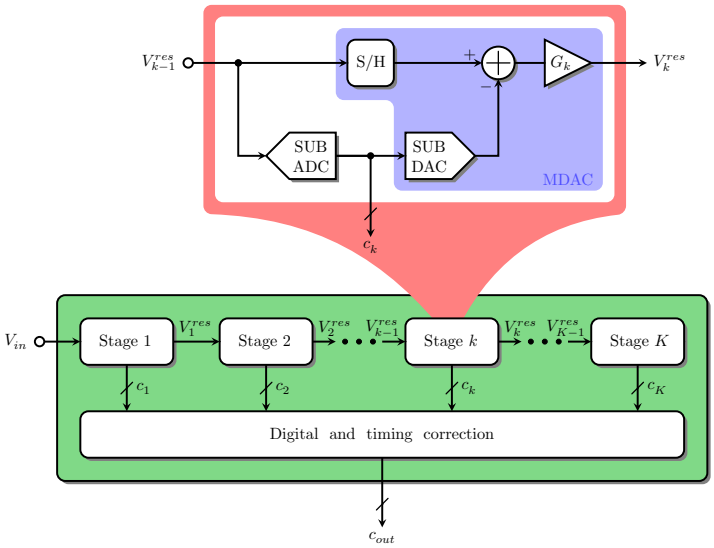
The basic principle underlying the Pipeline ADC is the estimation of a digital representation of the analog input by means of the application of a recursive algorithm [**?**]. Such ADC is a good candidate for high-resolution, high-speed applications.

3.3.1.1. Architecture

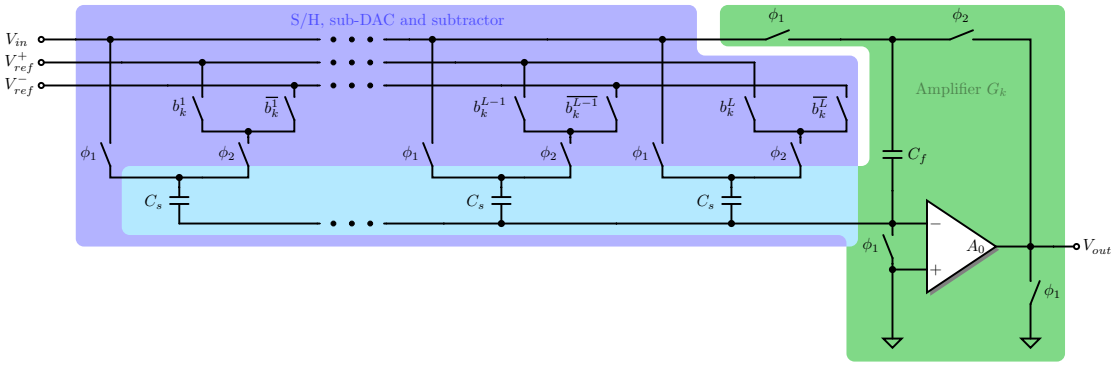
This converter is composed of a cascade of several similar blocks, called stages. A very simple model of each basic block is portrayed in Figure 3.10 (a). Each stage contains a sample-and-hold amplifier (SHA), a low-resolution sub-ADC, a low-resolution sub-DAC, a subtractor, and an amplifier. At the SHA sampling rate, each stage sequentially resolves a part of the final ADC resolution. Each sub-ADC gives a low-resolution digital value of the analog input. This digital result is fed to the sub-DAC, which outputs a coarse version of the analog input. The two voltages are then subtracted between them, and the amplifier scales the residue to the FSR of the pipeline ADC. The resulting analog value V_k^{res} is then fed to the next stage for similar operation. When all the stages have been exercised, the digital outputs of all stages are combined through a digital and timing correction block which produces the final high-resolution output code.

Generally, in the practical case, the sub-DAC, the subtractor and the amplifier are combined into a multiplying DAC (MDAC), as seen in Figure 3.10 (a), which performs all those operations.

Let's focus first on the nominal functioning of the pipeline ADC. As previously stated, the $K-1$ stages are identical and the $K-th$ stage is only a low-resolution ADC. They generate a low-resolution digital output c_k ($k = 1, ..., K$) of n_k bits, that defines the resolution of the sub-ADC and the sub-DAC of each stage. The sub-ADC is usually a simple flash ADC. A bank of L comparators, one per ADC transition, outputs a set of bits $\{b_k^l\}_{l=1,...,L}$



(a) N-bit, K-stages pipeline ADC architecture



(b) Multiplying-DAC architecture with simple sub-code thermometer codification

Figure 3.10: Architecture of an N-bit, K-stage pipeline ADC

and their complementary bits $\{\overline{b_k^l}\}_{l=1,\ldots,L}$. They are the results of the comparison of the analog input to a set of reference transitions $\{t_{id}^l\}_{l=1,\ldots,L}$, and represents the magnitude of the analog input as a thermometer code. Then a thermometer to binary encoder converts this set to a binary-format digital output c_k. The total number of transitions L for stage k is expressed as

$$L_k = 2^{n_k} - 1, \quad k = 1,2,\ldots K. \tag{3.14}$$

The MDAC of each stage, as seen in the singled-ended case in Figure 3.10 (b), has a switched-capacitor implementation in order to perform the sample and hold operation. It consists of two phases. In sampling phase ϕ_1, the L sampling capacitors C_s and the feedback capacitor C_f are charged to voltage V_{in}. In hold phase ϕ_2, the sampling capacitors are now connected to V_{ref}^+ and V_{ref}^- via the set of switches $\{b_k^l\}_{l=1,\ldots,L}$, ensuring the digital-to-

analog operation, and the feedback capacitor is connected to the output V_{out}, ensuring the amplification in closed loop. If we take $V_{k-1}^{res} = V_{in}$, $V_k^{res} = V_{out}$ and $V_{ref}^{+} = -V_{ref}^{-} = V_{ref}$, the transfer function of the MDAC can be expressed as

$$V_k^{res} = \frac{L_k \cdot C_s + C_f}{C_f} \cdot V_{k-1}^{res} - \frac{C_s}{C_f} \cdot \sum_{l=1}^{L_k} (b_k^l - \overline{b_k^l}) \cdot V_{ref}, \quad k = 1,2,...K \tag{3.15}$$

where $\{b_k^l\}_{l=1,...,L}$ and $\{\overline{b_k^l}\}_{l=1,...,L}$ are the same bits from the sub-ADC thermometric output that also activate the switches of the MDAC. In this way, the stages are synchronized since if stage k is in sampling mode, stage $k+1$ is in hold mode and no clock cycle is lost. Expression (3.15) can be re-evaluated as

$$V_k^{res} = G_k \cdot (V_{k-1}^{res} - V_k^{DAC}), \quad k = 1,2,...K \tag{3.16}$$

with

$$\nu_f = \frac{C_s}{C_f} \tag{3.17}$$

$$G_k = 1 + \nu_f \cdot L_k = 2^{n_k} \tag{3.18}$$

$$V_k^{DAC} = \frac{c_k}{1 + \nu_f \cdot L_k} \cdot V_{ref}. \tag{3.19}$$

In this case, $\nu_f = 1$, meaning $C_s = C_f$ in order to satisfy the equation.

The pipeline ADC makes available the sub-codes $\{c_k\}_{k=1,...,K}$ of all the stages within the same clock cycle by the means of time alignment. It consists in cascading delay blocks where the size of these stacks depends on the number of stages and is inversely proportional to the cumulative delay induced by each stage. The sub-codes are then combined using a weighted bit-shifting logic scheme by the means of bit shifters. The first stage sub-code is shifted with the highest weight and represents the Most Significant Bits (MSB) of the final output code c_{out} while the last stage sub-code stays unshifted and represents its LSBs. The total number of bits in the final code, namely the pipeline ADC resolution, is expressed as

$$N = \sum_{k=1}^{K} n_k \tag{3.20}$$

which is the length of the concatenation of all the sub-codes. From this point, the output code c_{out} is described in this case as

$$c_{out} = \sum_{k=1}^{K-1} 2^{w_k} \cdot c_k + c_K \tag{3.21}$$

with the weight attributed to each stage expressed as

$$\begin{aligned} w_k &= N - \sum_{j=1}^{k} n_j \\ &= \sum_{j=k+1}^{K} n_j, \quad k = 1,...,K-1 \end{aligned} \tag{3.22}$$

where the sub-codes and the final code can either be expressed with their decimal values or in binary codification.

3.3.1.2. Digital correction with redundancy

Unfortunately, in a real implementation, the nominal case does not apply since different sources of errors degrade the stage performance. Figure 3.11 details the different cases of the sub-ADC and sub-DAC nonlinear errors with their effect on the residue output. The errors are shown on a given transition voltage t_l of the sub-ADC. The sub-ADC exhibits errors due to offsets in its comparators, which leads to displacements of the transitions points with respect the ideal values, as depicted in 3.11 (b). In such a case, if a wrong decision of the output code c_k is made, then the resultant wrong V_{res}^k is likely to cause the next stage to make a wrong decision, with the same pattern going on until the last stage. In the MDAC, the op-amp is affected by the finite op-amp gain and its input offset. Moreover, capacitors are affected by mismatch and charge injection appears due to the non-idealities of the feedback switch. Capacitor mismatch and the finite op-amp gain can introduce an interstage gain error, as shown in Figure 3.11 (c) and in Figure 3.11 (d), charge injection error causes the transfer curve to shift vertically.

Equation (3.16) can be rewritten to a first order as

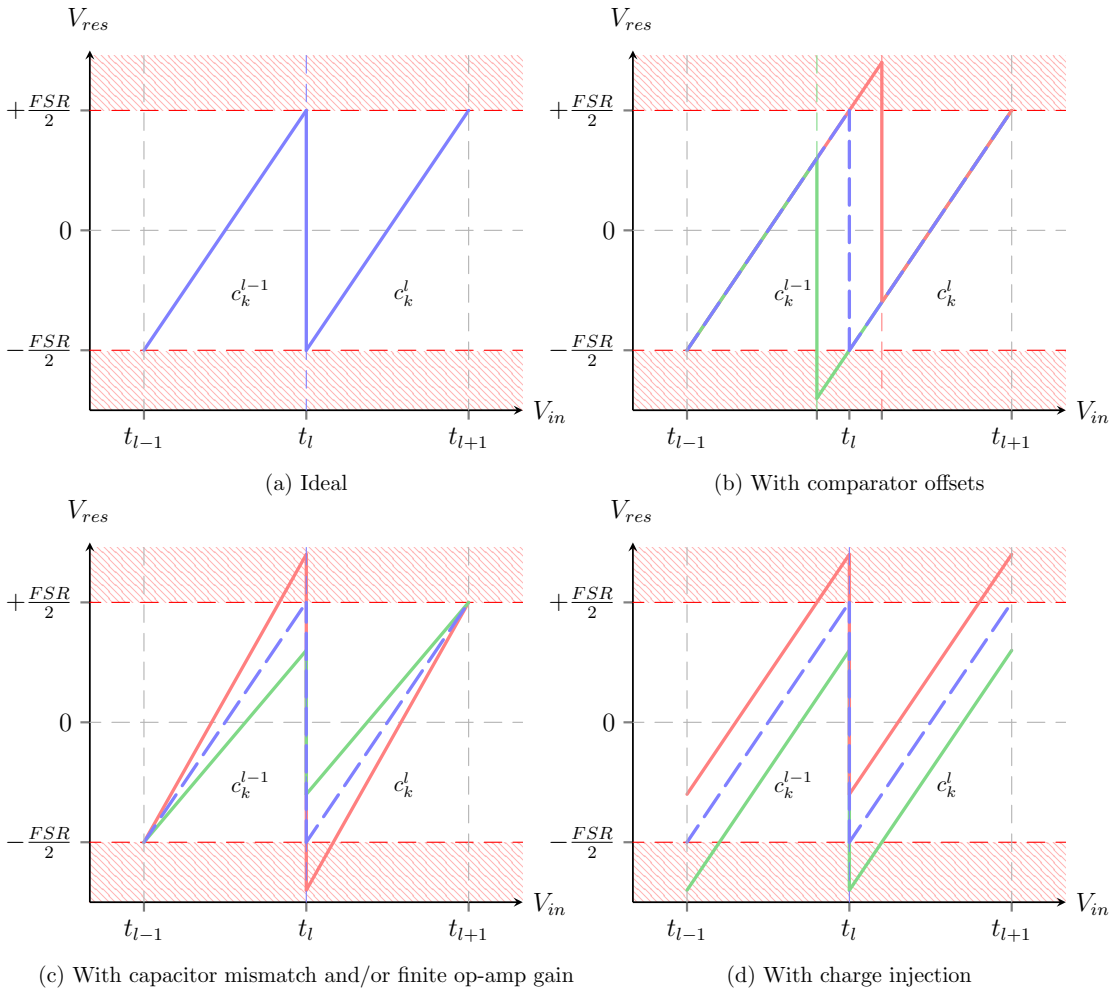
$$V_k^{res} = G_k' \cdot (V_{k-1}^{res} - V_k^{DAC}) + V_{os}(k), \quad k = 1,2,...K \tag{3.23}$$

with

$$G_k' = (1 + \nu_f \cdot L_k) \cdot (1 - e^{-t/\tau}) \cdot \left(\frac{1}{1 + \frac{1}{A_0 \cdot f}} \right) \tag{3.24}$$

$$\tau = \frac{1}{GBW} \tag{3.25}$$

$$f = \frac{C_f}{L_k \cdot C_s + C_f + C_p} \tag{3.26}$$



(a) Ideal

(b) With comparator offsets

(c) With capacitor mismatch and/or finite op-amp gain

(d) With charge injection

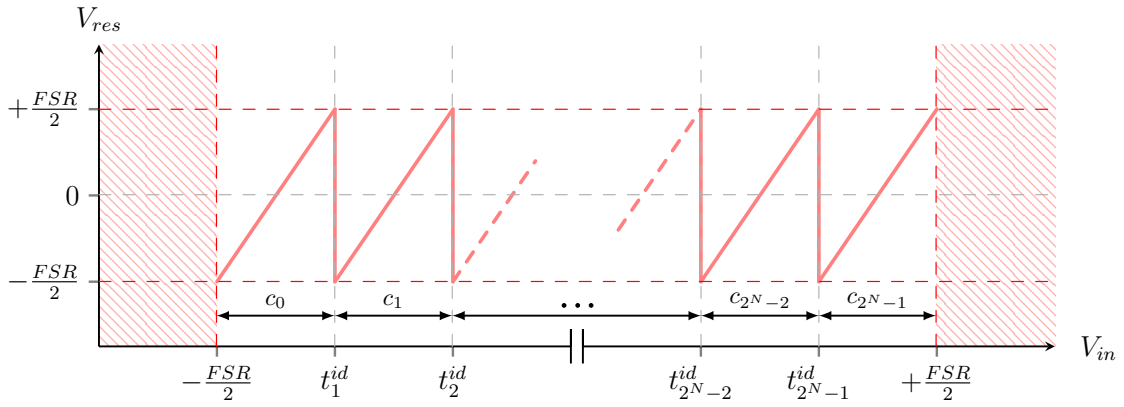
Figure 3.11: Transfer function of a pipeline stage when looking at transition t_l

where GBW is the gain bandwidth of the MDAC and C_p is the parasite capacitor at the minus node of the op-amp due to the ϕ_1 switch.

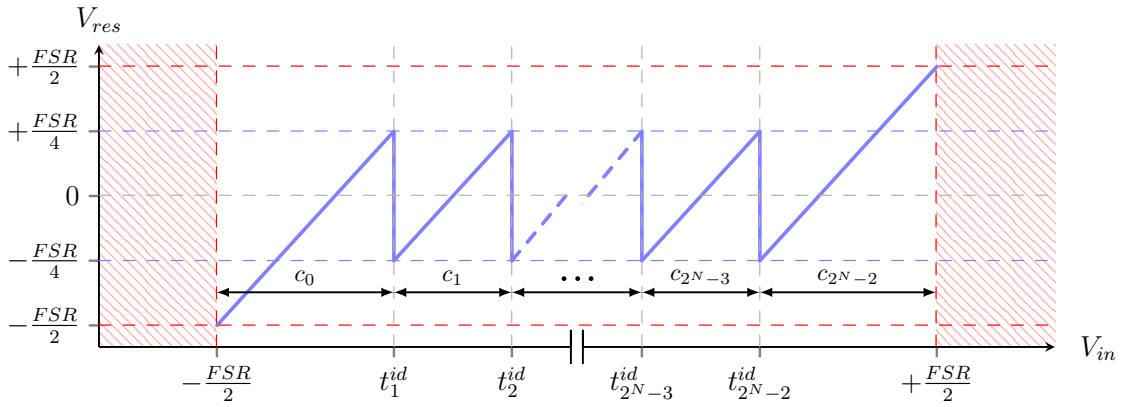
As observed in Figure 3.11, each error leads to an analog overranging of the MDAC output swing (being the ADC FSR), which translates into missing codes in the final code c_{out}. A correction is then mandatory in order to counter those unwanted effects. A digital correction is usually performed, relaxing the requirements on the sub-ADC comparators with respect to their offset, while ensuring the nominal static performance of the pipeline ADC.

This technique first consists in reducing the interstage analog gain G_k with respect to its original value. This prevents the residue voltage V_{res}^k to overrange the FSR. The reduced value of G_k is chosen so that its original value is divided by a factor, namely the scaling factor S_r, defined as

$$S_r = 2^r \tag{3.27}$$



(a) Without redundancy



(b) With redundancy

Figure 3.12: Ideal pipeline stage input-output transfer function

where r is an integer number greater than 1 and is designated as the redundancy factor. Dividing equation (3.18) by equation (3.27), the new gain is then defined as

$$G_k = 2^{n_k - r}. \tag{3.28}$$

Thus the input-output relationship is now described as

$$V_{res}^k = \left\{ \begin{array}{ll} G_k \cdot V_{res}^{k-1} & if \quad t_{l,k} < V_{res}^{k-1} \leqslant t_{l+1,k} \\ G_k \cdot (V_{res}^{k-1} - V_k^{DAC}/S_r) & otherwise \end{array} \right\}, \quad l = 1,2,...,L_k - 1 \tag{3.29}$$

where V_k^{DAC} has a different value depending on the sub-code value c_k^l.

The scaling factor is selected as a multiple of 2 so that $n_k - r \geqslant 0$. This not only provides an analog amplification greater than 1, but also allows hardware simplification. Indeed, any other value other than a multiple of 2 would induce extra hardware on the analog gain

block as well as the digital correction block, increasing the complexity of the circuit and dampening the interest for the correction technique.

Secondly, a digital gain correction is introduced at no extra cost for the original digital block. The modification only consists in modifying the weights of its bit shifters for each stage sub-code. As pictured in Figure 3.13 (b), the weights of the digital gains are reduced by the scaling factor S_r, allowing the redundancy scheme to operate. In this example, the sub-codes c_{K-1} and c_K are overlapped so that the two last bits of c_{K-1} are summed to the two first bits of c_K. Subsequently, the corrected final output code c_{out} is expressed as

$$c_{out} = \sum_{k=1}^{K} 2^{w'_k} \cdot c_k \tag{3.30}$$

with the corrected weight attributed to each stage expressed as

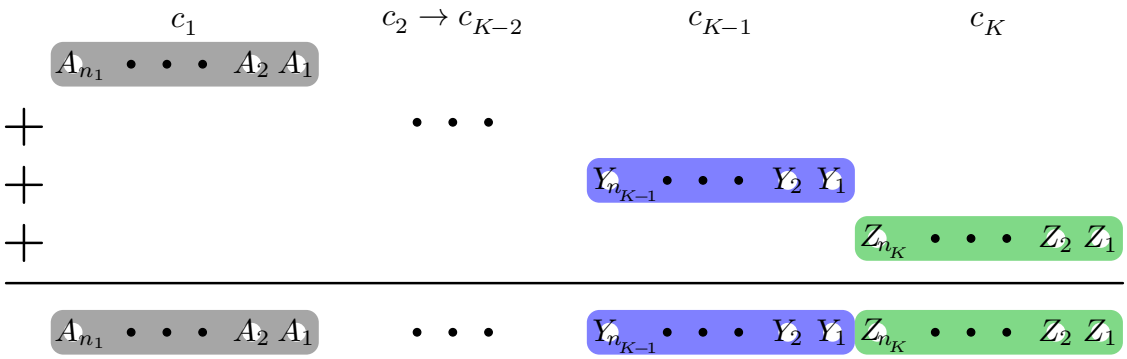
$$\begin{aligned} w'_k &= N - \sum_{j=1}^{k} n_j - (K-k) \cdot r \\ &= \sum_{j=1}^{K} (n_j - r) - \sum_{j=1}^{k} (n_j - r) \\ &= \sum_{j=k+1}^{K} (n_j - r), \quad k = 1, ..., K-1. \end{aligned} \tag{3.31}$$

From equation (3.30), it can be observed that the number of bits of final output code c_{out} is dependent on the number of redundant bits that are overlapped. Its resolution is now expressed as

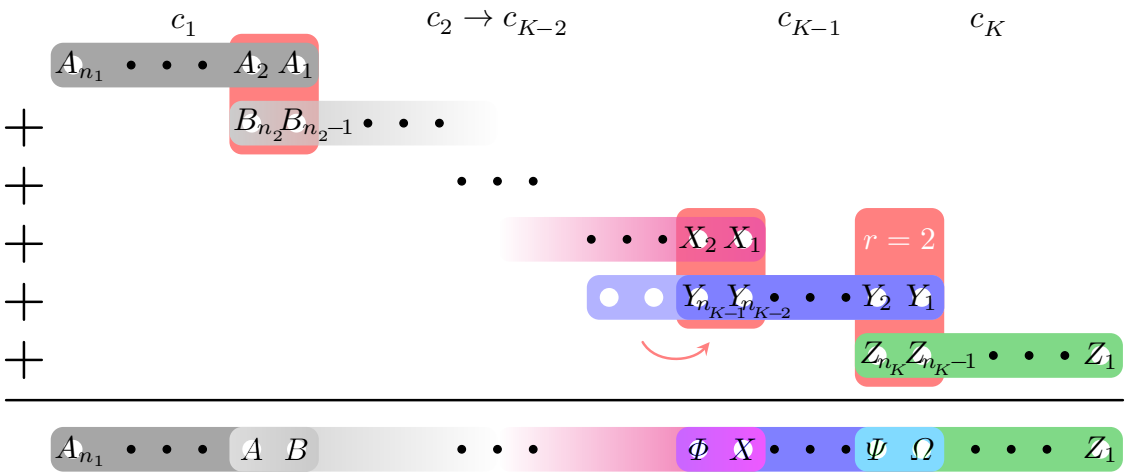
$$N_r = \sum_{k=1}^{K-1} (n_k - r) + n_K \tag{3.32}$$

showing that the final resolution is reduced by r bits for each additional stage in the pipeline ADC.

Through this digital correction, larger comparator offsets and gain errors can be tolerated before the residue overranges the ADC FSR. Nevertheless, the value of r must be chosen with care. Values of $r > 1$ lead to an increase of the number of stages in order to keep the same ADC resolution. Moreover, the increase of the number of stages implies an increase in the power consumption and the number of errors sources as well as a decrease in the ADC static performance. Using unary redundancy ($r = 1$) reduces by half the maximum excursion of the voltage residue V_{res}^k at the input of stage $k+1$. It is sufficient to deal with the nonlinear errors of the pipeline stages, thus preventing it from overranging



(a) Construction of output word c_{out} without redundancy



(b) Construction of output word c_{out} with redundancy ($r = 2$)

Figure 3.13: Digital correction logic scheme using redundancy

each stage FSR. For all those reasons, unary redundancy is often employed for this digital correction.

The redundancy scheme also allows a reduction of the hardware area. Indeed, the number of comparators in the sub-ADC can be reduced while keeping the same ADC performance. Note however that in some designs, the last stage is not affected by the hardware reduction in order to keep the exact number of output codes, which is one code short if applied to the last stage. The number of transitions L for stage k is now expressed as

$$L_k = 2^{n_k} - 2, \quad k = 1, 2, \ldots K. \tag{3.33}$$

This reduction in the number of comparators comes with a new mapping of the remaining

transitions in the sub-ADC. In the nominal case, the transitions are located at voltages

$$\begin{aligned}\{t_l^{id}\}_k &= \frac{FSR}{2^{n_k}} \cdot \left(l + \sum_{i=0}^{r} \frac{i}{2^i} \right) \\ &= \frac{FSR}{2^{n_k}} \cdot \left(l + 2 - \frac{r+2}{2^r} \right), \quad k = 1,2,...K, \quad l = 1,2,...L_k, \quad r = 0,1,...,n_k.\end{aligned} \tag{3.34}$$

which consists in the L_k transitions evenly distributed on the FSR of stage k, shifted by a fraction of the stage LSB depending on the value of the redundancy r.

3.3.2. Application of the RCLT to pipeline ADCs

3.3.2.1. Reduced code linearity testing principle

The repetitive structure of a pipeline ADC means that when traversing the input dynamic range of the converter, a comparator in a given pipeline stage is exercised several times (*i.e.* its threshold is crossed) for different ADC output transitions in between two consecutive codes. This property leads to two immediate statements. First, it means that we can establish groups of transitions that are due to the exercise of the same comparator of a particular pipeline stage. Secondly, it means that the errors (due to process variations) of this particular comparator will be spread to the group of related ADC transitions. Any nonideality in a pipeline stage (i.e. finite op-amp gain, capacitor mismatch, op-amp offset, sub-ADC comparators offset, etc.) is translated to a DNL error of all ADC output codes that involve an output transition that is due to the same comparator being exercised in this pipeline stage.

The reduced-code linearity testing method for pipeline ADCs achieves significant static test time reduction through the exploitation of this inherent property. By testing a reduced set of carefully chosen transitions covering every comparator of each pipeline stage, it is possible to evaluate the whole pipeline ADC nonlinearity. Each comparator in each pipeline stage needs to be represented only once in this set. The widths of the codes around each transition of the set are measured. The measured code widths are then attributed to the codes around the transitions that belong to the same group and were not initially selected. By applying this technique to all the groups of transitions, the whole ADC transfer function is covered and its nonlinearity can be evaluated.

This is illustrated by the transfer characteristic in Figure 3.14, where t_{ik} corresponds

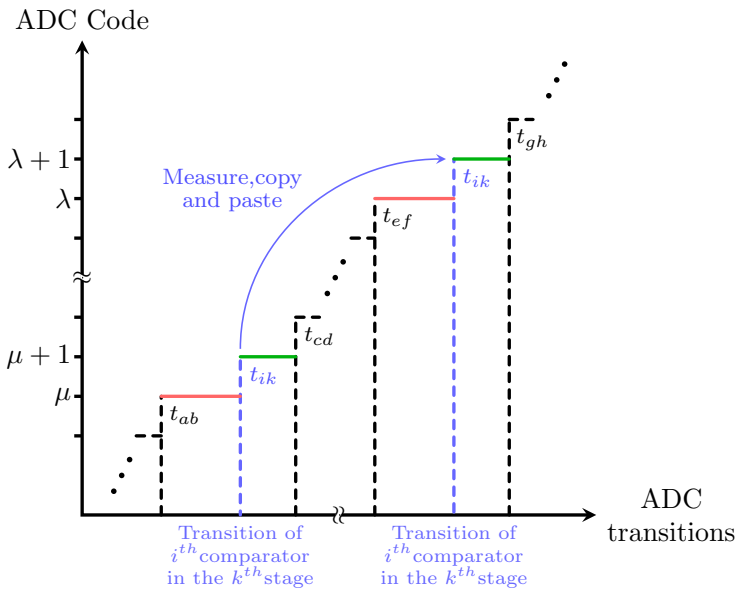


Figure 3.14: Reduced-code testing technique principle

to an output transition that is due to the i-th comparator in the k-th pipeline stage being exercised.

An ADC output code shares two adjacent ADC output transitions that involve two different comparators, possibly belonging to two different pipeline stages, as shown in the example of Figure 3.14. The stage that is closer to the front of the pipeline will dominate the DNL error that is finally produced. In the example of Figure 3.14, let us assume that the k-th stage is closer to the front of the pipeline compared to the b-th, d-th, f-th, and h-th stages. This means that the width of codes μ, $\mu+1$, λ, and $\lambda+1$ are principally affected by the nonidealities in the k-th stage. Furthermore, it means that the widths of the codes λ and $\lambda+1$ are practically equal to the widths of the codes μ and $\mu+1$, respectively. Thus, we need to measure the width of either μ or λ and the width of either $\mu+1$ or $\lambda+1$. Extending this argument, let us assume that we know the mapping between the transitions in the ADC output and the comparators in the pipeline stage that are being exercised to produce these transitions. If we measure only the codes around a representative set of ADC output transitions such that this set covers all comparators in all stages and each comparator is represented once in this set, then, by relying on the mapping, we can readily assign values to the widths of unmeasured codes around the unselected ADC output transitions. In other words, we measure a reduced number of codes in the histogram and we fill in the rest of the histogram automatically by relying on the information in the extracted mapping.

With regard to static test time, it is mainly governed by the transfer time of data from the ADC under test to the memory of the ATE and from the ATE to the workstation where the

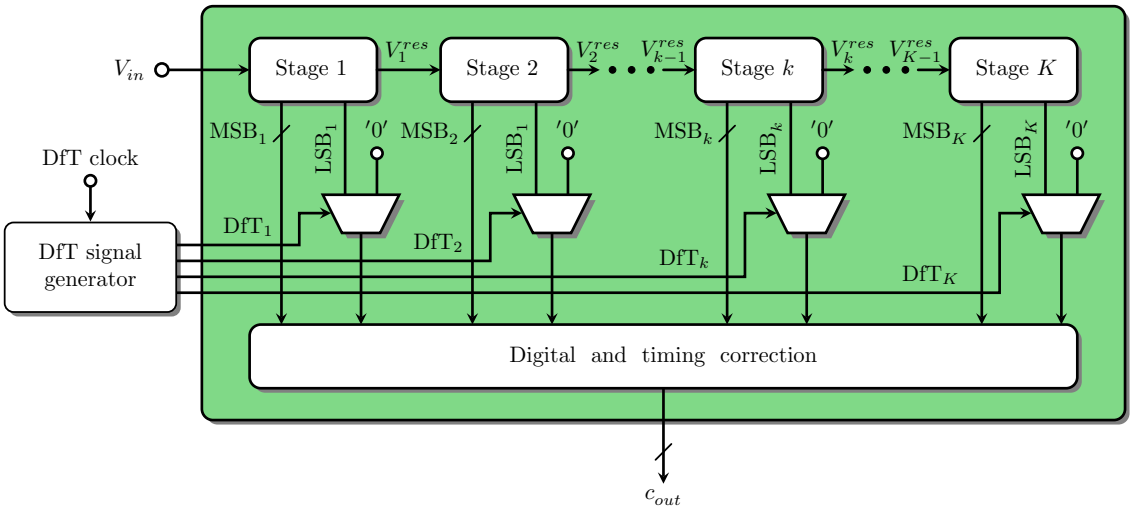


Figure 3.15: Transition-code based BIST method for pipeline ADCs [37]

data will be processed for constructing the histogram. During static test the excitation of dynamic phenomena should be avoided. This imposes strict limitations to the frequency of the test stimulus, which has to be necessarily slow to ensure static operation. Moreover, static test is very sensitive to noise, which makes it necessary to average the measurements across multiple runs. In a standard histogram test, these requirements translate to using a slow ramp as test stimulus (a typical slope may be around 100 hits/code) and applying this ramp multiple times to average noise (typically around 100 full-scale ramps are applied). As an example, for an 11-bit ADC, these conditions would produce $N = 2048 \times 10^4$ 11-bit values to be read out. Thus, in the case of the reduced code linearity test technique, if we measure $X\%$ of the codes, which compared to the standard histogram technique translates into measuring only $X\%$ of the data, then we drastically reduce the static test time. Overall, we expect to have a test time reduction slightly below $(100 - X)\%$ given that compared with the standard histogram technique we have the extra step of deriving the mapping. The derivation of the mapping needs to be carried on a chip-by-chip basis because the mapping may change due to process variations.

3.3.2.2. Reduced code linearity test technique

The method in [35], [36] and [37] details a reduced code linearity testing technique aimed at reducing the test time of pipeline ADCs. A conceptual block diagram of the technique is represented in Figure 3.15.

The proposed technique consists in forcing to '0' state the LSB of the digital output of a given pipeline stage in order to pinpoint the transitions related to each comparator in this same stage. It translates to apparent jumps in the ADC transfer function at the

transitions of the comparators of the stage under observation. Using the additional BIST circuitry, when one of the DfT_k, $k = 1,2,...,K$ (generated by a ring counter) is active, the corresponding multiplexer puts the LSB of the k-th stage to '0', as depicted in Figure 3.15. If no DfT signals are active, then the ADC is in nominal operation mode.

Next, a mapping of the expected distribution of the comparator transitions is performed. The maximum offset variation for each transition in each stage is measured. Those maximum values are determined so that the digital correction is still effective, and when the other nonidealities are neglected. When considering all error sources, the variation window for each transition becomes more or less large with respect to its initial ideal state. In order to deal with the gain errors, a large number of digital codes around the selected transition code are tested at the cost of a longer test time. The noise contribution is reduced by taking a higher number of samples per measured code. Measurements are focused on the offset of each comparator. If only the comparator offset error is taken into account, then each selected transition is a transition whose offset window does not overlap with those of the comparators in a previous stage.

However, these conditions do not hold in reality, and even more because of the effect of other error sources, such as noise. The demonstration of this method failing to correctly select the required codes has been shown in [38]. First, a 12-bit, 11-stage, 1.5-bit/stage pipeline ADC model is considered for the demonstration. The given model static performance is degraded with gain errors, comparators offsets and noise. It is shown that the transition of the second comparator in the first stage overlaps with the last transition in the second stage, proving that the selection principle is erroneous. Then a 12-bit, 2.5-bit/stage pipeline ADC model is considered, which has smaller offset window tolerance compared to the previous model. In this case, it is shown that the proposed method, when forcing the LSB of the output of the stage to '0', blindly selects the transitions and then associates them to what is thought to be the corresponding comparators, as it relies on an expected order of comparators transitions to perform the mapping. However, the real mapping of the comparators depends on the combination of error sources in the pipeline stages, and cannot be predetermined, also proving that this method is flawed. The interested reader may refer to [38] for further explanation.

3.3.2.3. Enhanced noise-insensitive technique

From the above discussion in section 3.3.2.1, it is clear that deriving an accurate mapping is a crucial step in the method. For this purpose, it has been proposed to monitor directly the outputs of the pipeline stages [38],[39] and to cancel out the noise in the transitions

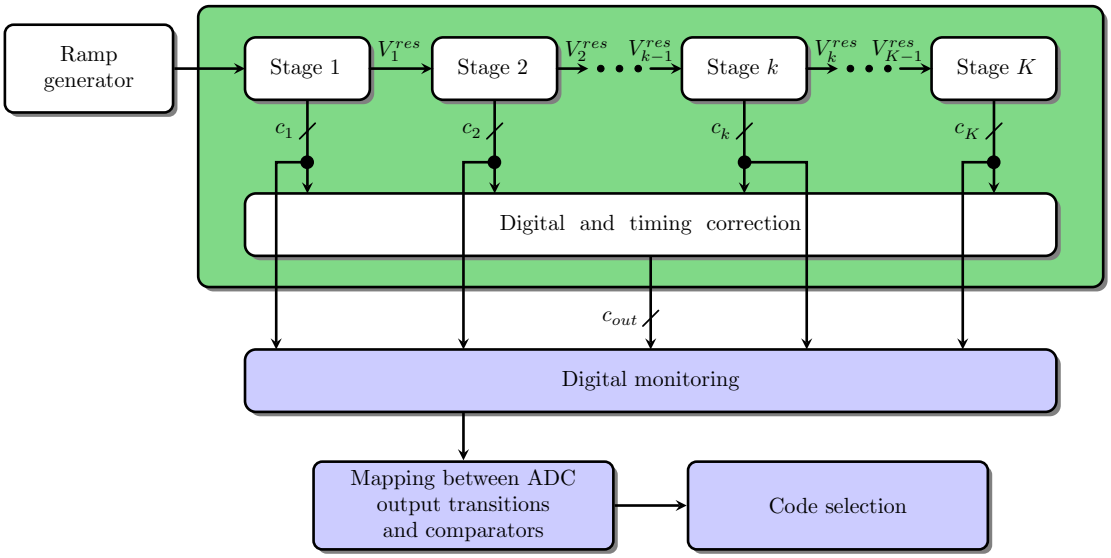


Figure 3.16: Digital monitoring of the digital outputs of the pipeline stages

of the pipeline stages [40], [41]. For the sake of completeness, we reproduce here the basics of the enhanced noise insensitive reduced code linearity test technique that has been developed in our lab and originally proposed in [38]. The interested reader is referred to [38]−[41] for a detailed analysis of the technique.

Two important recommendations are to be kept in mind in order to properly estimate the DNL and INL of the ADCUT. First, we must ensure that each selected transition is mapped to the right comparator. Secondly, avoiding selecting an ADC output transition for a comparator in a given target stage that involves in addition a comparator in one of the previous stages so as not to overshadow the DNL error produced by the target stage.

Unlike [35]−[37], the proposed method here consists in finding the comparators transitions without a predetermined mapping. In this method, each stage output codes are monitored before the digital correction block, as seen in Figure 3.16. Each time a comparator in a stage is exercised, its output transitions from the actual code to a new one. This transition gives a hint on which comparator has been exercised. Moreover, we can correlate the change in the output code of a given stage c_k with the change in the total output code c_{out} of the pipeline ADC, as performed by the digital correction block.

We define a natural transition as a transition of the digital output of a stage to the next value, due a comparator being exercised in this stage. On the contrary, we define a forced transition as a transition that is not due to a comparator being exercised in this stage.

When an increasing (resp. decreasing) ramp stimulus is fed to the pipeline ADC, if the digital output of a stage is increasing (resp. decreasing) one code at a time, then it corresponds to a natural transition. If the digital output of a stage is decreasing (resp.

Table 3.4: Possible transitions in a 2.5-bit pipeline stage

$c_k(n)/c_k(n-1)$	'000'	'001'	'010'	'011'	'100'	'101'	'110'
'000'		F_{10}	F_{20}	F_{30}	F_{40}	F_{50}	F_{60}
'001'	N_1		F_{21}	F_{31}	F_{41}	F_{51}	F_{61}
'010'	F_{02}	N_2		F_{32}	F_{42}	F_{52}	F_{62}
'011'	F_{03}	F_{13}	N_3		F_{43}	F_{53}	F_{63}
'100'	F_{04}	F_{14}	F_{24}	N_4		F_{54}	F_{64}
'101'	F_{05}	F_{15}	F_{25}	F_{35}	N_5		F_{65}
'110'	F_{06}	F_{16}	F_{26}	F_{36}	F_{46}	N_6	

increasing), or increasing (resp. decreasing) more than one code at a time, then it corresponds to a forced transition.

A forced transition in a given stage k is in fact due to the behavior of the previous stage $k-1$. If a natural transition happens in stage $k-1$, then a forced transition is observed at stage k output. This is explained by the fact that when the transition in stage $k-1$ happens, the output residue V_{k-1}^{res} suddenly becomes small and inferior to the comparator levels in stage k, causing a forced transition. For example, table 3.4 shows all the possible transitions to appear at the output of a 2.5-bit pipeline stage k when an increasing ramp stimulus is applied at the pipeline ADC input.

This table represents the digital output code of the pipeline stage at a given normalized time $(n-1)$, $c_k(n-1)$, and its status one clock period after, $c_k(n)$, when a natural or forced transition happens at time n. In this case, there are six possible natural transitions (N_1 to N_6) corresponding to the six comparators of the sub-ADC of the stage. Forced transitions have been represented by F_{ab}, where the output code of the stage is forced to switch from code a to code b. The red cells indicate that the output code value has not changed.

From this point, the mapping is done by applying a ramp stimulus to the pipeline ADC. The ramp is sampled by the ADC and the output codes of each stage as well as the final output code are saved. Then, for each sample in each stage, the mapping algorithm determines the mapping between the ADC output transitions and the comparators that are being exercised.

From this point, the selection of representative codes can be initiated. Each comparator in each stage should be represented once in this set. For each comparator in stages 2 to K, where K is the total number of stages, there is a number of transitions that we can select from. Considering a comparator in the k-th stage, we should avoid selecting an ADC

output transition that involves in addition to this comparator a comparator in one of the previous stages.

Next, the widths of the selected codes are measured using a static test method as the histogram method for example. By default, the two codes on the left and on the right of an ADC output transition are measured. However, we must consider measuring more than two codes around the ADC output transitions that involve comparators which belong to the stages that are closer to the front of the pipeline so as to account for large linearity errors.

The next step consists in a copy/paste of the measured code widths to the codes around the ADC output transitions that were not selected, as seen in Figure 3.14.

Finally, we can reconstruct the whole ADC transfer function and estimate the DNL and INL of the pipeline ADC.

As previously stated, the key step of the technique is to properly find the mapping between the ADC transitions and the stage comparators. The ADC codes can then be gathered in groups of codes with the same widths.

Unfortunately, the technique might be ineffective in presence of noise. Even a small input-referred noise of a stage causes its digital output to oscillate between two codes when a transition is about to happen, and leads to inaccuracies for the mapping between the transitions and the exercised comparators.

