POLITECNICO DI TORINO

Master Degree Course
in Electronic Engineering

Master Degree Thesis

Relaxation Digital to Analog Converter: Analysis and Design

Supervisor
prof. Paolo Stefano Crovetti

supervisor signature

.....

Candidate
Roberto Rubino

candidate signature

.....

A.Y. 2018-2019

Acknowledgements

I would like to thank first Prof. Paolo S. Crovetti for his kind guidance and feedback, attention to details, and endorsement throughout the whole project.
A special thank owes to Eng. Pedro De Toledo, for his friendly help and expertise-sharing, as well as the availability of Dr. Orazio Aiello, without whom this collaboration would probably never have begun.
Last but not least I want to thank my family, who gave me the background and robust support to carry on my academic career, and the closest friends who shared with me these years.

Abstract

Emerging Internet of Things (IoT) applications require ultra-low voltage, reconfigurable, highly energy and power efficient integrated circuits (ICs) for sensing, computing and communicating, which are extremely challenging to be designed by traditional techniques.
Interfaces towards the analog world, in particular, require medium-to-low resolution (8-12bit) data converters with a sample rate in the hundreds of kS/s range, capable to operate at supply voltages below 0.5V and achieving an energy figure of merit (FOM) in the order of 1fJ/(conversion step).
In this framework, the thesis addresses the analysis and design in 40nm CMOS technology process of a Digital-to-Analog Converter (DAC) suitable to the challenges of IoT applications. The converter, based on the novel-in-concept relaxation digital-to-analog conversion technique, exploits the impulse response of a first order RC network to generate and sum up binary weighted voltages to perform D/A conversion.
The followed approach begins with the theoretical analysis of the converter working principle, identifying non-idealities, modelling their effect on converter performance and developing design guidelines to minimize them.
A set of configurations of the analog block has been chosen to explore the space of design solutions and implemented within the 40nm PDK. Cadence simulations have been carried on the chosen configurations, initially driven by behavioural Verilog-A blocks and performed at the schematic level, allowing a complete formulation of a design strategy and compensation technique.
Post layout simulations of the analog block follows, allowing to choose the better configuration solutions, converging to few final test-cases.
Digital circuit which drives the converter and implements the compensation strategy has been designed in Verilog HDL code.
After synthesis and layout generation has been performed for the digital blocks, the thesis is concluded by evaluating post-layout performance of the whole converter.

The DAC presented in this work turns out to be a tiny, matching insensitive, almost fully-synthesizable architecture, allowing conversion rates up to MS/s with a state of the art energy figure of merit (FOM).
Future developments include digital design optimization for further conversion energy enhancement, circuit fabrication and post-fabrication performance validation, insertion of the Relaxation digital-to-analog converter as a part of more complex systems which would take advantage of the presented topology features.