- **Root codes extraction**

In order to circumvent this issue, it is proposed to use averaging. The first idea here is to find the root codes of each transition of the pipeline ADC. For the definition of a root code, let's first consider a pipeline ADC with K identical stages. We can make two groups out of its stages: the first group contains stages 1 to $k-1$, while the second group contains stages k to K. Furthermore, we define the function $f(k, comp_i^j, l)$ as the digital output of the k-th stage on the l side of the transition of the i-th comparator of the j-th stage. Each time the i-th comparator of the j-th stage is exercised, the digital output of this very stage changes from $f(j, comp_i^j, L)$ to $f(j, comp_i^j, R)$, meaning that its digital output goes from its value on the left side (L) of the transition of the i-th comparator to its value on the right side (R) of the same transition. The same reasoning can be applied to any other following stage k, provided they are identical. Finally, we can define two vectors of left and right digital outputs as

$$\left\{ \begin{array}{l} L_i^k = [f(k, comp_i^k, L), f(k+1, comp_i^k, L), ..., f(K, comp_i^k, L)] \\ R_i^k = [f(k, comp_i^k, R), f(k+1, comp_i^k, R), ..., f(K, comp_i^k, R)] \end{array} \right\}, \quad k = 1, 2, ..., K \tag{3.35}$$

From equation (3.35), the left root code LRC_i^k and the right root code RRC_i^k of the i-th comparator of the k-th stage is expressed as

$$\left\{ \begin{array}{l} LRC_i^k = \sum_{m=k}^{K} 2^{w'_m} \cdot f(m, comp_i^k, L) \\ RRC_i^k = \sum_{m=k}^{K} 2^{w'_m} \cdot f(m, comp_i^k, R) \end{array} \right\}, \quad k = 1, 2, ..., K. \tag{3.36}$$

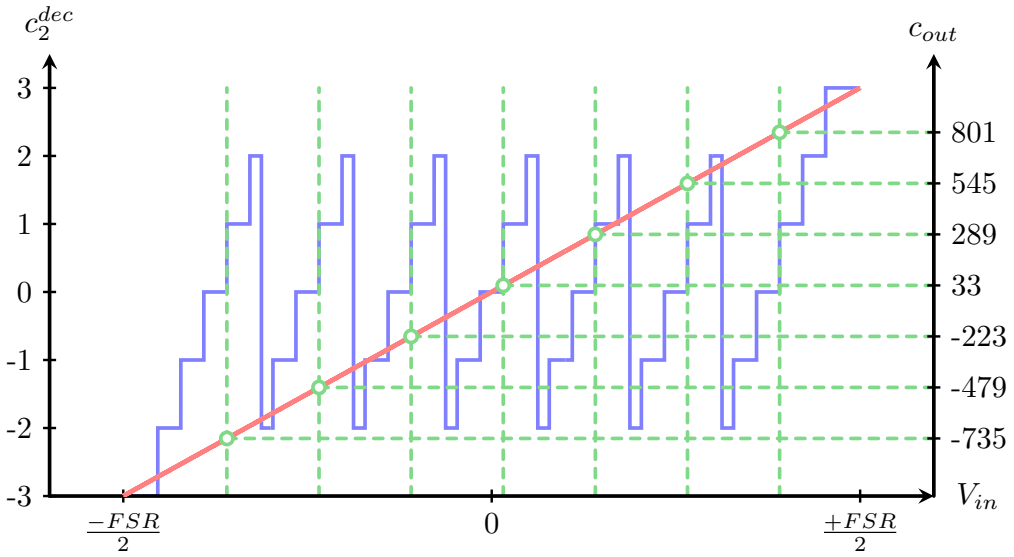


Figure 3.17: Second stage digital decimal output c_2^{dec} and corresponding digital decimal output c_{out} of a 2.5-bit/stage pipeline ADC model

For example, in Figure 3.17, a 11-bit pipeline ADC model of 2.5-bit/stage and a 3 bit final stage is considered. This example represents digital output of the second stage as well as the total output transfer function, with a focus on the fourth transition which corresponds to the fourth comparator of this stage. From this figure, we can identify the output code c_{out} related to each transition of the fourth comparator of the second stage.

Knowing that the weight of the first stage is $2^8 = 256$, we can write

$$\left\{\begin{array}{r} -735 = 33 - 2^8 \cdot 3 \\ -479 = 33 - 2^8 \cdot 2 \\ -223 = 33 - 2^8 \cdot 1 \\ 33 = 33 - 2^8 \cdot 0 \\ 289 = 33 + 2^8 \cdot 1 \\ 545 = 33 + 2^8 \cdot 2 \\ 801 = 33 + 2^8 \cdot 3 \end{array}\right\}$$

where 33 is the left root code of the fourth comparator of the second stage of this pipeline ADC model.

- **Noise cancellation**

Once the root codes are extracted for each comparator, the effect of the input-referred noise can be eliminated. If p is defined as the number of natural transitions of the i-th comparator in the k-th stage, one can list p L_i^k vectors and p R_i^k vectors for every possible value of k. Because of noise, the extracted values of the elements of L_i^k and R_i^k are not the same between the k-th stage and the following stages, meaning that the left and right root codes of natural transitions might differ. The noise is eliminated by using a voting scheme, considering the most frequent value out of the p available values for each element of L_i^k and R_i^k. The two vectors are rewritten as

$$\left\{\begin{array}{l} L_i^k = [i-1, f_m(k+1, comp_i^k, L), ..., f_m(K, comp_i^k, L)] \\ R_i^k = [i, f_m(k+1, comp_i^k, R), ..., f_m(K, comp_i^k, R)] \end{array}\right\}, \quad k = 1, 2, ..., K \qquad (3.37)$$

where $f_m(k+1, comp_i^k, L) = mode(f(k+1, comp_i^k, L))$ and $f_m(k+1, comp_i^k, R) = mode(f(k+1, comp_i^k, R))$ are the most frequent values of L_i^k and R_i^k, respectively.

Finally, from the noise-free L_i^k and R_i^k, the noise-free left and right root codes can be rebuilt, which allows a very precise mapping between the ADC transitions and comparators.

- **Code mapping from root codes**

The last step is to establish a list of all the related ADC output codes to a transition in a given stage using the root codes.

The root codes of the first stage are directly the ADC output codes. For all stages $k \geqslant 2$, we must find all the weighted combinations of the digital outputs of the previous stages added to the root codes in order to compute the related ADC output codes. For two tran-

sitions belonging to two consecutive stages k and $k+1$, three possible pairs of transitions can be found: (natural−natural), (natural−forced), and (forced−natural). While the digital output of the k first stages are at a fixed value, the digital output of the $(k+1)$-th stage ramps until the i-th comparator is exercised. At this moment, the resulting left and right ADC output codes correspond to the i-th comparator of this stage. If the transition in the k-stage is a natural i-th transition, then the value to consider is the second element of R_i^k, which is $f_m(k+1, comp_i^k, R)$, the code value in the $(k+1)$-th stage at the right of the i-th transition in the k-th stage. If the transition in the k-stage is a forced i-th transition, due to a natural transition in the $(k-1)$-th stage, then the value to consider is the third element of R_i^{k-1}, which is $f_m(k+2, comp_i^{k-1}, R)$, the code value in the $(k+1)$-th stage at the right of the i-th transition in the k-th stage.

The same process is done for all the transitions of the pipeline ADC. Then, from the list of selected ADC output codes for each stage, the code selection for DNL and INL estimation can be initiated, as explained earlier in this section.

3.4. Proposed servo-loop algorithm with RCLT for pipeline ADC BIST

3.4.1. Description

According to the reduced-code algorithm in [41], in order to characterize the complete static characteristic of the ADCUT we only need to measure the code widths around a reduced set of selected transitions. Specifically, we only need to measure the widths of 8 codes around the 6 transitions of the first stage, 6 codes around the 6 transitions of the second stage, 4 codes around the 6 transitions of the third stage, and 2 codes around the 6 transitions for the fourth and fifth stages. The interested reader is referred to [41] for a detailed description of the algorithm for relevant code selection.

Once the width of the target codes around the selected set of representative ADC transitions have been measured, we can reconstruct the full ADC static transfer characteristic by mapping these measurements to the rest of unmeasured codes in the ADCUT.

3.4.2. Comparison of the proposed servo-loop algorithm with and without RCLT

The proposed servo-loop technique combined with the RCLT technique is simulated in order to show its performance with respect to the proposed servo-loop technique without the RCLT technique. For this simulation, 1000 11-bit, 2.5-bit/stage, 2 V FSR pipeline ADCs are randomly generated using a parameterized realistic pipeline ADC model. The standard deviation of the ADC input-referred noise is fixed to $\sigma_{ADC} = 0.15$ LSB. The maximum swing of the generator output noise is ± 3 times its step size, so for a step size of $s = 0.1$ LSB, the standard deviation of the generator output noise is $\sigma_{ramp} = 0.1$ LSB. For each of the 1000 runs, 1024 samples are collected for each ADC code measurement. Each stage is randomly tuned for the generation of low, moderate and large INL values, especially the first stages which are responsible for the large INL drops and the main shape of the INL curve. These ADCs have an absolute maximum INL ranging from 0.2 LSB to 4.3 LSB. The range of the INL values is large in order to show to what extent the proposed technique with RCLT is able to estimate the INL of good and bad ADCs.

Figure 3.18 shows three graphs. The top graph represents the estimated INL value of the tested ADCs versus their actual INL value. The estimated INL using the RCLT technique is depicted by the blue circles, while the estimated INL when RCLT technique is used is plotted in red diamonds. The green line represents the ideal $y = x$ line, and the grey line represents the 1 LSB INL estimation error line. The blue histogram defines the dispersion of the maximum absolute difference between the estimated INL and the actual INL without the RCLT technique, while the red histogram shows the dispersion of the maximum absolute difference between the estimated INL and the actual INL with the RCLT technique.

Results in the case without RCLT show that while the actual INL ramps from 0.2 LSB to 4.3 LSB, the estimated INL varies from 0.79 LSB to 5.07 LSB, and the INL estimation error varies from 0 LSB to 1.7 LSB. The average INL estimation error is 0.69 LSB. 151 of the 1000 ADC/ramp generator pairs have been measured with an INL estimation error superior to 1 LSB, and 10 others have been measured with an INL estimation error below 0 LSB. With the RCLT technique, the estimated INL varies from 0.34 LSB to 5.4 LSB, and the INL estimation error varies from 0 LSB to 1.17 LSB. The average INL estimation error is 0.45 LSB. 15 of the 1000 ADC/ramp generator pairs have been measured with an INL estimation error superior to 1 LSB, and 9 others have been measured with an INL estimation error below 0 LSB. Moreover, we can see that the RCLT technique greatly reduces the variability of the proposed BIST technique.

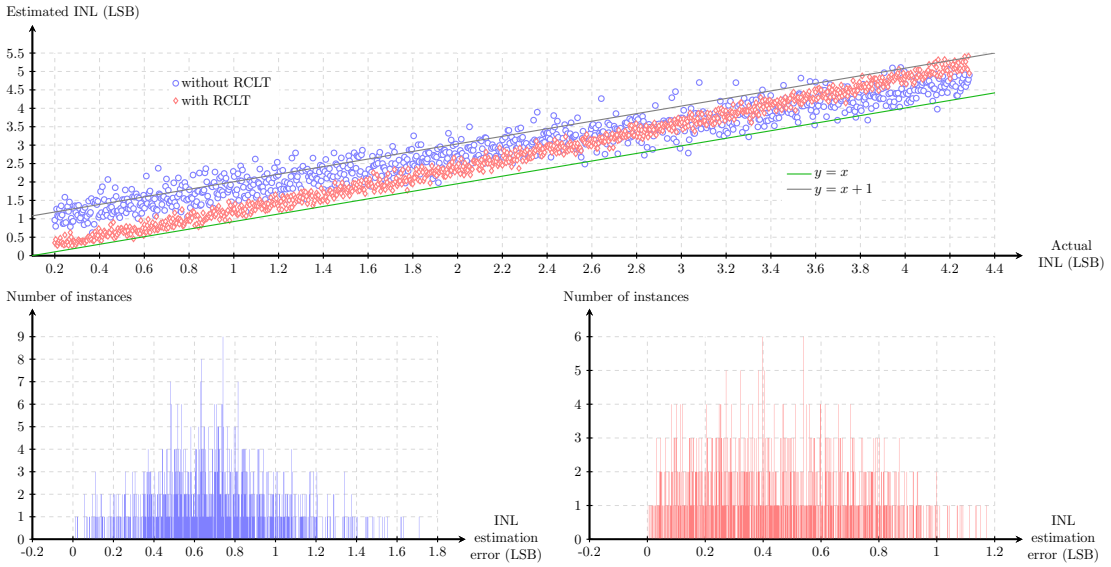


Figure 3.18: Estimated INL versus actual INL and respective histograms of the INL estimation error for 1000 ADC/ramp generator pairs using the proposed servo-loop technique without RCLT, then with RCLT

3.5. Discussion

In this chapter, we proposed a technique in order to adapt the well-known servo-loop technique to a BIST implementation of ADC static test. The proposed technique is compared to the classical servo-loop technique. Trade-offs are to be considered in order to keep an acceptable measurement accuracy and/or test time. The technique can be specifically applied to pipeline ADCs using the RCLT technique, drastically reducing the test time while enhancing its test quality. The pipeline ADC structure is presented and the reduced code testing technique for pipeline ADCs from [38]−[41] is explained.

In the next chapter, we propose a ramp generator that is necessary for a BIST implementation of the proposed servo-loop technique.

Chapter 4

Design of a ramp generator for ADC testing

This section describes the design of a novel ramp signal generator aimed at static linearity BIST applications. The proposed signal generation technique, detailed in the following subsections, is based on a discrete-time integrator with a very small integration gain. Such a system produces a step-wise ramp as response to a DC input reference.

4.1. Proposed signal generation technique

The proposed ramp generation technique, originally described in [**?**] and further developed in [**?**], is based on a two-input switched-capacitor (SC) integrator with a small capacitance difference in its input sampling capacitors, C_{i_1} and C_{i_2}. Figure 4.1 (a) shows a conceptual single-ended version of the proposed ramp generator in order to illustrate the signal generation technique. The integrator is based on a discrete-time implementation of the continuous-time RC integrator. The equivalent input resistance of the SC integrator, represented for example by the input branch containing C_{i_1}, can be written as

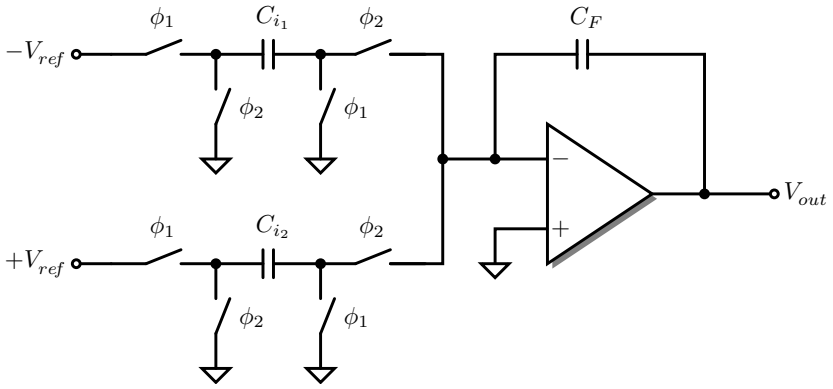
$$R_{eq} = \frac{T_{clk}}{C_{i_1}} \tag{4.1}$$

where T_{clk} is the integrator clock period. We can write from equation (4.1) the equivalent RC integrator time constant as

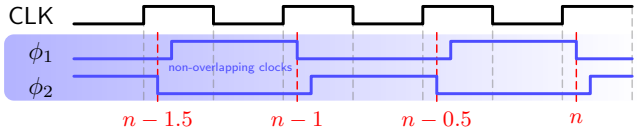
$$\tau_{eq} = R_{eq} \cdot C_F \tag{4.2}$$

$$= T_{clk} \cdot \frac{C_{i_1}}{C_{i_2}} \tag{4.3}$$

which means that the equivalent RC time constant is controlled by the ratio of the capacitors and the clock period. The real benefit of this implementation over the continuous one is that while the absolute value of capacitances and resistances have a poor tolerance



(a) Proposed single-ended SC integrator



(b) Timing of the generator

Figure 4.1: Proposed switched-capacitor ramp generator

and a poor matching, the matching between similar capacitances is very good (around 0.1 % in modern CMOS technologies). Moreover, the use of T_{clk} gives flexibility to the design. Consequently, this makes possible the implementation a relatively high precision integrated applications.

The core of the proposed ramp generator is a parasitic-insensitive SC integrator that operates with two non-overlapping clock phases, ϕ_1 and ϕ_2. Figure 4.1 (b) shows the associated timing for phases ϕ_1 and ϕ_2.

On sampling phase ϕ_1, as seen in Figure 4.2 (a), sampling capacitors C_{i_1} and C_{i_2} are charged to the input DC reference voltages $-V_{ref}$ and $+V_{ref}$, respectively, while the voltage on the integrating capacitor C_F remains constant. The voltage across each capacitor in this phase can be written as

$$V_{C_{i_1}}(n-1) = V_{in_1}(n-1) = -V_{ref} \tag{4.4}$$

$$V_{C_{i_2}}(n-1) = V_{in_2}(n-1) = +V_{ref} \tag{4.5}$$

$$V_{C_F}(n-1) = V_{out}(n-1) \tag{4.6}$$

where $n = t/T_{clk}$ is the normalized time with respect to the integrator clock period T_{clk}. In

consequence, the charges across each capacitor can be expressed as

$$Q_{C_{i_1}}(n-1) = -C_{i_1} \cdot V_{ref} \tag{4.7}$$
$$Q_{C_{i_2}}(n-1) = +C_{i_2} \cdot V_{ref} \tag{4.8}$$
$$Q_{C_F}(n-1) = C_F \cdot V_{out}(n-1). \tag{4.9}$$

On integrating phase ϕ_2, as seen in Figure 4.2 (b), capacitors C_{i_1} and C_{i_2} are now connected to the op-amp input and the integrating capacitor C_F. The charges Q_{i_1} and Q_{i_2} are transferred from the sampling capacitors to C_F and V_{out} settles to a new value. The sum of the two charges at the op-amp input node creates the required capacitance difference. Once the charges are settled at the end of phase ϕ_2, the charges across each capacitor can be expressed as

$$Q_{C_{i_1}}(n-0.5) = 0 \tag{4.10}$$
$$Q_{C_{i_2}}(n-0.5) = 0 \tag{4.11}$$
$$Q_{C_F}(n-0.5) = C_F \cdot V_{out}(n-1) + C_{i_2} \cdot V_{ref} - C_{i_1} \cdot V_{ref}. \tag{4.12}$$

We note that once phase ϕ_2 becomes inactive, the charge on C_F remains constant during the following ϕ_1 phase, until phase ϕ_2 becomes active again in the next cycle. Therefore, the charge on C_F at the end of phase ϕ_1, at time n, is equal to the one at time $(n-0.5)$. Then, we can write

$$Q_{C_F}(n) = Q_{C_F}(n-0.5). \tag{4.13}$$

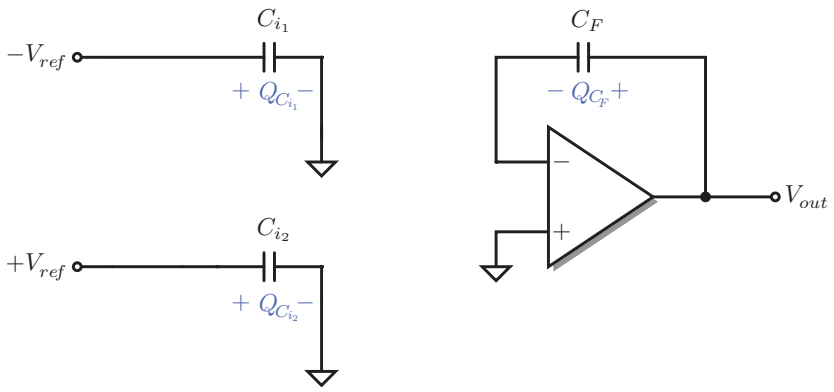
We can combine equations (4.12) and (4.13) in order to obtain the final charge relationship expressed as

$$C_F \cdot V_{out}(n) = C_F \cdot V_{out}(n-1) + C_{i_2} \cdot V_{ref} - C_{i_1} \cdot V_{ref}. \tag{4.14}$$

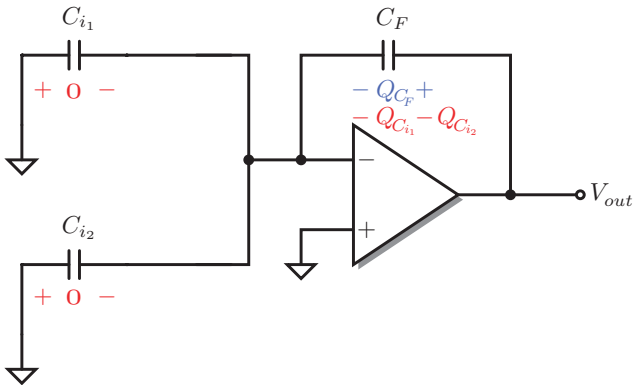
Finally, from equation (4.14), the output of the integrator, under ideal conditions, is given by

$$\begin{aligned} V_{out}(n) &= V_{out}(n-1) + \frac{C_{i_2} - C_{i_1}}{C_F} \cdot V_{ref} \\ &= V_{out}(n-1) + \frac{\delta C}{C_F} \cdot V_{ref}. \end{aligned} \tag{4.15}$$

As it can be derived from equation (4.15), the integrator output is a stepwise ramp,



(a) Sampling phase ϕ_1



(b) Integrating phase ϕ_2

Figure 4.2: Charges at phases ϕ_1 and ϕ_2

and the magnitude of the steps depends on the capacitance difference δC. This capacitance difference is implemented at layout level by deliberately creating a small difference between the input capacitors.

As it is well known, in a parasitic insensitive implementation as the one shown in Figure 4.1 (a), the parasitic capacitances across the switches around the input capacitors nodes and ground C_{p_1}, C_{p_2}, C'_{p_1}, C'_{p_2}, and the parasitic capacitances C_{p_3}, C_{p_4} at the terminals of C_F do not affect the operation of the circuit as in previous implementations. Figure 4.3 shows the parasitic-insensitive integrator with parasitic capacitors. C_{p_2} and C'_{p_2} are either connected to ground through the switch or to virtual ground through the switch. Therefore, since C_{p_2} and C'_{p_2} always remain discharged after settling, they do not affect the operation of the circuit. Finally, C_{p_1} and C'_{p_1} are continuously being charged to the associated input DC reference voltages and discharged to ground. However, when ϕ_1 is active, the fact that C_{p_1} and C'_{p_1} are also charged to respectively $-V_{ref}$ and $+V_{ref}$ does not affect the charge that is placed on C_{i_1} and C_{i_2}. When ϕ_2 is active, C_{p_1} and C'_{p_1} are discharged

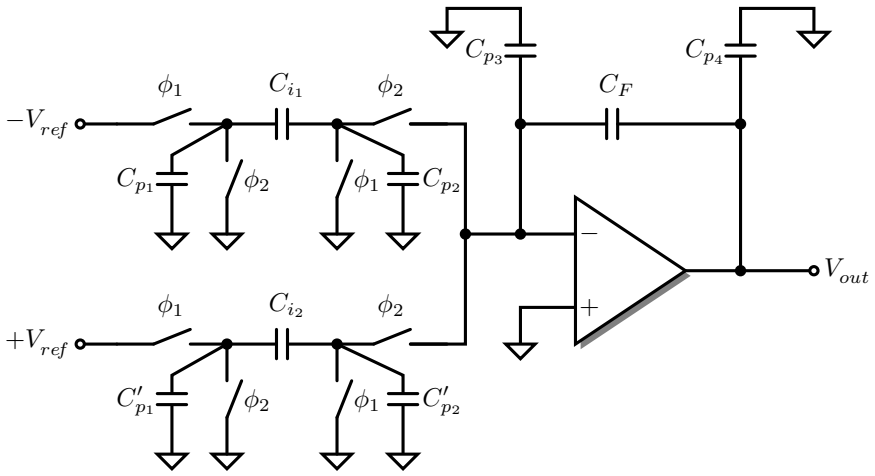


Figure 4.3: Proposed switched-capacitor ramp generator with parasitic capacitances shown

through the switch attached to their respective nodes and none of their discharging current passes through C_{i_1} and C_{i_2} to affect the charge accumulating on C_F. Therefore, they do not affect the circuit operation. In summary, while the parasitic capacitances may slow down the settling time behavior, they do not affect the discrete-time difference equation that occurs in the integrator shown in Figure 4.1 (a).

The proposed signal generator implementation, based on introducing a small difference between two capacitors, is prone to variations of the ramp step size due to random mismatch in the capacitors. Moreover, other non-idealities, such as noise, offset, integrator finite gain and leakage, switches charge injection and clock feedthrough, etc., have also an impact in the performance, i.e. resolution and linearity, of the ramp generator. In the next subsection, we analyze the effect of the main non-idealities in the ramp generator and we provide guidelines to reduce their impact in the generator performance.

4.2. Design considerations for a practical implementation of the proposed on-chip stimulus generator

4.2.1. Operational amplifier design

The operational amplifier is a standard fully-differential two-stage folded cascode topology with Miller compensation. Figure 4.4 depicts the global architecture of the op-amp while Figure 4.5 shows the topology of the amplifier (the bias voltage circuitry is not shown for simplicity, but it will be detailed later). The outputs of the two stages are connected to a dynamic switched-capacitor common-mode feedback (SCCMFB) network

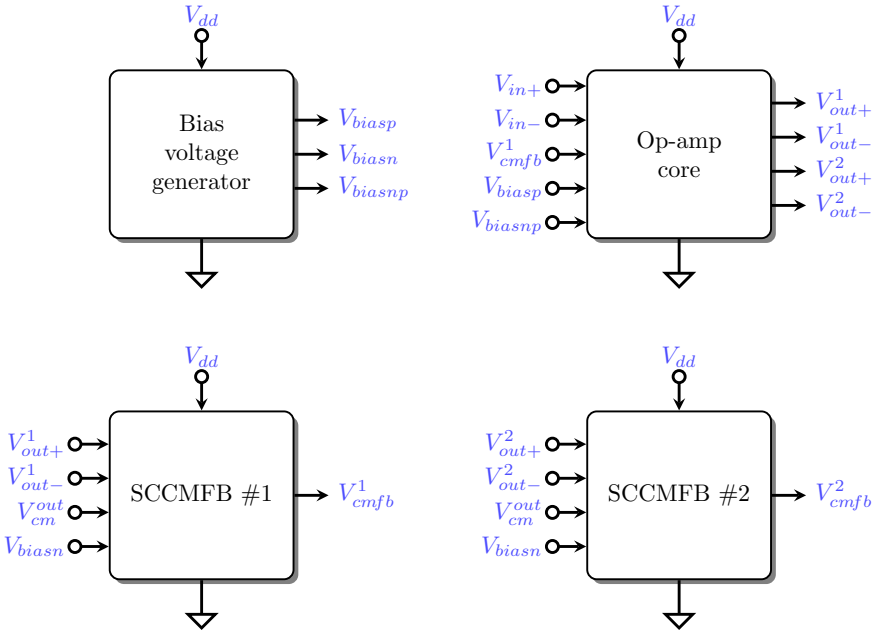


Figure 4.4: Architecture of the op-amp

[**?**], as shown in Figure 4.6, that defines the common-mode voltage for the differential output nodes. A differential configuration enables a high common-mode rejection ratio, improved immunity to charge injection and clock-feedthrough from the switches, and suppression of even harmonics, thus enhancing the linearity of the amplifier. Moreover, this topology with class-A second stage yields a higher gain and larger output swing compared to the folded cascode stage. Using a simple pole-zero analysis [**?**], the two-stage amplifier transfer function is modeled as

$$H_{OA}(\omega) = \frac{A_0 \cdot (1 + j\omega/z_1)}{(1 + j\omega/p_1)(1 + j\omega/p_2)}. \tag{4.16}$$

Assuming $1/p_1 >> 1/p_2$, equation (4.16) can be written as

$$H_{OA}(\omega) = \frac{A_0 \cdot (1 + j\omega/z_1)}{1 + j\omega/p_1 - \omega^2/(p_1 \cdot p_2)}. \tag{4.17}$$

From the circuit analysis, we can establish the equations of each parameter of the transfer function. The total gain of the op-amp is written as

$$\begin{aligned} A_0 &= A_{0_1} \cdot A_{0_2} \\ &= G_{m_1} \cdot R_{out_1} \cdot G_{m_2} \cdot R_{out_2} \\ &= \frac{g_{m_1} g_{m_3} g_{m_4}}{g_{m_4} g_{ds_3}(g_{ds_1} + g_{ds_2}) + g_{m_3} g_{ds_4} g_{ds_5}} \cdot \frac{g_{m_7}}{g_{ds_7} + g_{ds_8} + g_{ds_{7FB}} + g_{ds_{8FB}}} \end{aligned} \tag{4.18}$$

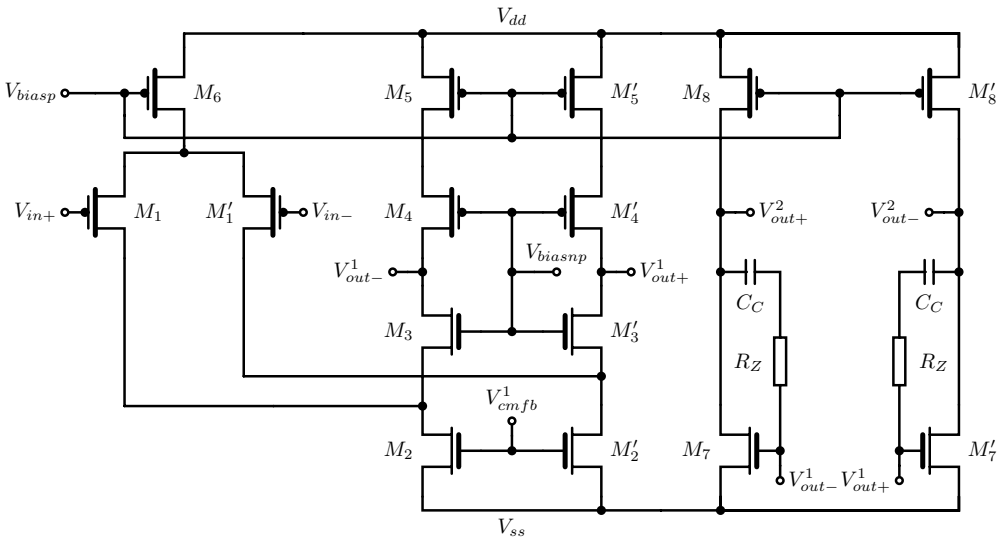


Figure 4.5: Fully differential two-stage folded cascode operational amplifier

where A_{0_1}, A_{0_2} are respectively the gains, G_{m_1}, G_{m_2} the transconductances, and R_{out_1}, R_{out_2} the output resistances of the first and second stages of the op-amp. The class-A second stage adds a non-negligible gain to the total gain value, but also provides a better output swing, which is required for our application where the output swing must be maximized to target the FSR of the ADC under test. For a single stage folded cascode op-amp, the output swing is about $V_{dd} - 4 \cdot V^{min}_{ds,sat}$, while the two stage op-amp output swing is about $V_{dd} - 2 \cdot V^{min}_{ds,sat}$, with $V^{min}_{ds,sat}$ the minimum voltage for which a transistor is in saturation. From equation (4.18), the gain can be maximized by increasing the transconductance and reducing the output conductance of transistors $M_1 - M'_1$, $M_3 - M'_3$, $M_4 - M'_4$, $M_7 - M'_7$.

Next, the dominant and non-dominant poles are expressed as

$$p_1 \approx \frac{g_{m_1}}{A_0 C_C} \tag{4.19}$$

$$p_2 \approx \frac{g_{m_7}}{C_L} \tag{4.20}$$

while the zero pole is denoted as

$$z_1 = \frac{g_{m_7} C_C}{(g_{m7} R_Z - 1)}. \tag{4.21}$$

The resistance R_Z is taken equal to the inverse of g_{m7} to suppress the zero pole. Alternatively, its value can be increased to enlarge the phase margin of the amplifier. The stability analysis of the op-amp is done considering the total load capacitance C_L^{TOT} which takes into account the load capacitance, the feedback capacitor and the sampling capacitors. The total load capacitance is then expressed as

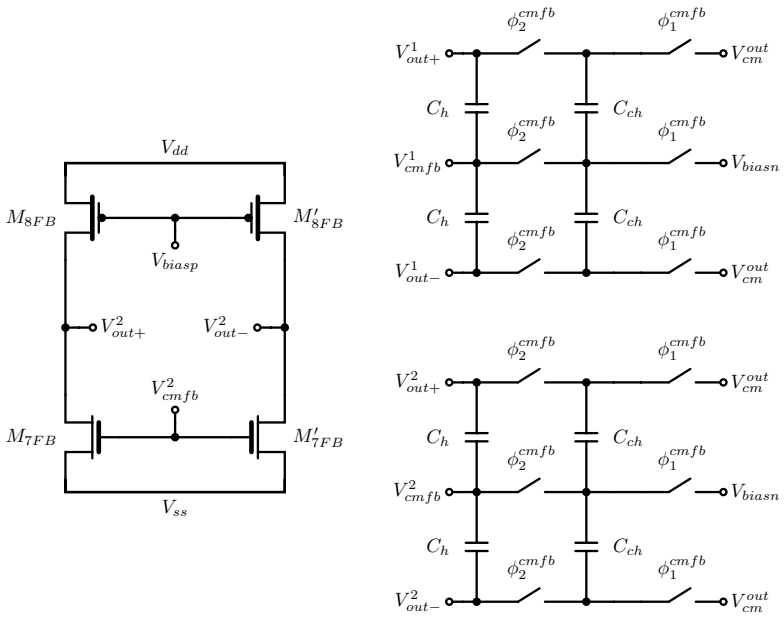


Figure 4.6: First-stage and second-stage SCCMFBs networks

$$C_L^{TOT} = C_L + \frac{(C_{i_1} + C_{i_2}) \cdot C_F}{C_{i_1} + C_{i_2} + C_F}. \tag{4.22}$$

From this capacitance value, parameters such as slew rate, bias current, gain bandwidth, and Miller compensation capacitance can be computed.

Finally, we write the gain-bandwidth of the op-amp as the product between the gain and the dominant pole of the op-amp

$$GBW = A_0 \cdot p_1 \approx \frac{g_{m1}}{C_C}. \tag{4.23}$$

Transconductance of transistors $M_7 - M'_7$ must be high enough to push the second pole to frequencies above the gain-bandwidth frequency.

The common-mode voltage for each stage is provided by dynamic SCCMFB networks. As shown in Figure 4.6, the two SCCMFB networks are controlled by non-overlapping clock phases ϕ_1^{cmfb} and ϕ_2^{cmfb}. The first SCCMFB samples the output voltage of the folded cascode stage. When ϕ_1^{cmfb} is active, each charge capacitor C_{ch} is charged to the differential voltage between the reference common-mode voltage V_{cm}^{out} and the bias voltage V_{biasn} of transistors $M_2 - M'_2$. Then during the ϕ_2^{cmfb} phase, the charge kept in the charge capacitors are transferred to the hold capacitors C_h whose voltage is updated and compensates the common-mode voltage difference between the output nodes until the next clock cycle. If the output common-mode is below its ideal value, the $M_2 - M'_2$ bias

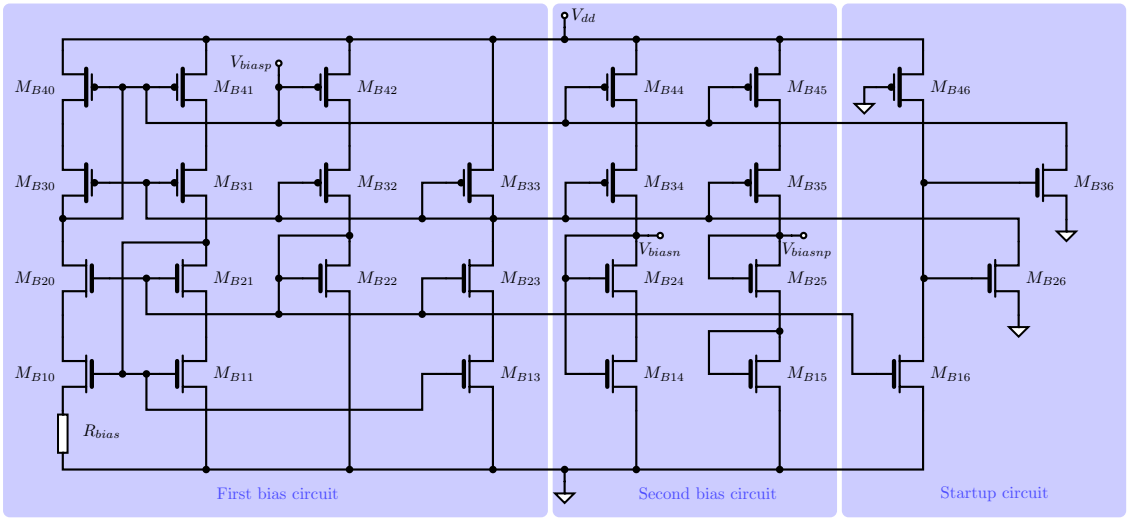


Figure 4.7: Self-biased bias generator with startup circuit

voltage is raised to counter the deviation, and vice versa. Similarly, the second SCCMFB samples the output voltages of the second stage [**?**]. The output common-mode voltage is then used to control two common source amplifiers, M_{7FB} and M'_{7FB}, that drive the output nodes by pumping or pushing current into the output branches. As it is described in [**?**], the common mode voltage settling time depends mainly on the ratio between the charge capacitor C_{ch} and the hold capacitor C_h. Their values are chosen to limit kT/C noise in the common-mode feedback branches and at the same time be reasonably small to reduce area overhead.

The bias voltages of the operational amplifier are generated with the bias generator pictured in Figure 4.7. The circuit is composed of two bias circuits. The first one is self-biased, meaning that the transistors transconductances have a very small dependency on the supply voltage value as well as process and temperature variations. In this way, the generated voltages are stable, which is mandatory for the proper operation of the op-amp. The bias resistor R_{bias} is set to 4.711 $k\Omega$ in order to generate the required bias current. This circuit generates the bias voltage of the input pair V_{biasp}. In order to enhance the PSRR of the circuit, a cascode current source implementation is considered using transistors M_{B20}, M_{B21}, M_{B22}, M_{B23}, M_{B30}, M_{B31}, M_{B32} and M_{B33}. The second bias circuit generates the bias voltage for the two CMFBs V_{biasn} and the bias voltage of the NMOS and PMOS cascodes V_{biasnp}. A delicate point in this structure is that once the power supplies are connected to the circuit, the circuit may stay in a stable state where no current is generated. This problem can be dealt with a startup circuit. As seen in Figure 4.7, this circuit is added to the bias generator using transistors M_{B16}, M_{B26}, M_{B36} and M_{B46}. The circuit forces a voltage on a node, or a current into a branch, in order to set the proper initial state after which normal operation can begin.

4.2.2. Finite gain and integrator leakage

A direct consequence of the finite amplifier gain is the integrator leakage. In each integration cycle, only a fraction α of the previous integrator output voltage is added to the input sample. The integrator leakage limits the gain of the integrator, as well as the ramp linearity. The Z-transform transfer function of the SC integrator with leakage can be expressed as

$$H_{INT}(z) = g \cdot \frac{z^{-1}}{1 - \alpha z^{-1}} \tag{4.24}$$

with

$$g \approx \frac{C_{i_2} - C_{i_1}}{C_F} \cdot \left(1 - \frac{1 + \frac{C_{i_2} - C_{i_1}}{C_F}}{A_0}\right), \tag{4.25}$$

$$\alpha \approx 1 - \frac{C_{i_2} - C_{i_1}}{A_0 C_F}. \tag{4.26}$$

These effects can be minimized by choosing a high gain A_0 as well as a small input capacitance difference $C_{i_2} - C_{i_1}$.

4.2.3. Gain bandwidth and slew rate

The proposed on-chip ramp generator is intended for ADC static linearity test applications. In order to characterize the linearity of an ADC, we must ensure that the ADC works under quasi-stationary conditions. This means that the ramp generator has to operate at a clock frequency that is significantly lower than the sampling rate of the ADC, such that practically the ADC samples a DC voltage. In other words, each output ramp step must settle completely on the integrating phase ϕ_2, such that the ADC digitizes a quasi-constant value when on its active phase. However, the clock frequency of the ramp generator should not be so slow to the point where the feedback capacitor starts discharging during the conversion significantly affecting the operation of the integrator. Given the above discussion, the GBW requirement can be relaxed while the slew rate (SR) requirement becomes more stringent. The SR can be expressed as

$$SR = \frac{I_{m_6}}{C_C} \tag{4.27}$$

where I_{m_6} is the current flowing through transistor M_6. From equation 4.27, we can select I_{m_6} and C_C to achieve a sufficiently high SR, keeping in mind that I_{m6} should be sufficiently small so as to avoid prohibitive power consumption (in any case, power consumption is not an important issue here since the ramp generator is powered on only in

test mode), and C_C should have a value that guarantees the stability of the amplifier while incurring the smallest area possible. Thereafter, from equation (4.23), g_{m1} is selected to be sufficiently high enough to achieve the required GBW.

4.2.4. Input-referred noise

Due to the switch-capacitor implementation of the input branches, all noise sources are uncorrelated during phase ϕ_1 but correlated during phase ϕ_2 because C_{i_1} and C_{i_2} are sampling the amplifier noise simultaneously. In the presence of noise, the Z-transform transfer function of a single-ended integrator can be expressed as

$$V_{out}(z) = \frac{C_{i_2} - C_{i_1}}{C_F} \cdot \frac{z^{-1}}{1 - z^{-1}} \cdot V_{ref} + N_{out}(z). \tag{4.28}$$

with

$$N_{out}(z) = \frac{C_{i_1}}{C_F} \cdot \frac{z^{-1} N_{i_1}^{\phi_1}(z) - N_{i_1}^{\phi_2}(z)}{1 - z^{-1}} + \frac{C_{i_2}}{C_F} \cdot \frac{z^{-1} N_{i_2}^{\phi_1}(z) - N_{i_2}^{\phi_2}(z)}{1 - z^{-1}} \tag{4.29}$$

where $N_{i_1}^{\phi_k}$ and $N_{i_2}^{\phi_k}$, $k \in \{1,2\}$ are the noise voltages sampled by the input capacitors C_{i_1} and C_{i_2} during phases ϕ_1 and ϕ_2.

By forming $N_{out}(z)N_{out}(z-1)$ and only keeping the correlated terms, the equation gives the output noise density. The input equivalent noise density is then calculated as the sum of the input noise densities for each input capacitor branch. Dividing by the square value of the signal transfer function in the frequency domain with respect to C_{i_1}, the corresponding input noise density is equal to

$$\begin{aligned} S_{in}(f)|_{C_{i_1}} &= S_{i_1}^{\phi_1}(f) + S_{i_1}^{\phi_2}(f) + \left(\frac{C_{i_2}}{C_{i_1}}\right)^2 \cdot (S_{i_2}^{\phi_1}(f) + S_{i_2}^{\phi_2}(f)) \\ &\quad + \frac{C_{i_2}}{C_{i_1}} \cdot (S_{i_{1,2}}^{\phi_1}(f) + S_{i_{2,1}}^{\phi_2}(f)) \end{aligned} \tag{4.30}$$

with $S_{ik}^{\phi_l}$, $\{k,l\} \in \{1,2\}$ the spectral densities of the noise sampled by C_{i_1} and C_{i_2} during phases ϕ_1 and ϕ_2, and $S_{i_{1,2}}^{\phi_2}$, $S_{i_{2,1}}^{\phi_2}$ the cross spectral densities of $N_{i_1}^{\phi_2}$ and $N_{i_2}^{\phi_2}$ during phase ϕ_2. Applying the same reasoning with the square value of the signal transfer function in the frequency domain with respect to C_{i_2}, we get

$$\begin{aligned} S_{in}(f)|_{C_{i_2}} &= S_{i_2}^{\phi_1}(f) + S_{i_2}^{\phi_2}(f) + \left(\frac{C_{i_1}}{C_{i_2}}\right)^2 \cdot (S_{i_1}^{\phi_1}(f) + S_{i_1}^{\phi_2}(f)) \\ &\quad + \frac{C_{i_1}}{C_{i_2}} \cdot (S_{i_{1,2}}^{\phi_1}(f) + S_{i_{2,1}}^{\phi_2}(f)) \end{aligned} \tag{4.31}$$

The amplifier closed-loop gain is considered as a second-order low-pass filter. Using the residue theorem [**?**], the spectral densities are expressed as

$$\begin{aligned} S_{i_m}^{\phi_1}(f) &= 4K_bT \cdot \left(\frac{T_{clk}}{2C_{i_m}}\right) \cdot sinc^2(fT_{clk}) \\ S_{i_m}^{\phi_2}(f) &= 4K_bT \cdot (2R_{on} + R_{OA}) \cdot \left(\frac{\beta T_{clk} GBW}{2}\right) sinc^2(fT_{clk}) \\ S_{i_{1,2}}^{\phi_2}(f) &= S_{i_{2,1}}^{\phi_2}(f) = 4K_bTR_{OA} \cdot \left(\frac{\beta T_{clk} GBW}{2}\right) \cdot sinc^2(fT_{clk}) \end{aligned} \tag{4.32}$$

with $m \in \{1,2\}$, K_b the Boltzmann constant, T the temperature in Kelvin, T_{clk} the period of phases ϕ_1 and ϕ_2, R_{on} the resistance value of the switches when closed, R_{OA} the equivalent input-referred noise resistance of the amplifier, $\beta = C_F/(C_F + C_{i_1} + C_{i_2})$ the feedback factor of the integrator, and assuming $T_{clk}GBW >> 1$.

Finally, the total input noise spectral density is integrated from 0 to $1/2T_{clk}$ and the equivalent input noise voltage in the differential case is given by

$$V_n^{in} = \sqrt{K_bT \cdot \left(\frac{\theta_{C_{i_1}}}{C_{i_1}} + \frac{\theta_{C_{i_2}}}{C_{i_2}}\right)} \tag{4.33}$$

with $\theta_{C_{i_1}}$ and $\theta_{C_{i_2}}$ being

$$\theta_{C_{i_1}} = 2 \cdot \left(1 + \beta GBW \cdot \left(2R_{on} \cdot \frac{C_{i_2}^2}{C_{i_1}} + \frac{R_{OA}}{2} \cdot \left(C_{i_2} + \frac{C_{i_2}^2}{C_{i_1}}\right)\right)\right) \tag{4.34}$$

$$\theta_{C_{i_2}} = 2 \cdot \left(1 + \beta GBW \cdot \left(2R_{on} \cdot \frac{C_{i_1}^2}{C_{i_2}} + \frac{R_{OA}}{2} \cdot \left(C_{i_1} + \frac{C_{i_1}^2}{C_{i_2}}\right)\right)\right) \tag{4.35}$$

where K_b is the Boltzmann constant, T the absolute temperature, T_{clk} the period of phases ϕ_1 and ϕ_2, R_{on} the resistance value of the switches when closed, R_{OA} the equivalent input-referred noise resistance of the amplifier, and $\beta = C_F/(C_F + C_{i_1} + C_{i_2})$ is the feedback factor of the integrator. It is interesting to notice that since we use a difference of capacitance to generate a small input capacitor, we can choose relatively high values for C_{i_1} and C_{i_2} with respect to the value of their difference, in order to limit kT/C noise. Moreover, the dimensions of the switches are increased to reduce R_{on}. R_{OA} can also be reduced,

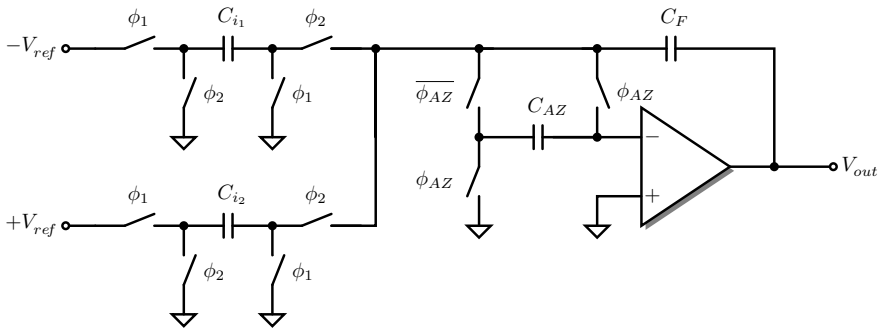


Figure 4.8: Auto zero implementation

knowing that the amplifier RMS input noise voltage is

$$\begin{aligned} V_n^{OA} &= \sqrt{V_{n_1}^2 + \left(\frac{g_{m_2}}{g_{m_1}}\right)^2 \cdot V_{n_2}^2} \\ &+ \sqrt{\left(\frac{g_{m_5}}{g_{m_1}}\right)^2 \cdot V_{n_5}^2 + \left(\frac{g_{m_7}}{g_{m1}A_{0_1}}\right)^2 \cdot V_{n_7}^2} \\ &= \sqrt{4K_bTR_{OA} \cdot (f_{max} - f_{min})}, \end{aligned} \tag{4.36}$$

where

$$\begin{aligned} V_{n_p}^2 &= \int_{f_{min}}^{f_{max}} \overline{V_{n_p}^2(f)} \cdot df \\ &= \frac{8K_bT}{3g_{m_p}} \cdot (f_{max} - f_{min}) + \frac{K_{M_p}}{WL \cdot C_{ox}} \cdot ln\left(\frac{f_{max}}{f_{min}}\right) \end{aligned} \tag{4.37}$$

$$R_{OA} \approx \frac{2}{3g_{m_1}} \cdot \left(1 + \frac{g_{m_2}}{g_{m_1}} + \frac{g_{m_5}}{g_{m_1}} + \frac{g_{m_7}}{g_{m_1}A_{0_1}^2}\right) \tag{4.38}$$

are the RMS squared noise voltage of transistor M_p (thermal and flicker noise) and the expression of R_{OA}, with K_{M_p} the process-dependent constant, C_{ox} the gate-oxide capacitance, WL the size of the transistor. Moreover, $f_{max} \approx \pi/2 \cdot f_{-3dB}$ and f_{min} is chosen around 1 to 100 Hz. A_{0_1} is supposed large, so the last term in the expression of R_{OA} can be neglected. Furthermore, g_{m_1} is chosen much larger than g_{m_2} and g_{m_5} to minimize the amplifier input noise voltage.

4.2.5. Operational amplifier input offset

Due to process mismatch between transistors, a non-negligible offset voltage can appear at the amplifier input, and might be integrated at each integration phase with the input signal if no correction is applied.

An offset is composed of a deterministic part and a random part. The former can be greatly reduced with careful design and layout. The latter is due to mismatch variations between transistors of the differential branches. Gate voltage offset is dominant over current offset. Transistors $M1-M1'$, $M2-M2'$, $M5-M5'$, $M7-M7'$ and $M8-M8'$ from Figure 4.5 participate in this effect. The input offset of the op-amp is then defined as

$$V_{os}^{OA} = \sqrt{\sigma^2(\delta V_{G_{1,1'}}) + \left(\frac{g_{m_2}}{g_{m_1}}\right)^2 \cdot \sigma^2(\delta V_{G_{2,2'}})} + \sqrt{\left(\frac{g_{m_5}}{g_{m_1}}\right)^2 \cdot \sigma^2(\delta V_{G_{5,5'}}) + \left(\frac{g_{m_8}}{g_{m_7} A_{0_1}}\right)^2 \cdot \sigma^2(\delta V_{G_{7,7'}})} \tag{4.39}$$

with

$$\sigma^2(\delta V_{G_{i,i'}}) = \sqrt{\sigma_T^2 + \left(\frac{I_{D_{sat,i}}}{g_{m_i}}\right)^2 \cdot \sigma_\beta^2} \tag{4.40}$$

$$\sigma_T = \frac{A_T}{\sqrt{WL}} \tag{4.41}$$

$$\sigma_\beta = \frac{A_\beta}{\sqrt{WL}} \tag{4.42}$$

where $\sigma(\delta V_{G_{i,i'}})$ is the standard deviation of the mismatched transistor pair, and A_T, A_β, WL, respectively the technology parameters and the size of the transistors.

The integrator output can be expressed during phase ϕ_2 as

$$V_{out}(n) = V_{out}(n-1) + \frac{C_{i_2} - C_{i_1}}{C_F} \cdot V_{ref} + \left(1 + \frac{C_{i_1} + C_{i_2}}{C_F}\right) \cdot V_{os}^{OA}. \tag{4.43}$$

The integrated offset can significantly affect the integrator step size as C_{i_1} and C_{i_2} are approximately equal. We can distinguish a deterministic and a random component in the amplifier offset. The former can be greatly reduced by careful design and dedicated layout techniques. The latter is due to mismatch variations between transistors of the differential branches and different techniques have been proposed for its cancellation in SC circuits [**?**].

The chopper technique for offset cancellation was considered first. This technique is based on modulating the input signal of the op-amp with a chopping frequency f_{ch}. This signal is then amplified by the op-amp and so is its input offset. At the op-amp output, the signal then goes through another chopping block for demodulation and offset cancellation. First, the relevant signal is demodulated, and secondly, the input offset is modulated with

the chopping frequency. Finally, the base-band signal is retrieved by inserting a low-pass filter at the output, which suppresses the harmonics related to the offset modulation. For best performance, the phase delay between the two modulator frequencies must be equal to the phase delay induced by the amplifier. Moreover, if the input is a constant voltage for example, the output will be an offset-free square wave, so the real output has to be retrieved with the use of a low-pass filter. These two drawbacks increase the design complexity and area overhead so this solution is not suitable for our application.

The proposed ramp generator design employs a simple closed-loop auto-zero technique to compensate the effect of the integrator offset. As detailed in [**?**], it is a classical offset compensation technique that relies on the use of a capacitor to store the offset and subtract it from the output. For the sake of simplicity, calculations are made on a single-ended version of the integrator. As it is shown in Figure 4.8, capacitor C_{AZ} is used to store the amplifier offset during the sampling phase and to subtract this offset in the integration phase. When ϕ_{AZ} is high, at the same frequency and phase as ϕ_1, C_{AZ} is charging between the offset and the output common-mode voltage as well as the C_F capacitor to compensate for the charge on the input capacitors. If V_{out}^{AZ} is the output in the ϕ_{AZ} phase, the voltage across C_{AZ} can be expressed as

$$V_{C_{AZ}} = V_{os}^{OA} - \frac{V_{out}^{AZ}}{A_0} \tag{4.44}$$

due to the finite gain of the amplifier. On phase $\overline{\phi_{AZ}}$, the three branches connect in node N and its voltage will drop by the same value. The new output voltage $V_{out}^{\overline{AZ}}$ is defined as

$$V_{out}^{\overline{AZ}} = V_{out}^{AZ} - \left(1 + \frac{C_{i_1} + C_{i_2}}{C_F}\right) \cdot \left(V_{os}^{OA} - \frac{V_{out}^{AZ}}{A_0}\right) \tag{4.45}$$

The non-inverting input is then described as

$$V^- = V_{os}^{OA} - \frac{V_{out}^{\overline{AZ}}}{A_0} \tag{4.46}$$

and finally the voltage at node N is

$$V_N = V^- - V_{C_{AZ}} = \left(1 + \frac{C_{i_2} + C_{i_1}}{C_F}\right) \cdot \frac{\left(V_{os}^{OA} - \frac{V_{out}^{AZ}}{A_0}\right)}{A_0} \tag{4.47}$$

allowing a quasi-perfect charge transfer between $C_{i_2} - C_{i_1}$ and C_F on phase ϕ_2. The fully-differential architecture, a high amplifier gain, as well as a careful layout contribute to reduce the effect of this residual uncompensated offset.

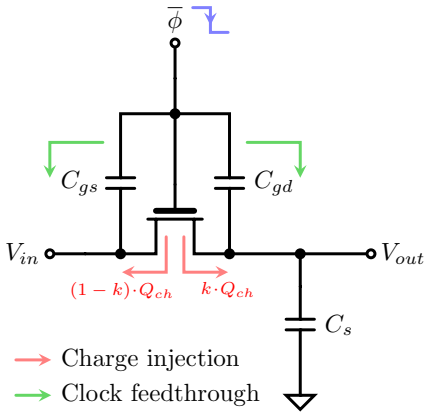


Figure 4.9: Sources of voltage errors on switches

4.2.6. Charge injection and clock feedthrough

Each switch is designed as a CMOS switch, consisting of a PMOS and a NMOS connected in parallel. The CMOS implementation is useful to reduce the R_{on} equivalent resistance of the switch which is expressed as

$$R_{on} = \frac{1}{\mu C_{ox} \frac{W}{L} \cdot (V_{dd} - V_{th} - V_{in})^2} \tag{4.48}$$

where C_{ox} is the gate-oxide capacitor.

Charge injection and clock feedthrough, shown in Figure 4.9, are the major error sources caused at the moment the switch turns off. Charge injection is caused by stray capacitance associated with the NMOS and PMOS transistors. The gate-drain capacitance associated with the PMOS device is about twice that of the NMOS device, because for both devices to have the same on-resistance, the PMOS device has about twice the area of the NMOS. During phase ϕ, the correspondent CMOS switches are closed for the input voltage sampling. However, when phase ϕ ends, they turn off. This is where charge injection occurs, due to mismatch between those stray capacitors. The charge exiting the drain gets injected in the input sampling capacitors, adding an offset to the sampled voltage. If the PMOS and NMOS switches had the same dimensions, the liberated charges would cancel each other due to their opposite signs. However, this is not true due to mismatch in the size of the transistors and because it is signal dependent. The switch-induced error voltage due to charge injection for both NMOS and PMOS transistors can

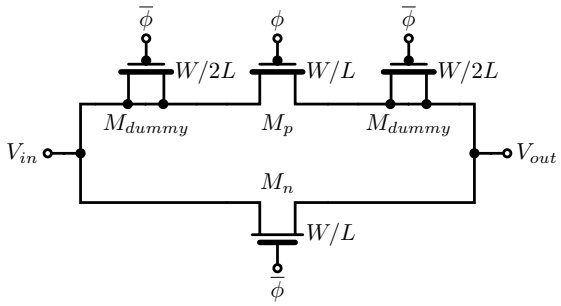


Figure 4.10: Implementation of CMOS switches with dummy transistors

be approximated as

$$\Delta V_n^{ci} = -\frac{kW_nL_nC_{ox} \cdot (V_{dd} - V_{thn} - V_{in})}{C_s} \tag{4.49}$$

$$\Delta V_p^{ci} = +\frac{kW_pL_pC_{ox} \cdot (V_{in} - |V_{thp}|)}{C_s} \tag{4.50}$$

where k is the fraction of charge injected on the output node, C_{ox} is the gate-oxide capacitor, and V_{thn} and V_{thp} are the threshold voltages of NMOS and PMOS transistors.

Nevertheless, charge injection can be limited by using several design techniques discussed in [**?**]. The proposed generator design employs two techniques to reduce the charge injection in the switches. The first technique is to connect dummy transistors on both sides of the CMOS switch to reduce the stray capacitor mismatch, as shown in Figure 4.10. They have the same length as the main switches but half their width and their drain and source are connected together. When the main switch turns off, the dummy switches turn on, and acquire the same amount of charge that the main switch has lost. The second technique is to use the bottom plate of the sampling capacitor. Using advanced clock phases with respect to the integrator clock, labeled ϕ_{1a} and ϕ_{2a} in Figure 4.16, the switch at the bottom plate of the capacitor opens before the top plate switch does. When the top plate switch opens, the bottom plate is floating and no charge is injected on the capacitor. These two techniques are implemented on each switch and on the clocking of the integrator. Additionally, the effects of charge injection are further reduced by the fully-differential architecture of the integrator, since charge injection in the positive and negative signal paths tend to compensate.

Clock feedthrough is due to the gate-to-source capacitors, interconnect parasitic capacitors of the switches and substrate coupling. They cause the clock signal to go through the analog line, and add an offset to the sampled voltage as well. It is reported that in most of the cases, this effect is dominated by clock injection effects [**?**]. The switch-induced error voltage due to clock feedthrough for both NMOS and PMOS transistors

can be approximated as

$$\Delta V_n^{cf} = -\frac{C_{gdn}}{C_s + C_{gdn}} \cdot V_{dd} \tag{4.51}$$

$$\Delta V_p^{cf} = +\frac{C_{gdp}}{C_s + C_{gdp}} \cdot V_{dd} \tag{4.52}$$

where C_{gdn} and C_{gdp} are the gate-drain overlap capacitance of NMOS and PMOS transistors.

Various solutions exist in order to counter this issue, but they are complex and take a lot of surface with respect to the switch itself. Area overhead and charge injection are reduced by increasing the sampling capacitor and using a fully-differential topology.

4.2.7. Clock generation

In this subsection, the digital part of the proposed generator is presented. It mainly consists of the several clock generators required to control the ramp generation.

4.2.7.1. Main clock generator

Several clock phases are required in order to pilot the different switches of the proposed generator. Figure 4.11 shows the clock generation circuit. This circuit generates clock phases ϕ_1, ϕ_2, ϕ_{1a}, ϕ_{2a}, ϕ_{AZ}, ϕ_{sample} and their respective complements. As seen at the beginning of Figure 4.11, this circuit is rendered inactive until phases ϕ_{reset} and ϕ_0 are inactive in order to prevent any ramp generation until the initial voltages are set at the integrator output. The digital flip-flop complementary output $\overline{\text{Q}}$ is cabled to input D to generate an output clock on Q that is half the frequency of the input clock. A second path is made to generate phase ϕ_{sample} and its complement. This path combines the half-frequency clock phase and a delayed version of the initial clock phase (using R_{d_1}, C_{d_1}) in order to produce a signal that has a duty cycle of 25 % for the output sampling phase, that happens when phase ϕ_2 is active. A second small RC block (R_{d_2}, C_{d_2}) are added to the path in order to delay the output sampling phase and limit the effect of spikes occurring at the integrator output when ϕ_2 is switched on. Furthermore, the 25 % duty cycle was chosen because it is easy to implement, avoids the spikes at the integrator output due to phase change between ϕ_1 going low and ϕ_2 going high, and keeps the integrator output stable during phase ϕ_{sample}.

The model of the digital flip-flop circuits used in the clock generators is shown in Figure

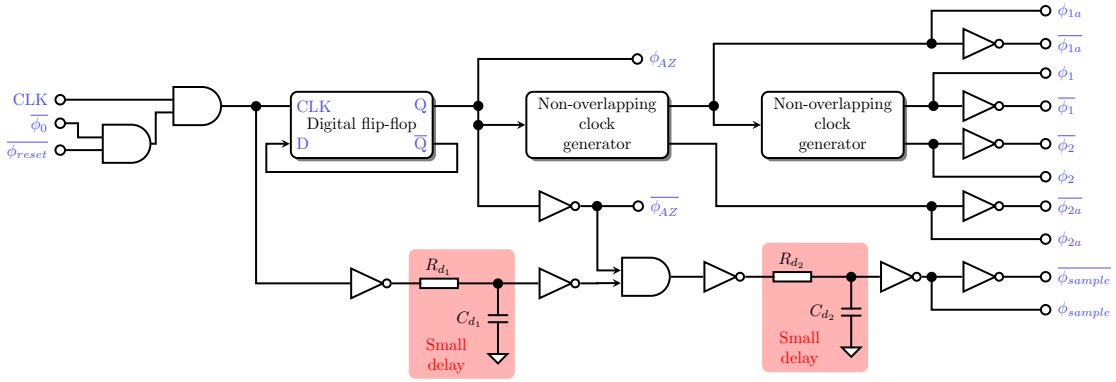


Figure 4.11: Clock generation circuit

4.12. It is a classical NAND-type flip-flop implementation in which each NAND cell is standard.

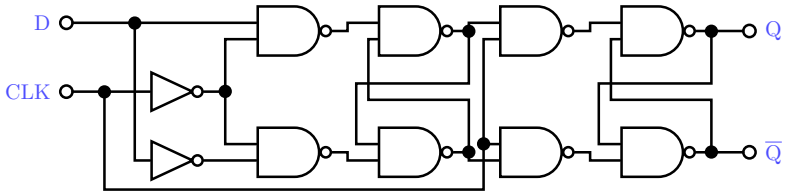


Figure 4.12: Digital flip-flop circuit

4.2.7.2. Non-overlapping clock generator

The circuit in Figure 4.13 generates two non-overlapping clock waveforms from an input clock. The two back-to-back NOR gates form an S-R latch. The S-R latch does not allow both outputs to be high at the same time. If T_{clk} is the period of the input clock of duty cycle 0.5, ΔI is the delay in the input inverter, and ΔN is the summed delay in each [NOR gate + buffer] group, then the duty cycles of ϕ_1 and ϕ_2, η_1 and η_2, are given by

$$\eta_1 = 0.5 - \frac{\Delta N + \Delta I}{T_{clk}} \tag{4.53}$$

$$\eta_2 = 0.5 - \frac{\Delta N - \Delta I}{T_{clk}} \tag{4.54}$$

The delay between the falling edge of ϕ_1 and the rising edge of ϕ_2, and vice-versa, is ΔN. The clock generating circuit is not perfectly symmetric and the duty-cycles in ϕ_1 and ϕ_2 are not equal. The inverter, required to generate the complement of the input clock, is the most critical from a symmetry point of view. The inverter is therefore sized to have a larger fanout, and the delay ΔI is minimized.

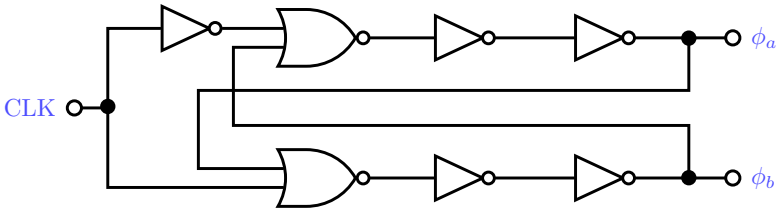


Figure 4.13: Non-overlapping clock generation circuit

4.2.7.3. SCCMFB clock generator

Figure 4.14 shows the generator for phases ϕ_1^{cmfb}, ϕ_2^{cmfb}, and their complements. The SCCMFB circuits do not need to have non-overlapping phases to work properly, but such a circuit was used because of its simplicity to generate two phases.

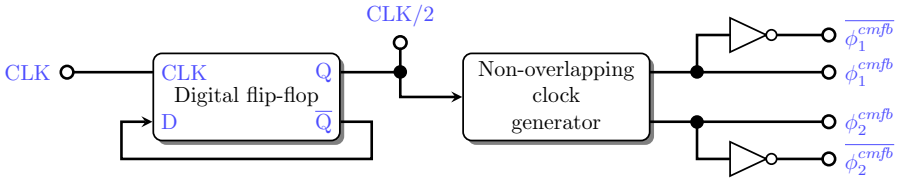


Figure 4.14: Generation of the phases for the SCCMFB circuits

4.2.7.4. ϕ_0 phase disabling circuit

The generator offers the possibility of charging an initial differential voltage V_a at its output. As previously stated, this feature is useful for calibration purposes, but could also be used for techniques such the double-histogram technique for ADC testing. The digital circuit in Figure 4.15 creates phases ϕ_{2ae}, ϕ_{AZe}, and their complementary phases from phases ϕ_0, ϕ_{2a}, and ϕ_{AZ}. The circuit simply disconnects the input branches and the auto zero of the generator from the output path while phase ϕ_0 is active. It prevents the generation of unwanted offsets due to ramp generation or offset cancellation during this phase.

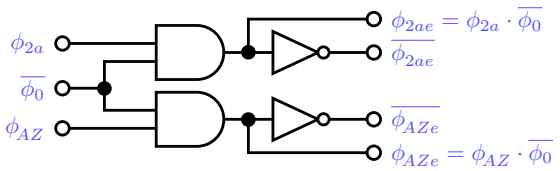


Figure 4.15: Circuit for integrator disabling during phase ϕ_0

4.3. Practical implementation

This section details the implementation of the proposed generator in a CMOS 65 nm technology with respect to a targeted ADC. The fully-differential design of the ramp generator is presented as well as the sizing of the circuit.

4.3.1. Fully-differential ramp generator for static linearity ADC test

The target of this design is a 11-bit, 5-stage, 80 Msps pipeline ADC designed in a CMOS 65 nm technology from STMicroelectronics. Following the guidelines considered in section 4.2, we can size the specifications for the generator such as gain and phase margin of the op-amp, GBW, SR, input and output range, etc. The list of the target specifications is shown in table 4.1.

Table 4.1: Goal specifications for the integrator

Parameter	Units	Design target
A_0	dB	>80
PM	°	>60
GBW	MHz	>90
Common mode input range	V	>2
Common mode output range	V	>2
SR	V/μs	>100
$t_{settling}$	ns	<100
Nominal clock frequency	kHz	200
Minimum clock frequency	kHz	1
I_{bias}	μA	lowest
Power consumption	mW	lowest
C_{i_1}	pF	0.999
C_{i_2}	pF	1
$C_{i_1} - C_{i_2}$	fF	1
C_F	pF	10
C_{AZ}	pF	1

A schematic of the fully-differential SC integrator is shown in Figure 4.16 and its associated timing is shown in Figure 4.17. The generator was designed in a CMOS 65 nm technology from STMicroelectronics with 7 levels of metal. Standard-V_t transistors were used for the design with a single-ended supply voltage of $V_{dd} = 2.5$ V.

In Figure 4.16, several elements are to be noted. Digital control signals ϕ_{up} and ϕ_{down} set the sign of the ramp slope for each integration step. Signal ϕ_{reset} performs a reset

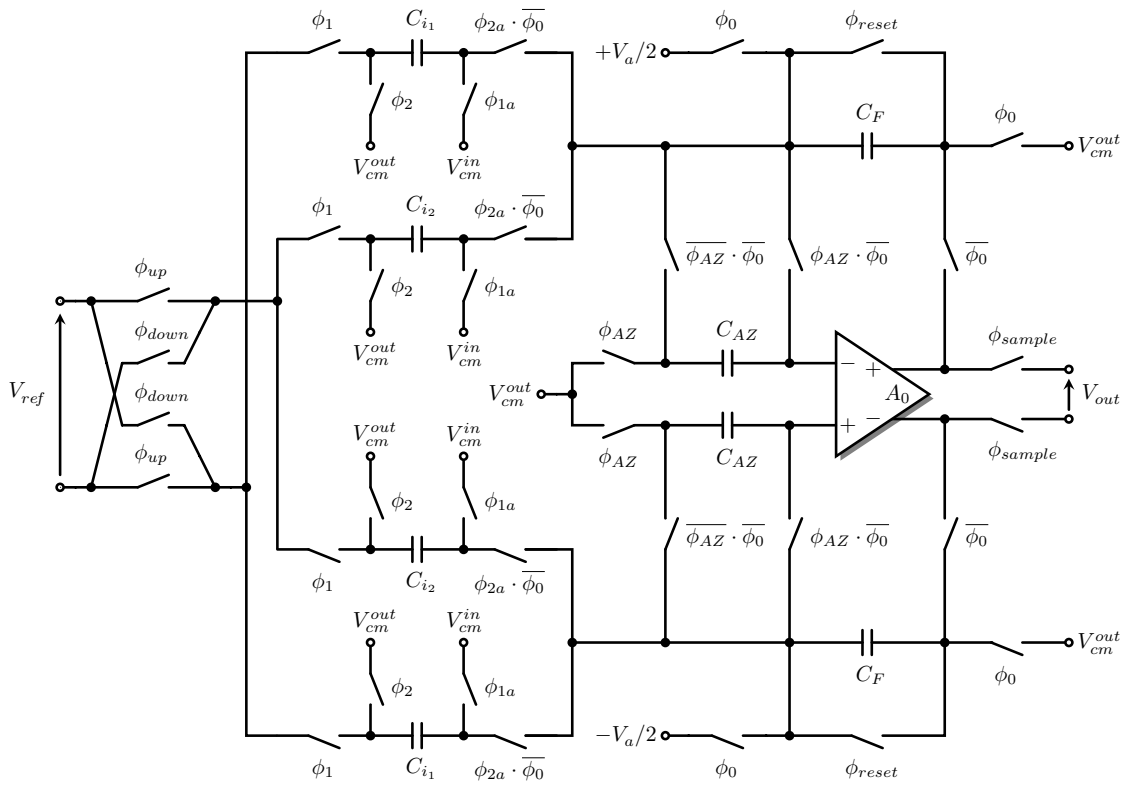


Figure 4.16: Proposed fully-differential switched-capacitor ramp generator

of the integrating capacitor, and signal ϕ_0 is added to charge the output of the integrator to a reference voltage V_a for measurement calibration purposes. The interested reader is referred to [**?**] for a detailed description of the measurement calibration procedure.

For the static linearity test of the ADCUT, the step size of the ramp generator is set to 1/5 of the LSB of the ADC. This is a good value in order to provide a good trade-off between ramp resolution and test time. With an ADC FSR equal to 2 V, the step size is approximately equal to 97.6 μV, so we chose a value of 100 μV, which does not degrade the performance of the generator and is easier to implement.

The integrator is driven by a maximum clock frequency of $T_{clk} = 200$ kHz. As depicted in Figure 4.17, the integrator output updates at half the frequency of T_{clk}.

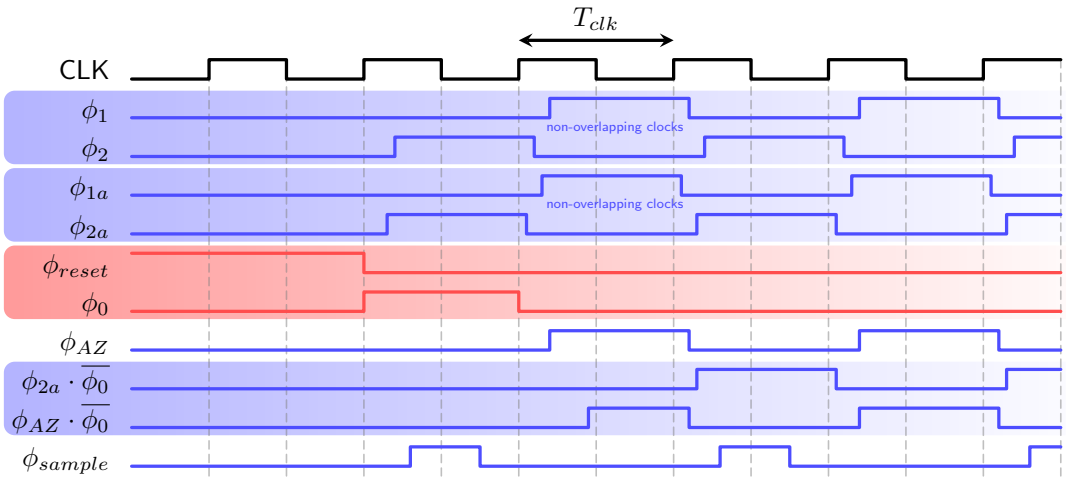


Figure 4.17: Timing for the SC integrator

The g_m/I_d methodology [**?**] has been used in order to size the transistors and other elements according to the specifications. The transistors in the gain stages have been designed in moderate inversion, whereas the load transistors were designed in order to properly reduce the op-amp noise [**?**], which is the best trade-off for limiting area overhead and power consumption. The size of each element composing the op-amp is given in Table 4.2.

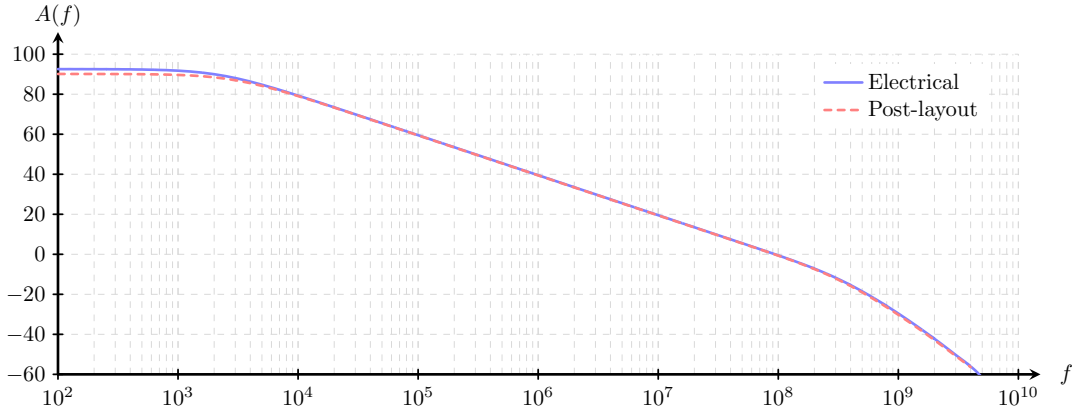
Table 4.2: Sizing of the op-amp and SCCMFBs

Instance	W/L (μm/μm)	Instance	W/L (μm/μm)	Instance	# of fingers (X/Y)
M1−M1'	100/0.3	**M7−M7'**	12/0.3	C_C	115/68×4 in //(2.94 pF)
M2−M2'	32/0.8	**M8−M8'**	300/0.3	C_{ch}	48/110 (495 fF)
M3−M3'	30/0.48	**M7FB−M7FB'**	0.5/0.3	C_h	48/110 (495 fF)
M4−M4'	310/1	**M8FB−M8FB'**	175/0.3		
M5−M5'	310/1	R_Z	3/9.9 (601 Ω)		
M6	5/2				

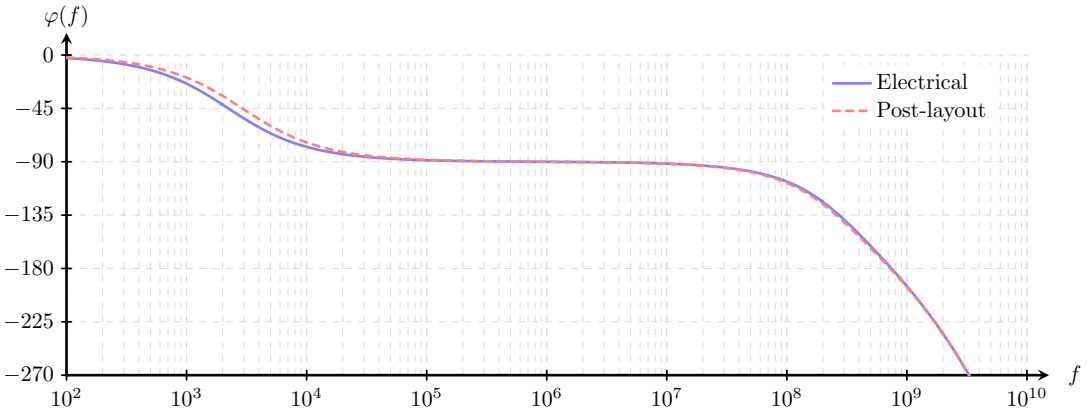
A Bode plot of the op-amp magnitude and phase is shown in Figure 4.18. The op-amp gain is equal to 92.5 dB, which is high enough for an effective reduction of the non-idealities discussed in section 4.2. The GBW is equal to 95 MHz, for a PM of 75 °.

Table 4.3: Sizing of the bias voltage generator

Instance	W/L (μm/μm)	Instance	W/L (μm/μm)	Instance	W/L (μm/μm)
M10	100/1	**M23**	25/2	**M36**	1/1
M20	25/2	**M33**	11.5/2	**M46**	1/10
M30	20/2	**M14**	40/5.5	R_{bias}	0.56/14
M40	31/1	**M24**	40/5.5		
M11	25/1	**M34**	50/2.8		
M21	25/2	**M44**	50/2.8		
M31	20/2	**M15**	10/1		
M41	31/1	**M25**	10/10		
M22	5/2	**M35**	30/3		
M32	20/2	**M45**	30/3		
M42	31/1	**M16**	10/1		
M13	25/1	**M26**	1/1		



(a) Magnitude plot



(b) Phase plot

Figure 4.18: Bode plot of the op-amp in open-loop operation

The sizes of the bias generator components are listed in table 4.3.

Figure 4.19 shows the total input-referred squared noise spectrum of the op-amp. The noise is composed of a flicker noise component and a thermal noise component. The

input-referred squared thermal noise is equal to $1.765 \cdot 10^{-15}$ V^2/Hz, while input-referred squared flicker noise is equal to $2.564 \cdot 10^{-12}$ V^2/Hz, giving an equivalent RMS input noise of 89.71 μV.

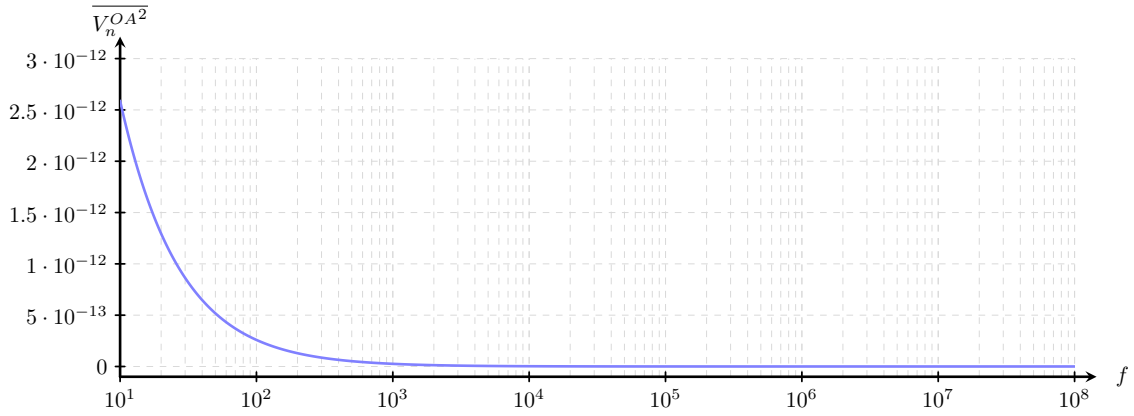


Figure 4.19: Input-referred noise of the op-amp

Table 4.4 lists the size of the transistors used for the CMOS switches.

Table 4.4: Sizing of the CMOS switches

Instance	W/L (μm/μm)
$\mathbf{M}_n$	0.8/0.3
$\mathbf{M}_p$	1/0.3
$\mathbf{M}_{dummy}$	0.5/0.3
$\mathbf{M}_n^{CMFB}$	0.3/0.2
$\mathbf{M}_p^{CMFB}$	2/0.3

Once all transistors are sized, electrical simulations and post-layout simulations of the extracted view is performed. The results are listed in table 4.5. The parasitics degrade the performance of the integrator, but the values are still well over the specifications and in very good agreement with the electrical simulation results.