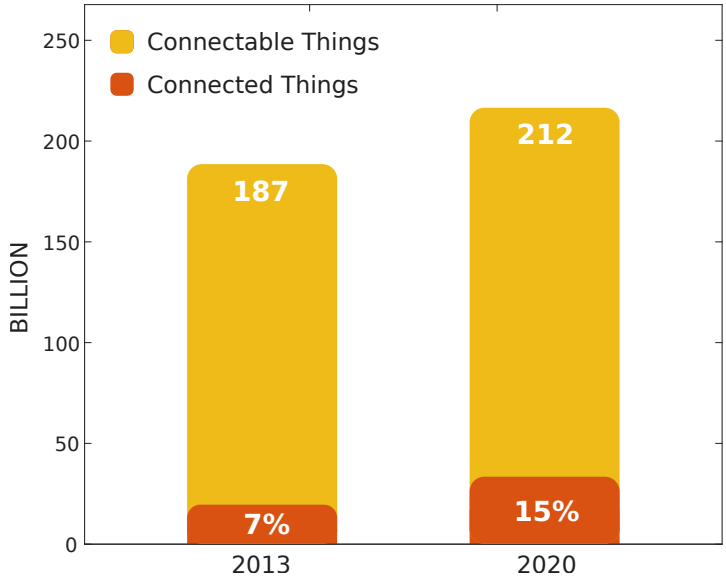
Contents

Chapter 1

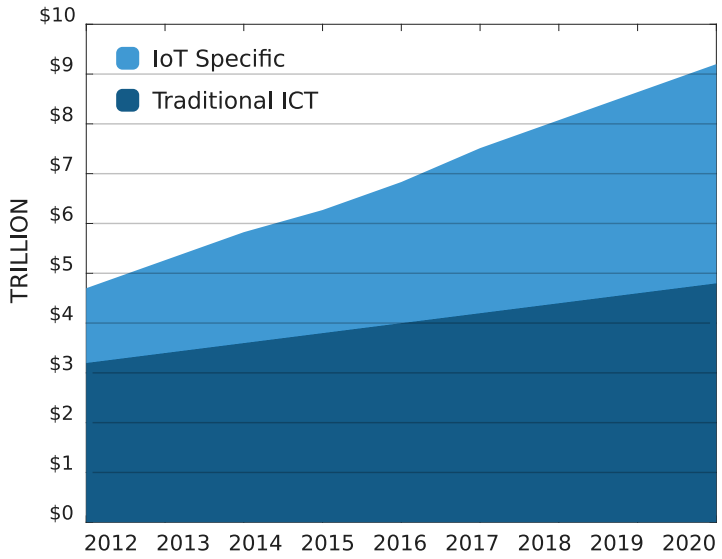
Introduction

In this chapter, IoT trends and circuits are presented, focusing on the general IoT node structure and functional specification. An overview of most commonly used data converters is exposed, followed by a description of the present work structure.

IoT is about establishing relation between the Internet and people's activity in the physical world, through a distributed network of sensing, computing and communicating nodes, which are provided with energy storage and harvesting capabilities.
The amount of connectable things is constantly increasing [1], among which, the number of actually connected devices is showing an even faster growth (Figure 1.1(a) on the following page).
From the economical perspective, the amount of vendor incomes related to IoT is also increasing faster than traditional ICT products (Figure 1.1(b) on the next page) and data generated by the sector is currently approaching 10% of the whole Digital Universe (DU) data [1] (Figure 1.2 page 3).



(a) The Internet of Things is Exploding.



(b) IoT will subsume the ICT industry.

FIGURE 1.1 – IoT trends (source [1]).