Table 4.5: Ramp generator parameters

Parameter	Units	Design target	Electrical simulation	Post-layout simulation
A_0	dB	> 80	92.5	90.1
PM	°	> 60	75	74
GBW	MHz	> 90	95	93
Common mode input range	V	> 2	$+2.39/-2.38$	$+2.27/-2.26$
Common mode output range	V	> 2	$+2.28/-2.27$	$+2.19/-2.17$
SR	V/μs	> 100	156	110
$t_{settling}$	ns	< 100	60	77
Nominal clock frequency	kHz	200	–	–
Minimum clock frequency	kHz	1	–	–
I_{bias}	μA	lowest	286	–
Power consumption	mW	lowest	23	–
C_{i_1}	pF	0.999	1.0246	–
C_{i_2}	pF	1	1.0256	–
$C_{i_2} - C_{i_1}$	fF	1	1	–
C_F	pF	10	0.966 $\times$10 in //	–
C_{AZ}	pF	1	1.0246	–

4.3.2. Layout of the generator

4.3.2.1. General considerations

The layout of the proposed circuit is shown in Figure 4.20, and a photomicrograph of the fabricated die is shown in Figure 4.21. Special attention must be taken on the parasitics which can notably degrade the performance of the generator. To that end, we followed standard layout design rules during the layout design phase. The circuit is designed to be the most symmetric possible with respect the vertical axis. Moreover, we separated the analog part (op-amp, bias voltage generator) from the digital part (main clock generator), while the mixed-signal blocks (SCCMFBs, capacitors and switches) are placed in between. The aim is to place the most sensitive parts far away from the noisy digital circuits and reduce the coupling noise from the substrate due to path crossing between analog and digital lines. If analog and digital lines were to cross each other, a grounded metal surface is put at the intersection between them at an intermediate layer so that their interaction with parasitic capacitances is minimized. Separated power supplies are considered for the digital and mixed-signal parts. The final surface of the circuit is 370 μm $\times$ 380 μm which can represent up to a 10 % of the core area of a commercial ADC die. This BIST area overhead is deemed acceptable by the industry.

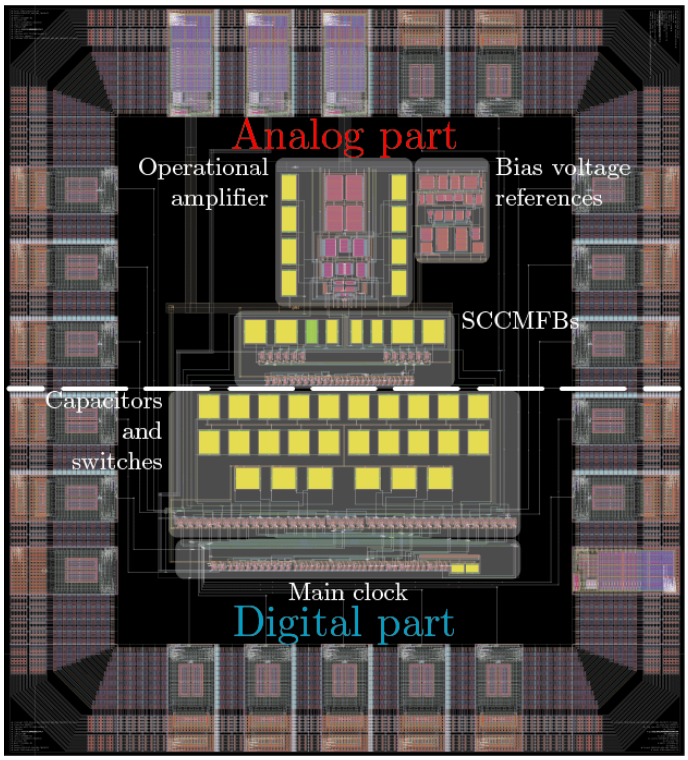


Figure 4.20: Layout of the proposed generator

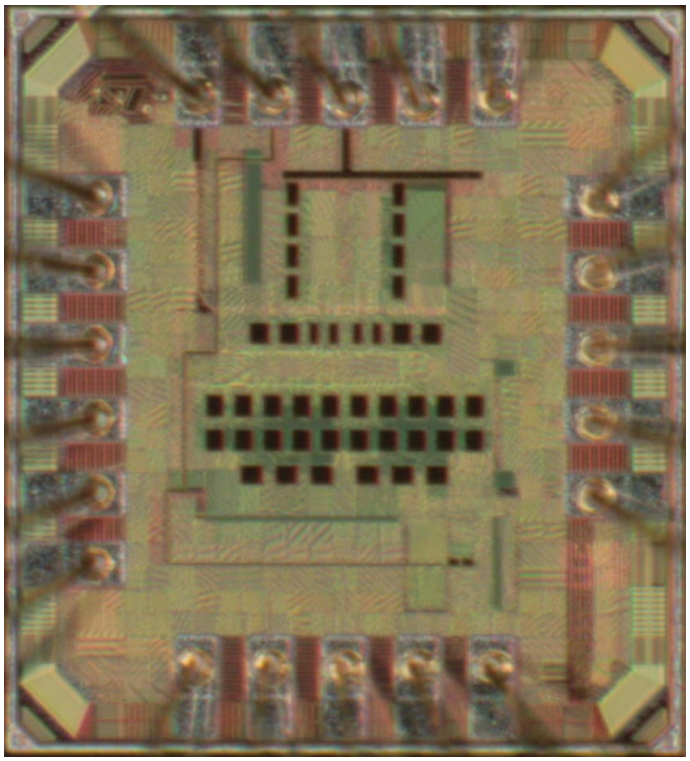
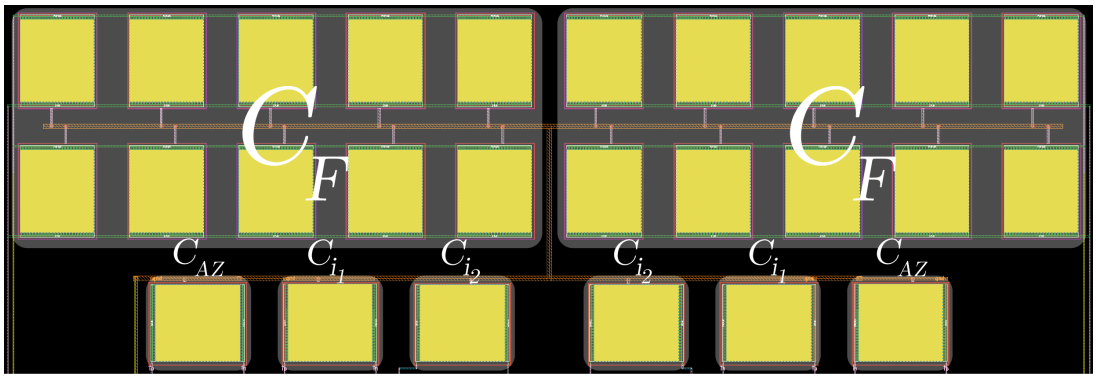


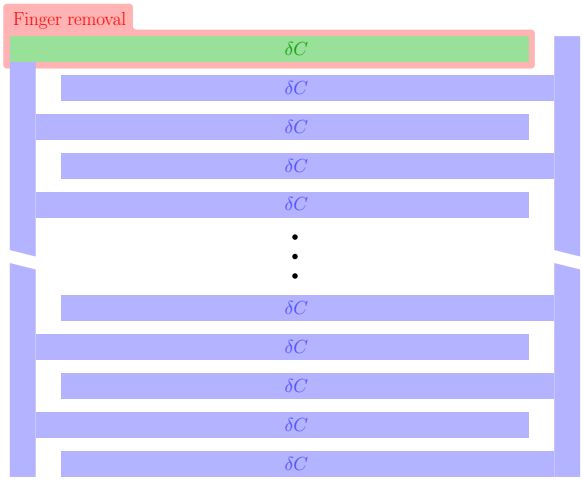
Figure 4.21: Die photomicrograph of the proposed generator

4.3.2.2. Capacitors

For the implementation of the capacitors of this circuit, we decided to use RF metal-on-metal capacitors (RF MOM). The linearity and matching of these devices are very good, so the performance of the generator is not compromised by the nonidealities of the MOM capacitors. The capacitor generation is simple with this type of capacitors, as we only have to play with the number of fingers in order to vary the capacitance, which was useful for layout purposes. Moreover, we designed the MOM capacitors such as the required capacitor difference, $C_{i_2} - C_{i_1} = \delta C$, corresponds exactly to the capacitance value of one finger. Figure 4.22 (a) shows the layout of capacitors C_{i_1}, C_{i_2}, C_{AZ}, and C_F, while Figure 4.22 (b) shows the finger removal from C_{i_1} in order to create the required virtual capacitor.



(a) Layout of capacitors C_{i_1}, C_{i_2}, C_{AZ}, and C_F



(b) Finger removal on C_{i_1} for the generation of a small capacitor difference δC

Figure 4.22: Top view of the capacitor layout

Chapter 5

Simulation and experimental results

5.1. Simulation results

The simulation results of the generator and the proposed BIST technique are presented in this section. The ramp generator simulation results consist in the characterization of the linearity of the generator output, while the results of the proposed BIST technique show its performance when combined with the reduced code testing algorithm developed in [41].

5.1.1. Ramp generator simulation results

As it was presented in the previous chapter, the proposed ramp generator was designed using STMicroelectronics 65nm CMOS technology. The ADCUT is an 11-bit 2.5 bits/stage pipeline ADC with a 1 V full scale. Given this target application, our design goal is a ramp generator that is able to generate a linear stepwise ramp with a step magnitude of around 100 μV, that is, around 0.2 LSB regarding the target ADC, in the voltage range from 0 to 1 V.

The ramp generator is simulated under typical process conditions, with a 200 kHz clock frequency, and using a 2 V differential DC input as voltage reference. The generated stepwise ramp has a nominal resolution of 13.3 bits, with an average step of 96.7 μV. As a measure of the ramp linearity, Figure 5.1 shows the difference between each integration step in the generated ramp and the average step. As it can be seen, the deviation of the step size is very well contained within the range $[-3, 3]$ μV along the considered voltage range. We define the slope error per step as the difference between the slope of the ramp connecting the middle points of two consecutive steps and the slope of a linear fit to an ideal ramp that crosses through the middle points of the steps. The average slope error, that is, the average of slope errors per step over all steps, is computed to be -0.040 %. In the same way, we can compute the INL of the generated ramp from the values in Figure 5.1, defined as the difference between the actual position of the middle point of

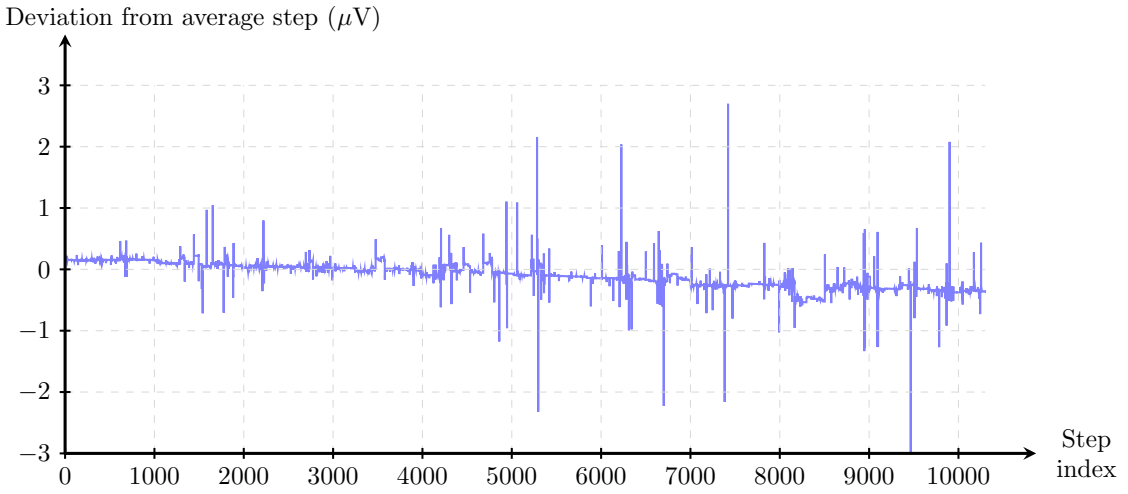


Figure 5.1: Deviation of the magnitude of the steps with respect to the average ramp step

each step and its ideal position. The complete INL characteristic of the generated ramp is shown in Figure 5.2. The maximum INL of the generated ramp is around 580 μV, that is, around 6 average steps. Figure 5.1 clearly shows a constant linear decrease of the ramp step size across the considered full-scale that can be explained by the effect of the integrator leakage. Consequently, the INL curve in Figure 5.2, obtained by integrating the DNL curve, shows a very characteristic parabolic shape. These results indicate that the main source of static non-linearity in the proposed ramp generator design comes from the integrator leakage and, to a lesser extent, from the residual integrator offset. It is also worth noticing that the proposed measurement strategy takes advantage of the smooth shape of the INL curve and the well-contained evolution of the ramp step size. Since the proposed reduced-code linearity test technique only requires the measurement of the widths of a limited set of codes, the ramp stimulus is only required to be locally linear in the vicinity of the target codes, as long as the ramp step size is relatively constant and only a small fraction of the LSB.

In order to further validate the performance of the proposed ramp generator under different operation conditions, Figure 5.3 shows the deviation of the step size of the generated ramp across the complete voltage range of the ramp for corner temperatures T = 0 °C and T = 75 °C and for the nominal temperature T = 27 °C. As it can be observed, the impact of temperature variations in the linearity of the ramp is well controlled. The resulting average step sizes are 96.9 μV and 97.1 μV for T = 0 °C and T = 75 °C, respectively. Compared to the 96.7 μV average step size for T = 27 °C, the change of the magnitude of the average step size due to stressed temperature operation is below 0.5 %.

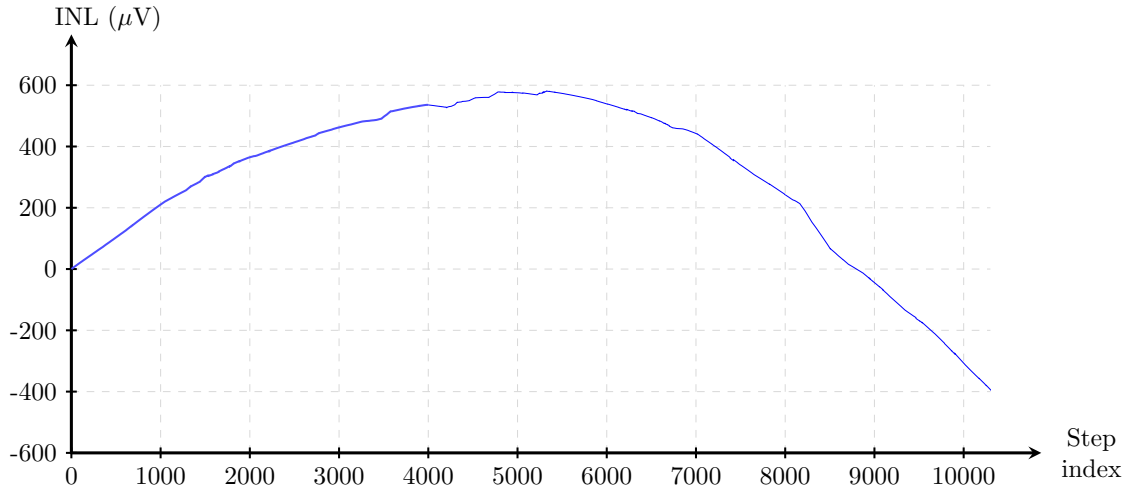


Figure 5.2: INL of the generated ramp

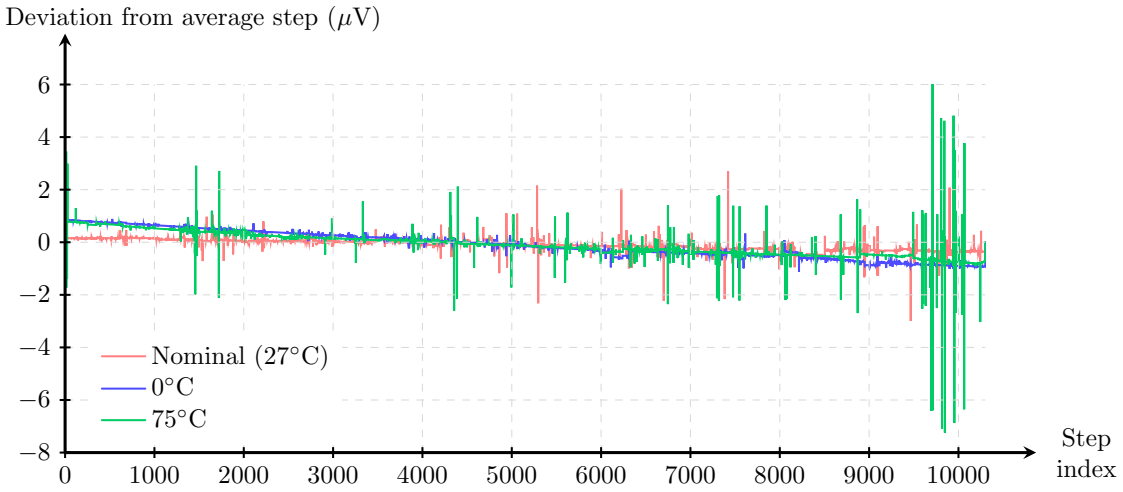


Figure 5.3: Deviation of the magnitude of the ramp steps with respect to the average ramp step at different operation temperatures

In order to show the impact of process variations and mismatch on the performance of the proposed ramp generator, we carried out a set of Monte Carlo transistor-level electrical simulations of the complete system using the actual Process Design Kit (PDK) of the technology. To speed up the simulation process, during each Monte Carlo iteration we monitor at fixed time instants the step size of the generated ramp (which is relatively stable as shown in Figure 5.3), and the iteration is stopped and skipped if the step size is small and well within the design goal. In this way, we bias the Monte Carlo simulation towards worst cases and we generate such worst cases with less computational effort.

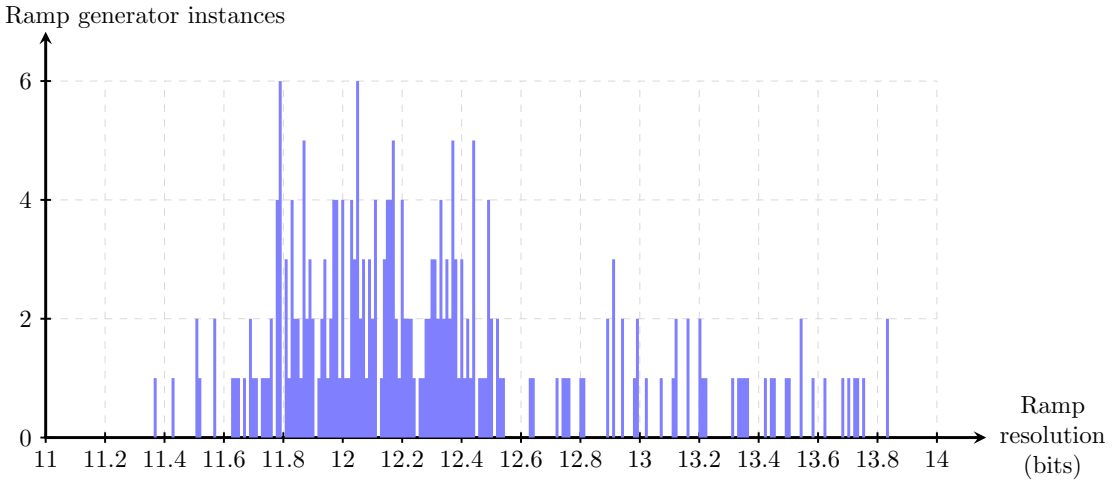


Figure 5.4: Histogram of the resolution of the generated ramp obtained by Monte Carlo process and mismatch simulations

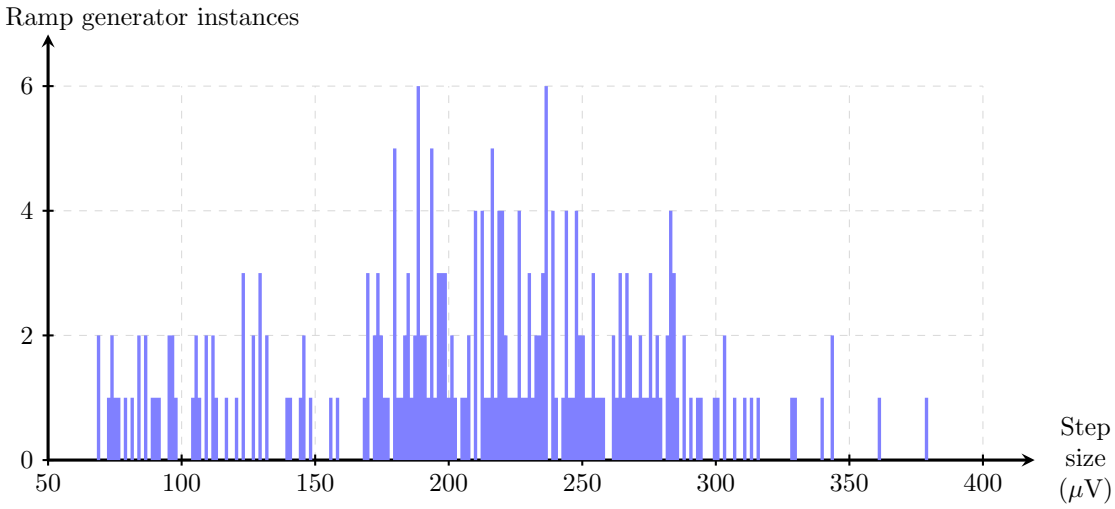


Figure 5.5: Histogram of the step size of the generated ramp obtained by Monte Carlo process and mismatch simulations

Figure 5.4 and Figure 5.5 show the histograms of the ramp resolution and average step size, respectively, for 250 Monte Carlo runs. As it can be seen, the obtained resolution and step size are well contained in the range of $11.4-13.8$ bits and $60-380\ \mu$V , respectively, which comply with the requirements of our target application. Figure 5.6 shows the histogram of the average ramp slope deviation for the 250 Monte Carlo runs. As it can be seen, this deviation is well centered around the -0.2 % value and the observed worst case value close to -6 % is still deemed very acceptable.

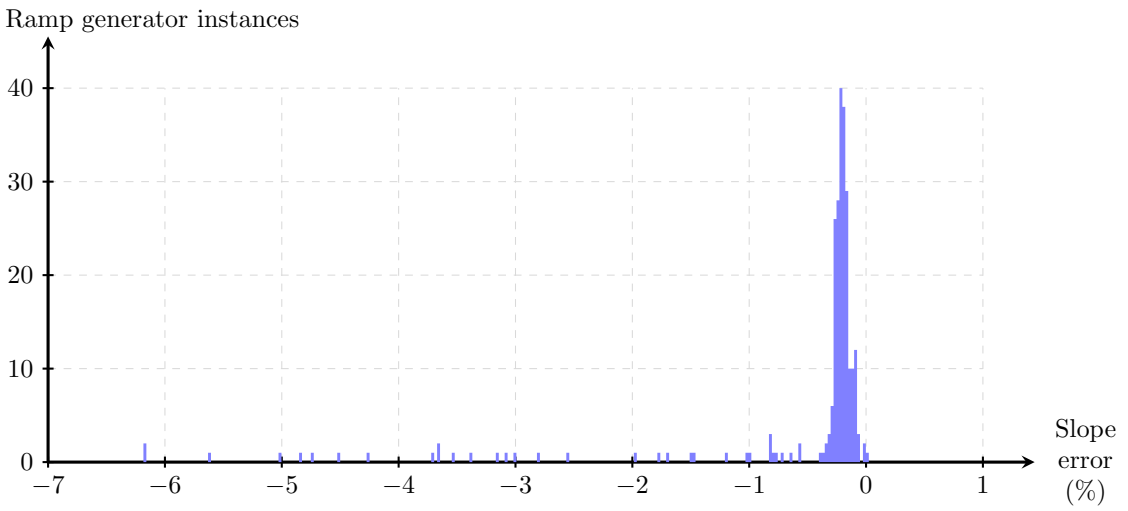


Figure 5.6: Histogram of the average slope error of the generated ramp obtained by Monte Carlo process and mismatch simulation

5.1.2. Servo-loop simulation results

In order to demonstrate the feasibility and performance of the proposed ramp generator in an actual test application, we employ the ramp generator to implement an on-chip servo-loop-based reduced-code linearity test technique for pipeline ADCs. The ADCUT is an 11-bit 2.5 bits/stage pipeline ADC IP provided by STMicroelectronics.

The complete test setup described in Figure 3.1 has been modeled in Matlab. Realistic behavioral models have been developed for the ADC and ramp generator. The ADC model has been derived from the actual static characteristic of the ADC, which was measured in the lab using the standard histogram technique on dedicated bench-top equipement. On the other hand, the ramp stimuli have been directly captured from the transistor-level electrical simulations of the complete ramp generator. The BIST engine, digital counter, and word comparator blocks are digital circuits and, thereby, for simplicity, have been considered as ideal blocks in this work.

As explained in subsection 3.4.2, we only need to measure the widths of 8 codes around the 6 transitions of the first stage, 6 codes around the 6 transitions of the second stage, 4 codes around the 6 transitions of the third stage, and 2 codes around the 6 transitions for the fourth and fifth stages for a total characterization of the converter. Only 132 out of 2046 codes of the ADCUT are measured, which represents about 6 % of the total number of codes, resulting in a significant test time reduction. Figure 5.7 shows the estimated INL using the proposed BIST scheme.

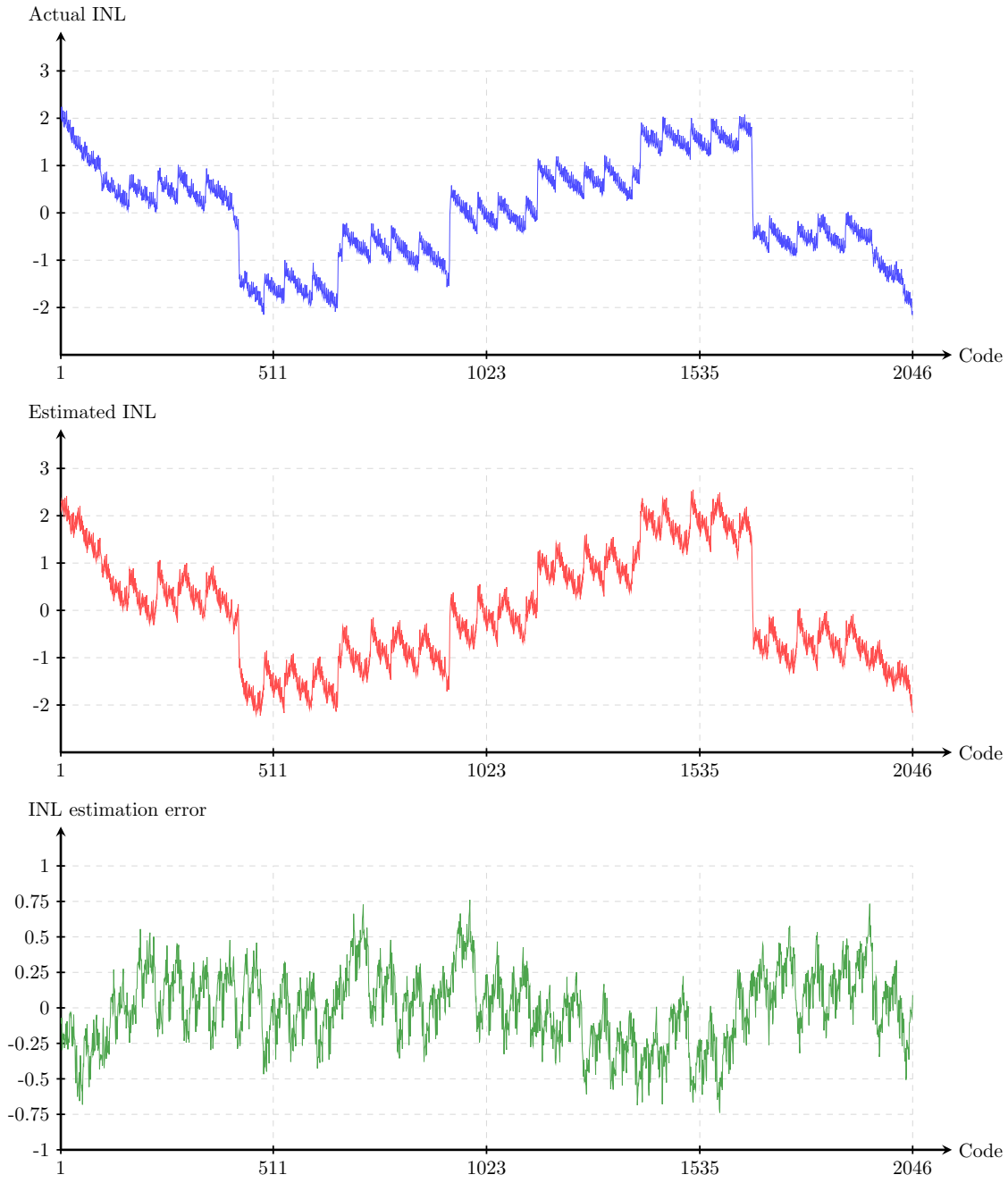


Figure 5.7: Actual INL obtained by standard histogram test, estimated INL obtained by BIST, and INL estimation error

A noise voltage of RMS value $\sigma_{ramp} = 0.2$ LSB (equivalent to 200 μV) is added to the generator output, while a noise voltage of RMS value $\sigma_{ADC} = 0.15$ LSB is added to the ADC input in order to simulate noisy transitions. Each code is traversed $r = 50$ times in the servo-loop configuration so as to average out noise effects. Figure 5.7 also shows the actual INL values obtained from a traditional histogram test for the purpose

of comparison, as well as the INL estimation error. As it can be seen, under nominal conditions the INL estimation error is well controlled in the range [−0.5,+0.5] LSB.

In order to estimate the robustness of the proposed BIST to process variations and mismatch, we performed Monte Carlo transistor-level electrical simulation of the ramp generator with 250 runs, and we employed the test set up of Figure 3.1 250 times, each time using a different ramp generator instance. Figure 5.8 shows the histogram of the maximum INL estimation error in absolute values recorded for each run. As it can be seen, the INL estimation error is always below 1 LSB with an average absolute value around 0.5 LSB. Results in [41] show that INL estimation error using an ideal ramp, that is, the INL estimation error that is exclusively due to the reduced-code linearity test technique itself, is in the range [−0.5,+0.5] LSB. According to our results in Figure 5.8, in the worst-case scenario, the non-linearity of the proposed ramp generator design only contributes to an additional 0.5 LSB to the INL estimation error introduced by the reduced-code linearity test technique.

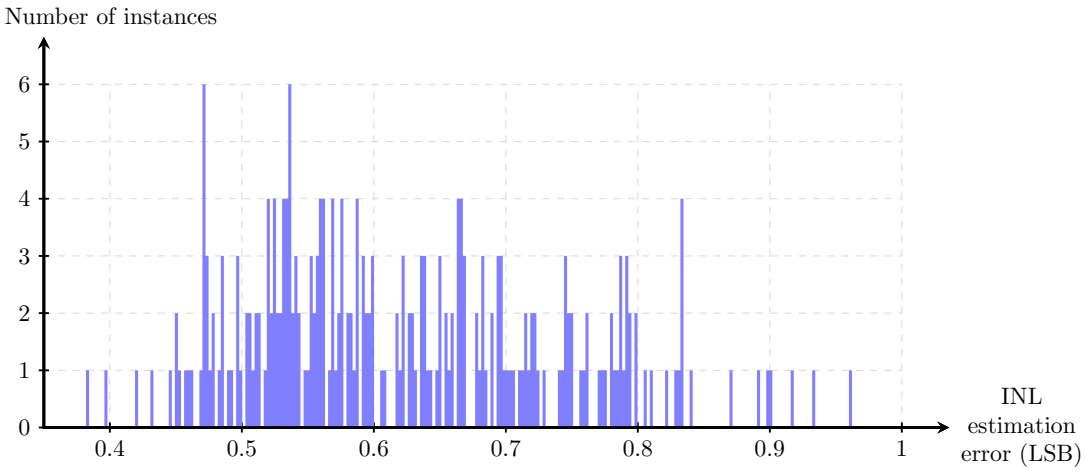


Figure 5.8: Histogram of the maximum absolute INL estimation error obtained by the BIST by assuming different Monte Carlo instances of the ramp generator

The test time for each run is computed as well. In the case where the RCLT is not used, each code of the ADCUT is measured, while only 6 % of those codes are measured when the RCLT is applied. The test time is then reduced to 6 % of the initial test time, but the test time reduction performance is degraded by the "waiting" time in between each code measurement. Indeed, the generator output must ramp between the groups of selected codes for measurement, resulting in additional test time. Moreover, the RCLT technique requires one sweep of the ADCUT transfer function in order to select the codes of interest. The ramp stimulus employed in this phase of the technique does not need to be linear, neither to be slow, but has to be monotonically increasing. However, in the case

where the ramp stimulus speed cannot be modulated, as in our case, the additional test time might be non negligible. Figure 5.9 shows an histogram of the ratio of the test time in the RCLT case over the test time when RCLT is not applied for an appreciation of the gained test time.

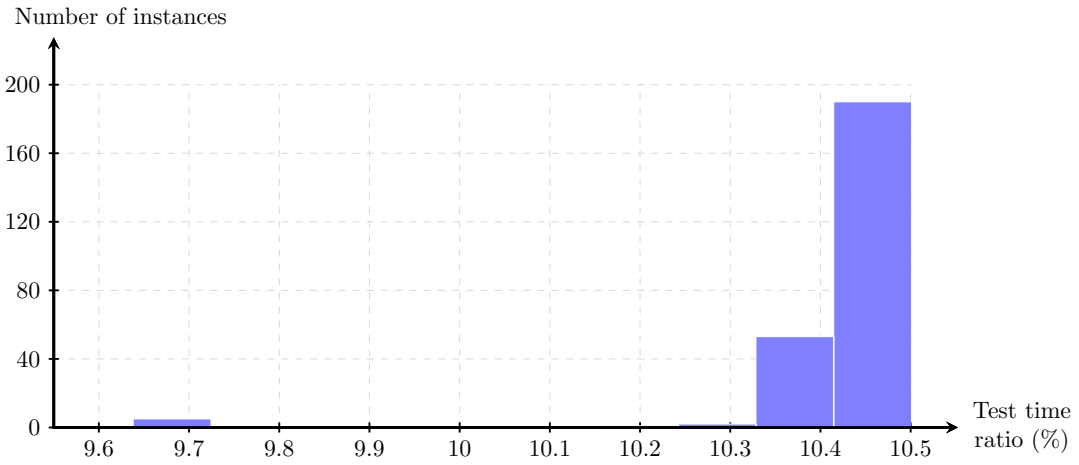


Figure 5.9: Histogram of the ratio of test times of the BIST with RCLT/without RCLT

As we can see in these results, most of the ratio instances are located around 10.4 %, while an outlier is located around 9.7 %. The computed ratios show that the test time with RCLT is ± 4.4 % higher than the expected 6% test time, which shows that the additional test time is not negligible in this case. Nevertheless, the test time reduction using the RCLT technique is very effective, as 90 % of the initial test time is saved, and can definitely compete with the histogram test technique at an equivalent accuracy level.

5.2. Experimental results

In this section, the experimental results for the ramp generator are presented. First, the ramp generator is characterized in terms of resolution, non-linearity, slope error, etc. Secondly, the ramp generator is coupled to a commercial ADC in order to assess its performance in a realistic test application.

5.2.1. Test setups

Two different test setups were devised in order to validate the performance of the integrated ramp generator. The first test setup was aimed at the characterization of the ramp generator performance. Figure 5.10 shows the proposed setup for the test of the ramp

generator. In this setup, a PC controls a National Instruments USB-6361 data acquisition card (DAQ). This card can capture a differential analog signal with a 16-bit linearity. The DAQ controls the UP, DOWN, RESET signals, and sets the frequency f_{CLK} of the ramp generator. The generator differential output is fed back to the DAQ which captures the signal at frequency f_{CLK}. The PC then retrieves the data which is processed in Matlab in order to characterize the ramp generator. An oscilloscope is also connected to observe and debug the evolution of the generator output.

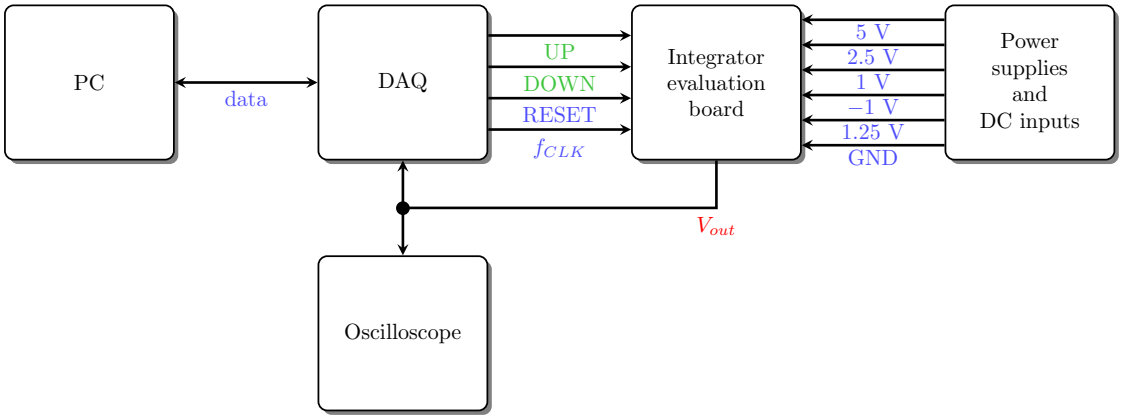


Figure 5.10: Schematic of the test setup for the characterization of the ramp generator

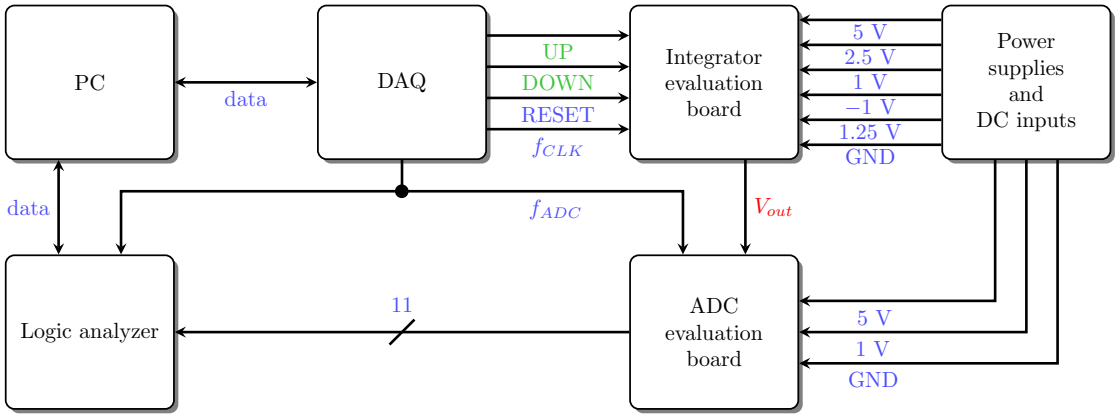


Figure 5.11: Schematic of the test setup for the ADC test

In order to employ the generator in an actual test application, an histogram-based ADC test was performed using a commercial ADC as DUT which reference is AD9240 from Analog Devices. The AD9240 is a 10 Msps, 14-bit, (4.5+3·3.5) bits/stage pipeline ADC, which FSR was set to 2 V for the experiment. Figure 5.11 shows the proposed setup for the ADC test case. In this setup, the DAQ still controls the integrator with the same signals, while the ADC evaluation PCB is connected at the differential buffer output of the generator board. The DAQ sets the operating frequency of the generator and the ADCUT,

respectively f_{CLK} and f_{ADC}. The ADC output is fed to a logic analyzer which captures the ADC output at a frequency rate of f_{ADC}, and the PC retrieves the collected data for further processing using Matlab and consisting in the DNL and INL evaluation.

Figures 5.12 and 5.13 show respectively the layout of the ramp generator and ADC evaluation PCBs, while Figures 5.14 and 5.15 detail the schematic for each device. Special care was taken for the routing of the differential paths, proper decoupling of the DC inputs and power lines, and proper grounding in order to minimize the influence of the PCBs on the measurements.