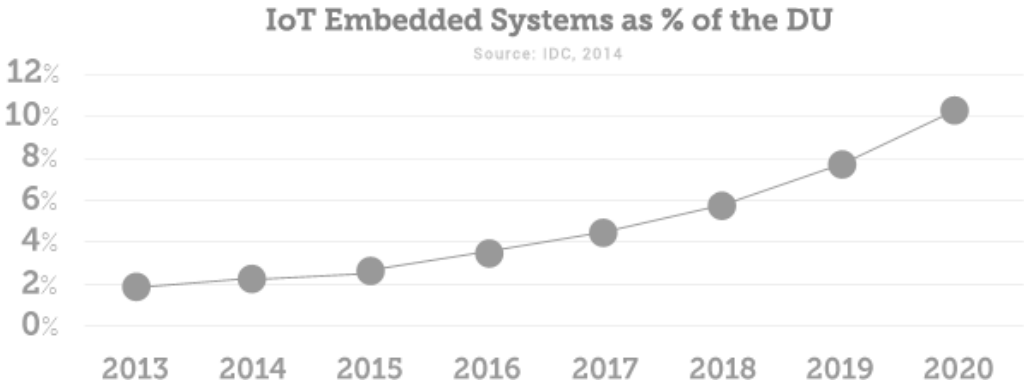
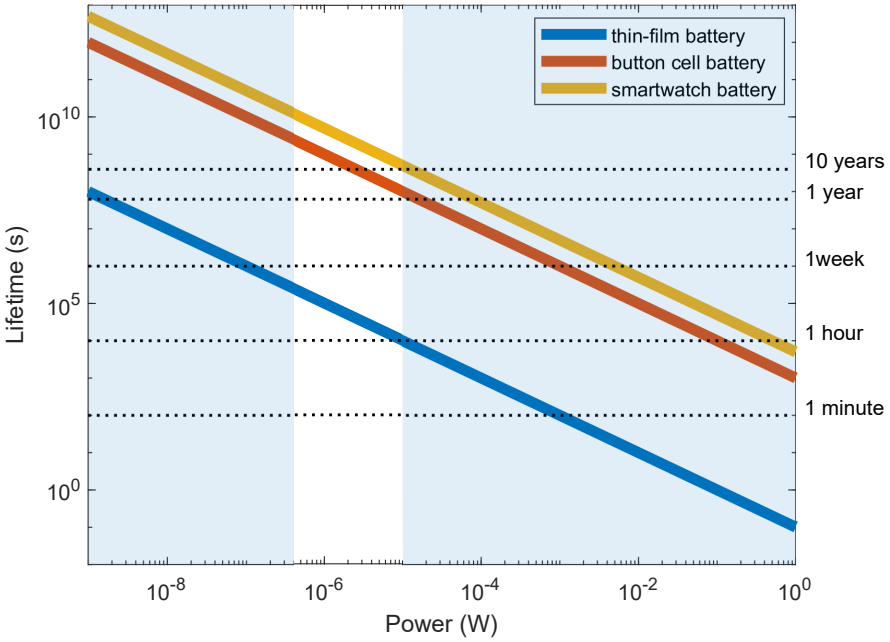


FIGURE 1.2 – The Internet of Things will Subsume the Information and Communication Technology IndustryIoT trends (source [1]).

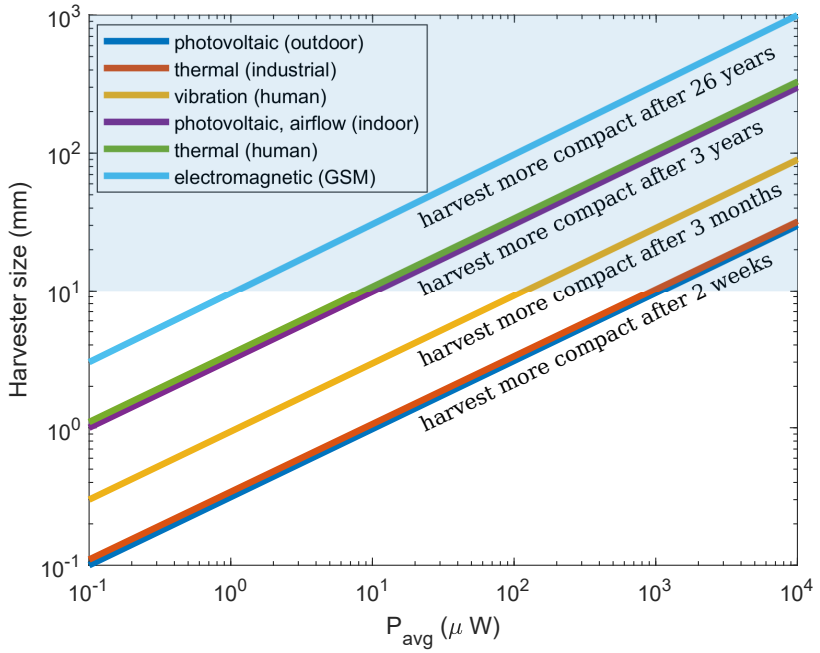
Due to the increasing number of connected devices, the need for low power consumption arises for sustainability reasons, in addition to the intrinsic energy constraint of IoT nodes. This latter requirement is related to lifetime of IoT devices, which is proportional to the capacity of battery and the average power consumption of the node itself [2], according to the expression:

$$t_{lifetime} = \frac{E_{battery}}{P_{avg}}$$

The lifetime of an IoT node is expected to be in the order of years without maintenance (i.e. without battery replacement) [2]. This means, with the available battery technology, that average power consumption must be below tens of μW (see Figure 1.3(a) on the next page).
If the battery alone is not enough to fulfil energy requirements for the specific application, the need for energy harvesting devices arises. Indeed, increasing lifetime constraint requires bigger and bigger batteries, so that, for long enough target lifetime energy harvesters are more compact than batteries, as shown in Figure 1.3(b) on the following page. In order to provide long lifetime, autonomous in energy systems, each part of the architecture must be highly optimized to the purpose of low voltage, low energy and size constraints.



(a) Lifetime vs average power consumption for different batteries



(b) Harvester size vs average power consumption for different harvester size

Figure 1.3 – IoT trends (source [1]).

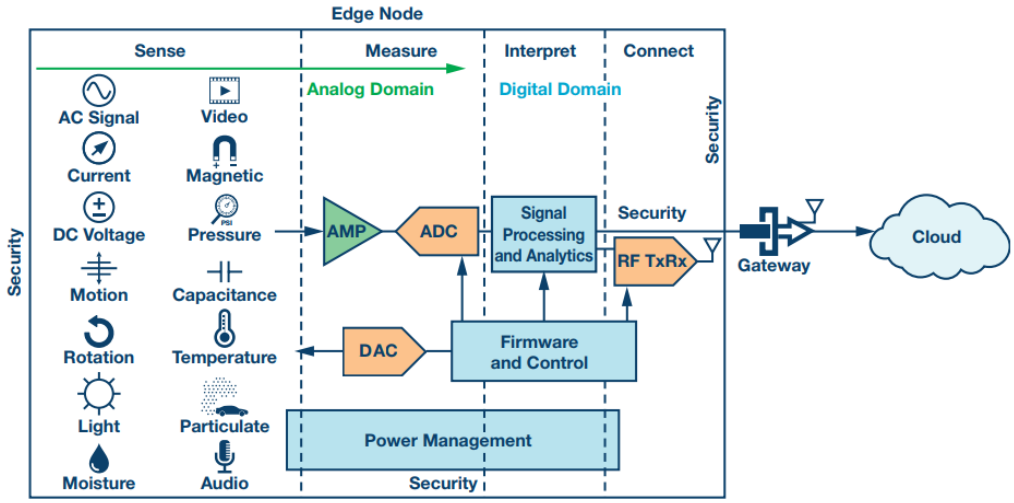
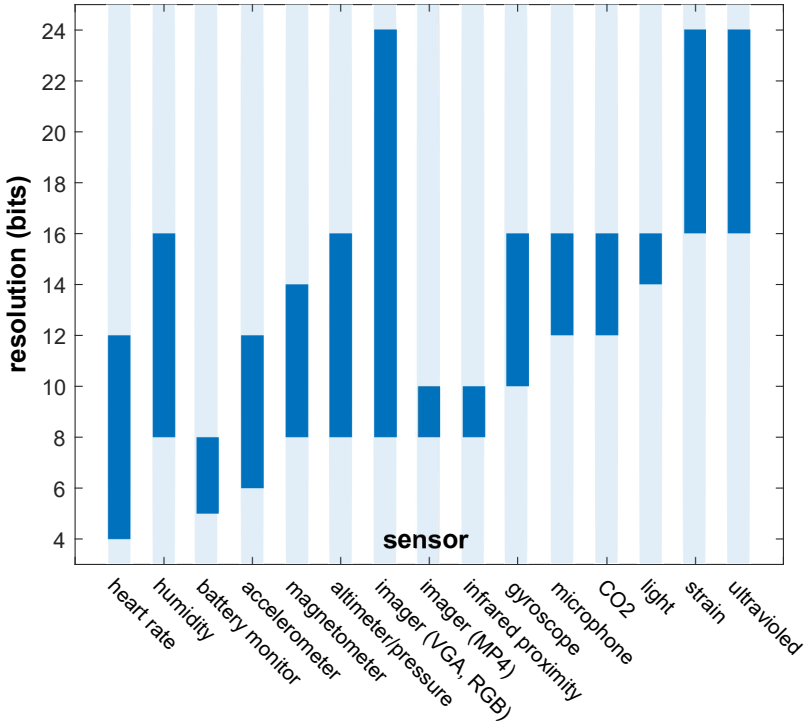