The ramp generator is encapsulated in a JLCC28 package, so a PLCC28 socket was placed on the evaluation PCB of the ramp generator for a quick insertion and removal between each ramp measurement. The DC nature of the ramp generator implies the need of an analog buffer at the ADCUT interface that provides driving capability and isolation. This requirement is mandatory when dealing with high-speed high-accuracy ADCs. Without this buffer, the ramp generator can only work with ADCs that either incorporate the buffer in their architecture, or have a continuous-time (CT) front-end, or operate at low sampling frequencies. Even in this last case, the generator must have a time constant significantly lower than the sampling time, requiring higher power consumption with respect to the situation with a dedicated buffer (since achieving high bandwidth is more efficient for buffers than for generator topologies [31], [32]). To that end, a commercial, low-noise, high-linearity differential buffer was added to the generator output on its evaluation PCB for effective kick-back noise reduction. The amplifier used in the buffer is referenced as AD8138 from Analog Devices. The AD8138 is a low-distortion, low-noise, differential ADC driver with a 4 V-p-p output swing which matches with the ramp generator output swing. The resistors are chosen low enough in order to minimize the noise at the buffer output without limiting the output swing.

The following supply voltages are fed to the evaluation PCB of the integrator chip: a 2.5 V single supply for the analog part of the integrator, a 2.5 V single supply for the digital part of the integrator, -1 V/1 V in order to generate a 2 V differential DC input, 1.25 V added to the 2 V input in order to establish the input common-mode voltage of the integrator, a 5 V single supply for the buffer completed with 2.5 V for its common-mode, and 5 V for the inverters on the digital signal path.

The evaluation PCB for the ADCUT was designed following the recommendations of its datasheet. The analog and digital grounds must be joined at some point to allow signals to be referred to a common potential. A star point, placed right under the ADCUT, was carefully chosen so as not to introduce digital currents into the ground of the analog part of the PCB, and prevent any digital noise from corrupting analog performance.

The following supply voltages are fed to the evaluation PCB of the ADC: a 5 V single supply for the analog part of the ADC, a 5 V single supply for the digital part of the ADC, a 1 V reference voltage in order to establish the FSR of the ADC.

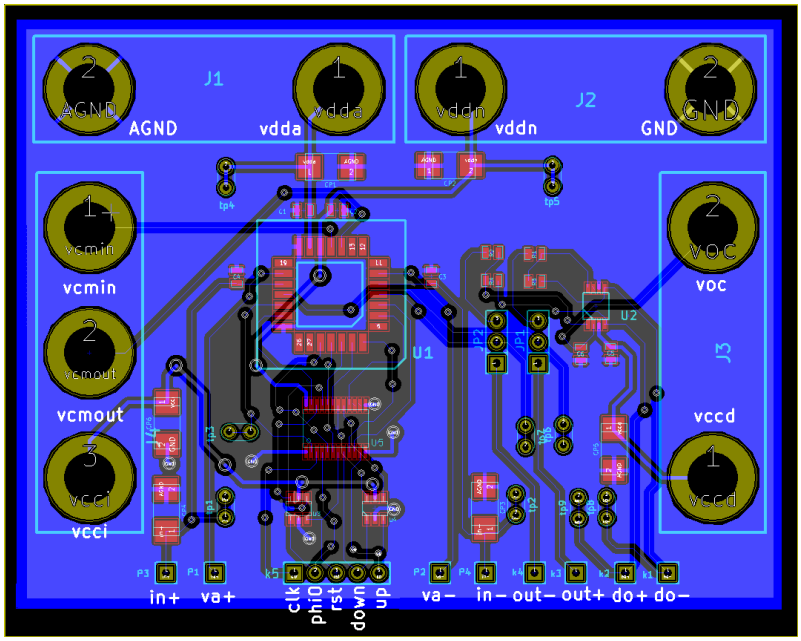


Figure 5.12: PCB layout for the ramp generator

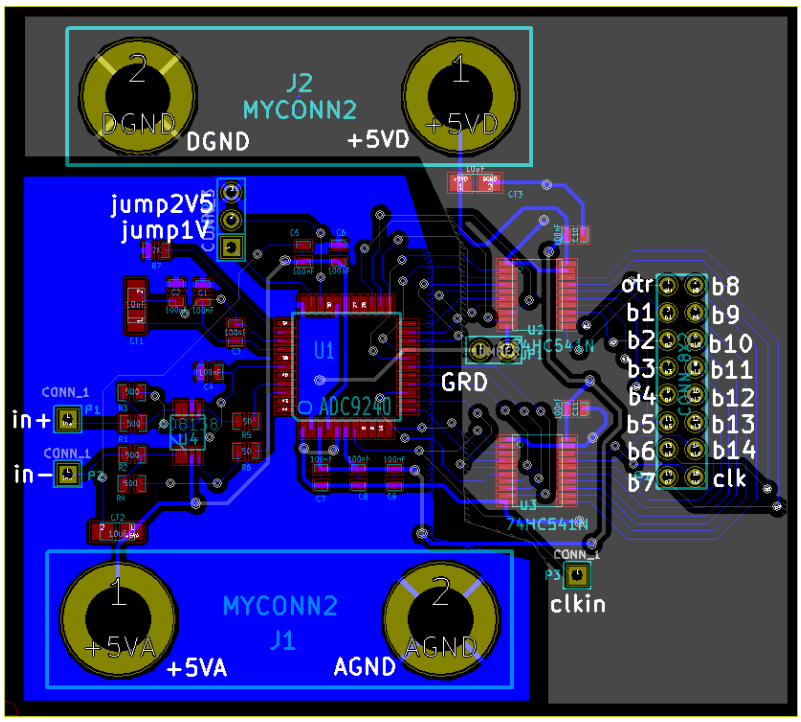


Figure 5.13: PCB layout for the ADC

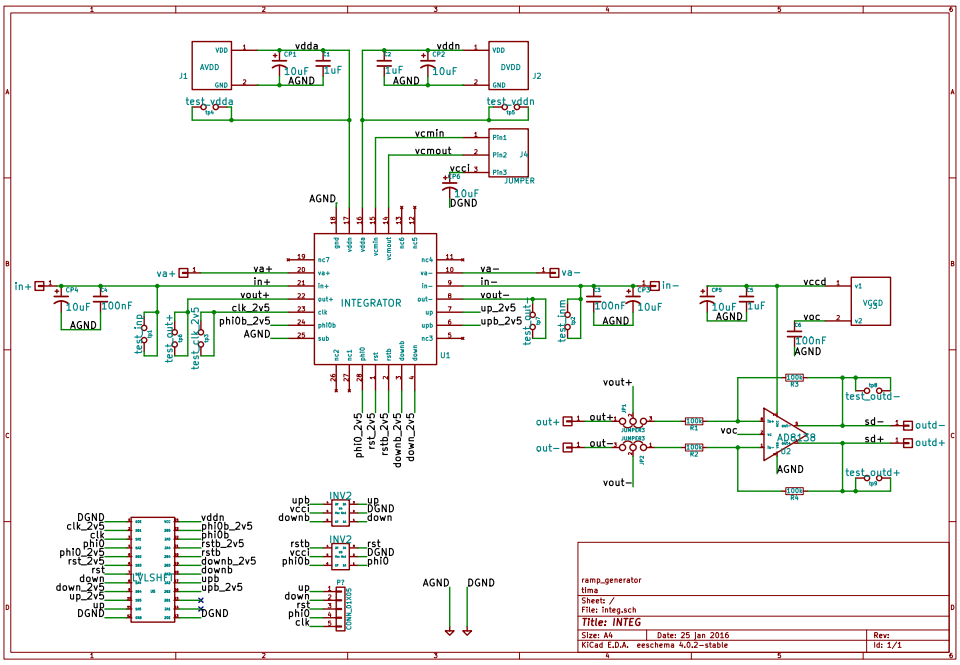


Figure 5.14: PCB schematic for the ramp generator

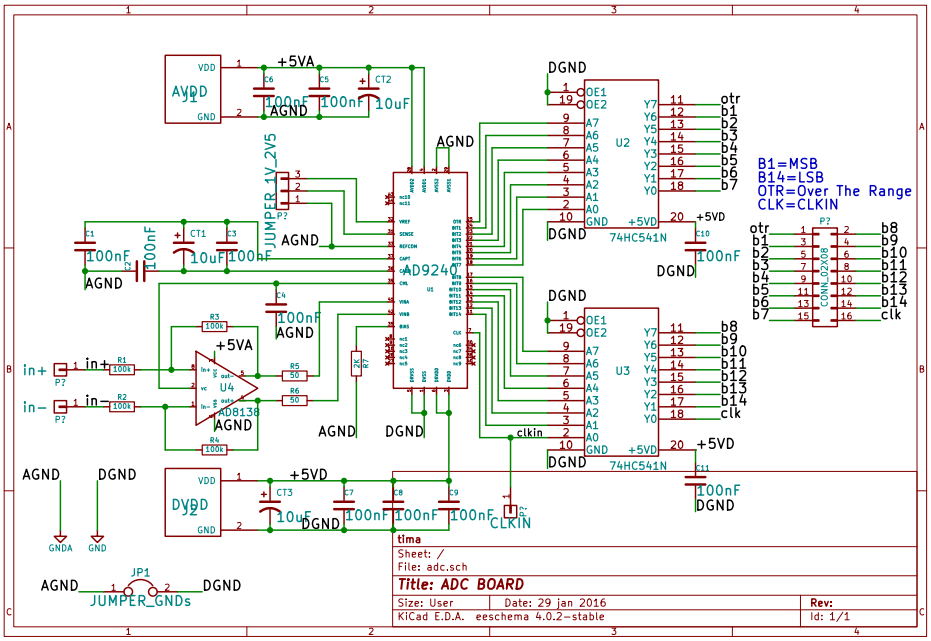


Figure 5.15: PCB schematic for the ADC

5.2.2. Ramp generator experimental results

We have a total of 15 packaged samples of the ramp generator available for characterization. The ramp generator frequency is set to $f_{CLK} = 20$ kHz. Each ramp stimulus is measured using the test setup shown in Figure 5.10 at a sampling rate of 1 MHz, which means the DAQ captures 100 samples per step. The collected data is then processed in Matlab. After averaging the 100 samples per step, the average step size, resolution, and slope error are computed. Each ramp stimulus is characterized on three different voltage ranges, that is: full range of the generator output, -1.5 V/$+1.5$ V (3 V range), and -1 V/$+1$ V (2 V range). Knowing that the measured average step size $\tilde{s}$ is computed as

$$\tilde{s} = \frac{V_{out}^{+} - V_{out}^{-}}{2^{N_r}}, \tag{5.1}$$

the ramp resolution N_r is expressed from equation (5.1) as

$$N_r = log_2\left(\frac{V_{out}^{+} - V_{out}^{-}}{\tilde{s}}\right) \tag{5.2}$$

where $V_{out}^{+} - V_{out}^{-}$ is the considered output range of the generator (full output range, -1.5 V/$+1.5$ V, or -1 V/$+1$ V), and the static equivalent number of bits of the ramp is expressed as

$$sENOB_r = log_2\left(\frac{V_{out}^{+} - V_{out}^{-}}{\max(|INL_r|)}\right). \tag{5.3}$$

Finally, the slope error is calculated as

$$\varepsilon_{slope} = \frac{\max(|DNL_r|)}{\tilde{s}} \cdot 100 \tag{5.4}$$

where $\max(|DNL_r|)$ is the maximum absolute value of the deviation from the average step in volts.

Figure 5.16 plots an histogram of the measured average step size, Figure 5.17 and Figure 5.18 show histograms of the measured resolution and static effective number of bits, while Figure 5.19 depicts an histogram of the measured slope error. Numbers are plotted in each bin in order to match the measured parameter value to the corresponding measured instance.

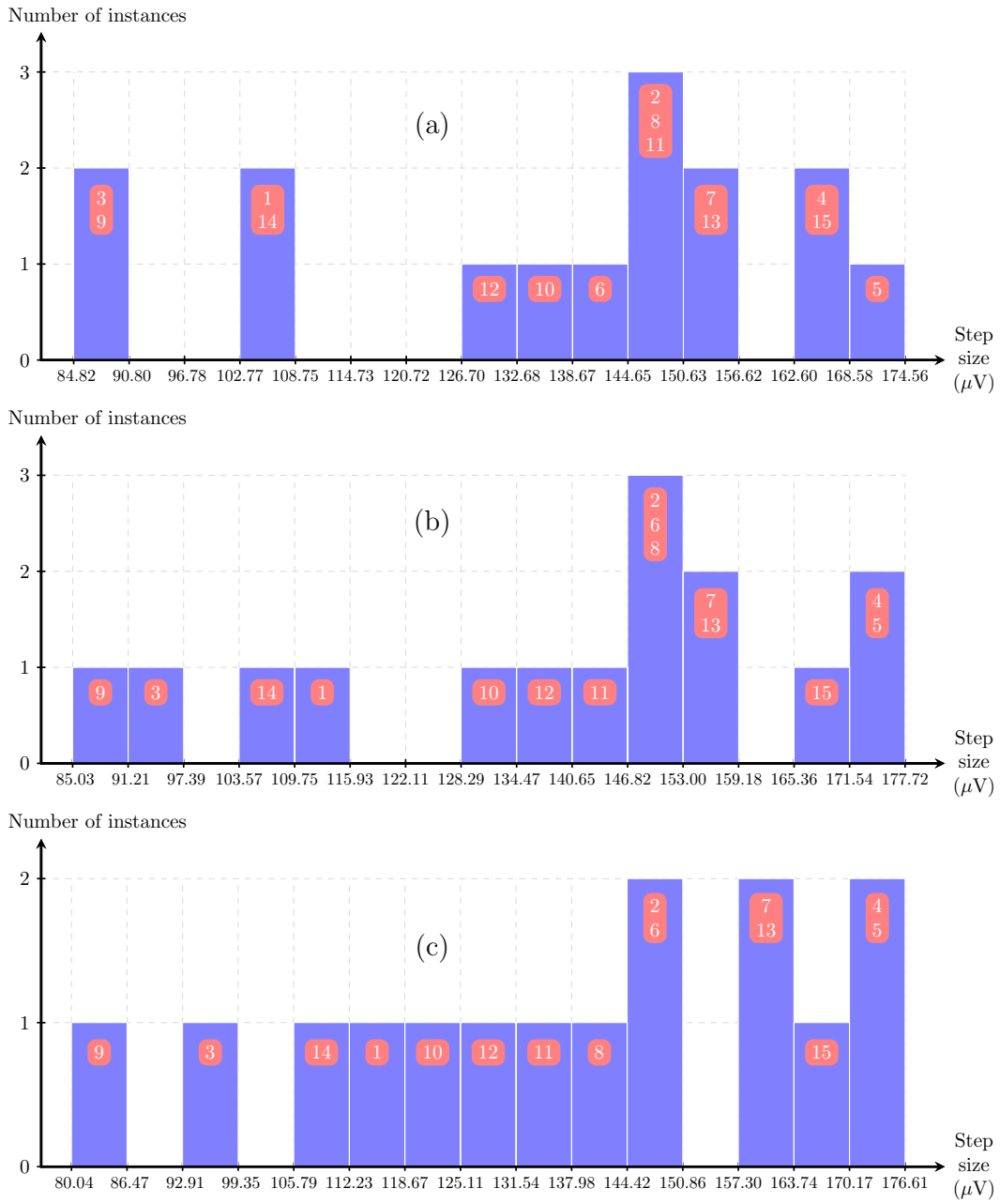


Figure 5.16: Histogram of the measured step size for the 15 samples: (a) full output range, (b) −1.5 V/+1.5 V, (c) −1 V/+1 V

Results in Figure 5.16 show that the measured average step size for the 15 samples is bound between 80 μV and 177 μV in the three considered cases. The average step size is relatively well contained around the specification, which complies with the requirements of our target application. This denotes a good capacitor matching and it shows that the concept of a very small capacitor difference is working. Moreover, we can see that the averaged step sizes of each instance do not vary much between the three cases, which indicates a good linearity for each ramp stimulus. The relatively high resolution obtained

with these samples is explained by the fact that the output range of the measured samples is large, around 4 V-p-p, moreover the measured step size is very small.

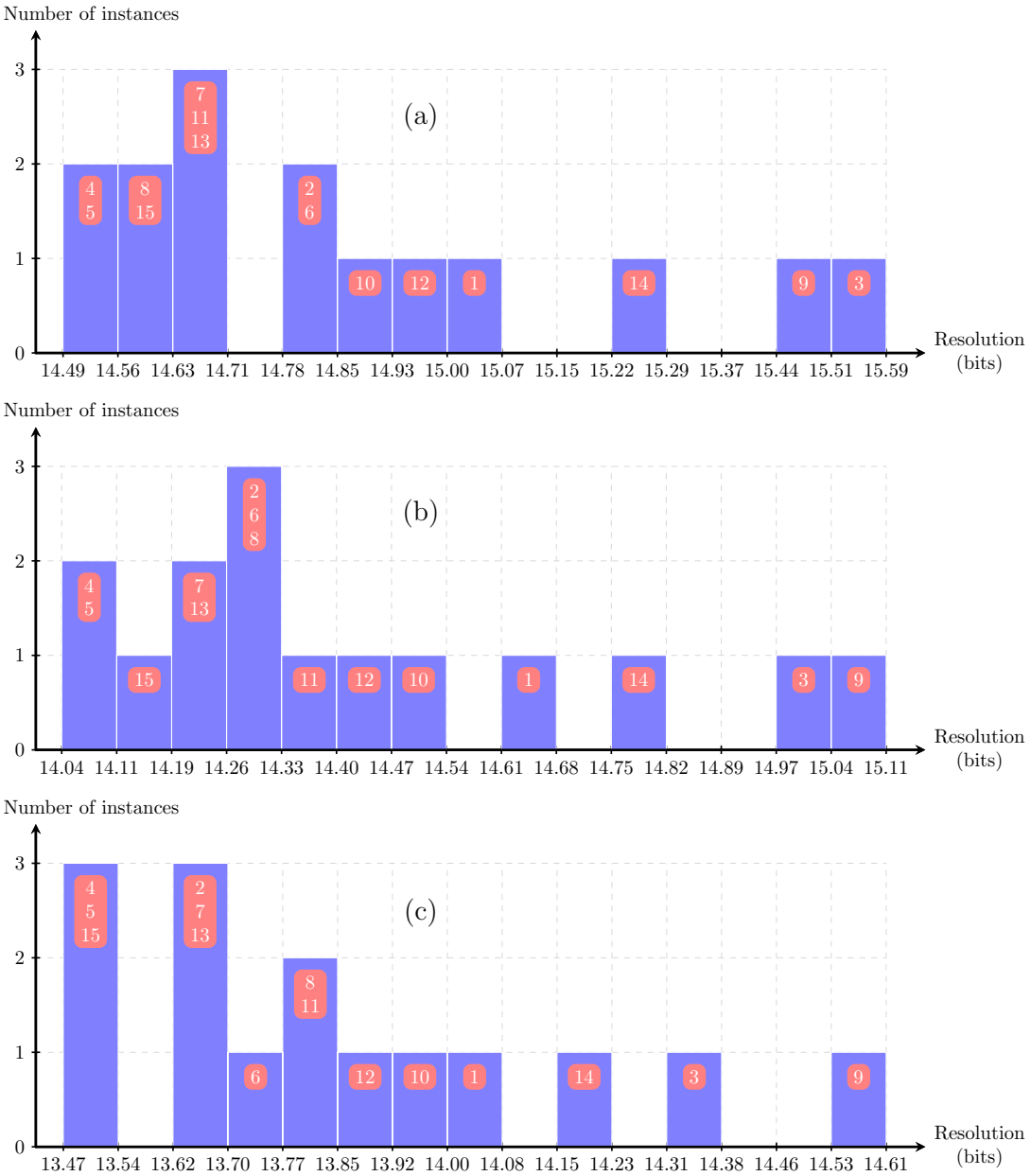


Figure 5.17: Histogram of the measured resolution for the 15 samples: (a) full output range, (b) −1.5 V/+1.5 V, (c) −1 V/+1 V

In Figure 5.17, the measured ramp resolution varies between 13.47 and 15.11 bits. The results show a good resolution for all the measured instances. The resolution becomes lower as we go toward the 2 V range case, which is explained by the fact that the range diminishes while the average step size remains constant. However, the lowest resolution

in this case is still very acceptable for the test of our target ADC, and in a general manner is clearly sufficient for testing 11−12 bits ADCs.

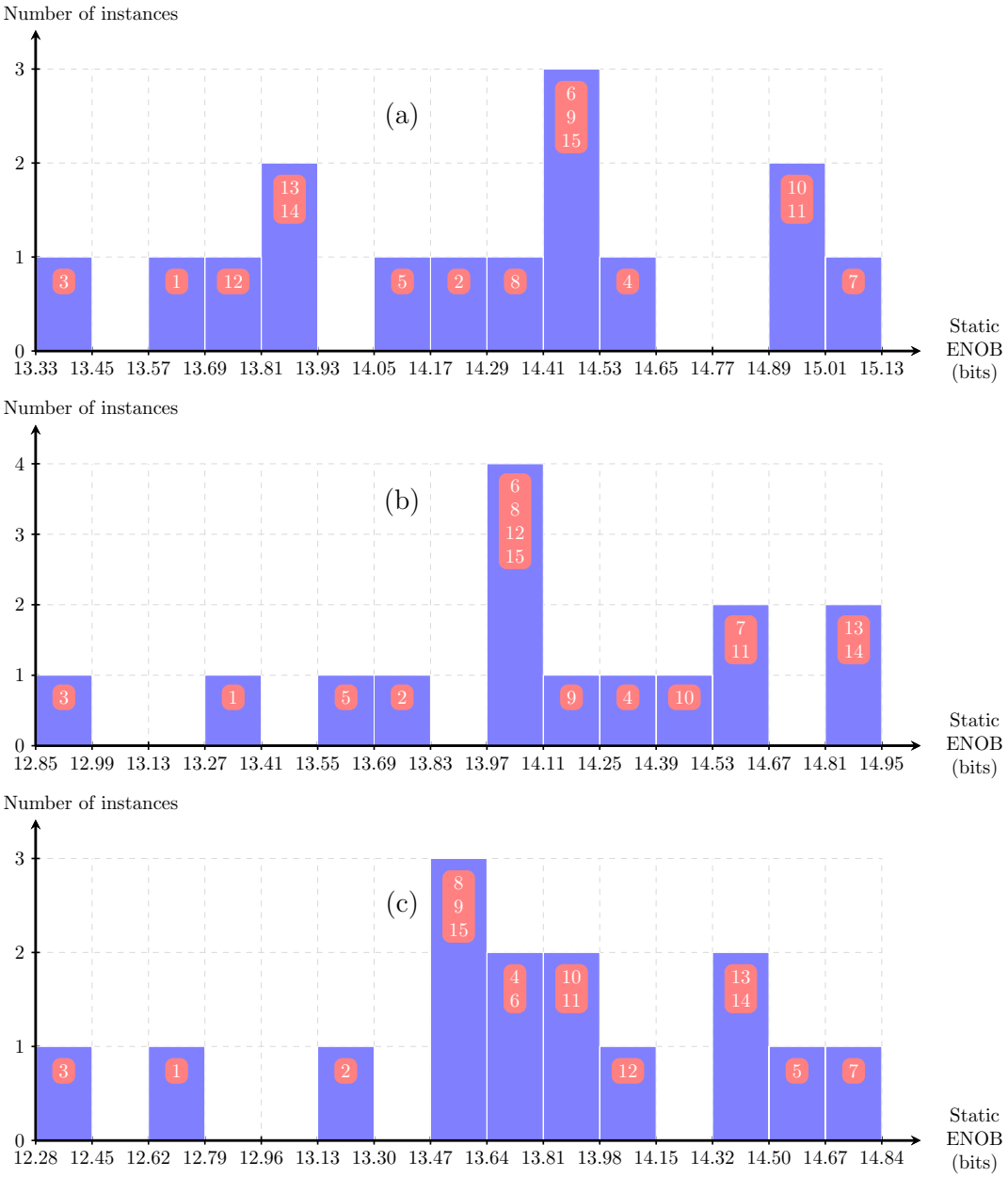


Figure 5.18: Histogram of the measured static effective number of bits for the 15 samples: (a) full output range, (b) −1.5 V/+1.5 V, (c) −1 V/+1 V

In Figure 5.18, the measured effective number of bits varies between 12.28 and 15.13 bits. The results show that if the static effective number of bits do not vary much over the three voltage ranges of interest, it means that the INL of the corresponding ramp remains relatively flat over its full range.

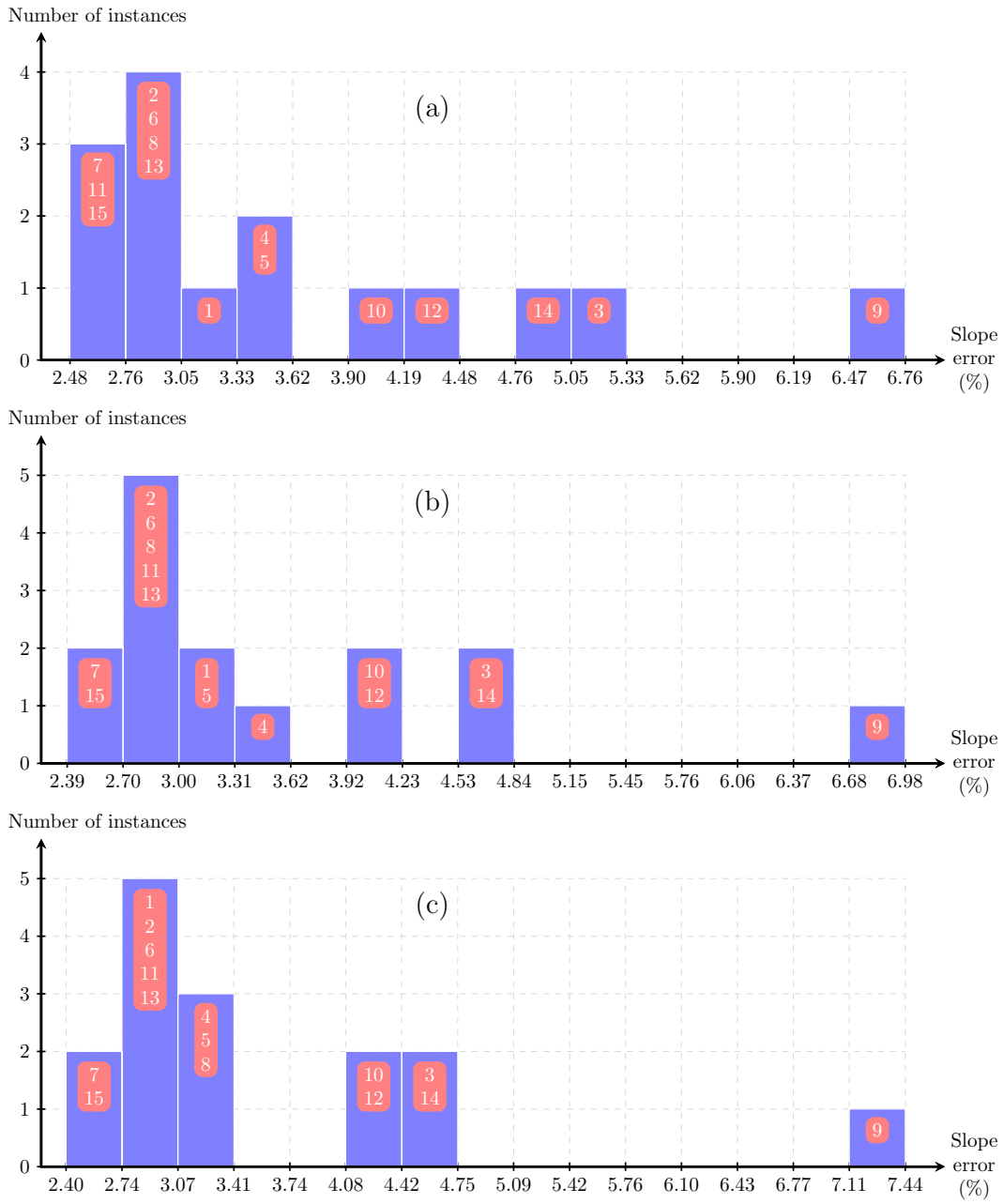


Figure 5.19: Histogram of the measured slope error for the 15 samples: (a) full output range, (b) −1.5 V/+1.5 V, (c) −1 V/+1 V

The slope error is calculated as the maximum DNL deviation of the ramp stimulus with respect to its middle point best-fit line. Consequently, in Figure 5.19, the measured slope error is found between 2.39 % and 7.44 %, but 10 over the 15 samples are around 3−4 %.

Table 5.1 lists the simulated and measured parameters of the different works on this topic along with the design proposed in this thesis. In the case of our work, the values of the linearity and the slope error are the mean values over the 15 tested ramp generator samples.

Table 5.1: Comparison of previous work on ramp generation with proposed solution

Reference	Technique	Complexity	Linearity (bits)		Slope error (%)	Output swing (V)	Frequency (kHz)
			Simulation	Measurement			
[56]	Automatic slope adjustment ramp generation	medium	N/A	8	N/A	−2/2	0.1−100
[57]	Current source-C ramp generation	low	14	N/A	N/A	0.1/0.9	1−10
	Relaxation oscillator architecture	medium	12	N/A	N/A	0.2/1.4	30−400
[59]	Adaptive triangle-wave generation	low	14	N/A	0.4	-1/1	10.1
[60]	Adaptive LMS integrators	low	N/A	N/A	0.6	N/A	N/A
		low	11	10	1.5	N/A	10
		medium	15	11	0.6	0.7/1.3	N/A
[61]	On-chip DAC and ADC testing	high	14	N/A	N/A	N/A	N/A
[62]	ΣΔ-based ramp generation	high	12	N/A	N/A	0.4/1.4	N/A
[64]	Dynamic Element Matching DAC	high	14	N/A	N/A	−1/1	1
This work	Discrete-time DC-input SC integrator	low	13.3 (1 V range)	14.97 (4 V-p-p range)	4.1	−2/2	0.2−200

5.2.3. ADC test experimental results

For the test setup, the ADC is set to a working frequency of $f_{ADC} = 10$ MHz while the ramp generator frequency is set to $f_{CLK} = 20$ kHz, meaning the ADC captures over 1000 samples per step. In this validation, and in the view of the resolution results of our generator in the 2 V FSR range, we will employ the first 11 most significant bits of the selected ADC in the test board to emulate an 11-bit ADC under test. Each ADC measurement iteration is performed according to the test setup shown in Figure 5.11. The data is processed in Matlab using the histogram test technique in order to extract the INL of the ADCUT. The actual INL is measured using the classical histogram technique. A slow 16-bit, 2 V FSR differential ramp stimulus (generated by the DAQ) is applied several times to the ADCUT input. Once the corresponding ADC output data is gathered, the INL is computed and stored as the reference for the comparison with the estimated INL. Figure 5.20 shows the histogram of the measured INL estimation error of the ADCUT.

The INL estimation error varies between 0.19 LSB and 0.61 LSB. Amongst these extremes, 12 of the 15 ADC tests have an INL estimation error below 0.5 LSB, which is very acceptable.

The instances with the highest step size values (such as 4, 5, 7, 13) are related to INL estimation errors that are not maximal, showing that step sizes higher than the specification can still give a relatively good INL estimation.

On the contrary, instance #8 is related to the worst INL estimation error despite having a good resolution, moderate step size and low slope error. This is because there is a limited number of collected samples in the experiment.

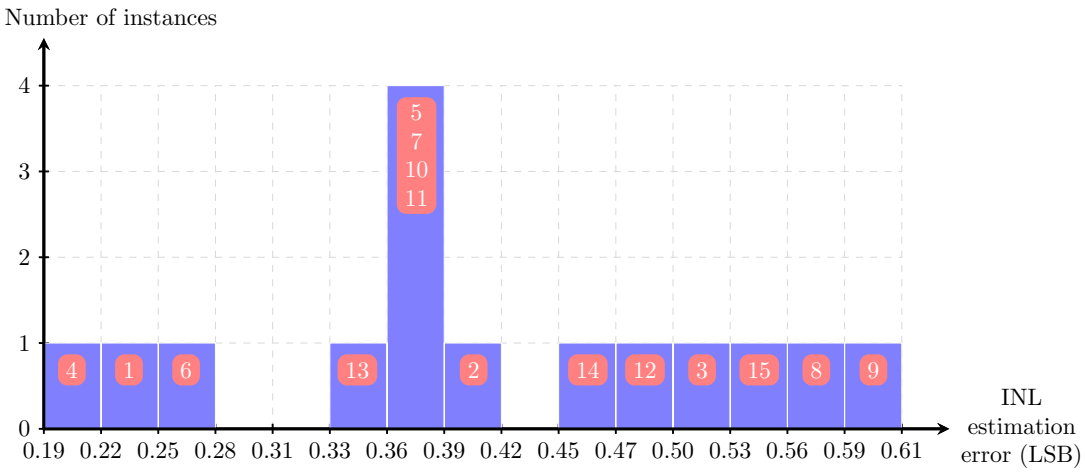


Figure 5.20: Histogram of the measured INL estimation error of the ADCUT using the 15 samples (histogram test technique)

Focusing on sample #5 that has a moderate INL estimation error, Figure 5.21 plots the actual DNL of the ADCUT, the estimated DNL, and the DNL estimation error, while Figure 5.22 shows the actual INL of the ADCUT, the estimated INL, and the INL estimation error. As we can see, there is a good agreement between the estimated and the actual value of the DNL and INL of the ADCUT.

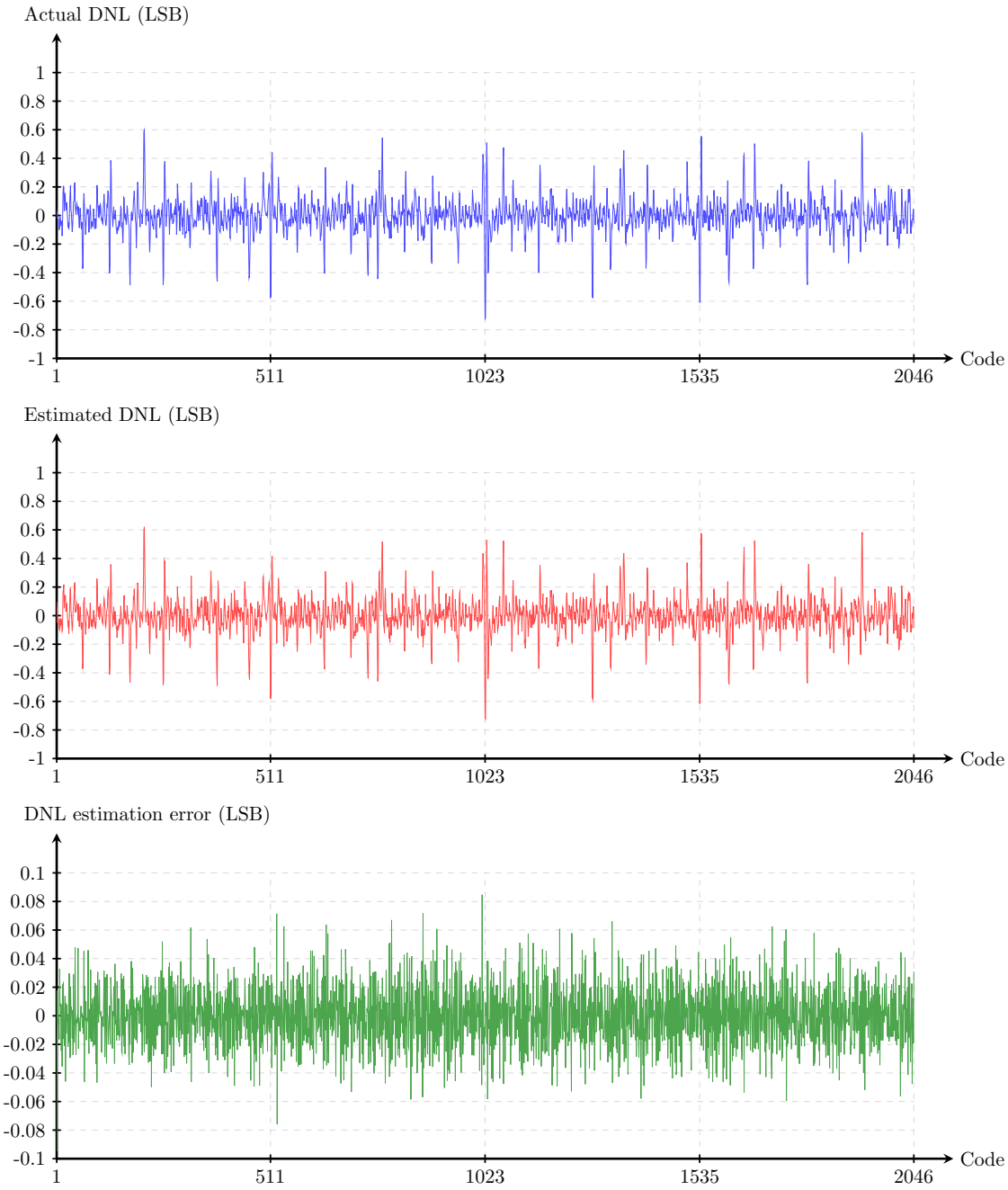


Figure 5.21: Actual DNL obtained by standard histogram test and high-linearity stimulus, estimated DNL obtained by standard histogram test with sample #5, and DNL estimation error

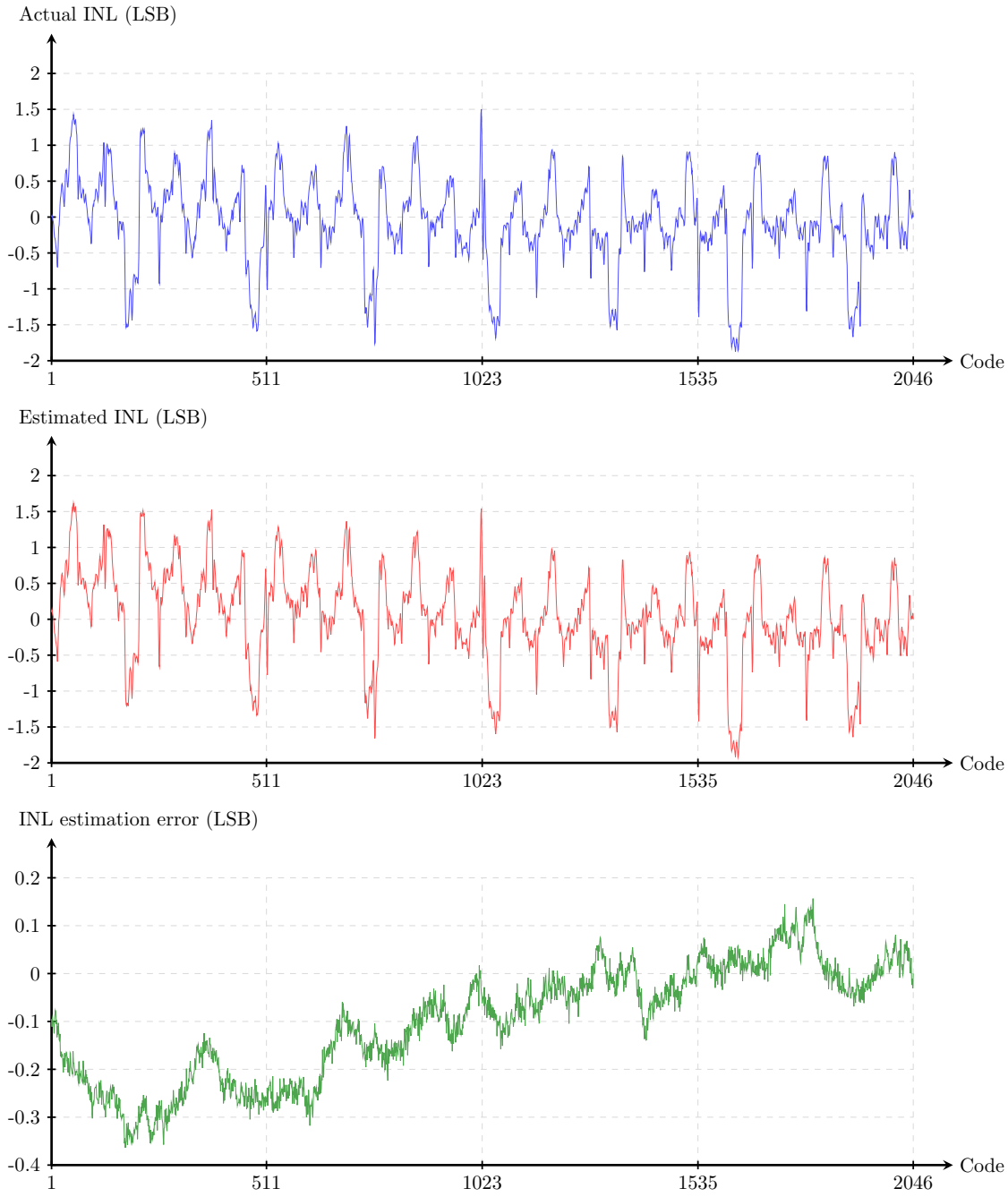


Figure 5.22: Actual INL obtained by standard histogram test and high-linearity stimulus, estimated INL obtained by standard histogram test with sample #5, and INL estimation error

Chapter 6

Conclusion

In this thesis, we presented a novel static linearity test strategy for ADCs, and particularly pipeline ADCs, using an on-chip high-linearity test stimulus generator with a modified servo-loop test technique combined with a reduced-code linearity test algorithm. The proposed technique is a low-cost approach for the BIST of pipeline ADCs, and is a trade-off between simplicity and accuracy for the evaluation of the static nonlinearities. Moreover, the technique is able to dramatically reduce the test time using the RCLT technique, where only a small fraction of the total number of codes are specifically selected and tested.

6.1. Summary of the contributions

First, we have investigated a novel way of implementing the well-known servo-loop technique on-chip with reduced analog resources. Indeed, the classical servo-loop requires a precise voltmeter in order to measure the mean value of the generator output oscillating around a target transition voltage. In this thesis, we propose a modified servo-loop implementation that provides a direct measurement of the width of a given ADC code without the need of an integrated voltmeter. The measurement strategy is based on using a controlled step-wise ramp stimulus generator for exciting the ADC under test in such a way that the measurement of a code width can be determined in the digital domain by simply counting the number of ramp steps between two consecutive ADC output transitions. Moreover, the ability of the proposed servo-loop to target a given ADC code makes it very suitable for implementing advanced reduced-code static test techniques.

Then, the proposed technique has been studied in order to compare its performance in terms of accuracy and test time with the classical servo-loop technique. The technique has also been assessed with respect to the INL estimation of generic ADCs. Next, the pipeline ADC architecture and the RCLT technique were explained. The modified servo-loop technique has then been combined with the RCLT technique in order to assess and validate the proposed BIST strategy.

One of the key contributions of this thesis is the proposal, design and silicon characterization of a novel on-chip ramp generator for ADC static BIST applications which is used for the stimulus generation in the proposed technique. The proposed ramp generator is based on a discrete-time SC integrator whose input stage has been modified to produce a very small integration gain. The main nonidealities affecting the performance of the ramp generator have been comprehensively discussed, and design trade-offs to alleviate these effects have been explored. The complete ramp generator has been designed in a 65 nm technology and its performance has been validated by transistor-level electrical simulations.

As a case study application, the proposed ramp generator has been used to enable the on-chip implementation of a reduced-code linearity test technique for pipeline ADCs. The proposed BIST strategy is based on a discrete-time integrator-based servo-loop that employs the developed ramp generator to provide the test stimulus. The proposed test technique has been validated by realistic behavioral simulations using actual data from an 11-bit 2.5 bits/stage pipeline ADC IP provided by STMicroelectronics. Obtained results show that the combination of the proposed ramp generator and the reduced-code linearity test technique yields an accurate estimation of the static linearity characteristic of the ADCUT across the complete conversion range, while only measuring the widths of 6% of the total number of codes.

Experimental results on 15 fabricated samples show a measured resolution for the generated ramp in the range of 13 to 15 bits for different voltage ranges, and a maximum output swing of 4 V-p-p. Linearity measurements show a maximum slope error well contained and centered around 3 %.

The fabricated samples were then employed for characterizing the INL and DNL of an 11-bit ADC. The maximum deviation of the estimated linearity parameters with respect to a functional characterization using high-precision external equipment is well below 0.6 LSB for all the measured samples.

Furthermore, the proposed servo-loop technique in combination with the RCLT technique effectively reduces the test time to $\sim$10 % of its initial value for the test of a 11-bit pipeline ADC, and gives an noticeable advantage to the proposed servo-loop technique with respect to the histogram technique.

6.2. Further work

Further work in this line include the integration of a complete BIST prototype including the generation of the test stimulus, analysis of the output response and external control, into a state-of-the-art ADC in collaboration with STMicroelectronics. Furthermore, we will explore the application of the proposed ramp generator and reduced-code linearity testing to other efficient static test strategies such as the double histogram technique [30], [29]. This will allow to separate the test of the ADC and the BIST circuitry, which is specially interesting for diagnosis purposes.

Bibliography

[1] M. Mahoney, *DSP-Based Testing of Analog and Mixed-Signal Circuits*. John Wiley & Sons, May 1987.

[2] "IEEE Standard for Terminology and Test Methods for Analog-To-Digital Converters," *IEEE Std 1241-2000*, 2001.

[3] P. D. Capofreddi and B. A. Wooley, "The efficiency of methods for measuring A/D converter linearity," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 3, pp. 763–769, Jun. 1999.

[4] S. Max, "Testing high speed high accuracy analog to digital converters embedded in systems on a chip," in *Test Conference, 1999. Proceedings. International*, 1999, pp. 763–771.

[5] ——, "Optimum measurement of adc code transitions using a feedback loop," in *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*, vol. 3, 1999, pp. 1415–1420 vol.3.

[6] A. Sabatini and P. Carbone, "Measurement of static ADC nonlinearities using the servoloop method," in *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference, 2004. IMTC 04*, vol. 1, May 2004, pp. 528–532.

[7] J. Doernberg, H.-S. Lee, and D. A. Hodges, "Full-speed testing of A/D converters," *Solid-State Circuits, IEEE Journal of*, vol. 19, no. 6, pp. 820–827, 1984.

[8] B. Ginetti and P. Jespers, "Reliability of code density test for high resolution ADCs," *Electronics Letters*, vol. 27, no. 24, pp. 2231–2233, Nov. 1991.

[9] R. C. Martins and A. C. Serra, "Automated ADC characterization using the histogram test stimulated by Gaussian noise," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 2, pp. 471–474, Apr. 1999.

[10] A. Di Nisio, N. Giaquinto, L. Fabbiano, G. Cavone, and M. Savino, "Improved Static Testing of A/D Converters for DC Measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 2, pp. 356–364, Feb. 2009.

[11] J. Blair, "Histogram measurement of ADC nonlinearities using sine waves," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 3, pp. 373–383, Jun. 1994.

[12] F. Adamo, F. Attivissimo, N. Giaquinto, and M. Savino, "FFT test of A/D converters to determine the integral nonlinearity," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 5, pp. 1050–1054, Oct. 2002.

[13] S. Bernard, M. Comte, F. Azaïs, Y. Bertrand, and M. Renovell, "Efficiency of Spectral-Based ADC Test Flows to Detect Static Errors," *Journal of Electronic Testing*, vol. 20, no. 3, pp. 257–267, Jun. 2004.

[14] F. Alegria, P. Arpaia, A. M. da Cruz Serra, and P. Daponte, "ADC histogram test by triangular small-waves," in *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, vol. 3, 2001, pp. 1690–1695 vol.3.

[15] F. Alegria, P. Arpaia, A. M. Serra, and P. Daponte, "Performance analysis of an ADC histogram test using small triangular waves," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 4, pp. 723–729, Aug. 2002.

[16] C. Wegener and M. P. Kennedy, "Linear model-based testing of ADC nonlinearities," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 1, pp. 213–217, 2004.

[17] A. C. Serra, M. F. Silva, P. M. Ramos, R. C. Martins, L. Michaeli, and J. Saliga, "Combined spectral and histogram analysis for fast ADC testing," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 4, pp. 1617–1623, Aug. 2005.

[18] Z. Yu, D. Chen, R. Geiger, and Y. Papantonopoulos, "Pipeline ADC linearity testing with dramatically reduced data capture time," in *2005 IEEE International Symposium on Circuits and Systems*, vol. 1, May 2005, pp. 792–795.

[19] Z. Yu and D. Chen, "Algorithm for dramatically improved efficiency in adc linearity test," in *2012 IEEE International Test Conference*, Nov 2012, pp. 1–10.

[20] K. L. Parthasarathy, L. Jin, T. Kuyel, D. Price, D. Chen, and R. Geiger, "Experimental evaluation and validation of a bist algorithm for characterization of a/d converter performance," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 5, May 2003, pp. 537–540.

[21] L. Jin, D. Chen, and R. Geiger, "A digital self-calibration algorithm for ADCs based on histogram test using low-linearity input signals," in *2005 IEEE International Symposium on Circuits and Systems*, vol. 2, May 2005, pp. 1378–1381.

[22] L. Jin, K. Parthasarathy, T. Kuyel, D. Chen, and R. L. Geiger, "Accurate testing of analog-to-digital converters using low linearity signals with stimulus error iden-

tification and removal," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1188–1199, June 2005.

[23] E. Korhonen, J. Hakkinen, and J. Kostamovaara, "A Robust Algorithm to Identify the Test Stimulus in Histogram-Based A/D Converter Testing," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2369–2374, Dec. 2007.

[24] E. Korhonen and J. Kostamovaara, "An Improved Algorithm to Identify the Test Stimulus in Histogram-Based A/D Converter Testing," in *2008 13th European Test Symposium*, May 2008, pp. 149–154.

[25] E. Korhonen, C. Wegener, and J. Kostamovaara, "Combining the Standard Histogram Method and a Stimulus Identification Algorithm for A/D Converter INL Testing With a Low-Quality Sine Wave Stimulus," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1166–1174, Jun. 2010.

[26] E. Korhonen and J. Kostamovaara, "A loopback-based INL test method for D/A and A/D converters employing a stimulus identification technique," in *Automation Test in Europe Conference Exhibition 2009 Design*, Apr. 2009, pp. 1650–1655.

[27] ——, "Memory Optimized Two-Stimuli INL Test Method for DAC-ADC Pairs," in *2011 Sixteenth IEEE European Test Symposium*, May 2011, pp. 25–32.

[28] S. C. Vora and L. Satish, "ADC Static Characterization Using Nonlinear Ramp Signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2115–2122, Aug. 2010.

[29] M. A. Jalón and E. Peralías, "ADC non-linearity low-cost test through a simplified Double-Histogram method," in *Mixed-Signals, Sensors, and Systems Test Workshop, 2009. IMS3TW '09. IEEE 15th International*, Jun. 2009, pp. 1–6.

[30] M. A. Jalón, A. Rueda, and E. Peralías, "Enhanced double-histogram test," *Electronics Letters*, vol. 45, no. 7, pp. 349–351, Mar. 2009.

[31] A. J. Ginés, E. J. Peralías, G. Leger, A. Rueda, G. Renaud, M. J. Barragán, and S. Mir, "Linearity test of high-speed high-performance ADCs using a self-testable on-chip generator," in *Test Symposium (ETS), 2016 21th IEEE European*, May 2016, pp. 1–6.

[32] A. J. Ginés, E. Peralías, G. Leger, A. Rueda, G. Renaud, M. J. Barragán, and S. Mir, "Design Trade-offs for On-chip Driving of High-speed High-performance ADCs in Static BIST Applications," in *Mixed-Signal Testing Workshop (IMSTW), 2016 21th International*, Jul. 2016, pp. 1–6.

[33] A. J. Ginés, E. J. Peralías, and A. Rueda, "Blind Adaptive Estimation of Integral Nonlinear Errors in ADCs Using Arbitrary Input Stimulus," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 452–461, Feb. 2011.

[34] S. Goyal, A. Chatterjee, M. Atia, H. Iglehart, C. Y. Chen, B. Shenouda, N. Khouzam, and H. Haggag, "Test time reduction of successive approximation register a/d converter by selective code measurement," in *IEEE International Conference on Test, 2005.*, Nov 2005, pp. 8 pp.–225.

[35] J. F. Lin, T. C. Kung, and S. J. Chang, "A Reduced Code Linearity Test Method for Pipelined A/D Converters," in *2008 17th Asian Test Symposium*, Nov. 2008, pp. 111–116.

[36] J. F. Lin, S. J. Chang, and C. H. Huang, "Design-for-Test Circuit for the Reduced Code Based Linearity Test Method in Pipelined ADCs with Digital Error Correction Technique," in *2009 Asian Test Symposium*, Nov 2009, pp. 57–62.

[37] J.-F. Lin, S.-J. Chang, T.-C. Kung, H.-W. Ting, and C.-H. Huang, "Transition-Code Based Linearity Test Method for Pipelined ADCs With Digital Error Correction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 12, pp. 2158–2169, Dec. 2011.

[38] A. Laraba, H.-G. Stratigopoulos, S. Mir, H. Naudet, and C. Forel, "Enhanced reduced code linearity test technique for multi-bit/stage pipeline ADCs," in *Test Symposium (ETS), 2012 17th IEEE European*, May 2012, pp. 1–6.

[39] A. Laraba, H.-G. Stratigopoulos, S. Mir, H. Naudet, and G. Bret, "Reduced code linearity testing of pipeline ADCs in the presence of noise," in *VLSI Test Symposium (VTS), 2013 IEEE 31st*, Apr. 2013, pp. 1–6.

[40] ——, "Reduced-Code Linearity Testing of Pipeline ADCs," *IEEE Design Test*, vol. 30, no. 6, pp. 80–88, Dec. 2013.

[41] A. Laraba, H.-G. Stratigopoulos, S. Mir, and H. Naudet, "Exploiting Pipeline ADC Properties for a Reduced-Code Linearity Test Technique," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2391–2400, Oct. 2015.

[42] E. Peralías, A. Ginés, and A. Rueda, "INL systematic reduced-test technique for Pipeline ADCs," in *2014 19th IEEE European Test Symposium (ETS)*, May 2014, pp. 1–6.

[43] M. Ehsanian, B. Kaminska, and K. Arabi, "A new digital test approach for analog-to-digital converter testing," in *VLSI Test Symposium, 1996., Proceedings of 14th*, Apr. 1996, pp. 60–65.

[44] K. Arabi and B. Kaminska, "Efficient and accurate testing of analog-to-digital converters using oscillation-test method," in *European Design and Test Conference, 1997. ED TC 97. Proceedings*, Mar 1997, pp. 348–352.

[45] Z. Zhao and A. IvaNov, "Embedded servo loop for ADC linearity testing," *Microelectronics Journal*, vol. 33, no. 10, pp. 773–780, Oct. 2002.

[46] R. de Vries, T. Zwemstra, E. M. J. G. Bruls, and P. P. L. Regtien, "Built-in self-test methodology for A/D converters," in *Proceedings of the 1997 European conference on Design and Test*, Mar 1997, pp. 353–358.

[47] Y.-C. Wen and K.-J. Lee, "An on-chip ADC test structure," in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, 2000, pp. 221–225.

[48] Y.-C. Wen, "A bist scheme for testing analog-to-digital converters with digital response analyses," in *23rd IEEE VLSI Test Symposium (VTS'05)*, May 2005, pp. 383–388.

[49] M. Renovell, F. Azaïs, S. Bernard, and Y. Bertrand, "Hardware resource minimization for histogram-based ADC BIST," in *18th IEEE VLSI Test Symposium, 2000. Proceedings*, 2000, pp. 247–252.

[50] F. Azaïs, S. Bernard, Y. Bertrand, and M. Renovell, "Implementation of a linear histogram BIST for ADCs," in *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings*, 2001, pp. 590–595.

[51] F. Azaïs, S. Bernard, Y. Betrand, and M. Renovell, "Towards an ADC BIST scheme using the histogram test technique," in *Test Workshop, 2000. Proceedings. IEEE European*, 2000, pp. 53–58.

[52] H. W. Ting, B. D. Liu, and S. J. Chang, "A Histogram-Based Testing Method for Estimating A/D Converter Performance," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 2, pp. 420–427, Feb. 2008.

[53] J. Ren, J. Feng, and H. Ye, "A novel linear histogram BIST for ADC," in *9th International Conference on Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008*, Oct. 2008, pp. 2099–2102.

[54] J. Duan, D. Chen, and R. Geiger, "Phase control of triangular stimulus generator for ADC BIST," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1935–1938.

[55] W. Jiang and V. D. Agrawal, "A DSP-based ramp test for on-chip high-resolution ADC," in *2011 IEEE 43rd Southeastern Symposium on System Theory*, Mar. 2011, pp. 203–207.

[56] C. Jansson, K. Chen, and C. Svensson, "Linear, polynomial and exponential ramp generators with automatic slope adjustment," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 2, pp. 181–185, Feb. 1994.

[57] L.-C. Wang, E. Sanchez-Sinencio, and F. Maloberti, "Very linear ramp-generators for high resolution ADC BIST and calibration," in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000*, vol. 2, 2000, pp. 908–911.

[58] F. Azaïs, S. Bernard, Y. Bertrand, X. Michel, and M. Renovell, "A low-cost adaptive ramp generator for analog BIST applications," in *VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001*, 2001, pp. 266–271.

[59] S. Bernard, F. Azaïs, Y. Bertrand, and M. Renovell, "A high accuracy triangle-wave signal generator for on-chip ADC testing," in *Test Workshop, 2002. Proceedings. The Seventh IEEE European*, 2002, pp. 89–94.

[60] B. Provost and E. Sanchez-Sinencio, "On-chip ramp generators for mixed-signal BIST and ADC self-test," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 263–273, Feb. 2003.

[61] J.-L. Huang, C.-K. Ong, and K.-T. Cheng, "A BIST scheme for on-chip ADC and DAC testing," in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, 2000, pp. 216–220.

[62] Y.-S. Wang, J.-S. Wang, F.-S. Lai, and Y.-S. Ye, "A low-cost BIST scheme for ADC testing," in *2005 6th International Conference on ASIC*, vol. 2, Oct. 2005, pp. 694–698.

[63] B. Olleta, H. Jiang, D. Chen, and R. L. Geiger, "A deterministic dynamic element matching approach for testing high-resolution ADCs with low-accuracy excitations," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 3, pp. 902–915, Jun. 2006.

[64] H. Jiang, B. Olleta, D. Chen, and R. L. Geiger, "Testing High-Resolution ADCs With Low-Resolution/Accuracy Deterministic Dynamic Element Matched DACs," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1753–1762, Oct. 2007.

[65] H. Xing, H. Jiang, D. Chen, and R. L. Geiger, "High-Resolution ADC Linearity Testing Using a Fully Digital-Compatible BIST Strategy," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2697–2705, Aug. 2009.

Publications

[1] G. Renaud, M. J. Barragán, S. Mir, and M. Sabut, "On-Chip Implementation of an Integrator-Based Servo-Loop for ADC Static Linearity Test," in *Test Symposium (ATS), 2014 IEEE 23rd Asian*, Nov. 2014, pp. 212−217.

[2] G. Renaud, M. J. Barragán, and S. Mir, "Design of an on-chip stepwise ramp generator for ADC static BIST applications, " in *Mixed-Signal Testing Workshop (IMSTW), 2015 20th International*, Jun. 2015, pp. 1−6.

[3] G. Renaud, M. J. Barragán, A. Laraba, H.-G. Stratigopoulos, S. Mir, H. Le-Gall, and H. Naudet, "A 65nm CMOS Ramp Generator Design and its Application Towards a BIST Implementation of the Reduced-Code Static Linearity Test Technique for Pipeline ADCs, " *Journal of Electronic Testing*, pp. 1−15, 2016.

[4] A. J. Ginés, E. J. Peralías, G. Leger, A. Rueda, G. Renaud, M. J. Barragán, and S. Mir, "Linearity test of high-speed high-performance ADCs using a self-testable on-chip generator," in *Test Symposium (ETS), 2016 21th IEEE European*, May 2016, pp. 1−6.

[5] A. J. Ginés, E. J. Peralías, G. Leger, A. Rueda, G. Renaud, M. J. Barragán, and S. Mir, "Design Trade-offs for On-chip Driving of High-speed High-performance ADCs in Static BIST Applications," in *Mixed-Signal Testing Workshop (IMSTW), 2016 21th International*, Jul. 2016, pp. 1−6.

Appendix

A

Résumé en français

A.1. Introduction

A.1.1. Contexte

Aujourd'hui, les circuits intégrés sont utilisés largement utilisés dans un large domaine d'activités : électronique grand public, domaine automobile, télécommunications, etc. Cependant, bien que le coût de fabrication s'est réduit de manière constante au cours des 50 dernières années, le coût du test est resté constant. En effet, la complexité croissante des circuits intégrés les rendent de plus en plus difficiles à tester, en particulier pour les circuits intégrés analogiques et mixtes. Le rôle principal du test de production est de s'assurer qu'aucun circuit défectueux ne soit délivré au client. En conséquence, plus de 50 % du coût total de production peut être alloué au test de circuits intégrés durant cette phase cruciale [**?**].

Prenons par exemple le cas des systèmes sur puce, qui contiennent des circuits analogiques, mixtes, numériques et radiofréquences. Si on regarde le temps de test par type de circuits pour un système sur puce de téléphone mobile affiché en Figure A.1, on peut voir que plus d'un tiers du temps de test est alloué au test des circuits mixtes.

Le test de circuits analogiques, mixtes, ou radiofréquences repose encore sur la technique de caractérisation fonctionnelle, qui n'est pas standardisée (malgré certaines avancées intéressantes dans le domaine [**?**],[**?**],[**?**]), avec peu de techniques de test intégrées, et donc très coûteuse.

La technique de test intégré, qui consiste à intégrer quelques fonctionnalités des équipements automatiques de test au sein même des composants (génération de stimuli de test, évaluation de la réponse, et contrôle du protocole de test), est une solution prometteuse afin de réduire le coût du test de ces derniers. En effet, cette technique permet une meilleure testabilité de la plaquette au système, une bonne qualité de test, un test rapide, moins de dépendance vis-à-vis des équipements de test, une réduction du temps de dé-

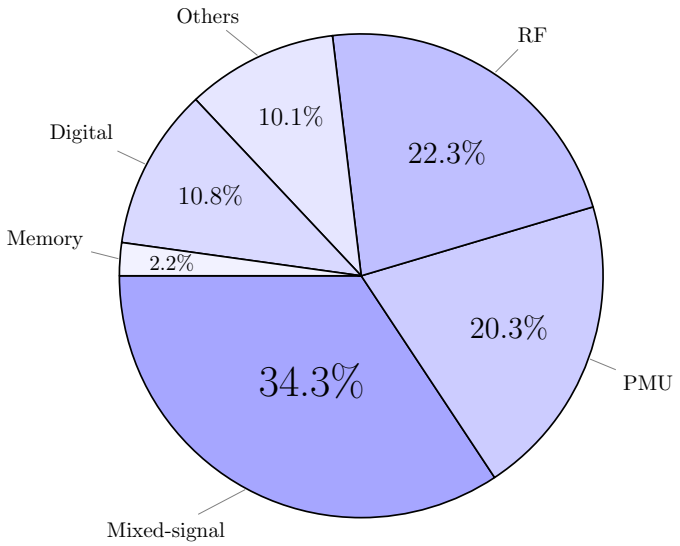


Figure A.1: Diagramme circulaire du temps de test respectif pour différents types de circuits d'un système sur puce de téléphone mobile [**?**]

veloppement, un test moins cher et moins long, et un délai de commercialisation moins long. Le test intégré permet aussi de vérifier l'état du composant durant toute sa durée de fonctionnement, ce qui peut être utile dans des cas où le composant est inaccessible, ou dans des environnements hostiles pour l'homme. Cependant, le test intégré implique une augmentation de la surface de silicium nécessaire au composant, une dégradation des performances, des exigences sur la qualité du test pour des applications visant l'état de l'art, et l'obligation de réduire le temps du test, ce qui pose beaucoup de défis à relever.

A.1.2. Motivation et description du travail de thèse

Dans ce travail de thèse, nous avons décidé de nous concentrer sur le test des convertisseurs analogique-numérique (CAN), et particulièrement sur le test statique des convertisseurs analogique-numérique de type pipeline. Les CANs sont des composants indispensables pour tout système électronique à signaux mixtes. Ils sont la liaison entre le monde analogique et numérique. Deux types de test sont généralement effectués : le test statique, qui consiste à mesurer les spécifications statiques telles que la Non-Linéarité Différentielle (NLD) et la Non-Linéarité Intégrale (NLI), l'erreur de gain et l'offset, alors que le test dynamique consiste à mesurer les spécifications dynamiques telles que le nombre effectif de bits, le rapport signal à bruit, le rapport signal à bruit avec distorsion, le taux de distorsion harmonique, etc. Pendant le test de production des CANs, les deux métriques principalement mesurées sont la NLD et la NLI.

Afin d'évaluer ces paramètres, les techniques de test standard utilisent deux méthodes

statistiques, qui sont problématiques pour des CANs de haute résolution. En effet, le signal de test doit être le plus linéaire possible, ce qui augmente considérablement le coût de l'équipement de test qui le génère. De plus, le pas de quantification des CANs devenant de plus en plus petit, les mesures sont de plus en plus sensibles au bruit. Afin d'obtenir une bonne NLI, un grand nombre d'échantillons doit être collecté et analysé, ce qui augmente considérablement le temps de test et peut être prohibitif.

Une solution est d'intégrer le générateur de signaux analogiques de test dans le CAN, ce qui a pour effet de réduire de manière significative le bruit lors des mesures. Le développement d'une méthode de test statique intégré est une solution prometteuse afin de réduire le coût du test statique. Le problème de l'accès au circuit intégré est éliminé, ce qui réduit encore plus le coût du test.

Ce travail de thèse vise à chercher et développer une nouvelle technique de test intégré pour le test des CANs. Le but consiste à développer un générateur de stimuli analogiques à bas coût en combinaison avec des techniques de mesure efficaces pour l'évaluation des performances statiques des CANs à haute résolution, et en particulier les CANs pipeline. Une solution employant une boucle de retour utilisant une technique adaptée de servo-loop est proposée et est appliquée en combinaison avec une technique de réduction de code.

A.2. Etat de l'art du test statique des CANs

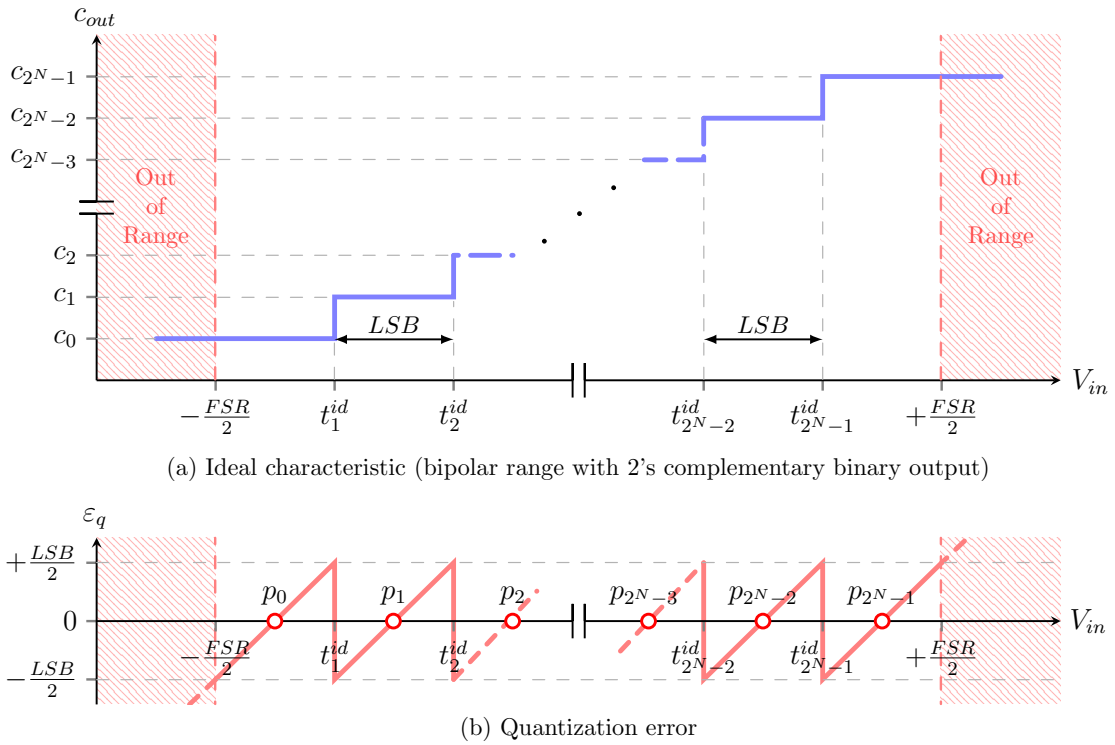
A.2.1. Définitions pour le test statique des CANs

La fonction de transfert idéale d'un CAN de N bits est montrée sur la Figure A.2. Lorsque le CAN est idéal tous les codes ont la même largeur, appelé LSB, qui est défini comme suit :

$$LSB = t_{k+1}^{id} - t_k^{id} = \frac{V_{ref}^{+} - V_{ref}^{-}}{2^N} \tag{A.1}$$

où t_k^{id} et t_{k+1}^{id} sont respectivement les transitions idéales des codes c_k et c_{k+1}, N est le nombre de bits du CAN, et V_{ref}^{+} et V_{ref}^{-} sont respectivement les tensions d'entrée maximum et minimum définissant la pleine échelle du CAN.

Les erreurs statiques les plus importantes sont la Non-Linéarité Différentielle (NLD) et la Non-Linéarité Intégrale (NLI), illustrées respectivement en Figure A.3 et Figure A.4.



(a) Ideal characteristic (bipolar range with 2's complementary binary output)

(b) Quantization error

Figure A.2: Fonction de transfert idéale d'un CAN de N bits

La NLD est définie telle que

$$DNL_k = \frac{W_k - LSB}{LSB}, \quad k = 1,2,...2^N - 2 \tag{A.2}$$

où $W_k = t_{k+1} - t_k$ est la largeur du code c_k. La NLI est définie telle que

$$INL_k = \frac{t_k - t_k^{id}}{LSB}, \quad k = 1,2,...2^N - 2. \tag{A.3}$$

A.2.2. Techniques standard de test statique de CAN

Deux techniques standard de test sont très usitées en industrie : la technique d'histogramme et la technique de servo-loop.

La technique d'histogramme consiste à appliquer un signal analogique ayant une distribution connue à l'entrée du CAN, et à récupérer le nombre d'occurrences de chacun des codes exercés. À partir de cet histogramme, les performances statiques du CAN peuvent être calculées en considérant l?histogramme de référence du signal utilisé en entrée.

Pour un stimulus de rampe, si H_k^r est le nombre d'occurrences par code pour le code c_k,

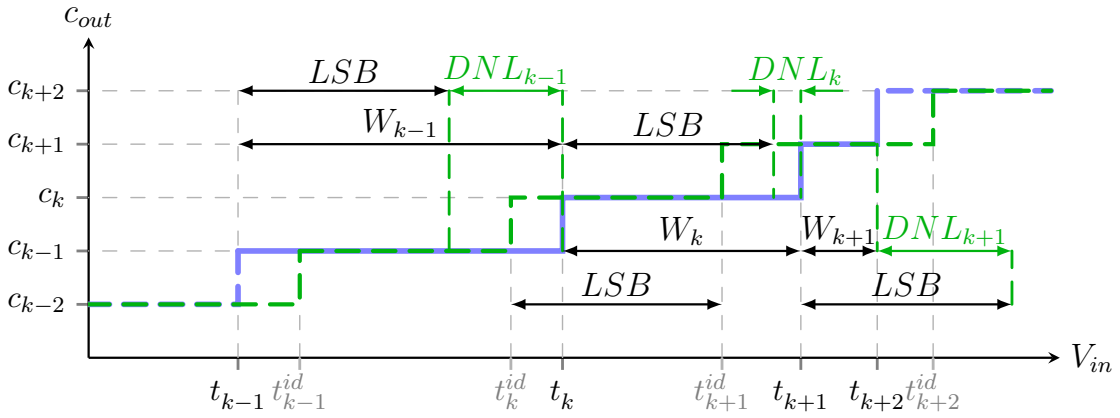


Figure A.3: Représentation de la NLD d'un CAN de N bits

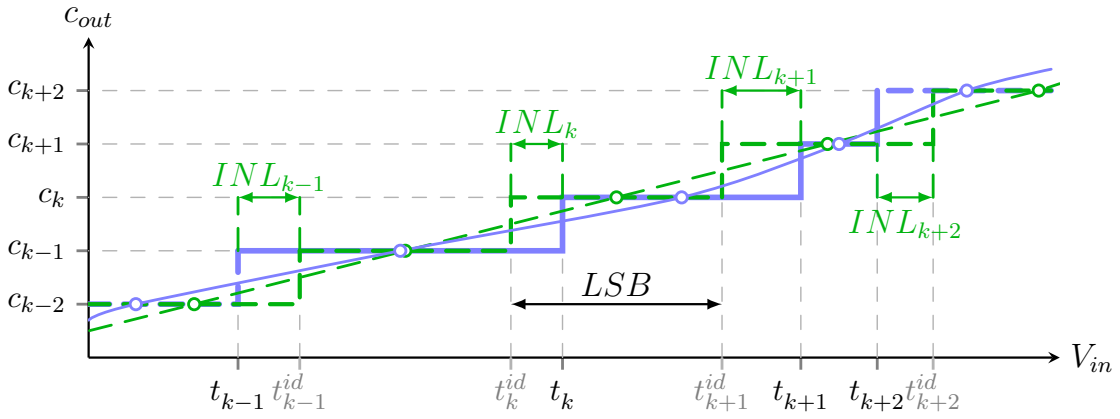


Figure A.4: Représentation de la NLI d'un CAN de N bits

alors le nombre idéal de'occurrences par code H^r_{ref} est exprimé tel que

$$H^r_{ref}(k) = n_T \cdot \frac{FSR}{2^N \cdot A_r}, \quad k = 1,2,...2^N - 2 \tag{A.4}$$

où n_T est le nombre total d'échantillons collectés sur la dynamique d'entrée du CAN et A_r est l'amplitude de la rampe idéale, dépassant les limites de la pleine échelle du CAN. À partir de l'équation (A.4), la largeur du code c_k est calculée telle que

$$W_k = \frac{H^r_k}{H^r_{ref}}, \quad k = 1,2,...2^N - 2. \tag{A.5}$$

La NLD et la NLI du code c_k sont alors calculées telles que

$$DNL_k = W_k - 1\ LSB, \quad k = 1,2,...2^N - 2 \tag{A.6}$$

et

$$INL_k = \sum_{k=1}^{k} DNL_k, \quad k = 1,2,...2^N - 2. \tag{A.7}$$

Pour un signal sinusoïdal en entrée, tous les codes n'ont pas le même histogramme de référence. Dans le cas où le milieu du signal sinusoïdal correspond au milieu de la dynamique d'entrée du CAN, les histogrammes de référence sont donnés par

$$\begin{aligned} H^s_{ref}(k) = &\frac{n_T}{\pi} \cdot \left(\arcsin\left[\frac{k-(2^{N-1}-1)-V^s_{os}}{A_s}\right] \right. \\ &\left. - \arcsin\left[\frac{k-2^{N-1}-V^s_{os}}{A_s}\right]\right), \quad k = 1,2,...2^N-2 \end{aligned} \tag{A.8}$$

où V^s_{os} et A_s sont respectivement l'offset et le gain du stimulus sinusoïdal. En considérant W_l et W_u les occurrences respectives du premier code et du dernier code du CAN qui sont reliés au valeurs la plus basse et la plus haute du stimulus sinusoïdal, il est possible d'en déduire l'offset V^s_{os} et l'amplitude A_s :

$$V^s_{os} = (2^{N-1}-1) \cdot \left(\frac{\cos(\pi \cdot W_l/n_T) - \cos(\pi \cdot W_u/n_T)}{\cos(\pi \cdot W_l/n_T) + \cos(\pi \cdot W_u/n_T)} \right) \tag{A.9}$$

et

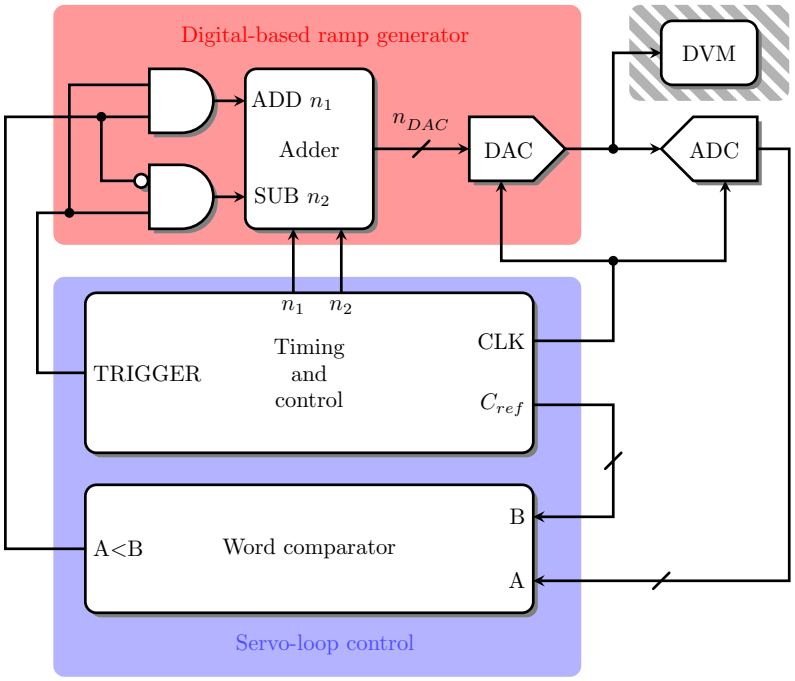
$$A_s = \frac{(2^{N-1}-1) - V^s_{os}}{\cos(\pi \cdot W_u/n_T)}. \tag{A.10}$$

Le même raisonnement est appliqué comme dans le cas d'un stimulus de rampe afin de calculer la NLD et la NLI en utilisant les équations (A.5), (A.6) et (A.7).

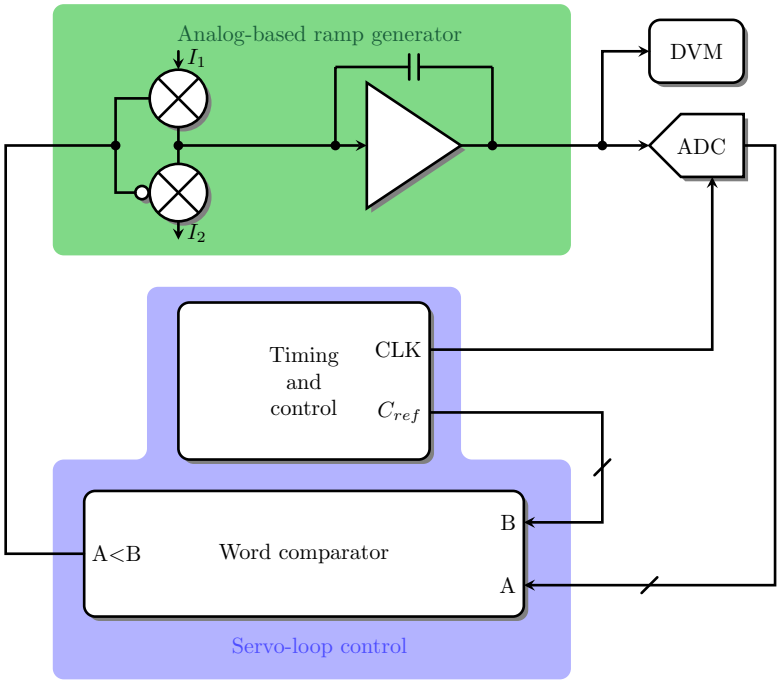
La technique de servo-loop est une approche en boucle fermée. Le principe est illustré en Figure A.5. Un signal est appliqué au CAN sous test et sa sortie est comparée à un code de référence $C_{ref}(k)$, qui spécifie la transition t_k qui doit être déterminée par la mesure.

En définissant une transition cible t_k, si la sortie du CAN est en dessous du code cible de référence $C_{ref}(k)$, alors l'entrée du CAN augmente jusqu'à ce que la transition t_k soit dépassée. Si la sortie du CAN est supérieure ou égale au code de référence, l'entrée du CAN diminue jusqu'à ce que la transition cible soit croisée. Une fois que la transition t_k est atteinte, l'entrée du CAN oscille autour de cette transition. La mesure de la tension de cette transition consiste à filter le signal triangulaire dû à l'oscillation du stimulus à l'entrée du CAN afin d'obtenir une valeur fixe. Cette valeur estimée est ensuite comparée à la valeur idéale de la transition cible afin d'estimer la NLI du code cible $C(k)$.

Deux types de servo-loop peuvent être employées pour le test statique d'un CAN. La



(a) Version numérique (voltmètre optionnel)



(b) Version analogique (voltmètre nécessaire)

Figure A.5: Deux implémentations du test de servo-loop [2]

Figure A.5 (a) montre la version numérique de la servo-loop, alors que la Figure A.5 (b) montre la version analogique de la servo-loop.

La complexité et la résolution des CANs augmente, ce qui implique la génération d'un

stimulus de test de grande qualité. Les techniques de test actuelles peuvent couvrir ces besoins, mais nécessitent un équipement de test onéreux et un temps de test toujours plus long. Afin de s'affranchir de ces problèmes, différents travaux visant la réduction du temps et/ou coût du test ont été entrepris. On peut les classifier en deux groupes : techniques de conception en vue du test (DfT) et techniques de test intégré (BIST).

A.2.3. Techniques de conception en vue du test

Une technique de conception en vue du test est définie comme toute technique au niveau de la conception permettant de faciliter le test d'un circuit.

Dans [7], l'auteur propose une technique de caractérisation de CAN en boucle ouverte par ordinateur reposant sur la technique de test de densité de codes et une analyse spectrale utilisant la transformée de Fourier rapide (FFT).

Dans [8], [9], et [10], les auteurs proposent d'employer un signal de bruit en tant que stimulus pour le test statique de linéarité des CANs.

Les techniques proposées dans [11], [12] et [13] utilisent un stimulus sinusoïdal pour le test statique de linéarité des CANs.

Les travaux de [14] et [15] présentent une technique d'histogrammes se reposant sur une série de stimuli triangulaires à faible amplitude.

Dans [16], [17], [18], et [19], les auteurs dévoilent une technique de test statique de CANs se reposant sur une approche haut niveau, à l'aide d'un modèle précis du CAN sous test.

Les auteurs dans [20], [21], [22], [23], [24], [26], [25], [27], [28], [29], [30], et [31] proposent des techniques consistant à relâcher les contraintes de linéarité du stimulus pour le test statique de CANs.

Les travaux dans [29], [30], [31], et [32] proposent une technique de test de double histogramme test technique visant aussi à relâcher les contraintes de linéarité du stimulus pour le test statique de CANs.

Dans [34], [35], [36], [37], [38], [39], [40], [41], et [42], les auteurs proposent une technique de test utilisant un algorithme de réduction de codes pour le test de CANs ayant une architecture répétitive comme les CANs pipeline ou à approximation successive (SAR).

A.2.4. Techniques de test intégré

Afin de réduire davantage la complexité et le coût du test statique de CAN, le développement d'une technique de test intégré peut être envisagée. Dans ce cas, la génération du stimulus de test et l'analyse de la réponse à ce stimulus sont intégrés dans le circuit à tester.

Dans [43] et [37], les auteurs proposent une approche de test intégré totalement numérique pour le test statique de CAN.

Les travaux dans [44] et [45] proposent sur une technique de test intégré reposant sur un stimulus oscillant et la technique de servo-loop pour le test statique de CAN.

Dans [46], [47], et [48] proposent une technique de test intégré utilisant un compteur numérique pour le test statique de CAN.

Les travaux dans [49], [50], [51], [52], et [53] proposent une technique de test intégré utilisant la technique d'histogramme avec une réduction des contraintes matérielles imposées sur le circuit intégré de test pour le test statique de CAN.

Les auteurs dans [54] proposent une technique de test intégré qui consiste à retarder la phase du stimulus de test sur chaque période afin d'échantillonner de manière cohérente.

Le travail présenté dans [55] repose sur l'utilisation d'une unité de calcul numérique (DSP) intégrée afin de réduire le temps de test.

A.2.5. Génération d'un stimulus de test

L'intégration d'un générateur de stimulus de test approprié au CAN à tester dont on veut connaître la linéarité est l'un des plus grands défis à relever. Dans l'industrie, le stimulus de test doit être 2 à 3 bits plus linéaire que le CAN à tester. Cependant, la résolution des CANs ne cessant d'augmenter, cette contrainte devient de plus en plus difficile à honorer.

La génération de stimulus à haute linéarité pour le test statique de CANs reste une question ouverte à la recherche, bien que plusieurs travaux aient été présentés depuis un grand nombre d'années. De manière générale, deux types de génération de stimulus pour le test statique de CAN peuvent être trouvés dans la littérature : la génération de stimulus par intégration de signal et la génération de stimulus avec un convertisseur numérique-analogique (CNA). Le Tableau A.1 recense les différents travaux effectués dans le domaine au cours des dernières années.

Table A.1: Comparaison des techniques de génération de stimulus de rampe

Référence	Technique	Complexité	Linéarité (bits)		Erreur de pente (%)
			Simulation	Mesure	
[56]	Génération de rampe avec ajustement automatique de pente	moyenne	N/A	8	N/A
[57]	Génération de rampe avec source de courant chargeant une capacité	basse	14	N/A	N/A
	Architecture d'oscillateur à relaxation	moyenne	12	N/A	N/A
[59]	Génération adaptative d'un signal triangulaire	basse	14	N/A	0.4
[60]	Intégrateurs adaptatifs par la méthode des moindres carrés	basse	N/A	N/A	0.6
		basse	11	10	1.5
		moyenne	15	11	0.6
[61]	Test intégré et simultané de CAN et CNA	haute	14	N/A	N/A
[62]	Génération de rampe à l'aide de CAN ΣΔ	haute	12	N/A	N/A
[64]	CNA avec technique d'appariement dynamique d'éléments	haute	14	N/A	N/A

A.3. Technique de servo-loop proposée

La technique de servo-loop classique, définie dans [2], repose sur l'utilisation d'un voltmètre de haute résolution pour la mesure précise de la tension de chaque transition visée. Dans une implémentation de test intégré, il est quasiment impossible de concevoir un circuit de test compact et assez précis pour obtenir une précision de test suffisante. Nous proposons donc dans cette section une nouvelle technique de test intégré s'inspirant de la technique de servo-loop où la mesure de la largeur de chaque code est effectuée sans voltmètre, à l'aide de blocs totalement numériques. La technique est ensuite combinée avec une technique de réduction de codes qui permet un gain non négligeable sur le temps de test.

A.3.1. Description

La technique de servo-loop proposée est une version modifiée de la technique de servo-loop classique [2], [44]. Comme illustré en Figure A.6, le générateur de rampe à temps continu est remplacé par un générateur de rampe à temps discret proposé dans ce travail de thèse. Le fonctionnement de la technique proposée est très similaire à la technique de servo-loop classique, mais au lieu de mesurer la tension des transitions du CAN sous test, on mesure directement la largeur de chaque code correspondant. La largeur W_k d'un code c_k ($k = 1, 2, ..., 2^N - 1$) est mesurée en comptant, à l'aide d'un compteur numérique, le nombre de pas d'intégration du générateur de rampe qu'il faut pour traverser la différence de tension entre les transitions t_k et t_{k+1} qui correspondent au code visé. En partant de la transition t_k à gauche du code à mesurer c_k, si le code de sortie du CAN est inférieure ou

égale au code visé, alors l'entrée du CAN augmente pas à pas, et le compteur commence à incrémenter jusqu'à ce que la transition t_{k+1}, à droite du code visé, est rencontrée, ce qui veut dire que dans ce cas la sortie du CAN est égale au code supérieur, c_{k+1}. Ensuite, l'entrée du CAN descend pas à pas, et le compteur continue de s'incrémenter jusqu'à ce que la transition t_k soit croisée, ce qui veut dire dans ce cas que la sortie du CAN est égale au code inférieur, c_{k-1}. Le processus est répété r fois en faisant osciller l'entrée du CAN autour de la dynamique de tension qui correspond à la largeur du code c_k pour moyenner les bruits combinés de la sortie du générateur et de l'entrée du CAN. Le nombre de pas par code pour le code c_k s'exprime de la manière suivante :

$$\widehat{m_k} = \frac{\sum_{i=1}^{r} n_i}{r}, \quad k = 1,2,...,2^N - 1 \tag{A.11}$$

où n_i est le nombre de pas comptés pendant l'excursion i. Les comptes des excursions vers le haut et vers le bas sont sommées, et l'expression est divisée par le nombre d'excursions r, pour donner une estimation moyenne du nombre de pas par code pour la largeur W_k du code c_k. Le nombre de pas par code $\widehat{m_k}$ est ensuite multiplié par la hauteur estimée du pas d'intégration $\widehat{s}$ afin d'évaluer la largeur W_k. L'opération est répétée pour chaque code de la caractéristique à mesurer. Finalement, la NLD et la NLI peuvent être facilement déduites de la caractéristique du CAN en utilisant leurs définitions standard. Leurs expressions sont :

$$\left\{ \begin{array}{ll} \widehat{DNL_k} & = \widehat{m_k} \cdot \widehat{s} - 1 \; LSB \\ \widehat{INL_k} & = \sum_{i=1}^{k} \widehat{DNL_k} \end{array} \right\}, \quad k = 1,2,...,2^N - 1 \tag{A.12}$$

où

$$\widehat{s} = \frac{\sum_{k=1}^{2^N-1} \widehat{m_k}}{2^N - 1}. \tag{A.13}$$

L'approximation sur la mesure et le pas d'intégration du générateur implique une erreur statique sur l'estimation de la NLD de chaque code mesuré. Cette erreur s'accumule dans l'estimation de la NLI, mais peut être éliminée en utilisant un algorithme de meilleur ajustement [1] qui peut facilement être adapté à notre approche de test intégré.

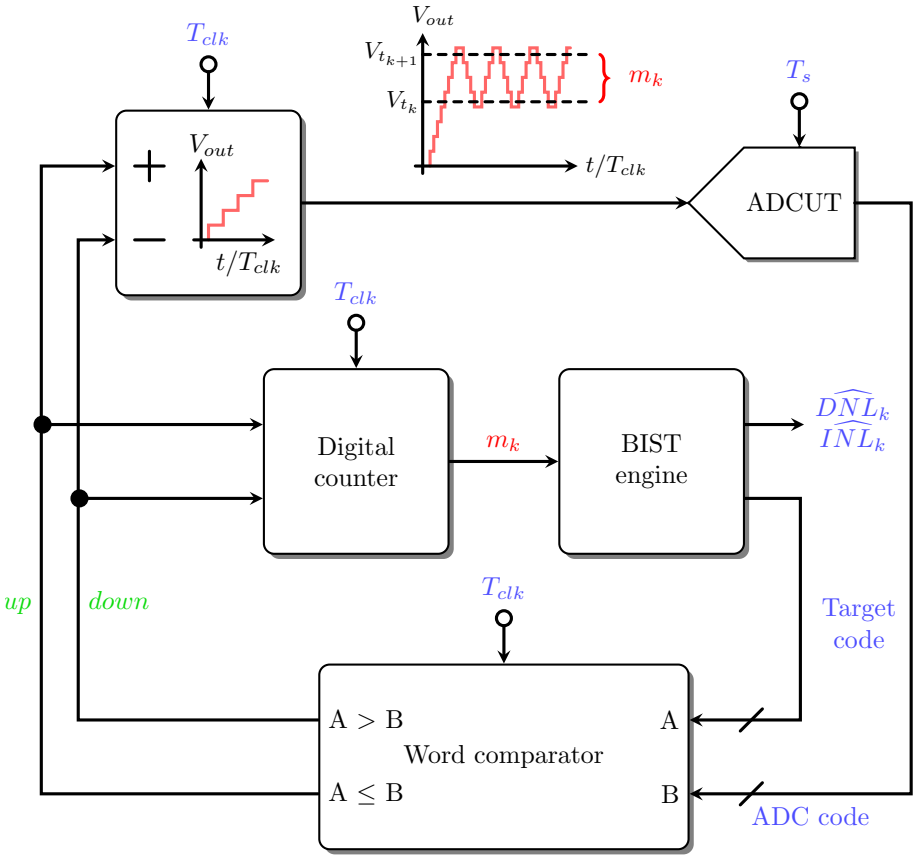


Figure A.6: Schéma du circuit de servo-loop proposé

A.3.2. Combinaison de la technique de servo-loop proposée avec la technique de réduction de codes

En reprenant le travail décrit dans [41] et expliqué dans la sous-section 3.3.2, il est possible de mesurer la caractéristique complète d'un CAN de type pipeline en ne mesurant qu'un nombre réduit de codes judicieusement sélectionnés. La Figure A.7 illustre la technique expliquée. Par exemple, pour un CAN pipeline de 11 bits à 5 étages de 2.5 bits/étage, il n'est nécessaire de mesurer que les largeurs des 8 codes autour des 6 transitions pour le premier étage, les largeurs des 6 codes autour des 6 transitions pour le second étage, les largeurs des 4 codes autour des 6 transitions pour le troisième étage, et les largeurs des 2 codes autour des 6 transitions pour le quatrième et le cinquième étage. La technique est expliquée plus en détail dans [41].

Pour résumer, les étapes de la technique sont comme suit :

• Trouver la correspondance entre les transitions de sortie du CAN et les comparateurs qui sont exercés.

• Sélectionner un ensemble représentatif de transitions de sortie du CAN.

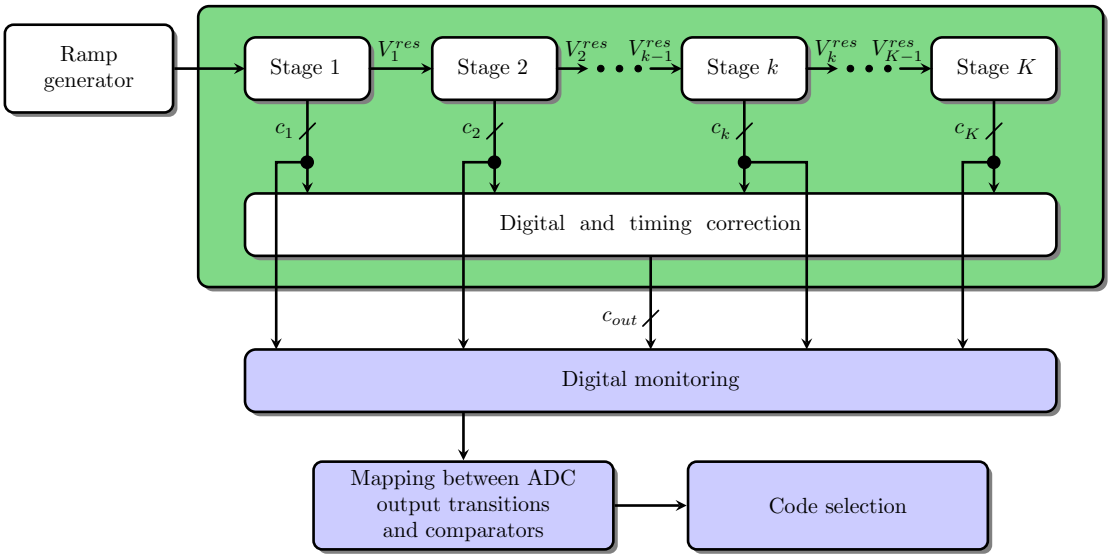


Figure A.7: Suivi numérique des sorties numériques des étages du CAN de type pipeline

• Mesurer les largeurs des codes autour des transitions de sortie du CAN qui étaient sélectionnées en utilisant la technique de servo-loop proposée.
• Déduire les largeurs des codes autour des transitions de sortie du CAN qui n'ont pas été sélectionnées dans l'étape 2.
• Calculer la NLD et la NLI à partir des largeurs de code obtenues.

La technique de servo-loop proposée, combinée avec la technique de réduction de codes, est simulée afin de montrer ses performances vis-à-vis de la même technique, mais sans la technique de réduction de codes. Pour cette simulation, 1000 CANs pipeline de 11 bits avec 5 étages de 2.5 bits/étage et une pleine échelle de 2 V sont générés aléatoirement en utilisant un modèle de CAN de type pipeline réaliste et paramétrable. L'écart type du bruit d'entrée du CAN est fixé à $\sigma_{ADC} = 0.15$ LSB. La variation maximum du bruit de sortie du générateur est de ± 3 fois son pas d'intégration, donc pour un pas de $s = 0.1$ LSB, l'écart type du bruit de sortie du générateur est de $\sigma_{ramp} = 0.1$ LSB. Pour chacune des 1000 simulations lancées, 1024 échantillons sont collectés pour la mesure de chaque code sélectionné. Les CANs générés ont une NLI variant entre 0.2 LSB et 4.3 LSB. La variation de la NLI est large afin de montrer l'efficacité de la technique proposée avec la technique de réduction de codes à estimer la NLI de CANs avec de bonnes et mauvaises performances.

La Figure A.8 montre trois graphiques. Le graphique du haut représente la NLI estimée des CANs testés contre leur NLI réelle. La NLI estimée avec la technique de réduction de codes est illustrée par les cercles bleus, alors que la NLI estimée sans la technique de réduction de codes est illustrée par les losanges rouges. La ligne verte représente la

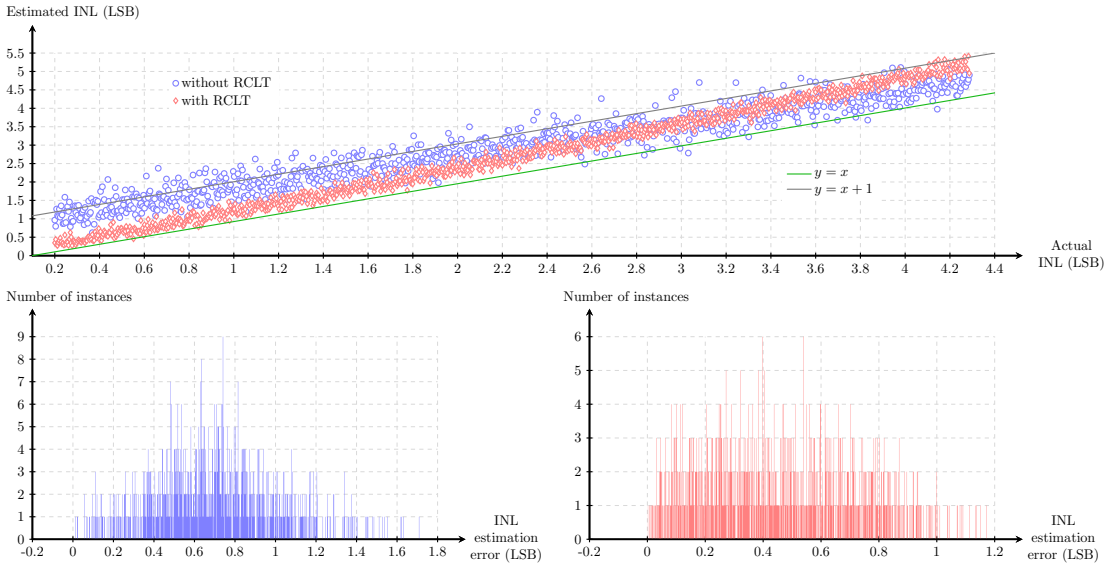


Figure A.8: NLI estimée contre la NLI réelle et histogrammes respectifs de l'erreur d'estimation de la NLI pour la simulation de 1000 couples CAN pipeline/générateur de rampe en utilisant la technique de test proposée tout d'abord sans, puis avec la technique de réduction de codes

ligne idéale $y = x$, et la ligne grise représente la ligne d'erreur d'estimation de 1 LSB. L'histogramme en bleu définit la dispersion de la différence absolue maximale entre la NLI estimée et la NLI réelle sans l'utilisation de la technique de réduction de codes, alors que l'histogramme en rouge montre la dispersion de la différence absolue maximale entre la NLI estimée et la NLI réelle en utilisant cette fois-ci la technique de réduction de codes.

Les résultats montrent que dans le cas sans la technique de réduction de codes, tandis que les valeurs de la NLI varient entre 0.2 LSB et 4.3 LSB, les valeurs des estimations de la NLI varient entre 0.79 LSB et 5.07 LSB, et l'erreur d'estimation de la NLI varie entre 0 LSB et 1.7 LSB. La valeur moyenne de l'erreur d'estimation de la NLI est de 0.69 LSB. 151 des 1000 couples CAN pipeline/générateur de rampe ont été mesurés avec une erreur d'estimation de la NLI supérieure à 1 LSB, alors que 10 autres ont été mesurés avec une erreur en dessous de 0 LSB. Dans le cas avec la technique de réduction de codes, tandis que les valeurs de la NLI varient entre 0.2 LSB et 4.3 LSB, les valeurs des estimations de la NLI varient entre 0.34 LSB et 5.4 LSB, et l'erreur d'estimation de la NLI varie entre 0 LSB et 1.17 LSB. La valeur moyenne de l'erreur d'estimation de la NLI est de 0.45 LSB. 15 des 1000 couples CAN pipeline/générateur de rampe ont été mesurés avec une erreur d'estimation de la NLI supérieure à 1 LSB, alors que 9 autres ont été mesurés avec une erreur en dessous de 0 LSB. De plus, on peut observer que la technique de réduction de codes réduit la variabilité de la technique de test intégré proposée.

A.4. Conception d'un générateur de rampe pour le test de CANs

La cible de ce générateur est un CAN de type pipeline de 11 bits à 5 étages et échantillonnant à 80 Msps, conçu dans une technologie CMOS 65 nm de STMicroelectronics. En suivant les lignes directrices données dans la section 4.2, nous pouvons établir les spécifications que le générateur doit respecter tels que le gain ou la marge de phase de l'amplificateur opérationnel, le produit gain-bande, la vitesse de balayage, les plages d'entrée et de sortie, etc. La liste des spécifications définies sont listées dans le Tableau A.2.

Un schéma du générateur complet est montré en Figure A.9, ainsi que le timing associé en Figure A.10. C'est un intégrateur à capacités commutées totalement différentiel conçu en technologie CMOS 65 nm avec 7 niveaux de métaux de STMicroelectronics. Des transistors à V_t standard ont été utilisés dans la conception de ce circuit, avec une alimentation unique de $V_{dd} = 2.5$ V.

La sortie de l'intégrateur, en conditions idéales, est donnée par :

$$\begin{aligned} V_{out}(n) &= V_{out}(n-1) + \frac{C_{i_2} - C_{i_1}}{C_F} \cdot V_{ref} \\ &= V_{out}(n-1) + \frac{\delta C}{C_F} \cdot V_{ref}. \end{aligned} \tag{A.14}$$

au temps $t = n \cdot T_{clk}$, où T_{clk} est la période d'horloge de l'intégrateur, V_{ref} est une tension d'entrée constante de référence, et C_F est la capacité de la boucle de retour de l'intégrateur. Par ailleurs, comme observé dans l'équation (A.14), la sortie de l'intégrateur est une rampe à pas discrets, et l'amplitude de ses pas dépend de la différence de capacités δC. Cette valeur est implémentée au niveau de la disposition physique du circuit en créant délibérément une petite différence de capacités entre les capacités d'entrée de l'intégrateur.

Dans la Figure A.9, plusieurs éléments sont à noter. Les contrôles numériques ϕ_{up} et ϕ_{down} permettent de piloter la pente de la rampe à chaque pas d'intégration. Le signal ϕ_{reset} effectue une remise à zéro de la capacité d'intégration, et le signal ϕ_0 est ajoutée afin de pouvoir charger la sortie de l'intégrateur à une valeur initiale de référence V_a à des fins de calibration pour les mesures. Pour plus d'informations, la conception du circuit ainsi que la procédure de calibration sont expliquées en détail dans [**?**].

Pour le test statique de linéarité du CAN sous test, la taille du pas du générateur de rampe est fixé à 1/5 du LSB du CAN. C'est une bonne valeur qui permet d'avoir un bon

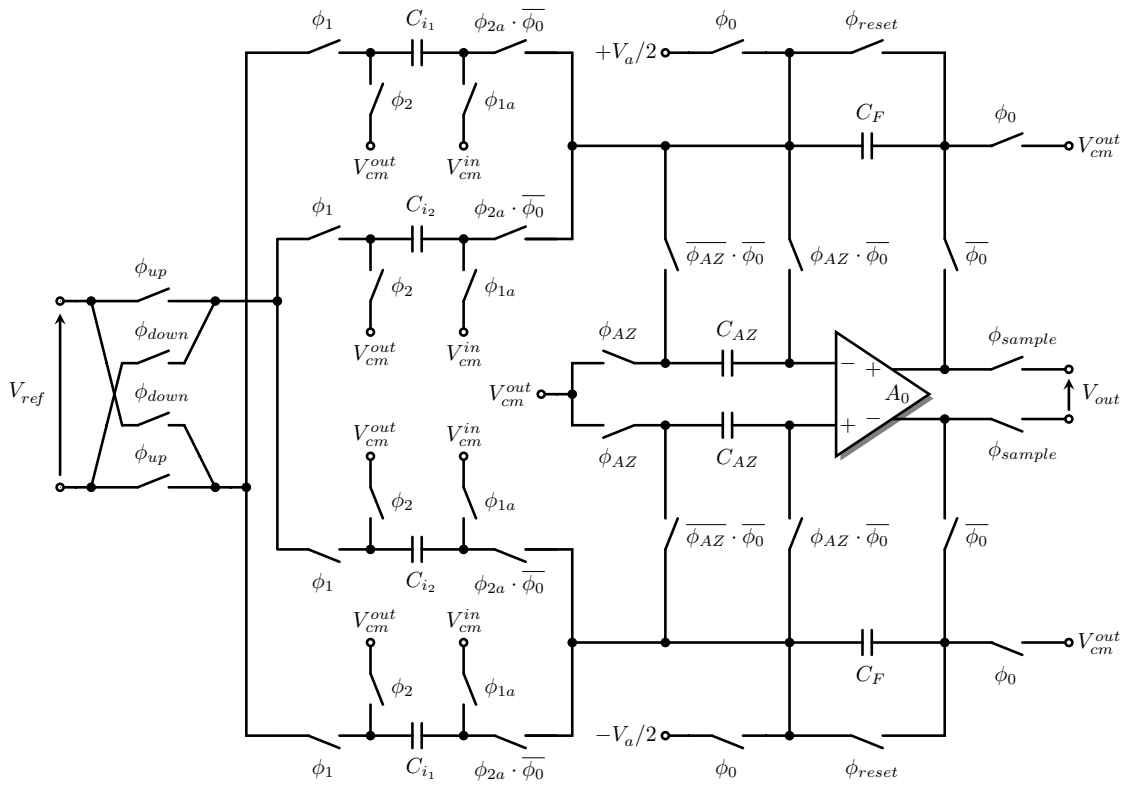


Figure A.9: Proposition de générateur de rampe à capacités commutées totalement différentiel

compromis entre la résolution de la rampe et le temps de test. De plus, avec une pleine échelle du CAN de 2 V, le pas est fixée approximativement à 97.6 μV, arrondi à 100 μV pour faciliter la conception sans dégrader les performances du générateur.

L'intégrateur est piloté par une fréquence d'opération maximale de $T_{clk} = 200$ kHz. Comme décrit dans la Figure A.10, la sortie de l'intégrateur est mise à jour à une fréquence deux fois plus faible.

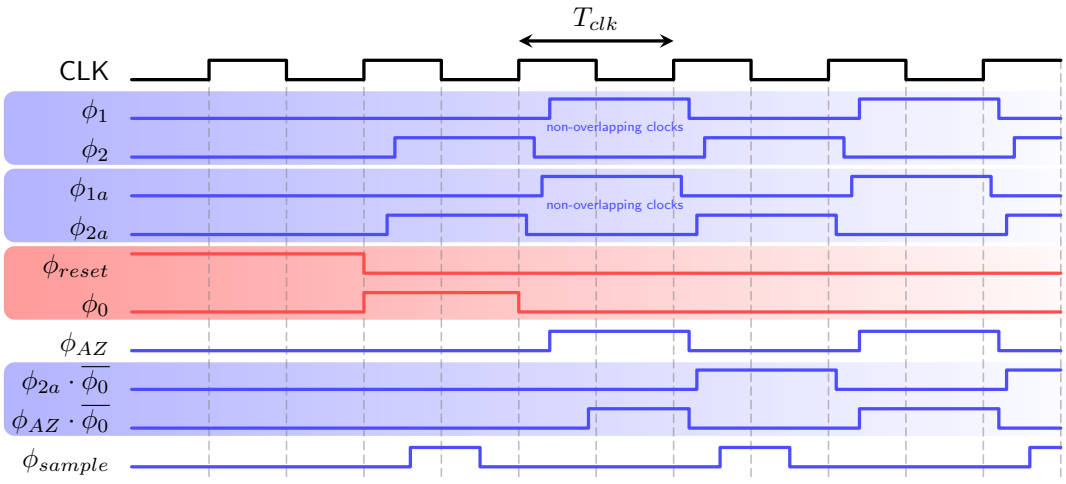


Figure A.10: Timing de l'intégrateur à capacités commutées proposé

Table A.2: Spécifications requises et paramètres du générateur de rampe proposé

Paramètre	Unité	Objectif	Simulation électrique	Simulation post-layout
A_0	dB	> 80	92.5	90.1
PM	°	> 60	75	74
GBW	MHz	> 90	95	93
Plage d'entrée de mode commun	V	> 2	$+2.39/-2.38$	$+2.27/-2.26$
plage de sortie de mode commun	V	> 2	$+2.28/-2.27$	$+2.19/-2.17$
SR	V/μs	> 100	156	110
$t_{settling}$	ns	< 100	60	77
Fréquence d'horloge nominale	kHz	200	–	–
Fréquence d'horloge minimale	kHz	1	–	–
I_{bias}	μA	la plus basse	286	–
Consommation	mW	la plus basse	23	–
C_{i_1}	pF	0.999	1.0246	–
C_{i_2}	pF	1	1.0256	–
$C_{i_2} - C_{i_1}$	fF	1	1	–
C_F	pF	10	0.966 $\times$10 en //	–
C_{AZ}	pF	1	1.0246	–

La disposition physique du circuit proposé est montré en Figure A.11, et une photomicrographie du circuit fabriqué est montré en Figure A.12. Une attention toute particulière doit être prise vis-à-vis des éléments parasites qui peuvent dégrader notablement les performances du générateur. C'est pourquoi nous avons suivi des règles de conception standard pour la disposition physique du circuit. Le circuit a été conçu de manière la plus symétrique possible par rapport à l'axe vertical. De plus, la partie analogique et la partie numérique ont été séparées afin de limiter les couplages de bruit entre les deux domaines. Des plots d'alimentation ont été dédiés pour chaque domaine également. La surface finale du circuit est de 370 μm $\times$ 380 μm, ce qui peut représenter environ 10 % de l'aire centrale d'un CAN commercial. L'aire allouée au circuit de test intégré est donc jugé acceptable d'un point de vue industriel.

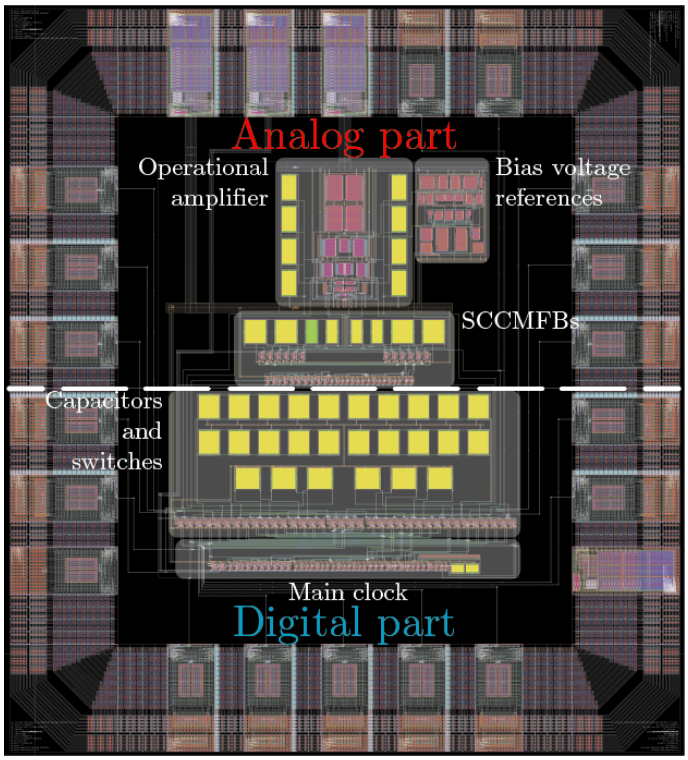


Figure A.11: Layout of the proposed generator

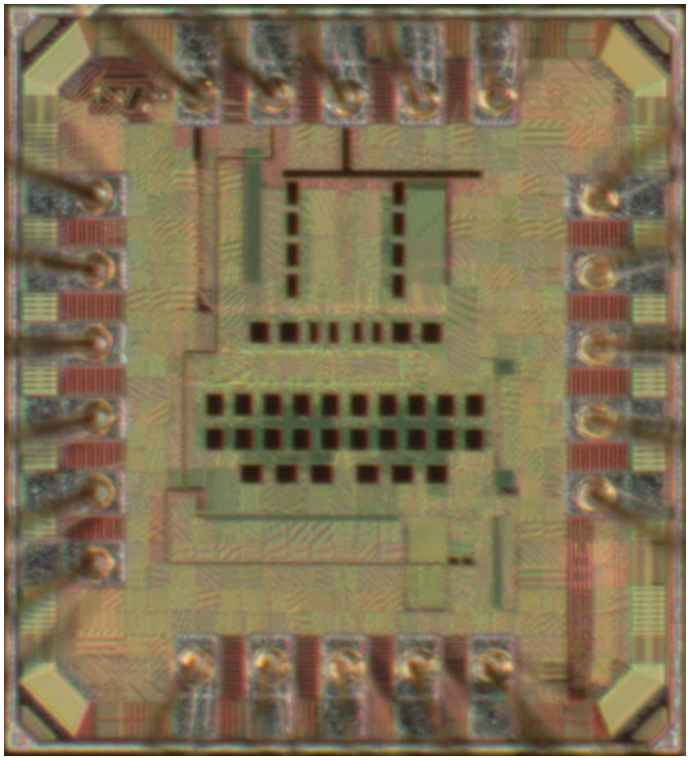
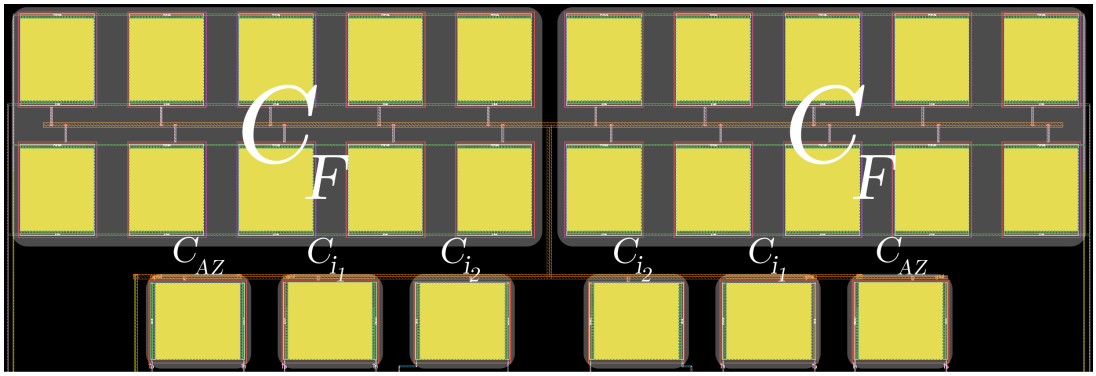


Figure A.12: Die photomicrograph of the proposed generator

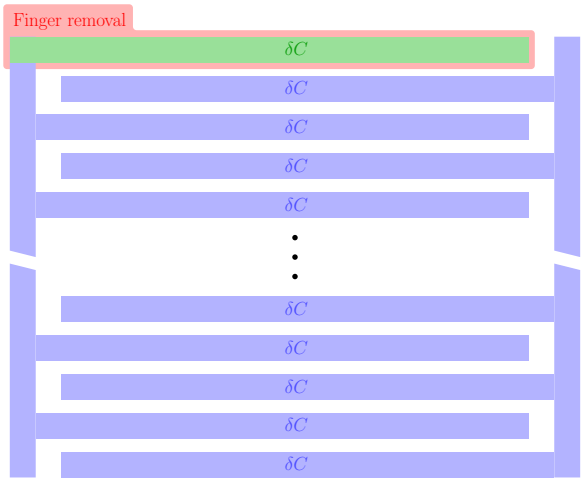
Pour l'implémentation des capacités du circuit, nous avons décidé d'utiliser des capa-

cités métal sur métal radiofréquence (RF MOM), car la linéarité et l'appariement de ces capacités sont très bons. Les performances du générateur ne sont donc pas mises en cause par les non idéalités des capacités MOM. Il est très simple de concevoir ce type de capacités, puisqu'il faut seulement choisir le nombre de doigts nécessaires pour une valeur de capacité donnée, ce qui est d'autant plus utile et pratique pour la disposition physique de ces dernières. Par ailleurs, nous avons conçu ces capacités MOM telles que la différence de capacités requise, $C_{i_2} - C_{i_1} = \delta C$, puisse correspondre exactement à la capacité d'un doigt.

La Figure A.13 (a) montre la disposition physique des capacités C_{i_1}, C_{i_2}, C_{AZ}, et C_F, alors que la Figure A.13 (b) montre la suppression du doigt sur C_{i_1} afin de créer la valeur recherchée de capacité virtuelle.



(a) Disposition physique des capacités C_{i_1}, C_{i_2}, C_{AZ}, et C_F



(b) Suppression d'un doigt sur C_{i_1} pour la génération d'une petite différence de capacités δC

Figure A.13: Vue de dessus de la disposition physique des capacités

A.5. Résultats de simulation et résultats expérimentaux

A.5.1. Résultats de simulation

Comme présenté dans la section précédente, le générateur de rampe proposé a été conçu en technologie CMOS 65 nm de STMicroelectronics. Le CAN sous test est un modèle de CAN de type pipeline de 11 bits avec 5 étages de 2.5 bits/étage et une pleine échelle de 1 V. Avec cette cible, le générateur a été conçu afin d'avoir un pas d'intégration de 100 μV, soit environ 0.2 LSB compte tenu de la pleine échelle du CAN sous test, dans la plage de 0 à 1 V.

Le générateur de rampe a été simulé sous des conditions de procédé de fabrication standard, avec une fréquence d'opération de 200 kHz et une tension d'entrée différentielle et continue de 2 V comme tension de référence. La rampe pas à pas ainsi générée a une résolution nominale de 13.3 bits, avec un pas moyen de 96.7 μV. Afin de mesurer la linéarité de la rampe, la Figure A.14 montre la différence entre chaque pas d'intégration de la rampe générée et le pas moyen. Comme observé, la déviation de la taille du pas est bien contenue dans la plage $[-3, 3]$ μV sur la plage de tension considérée. L'erreur de pente par pas étant définie comme la différence entre la pente de la rampe connectant les points milieux de deux pas consécutifs et la pente de la droite de meilleur ajustement qui passe par les points milieux des pas. L'erreur moyenne de pente, qui est en réalité la moyenne des erreurs moyennes des pentes sur tous les pas, est calculée avec une valeur de -0.040 %. On peut de plus calculer la NLI de la rampe générée à partir des valeurs de la Figure A.14, et qui se définit comme la différence entre la position réelle des points milieux de chaque pas et leur position idéale. La caractéristique de NLI complète est montrée en Figure A.15. La NLI maximum de la rampe générée est d'environ 580 μV, ce qui correspond à 6 pas moyens d'intégration. La Figure A.14 montre clairement une décroissance constante du pas d'intégration sur tout le long de la pleine échelle, ce qui est dû aux fuites de courant de l'intégrateur. En conséquence, la courbe de NLI en Figure A.15, obtenue par intégration de la courbe de NLD de la rampe, a une forme de parabole très caractéristique. Ces résultats montrent que la principale source de non linéarité statique, dans le générateur proposé, vient tout d'abord des fuites de courant de l'intégrateur, et dans une moindre mesure, vient du résidu de tension restant après l'auto-zéro de l'offset de l'amplificateur opérationnel du générateur. Notons de plus que la stratégie de mesure proposée tire parti de la forme lisse de la courbe de NLI et l'évolution bien maîtrisée de la taille du pas d'intégration. Puisque la technique de réduction de codes ne nécessite de mesurer qu'une petite partie des largeurs des codes du CAN sous test, le stimulus de

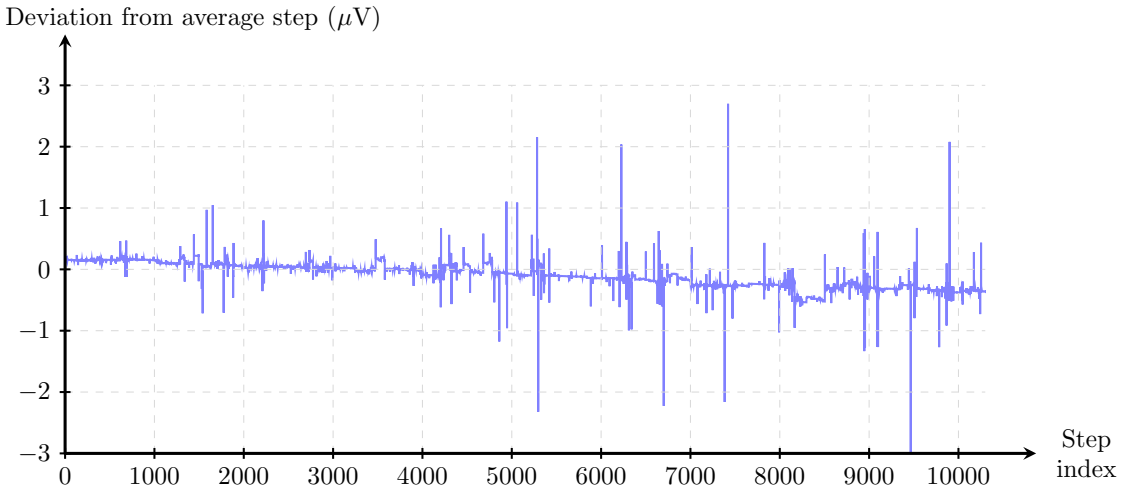


Figure A.14: Déviation de la taille du pas d'intégration par rapport au pas moyen

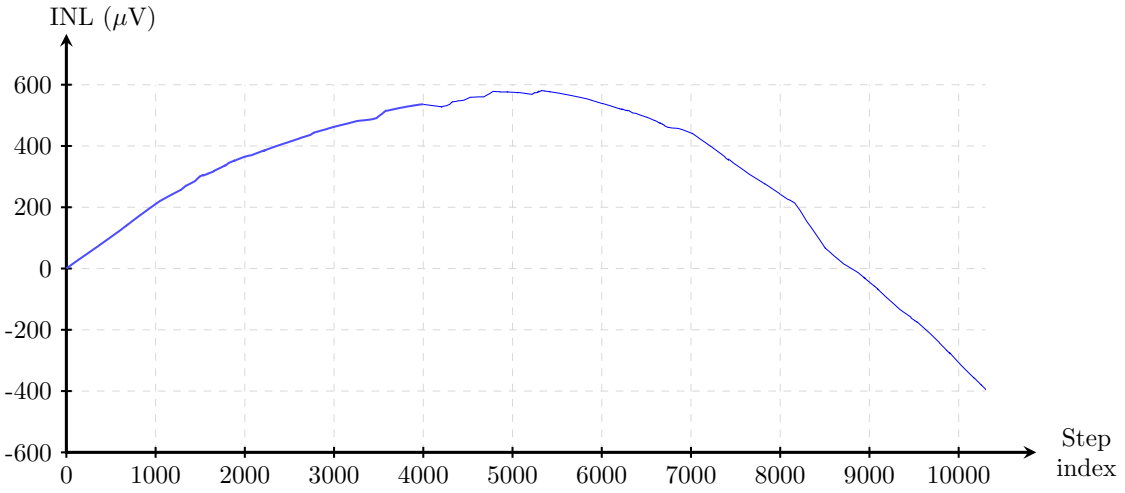


Figure A.15: NLI de la rampe générée

rampe se doit d'être linéaire localement aux alentours des codes ciblés, tant que le pas de la rampe générée est relativement constant et reste une fraction du LSB du CAN sous test.

Afin d'aller plus loin dans la validation des performances du générateur proposé sous différentes conditions d'utilisation, la Figure A.16 montre la déviation du pas d'intégration sur toute la plage de tension de la rampe quand la température varie, soit pour T = 0 °C, T = 75 °C et pour la température nominale de T = 27 °C. On peut observer que l'impact des variations de température sur la linéarité de la rampe est bien maîtrisé. Les résultats montrent respectivement un pas moyen de 96.9 μV et 97.1 μV pour T = 0 °C et T = 75 °C. Comparés aux 96.7 μV du pas moyen à T = 27 °C, la variation du pas moyen dû au stress thermique est évalué à moins de 0.5 %.

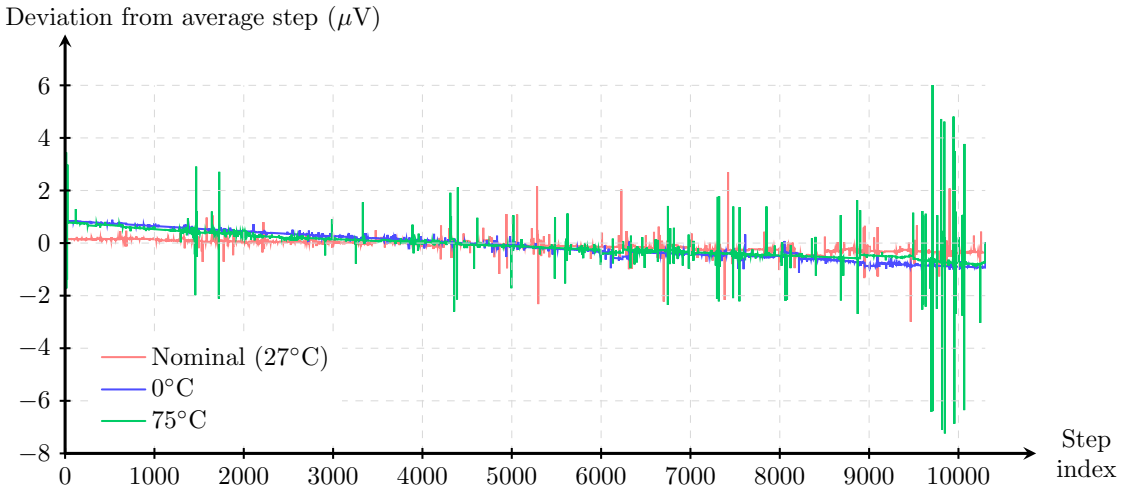


Figure A.16: Déviation de la taille du pas de la rampe vis-à-vis des différentes températures considérées

Afin de montrer l'impact des variations de procédés de fabrication et d'appariement sur les performances du générateur proposé, une campagne de simulations de type Monte Carlo a été entreprise au niveau transistor sur le système complet en utilisant le kit de conception du procédé de la technologie considérée. Afin d'obtenir rapidement des résultats, les itérations des simulations Monte Carlo ont été stoppées et supprimées si le pas d'intégration était trop petit et bien au delà des spécifications requises. De cette manière, nous avons pu nous focaliser sur les cas limites et les pires cas tout en nécessitant moins de ressources de calcul.

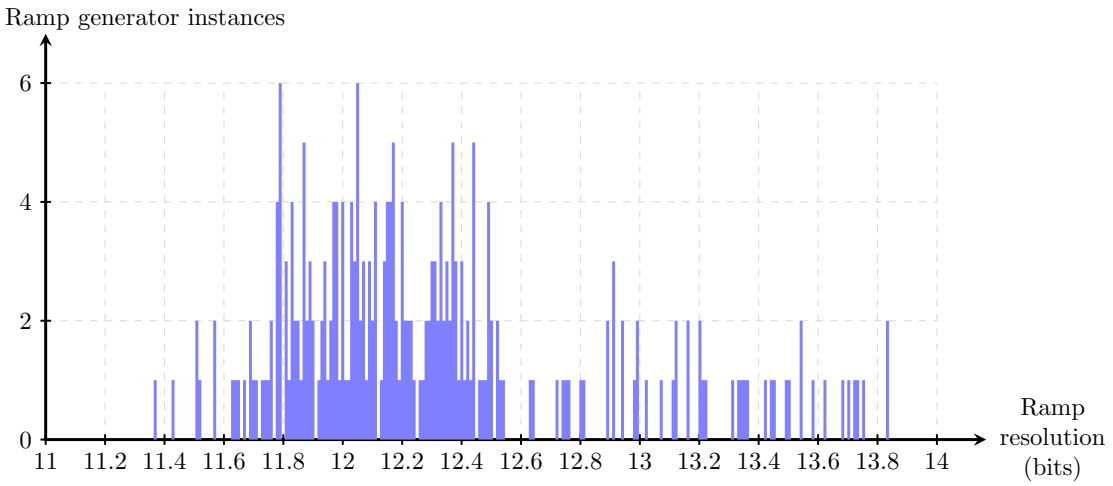


Figure A.17: Histogramme de la résolution de la rampe générée obtenue par simulations Monte Carlo faisant varier le procédé de fabrication et l'appariement

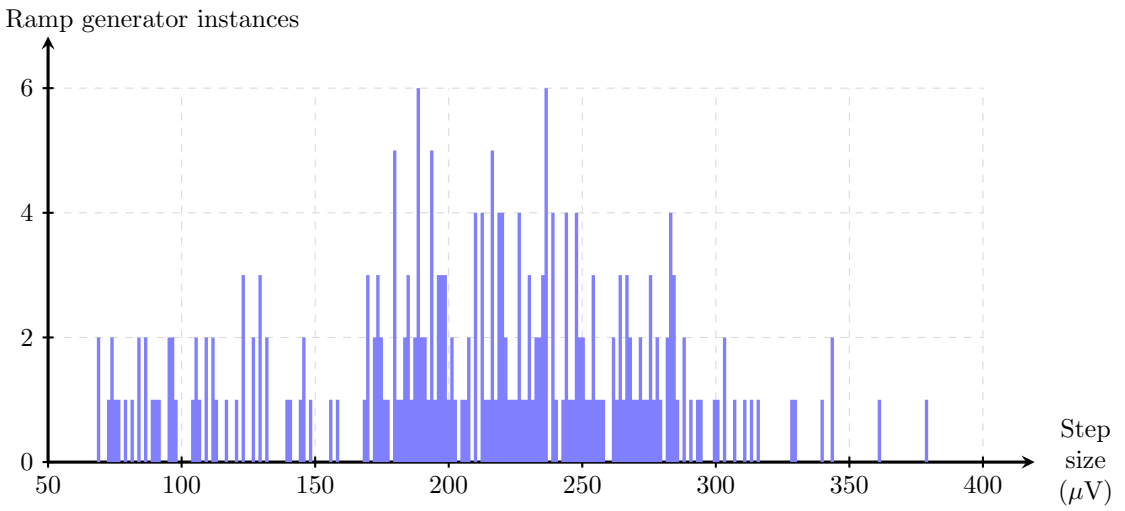


Figure A.18: Histogramme de la taille du pas rampe générée obtenue par simulations Monte Carlo faisant varier le procédé de fabrication et l'appariement

La Figure A.17 et la Figure A.18 montrent respectivement les histogrammes de la résolution de la rampe et du pas moyend d'intégration pour 250 simulations Monte Carlo. Nous pouvons noter que la résolution et la taille du pas obtenues sont bien maîtrisées dans la plage, respectivement, de $11.4-13.8$ bits et $60-380$ μV, ce qui se conforme aux spécifications requises pour le CAN ciblé. La Figure A.19 montre l'histogramme de la déviation moyenne de la pente de la rampe pour 250 simulations Monte Carlo. Nous pouvons remarquer que la déviation est contenue autour de -0.2 %, et le pire cas est autour de -6 %, ce qui encore jugé comme très acceptable.

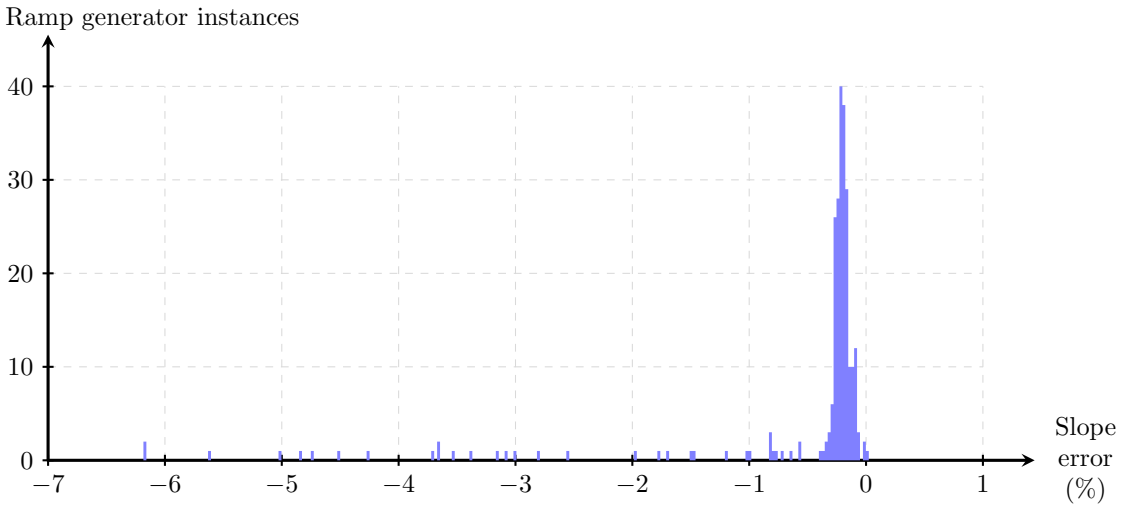


Figure A.19: Histogramme de l'erreur moyenne de pente de la rampe générée obtenue par simulations Monte Carlo faisant varier le procédé de fabrication et l'appariement

La faisabilité et les performances du générateur proposé sont ensuite évaluées dans un

cas réel. Pour ce faire, nous avons utilisé le générateur de rampe afin d'implémenter une technique de test intégré de servo-loop en combinaison avec la technique de réduction de codes sur des données mesurées d'un CAN de type pipeline de 11 bits avec 5 étages de 2.5 bits/étage de STMicroelectronics.

La configuration complète du test, décrite en Figure 3.1, a été modélisée sous Matlab. Des modèles comportementaux réalistes ont été développés pour le CAN sous test et le générateur de rampe. Comme indiqué, le modèle du CAN a été générée à partir de données physiques d'un CAN dont la linéarité statique a été mesurée à l'aide de la technique d'histogramme sur un banc de test dédié. D'autre part, les stimuli de rampe ont été directement enregistrés à partir des simulations électriques Monte Carlo au niveau transistor du générateur complet. Le moteur de test intégré, le compteur numérique, et le bloc de comparaison des mots numériques sont totalement numériques et donc, par conséquent, ont été considérés comme idéaux dans ce travail de thèse. Seuls 132 des 2046 codes de sortie du CAN sous test ont été mesurés suivant la technique de réduction de codes, ce qui représente environ 6 % du nombre total des codes, et résulte en une réduction significative du temps de test. La Figure A.20 montre la NLI estimée du CAN en utilisant la technique de test intégré proposée.

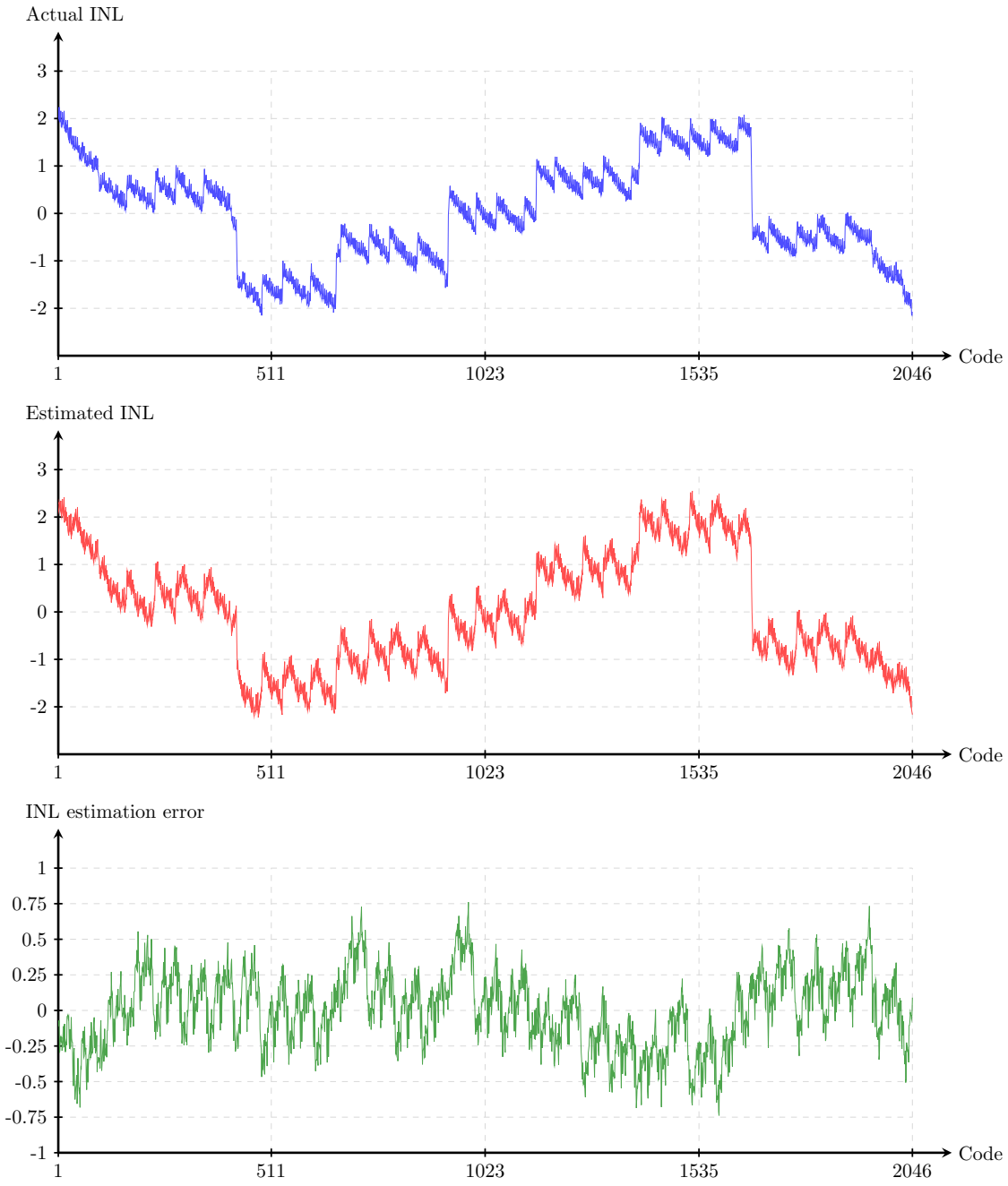


Figure A.20: NLI réelle obtenue par la technique standard d'histogramme, NLI estimée par la technique de test intégré proposée, et erreur d'estimation de la NLI

Un bruit de $\sigma_{ramp} = 0.2$ LSB Root Mean Square (RMS) est ajoutée à la sortie du générateur, et un bruit de $\sigma_{ADC} = 0.15$ LSB RMS est ajoutée à l'entrée du CAN sous test afin de simuler des transitions bruyantes. Chaque code est traversé $r = 50$ fois avec la technique de servo-loop proposée afin de moyenner le bruit. La Figure A.20 montre aussi les valeurs réelles de la NLI obtenues avec la technique de test par histogramme et l'erreur

d'estimation de la NLI à des fins de comparaison. Sous des conditions nominales, l'erreur d'estimation de la NLI est bien maîtrisée dans la plage [−0.5,+0.5] LSB.

Afin d'estimer la robustesse de la technique de test intégré proposée vis-à-vis des variations du procédé de fabrication et d'appariement, nous avons effectué 250 simulations électriques Monte Carlo au niveau transistor du générateur de rampe proposé, et nous avons employé chaque rampe générée avec la technique de servo-loop proposée. La Figure A.21 montre l'histogramme de l'erreur maximale d'estimation de la NLI enregistrée pour chaque simulation. L'erreur d'estimation de la NLI est toujours en dessous du LSB avec une valeur moyenne autour de 0.5 LSB. Les résultats présentés dans [41] montrent qu'en utilisant une rampe idéale, l'erreur d'estimation de la NLI, due exclusivement à la technique de réduction de codes elle-même, représente une erreur comprise dans la plage [−0.5,+0.5] LSB. Au vu de nos résultats en Figure A.21, dans le pire cas, la non linéarité du générateur de rampe proposé ne contribue seulement qu'à 0.5 LSB supplémentaires dans l'erreur d'estimation de la NLI introduite par la technique de réduction de codes.

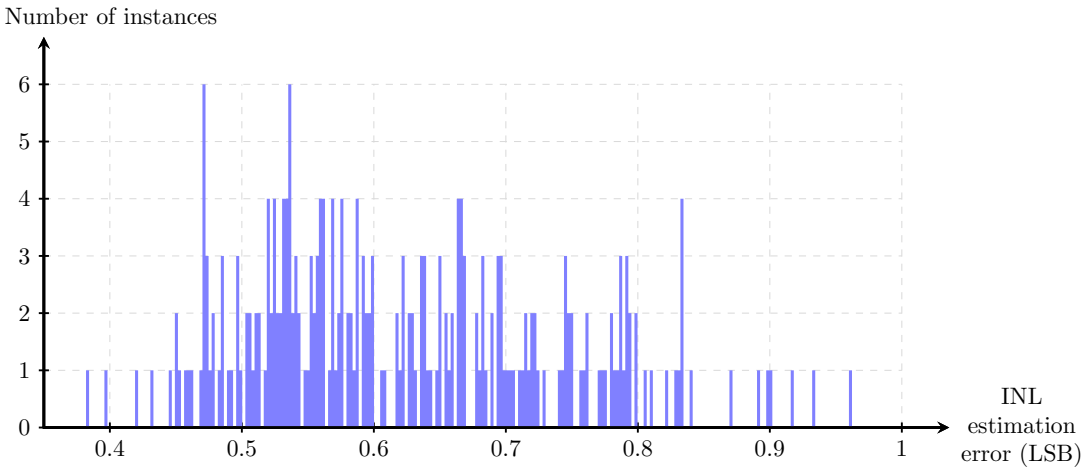


Figure A.21: Histogramme de l'erreur absolue maximale d'estimation de la NLI obtenue par la technique de test intégré proposée en considérant différents générateurs de rampe générés par simulation Monte Carlo

De même, le temps de test est calculé pour chaque simulation. Dans le cas où la technique de réduction de codes n'est pas employée, chaque code du CAN sous test est mesuré, alors que dans le cas où la technique est appliquée, seulement 6 % des codes sont mesurés. Le temps de test est réduit à 6 % du temps de test initial, mais le gain de temps est dégradé par le temps d'attente entre deux mesures. En effet, la sortie du générateur doit transiter entre les groupes de codes à mesurer, ce qui résulte en un temps de test additionnel. De plus, la technique de réduction de codes nécessite l'application d'une première rampe sur la dynamique d'entrée du CAN sous test afin de trouver et

sélectionner les codes à mesurer. Le stimulus de rampe employé pour cette tâche ne nécessite pas d'être linéaire ou lent, mais doit être monotoniquement croissant pour que l'algorithme de sélection de codes fonctionne. Cependant, dans notre cas, la vitesse de la rampe ne peut pas être augmentée, ce qui veut dire que le temps alloué à cette action peut être non négligeable. La Figure A.22 montre l'histogramme du ratio du temps de test dans le cas où la technique de réduction de codes est employée sur le temps de test quand elle ne l'est pas, afin d'apprécier le gain de temps entre les deux cas de figure.

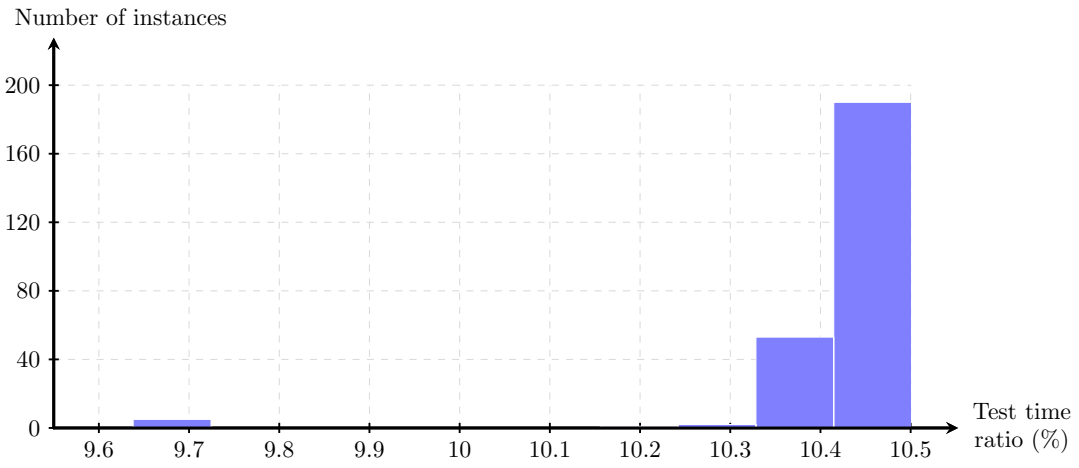


Figure A.22: Histogramme du ratio des temps de test avec/sans la technique de test intégré proposée

Comme nous pouvons le voir dans les résultats, la quasi totalité des simulations indiquent un ratio autour de 10.4 %, alors que quelques valeurs sont autour de 9.7 %. Les ratios calculés montrent que le temps de test avec la technique de réduction de codes est $\pm$ 4.4 % supérieur aux 6 % du temps de test total annoncés, ce qui montre que le temps de test additionnel n'est pas négligeable dans ce cas-ci. Cependant, la réduction du temps de test est très effective, puisque près de 90 % du temps de test initial à été supprimé, ce qui montre que la technique de test intégré proposée, avec la technique de réduction de codes, peut se mesurer à la technique d'histogramme standard pour des niveaux de précision équivalents.

A.5.2. Résultats expérimentaux

Nous avons caractérisé un total de 15 échantillons de générateurs de rampe fabriqués. La fréquence du générateur a été fixé à $f_{CLK} = 20$ kHz. Chaque stimulus de rampe a ét mesuré avec la configuration de test illustrée en Figure 5.10 à une fréquence d'échantillonnage de 1 MHz, ce qui veut dire que la carte d'acquisition (DAQ) a cap-

turé 100 échantillons par pas d'intégration de la rampe mesuré. Les données collectées ont ensuite été traitées dans Matlab. Après un moyennage sur les 100 échantillons par pas, le pas moyen, la résolution, et l'erreur de pente de la rampe ont été évalués. Chaque stimulus de rampe mesuré a été caractérisé sur trois différentes plages de tension, soit : la pleine échelle du générateur (plage d'environ 4 V), -1.5 V/$+1.5$ V (plage de 3 V), et -1 V/$+1$ V (plage de 2 V). En sachant que le pas moyen mesuré $\tilde{s}$ est calculé comme :

$$\tilde{s} = \frac{V_{out}^{+} - V_{out}^{-}}{2^{N_r}}, \tag{A.15}$$

la résolution de la rampe N_r s'exprime avec l'équation (A.15) telle que :

$$N_r = log_2\left(\frac{V_{out}^{+} - V_{out}^{-}}{\tilde{s}}\right) \tag{A.16}$$

où $V_{out}^{+} - V_{out}^{-}$ est la plage de sortie considérée (pleine échelle, -1.5 V/$+1.5$ V, ou -1 V/$+1$ V), et le nombre effectif de bits statique s'exprime tel que :

$$sENOB_r = log_2\left(\frac{V_{out}^{+} - V_{out}^{-}}{\max(|INL_r|)}\right). \tag{A.17}$$

Finalement, l'erreur de pente est calculée telle que :

$$\varepsilon_{slope} = \frac{\max(|DNL_r|)}{\tilde{s}} \cdot 100 \tag{A.18}$$

où $\max(|DNL_r|)$ est la value absolue maximale de la déviation du pas de la rampe du pas moyen en volts.

La Table A.3 liste les travaux effectués précédemment dans le domaine afin de comparer leurs performances avec nos résultats.

Table A.3: Comparaison des travaux précédents sur la génération de rampe avec la solution proposée

Référence	Technique	Complexité	Linéarité (bits)		Erreur de pente (%)	Plage de sortie (V)	Fréquence (kHz)
			Simulation	Mesure			
[56]	Génération de rampe avec ajustement automatique de pente	moyenne	N/A	8	N/A	$-2/2$	$0.1-100$
[57]	Génération de rampe avec source de courant chargeant une capacité	basse	14	N/A	N/A	0.1/0.9	$1-10$
	Architecture d'oscillateur à relaxation	moyenne	12	N/A	N/A	0.2/1.4	$30-400$
[59]	Génération adaptative d'un signal triangulaire	basse	14	N/A	0.4	-1/1	10.1
[60]	Intégrateurs adaptatifs par la méthode des moindres carrés	basse	N/A	N/A	0.6	N/A	N/A
		basse	11	10	1.5	N/A	10
		moyenne	15	11	0.6	0.7/1.3	N/A
[61]	Test intégré et simultané de CAN et CNA	haute	14	N/A	N/A	N/A	N/A
[62]	Génération de rampe à l'aide de CAN ΣΔ	haute	12	N/A	N/A	0.4/1.4	N/A
[64]	CNA avec technique d'appariement dynamique d'éléments	haute	14	N/A	N/A	$-1/1$	1
Ce travail	Intégrateur à capacités commutées et temps discret avec entrée constante	basse	13.3 (plage de 1 V)	14.97 (plage de 4 V-p-p)	4.1	$-2/2$	$0.2-200$

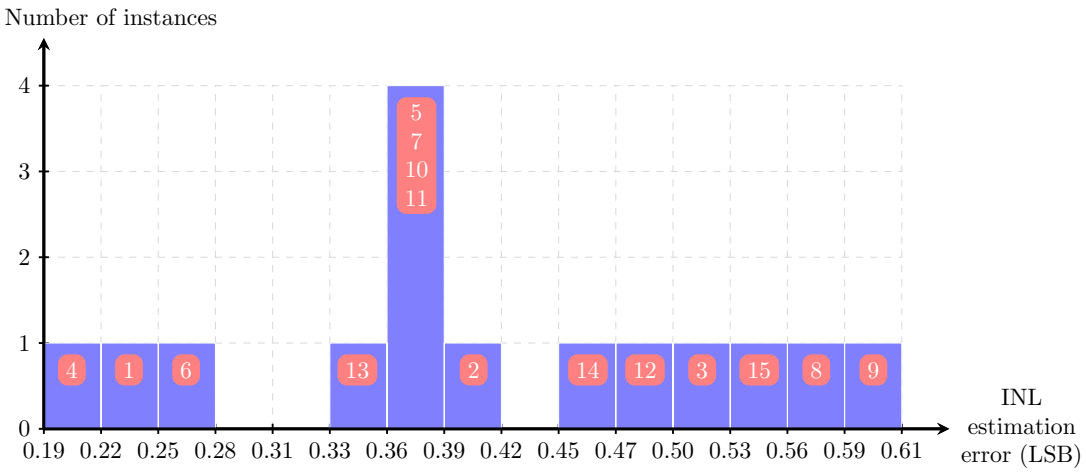


Figure A.23: Histogramme de l'erreur d'estimation de la NLI mesurée du CAN sous test en utilisant les 15 échantillons (technique d'histogramme)

Les échantillons de générateur de rampe ont ensuite été utilisés afin de caractériser un CAN pipeline de 14 bits du commerce et ainsi valider le travail de thèse sur un cas concret. Pour cette configuration de test, la fréquence du CAN sous test a été fixée à $f_{ADC} =$ 10 MHz alors que la fréquence du générateur a été fixée à $f_{CLK} = 20$ kHz, permettant de récupérer 1000 échantillons par pas du générateur. Nous avons limité la sortie du CAN sous test aux 11 premiers bits qui sont les plus significatifs (MSB) afin d'émuler un CAN de 11 bits et ainsi rester entre 2 et 3 bits en dessous de la résolution des rampes mesurées sur la plage de 2 V. Chaque itération de la mesure du CAN a été effectuée selon la configuration de test illustrée en Figure 5.10. Les données sont traitées dans Matlab en utilisant la technique d'histogramme afin d'extraire la NLI du CAN sous test. La valeur réelle de la NLI du CAN, qui sert de référence, est aussi extraite en appliquant un signal de 16 bits et 2 V différentiel très lent à son entrée et en traitant les données collectées à sa sortie avec la technique d'histogramme. La Figure A.23 montre l'histogramme de l'erreur d'estimation de la NLI mesurée.

L'erreur d'estimation de la NLI varie entre 0.19 LSB et 0.61 LSB. Parmi ces valeurs, 12 des 15 mesures montrent une erreur d'estimation de la NLI en dessous de 0.5 LSB, ce qui est très acceptable.

Les cas où les valeurs du pas d'intégration sont les plus hautes sont reliés aux erreurs d'estimation de la NLI qui ne sont pas maximales, ce qui montre que des pas d'intégration plus grands que la valeur limite requise peuvent tout de même mener à une estimation correcte de la NLI du CAN sous test.

En se focalisant sur l'échantillon #5 qui présente une erreur d'estimation de NLI modérée, la Figure A.24 trace la NLD réelle, la NLD estimée et l'erreur d'estimation de la

NLD du CAN sous test, alors que la Figure A.25 montre la NLI réelle, la NLI estimée et l'erreur d'estimation de la NLI du CAN sous test. Comme nous pouvons observer, il y a un bon accord entre la valeur estimée et la valeur réelle de la NLD et la NLI du CAN sous test.

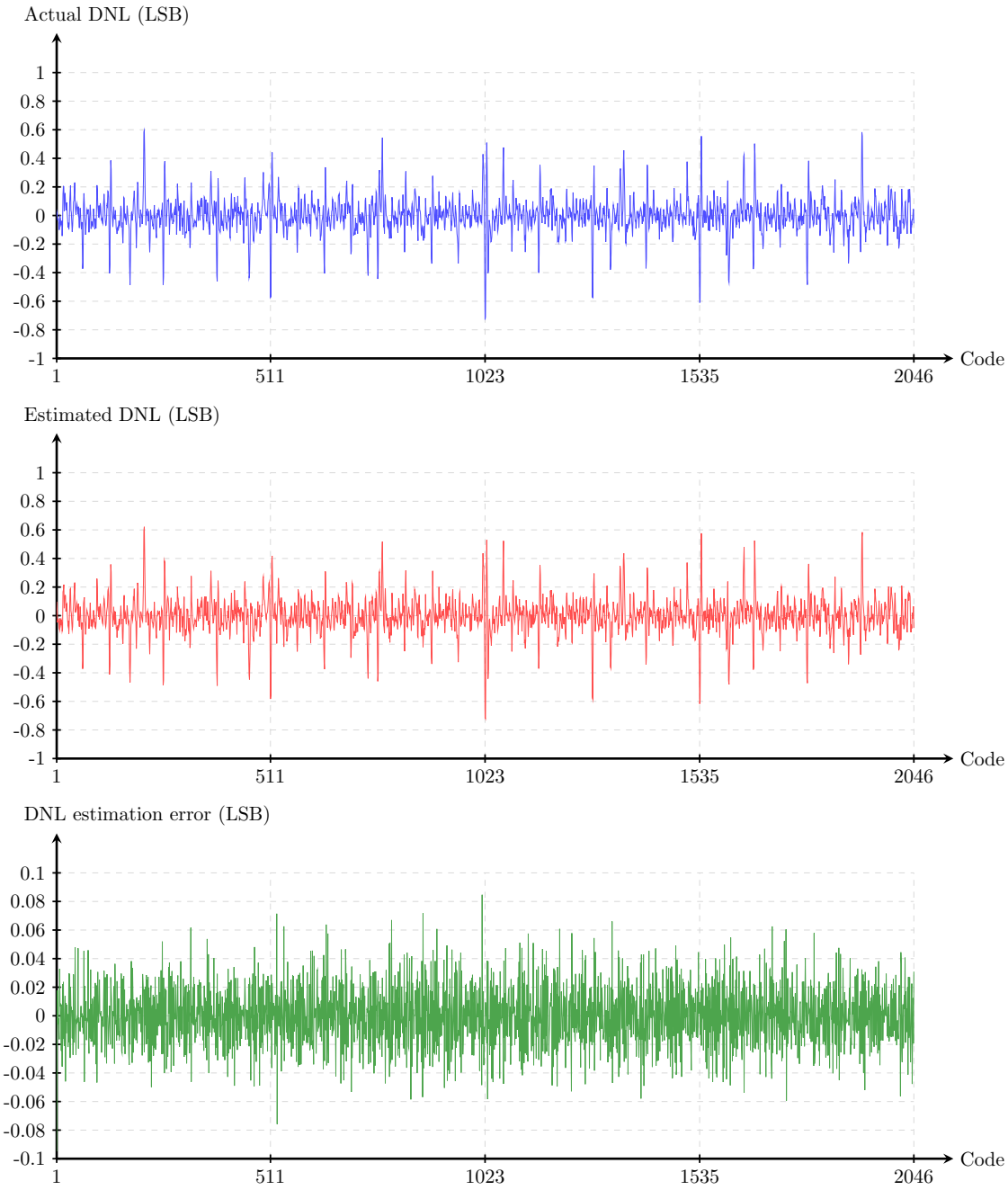


Figure A.24: NLD réelle obtenue par la technique d'histogramme et un stimulus de haute linéarité, NLD estimée obtenue par la technique d'histogramme et l'échantillon #5, et erreur d'estimation de la NLD

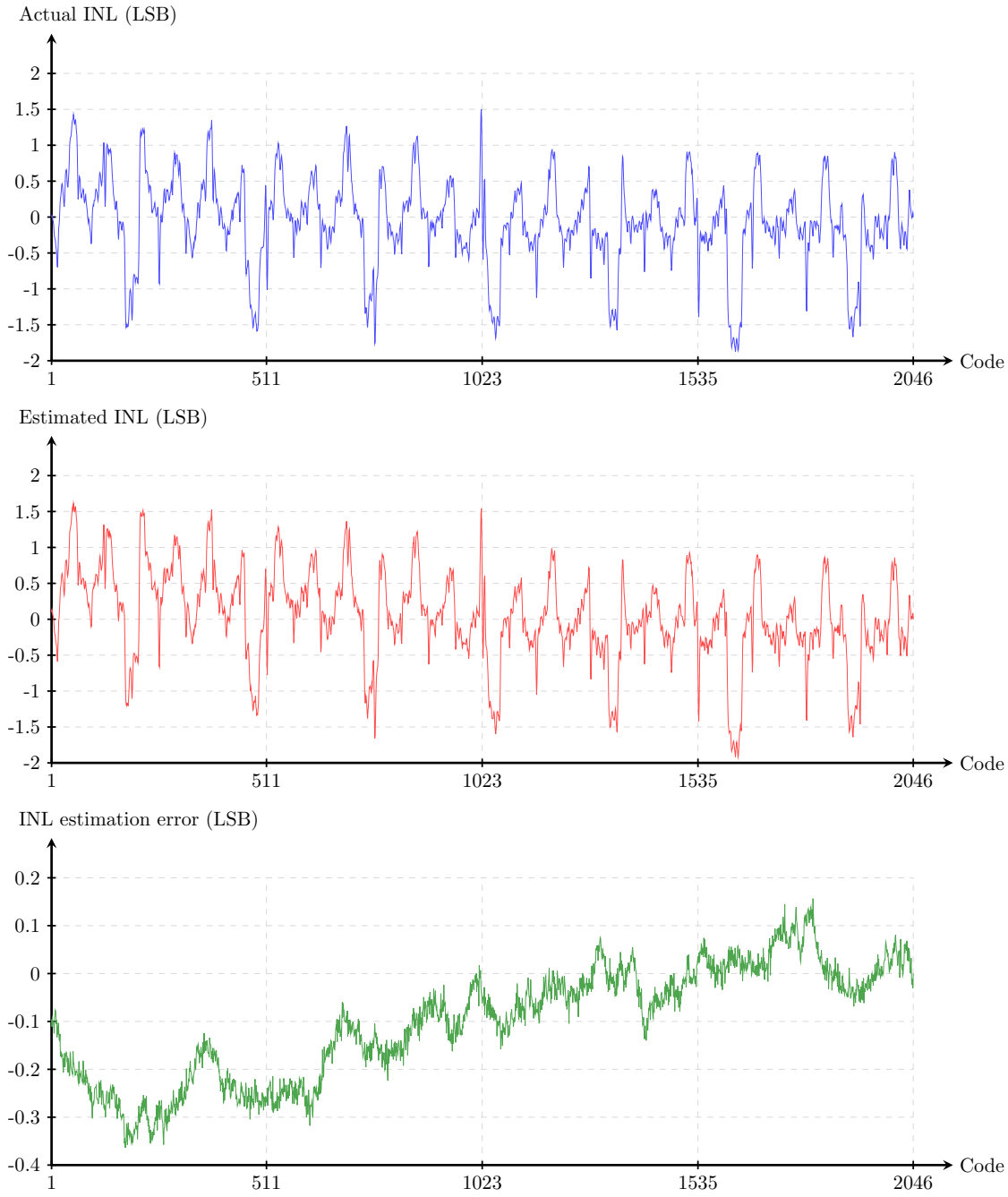


Figure A.25: NLI réelle obtenue par la technique d'histogramme et un stimulus de haute linéarité, NLI estimée obtenue par la technique d'histogramme et l'échantillon #5, et erreur d'estimation de la NLI

Abstract

Pipeline ADC Built-In-Self-Test

Abstract: This PhD thesis is aimed at exploring new Built-In-Self-Test (BIST) techniques for static linearity characterization of pipeline ADCs. During the production phase, the static and dynamic performances of the ADCs are tested. Static linearity test techniques are one of the more expensive test procedures that are performed at production line. The measurement of the static linearity performance requires the application of a low frequency high linearity stimulus and the collection of a high volume of output samples for noise averaging, usually using a histogram-based test setup. Thus, as the resolution of state-of-the-art ADCs increases, test time for static linearity characterization increases exponentially. For this reason, the reduction of the ADC test time is a hot topic that has gained an increasing interest over the past years. New techniques have recently been proposed to effectively reduce test time, but no BIST technique has yet been developed that considers a high resolution signal generator in combination with an on-chip analysis technique that dramatically reduces the amount of data. In this thesis, static linearity BIST techniques will be investigated for pipeline ADCs. In particular, this thesis presents a novel high-linearity on-chip test stimulus generator and a modified servo-loop technique that, in combination with reduced-code linearity test algorithms, lead to the definition of an efficient and accurate BIST strategy for pipeline ADCs. The work includes the experimental validation of the proposed techniques in collaboration with STMicroelectronics, Grenoble.

Keywords: Design-For-Test, Built-In-Self-Test, pipeline ADC.

Auto test de convertisseurs de signal de type pipeline

Résumé: Cette thèse vise l'étude de nouvelles architectures d'auto test pour les convertisseurs de type pipeline. En production, les convertisseurs sont testés en fonctionnement statique et dynamique. Les techniques de test statique de linéarité sont les techniques les plus coûteuses durant la phase de production. La mesure des performances statiques utilise un stimulus à haute linéarité et très basse fréquence et la méthode de l'histogramme, nécessitant la collecte d'un grand nombre d'échantillons en sortie afin de moyenner le bruit. Ainsi, la quantité de données nécessaire augmente exponentiellement avec la résolution du CAN sous test. Pour cette raison, la réduction du temps de test des CANs est un domaine de recherche qui attire de plus en plus d'attention. Récemment, des nouvelles solutions ont été mises au point pour réduire de façon importante le temps de test, mais aucune solution d'auto test considérant un générateur de signal de haute résolution en

combinaison avec une technique d'analyse intégrée, réduisant considérablement la quantité de données, n'a encore été développée. Dans le cadre de cette thèse, on envisage l'étude de techniques d'auto test statique pour ce type de convertisseurs. En particulier, cette thèse présente un générateur de stimulus de test intégré à haute linéarité et une technique modifiée de servo-loop qui, en combinaison avec un algorithme de test de linéarité avec réduction de codes, conduit à la définition d'une stratégie efficace et précise de test intégré pour les CANs de type pipeline. La thèse inclut la validation expérimentale des techniques proposées, en coopération avec ST Microelectronics, Grenoble.

Mots-clefs: Conception en vue du test, auto test, CAN de type pipeline.

Numéro ISBN: 978-2-11-129220-8