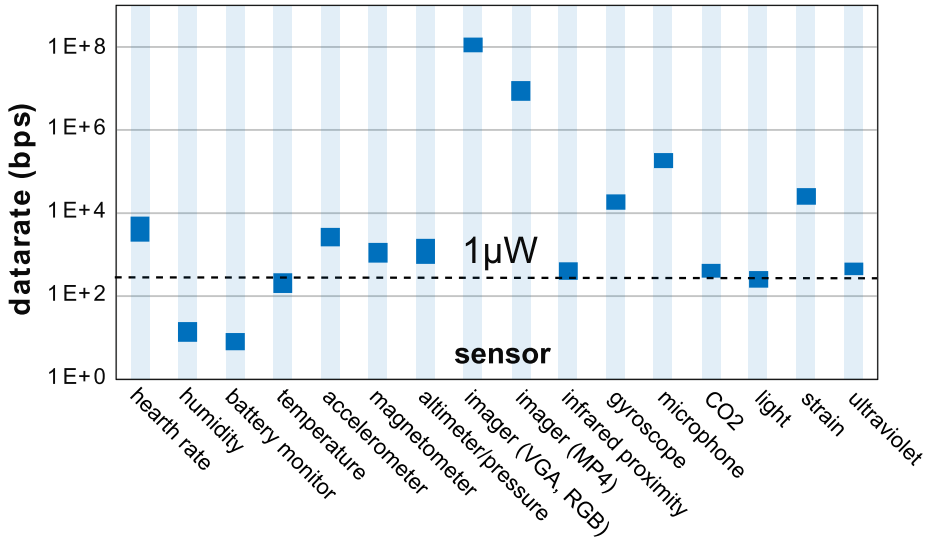
FIGURE 1.4 – IoT node architecture (source [3]).

1.0.1 Circuits and data converters for IoT

The general architecture of an IoT node is shown in (Figure 1.4), where analog and digital circuits shrinked together in a low power System on a Chip (SoC), are playing a key role.
Every information coming from the sensing devices undergoes the conversion process "physical signal" → "digital signal" → "data". The reverse process happens for each operation which needs to act to the world ("digital signal" → "control signal"→"thing behaviour") [4]. This process is enabled by data converters, i.e. ADC and DAC, which have to provide, for the generic IoT application, a wide resolution range (8÷16 bits). This requirement reduces to 8÷12 bits (for ADC) for most of the practical applications and it reduces to an even lower range for DACs if the node is implemented on a Microcontroller Unit (MCU) [2]. Resolution reduction must be tailored to the specific application, avoiding using general-purpose solutions, and it is necessary in order to minimize energy requirements, since energy dissipation increases exponentially with resolution [5]. This is the main reason why power consumption for data converters of Figure 1.6 page 7 spreads over a wide range. A sum-up of common resolution ranges for different sensors is shown in figure Figure 1.5(a) on the next page.
Concerning data rate constraints for IoT, most applications need hundreds to some thousands of bits per second, and very few applications (i.e. image and audio streaming) need higher data rates. This is resumed in Figure 1.5(b) on the following page.



(a) Converter resolution vs type of sensor



(b) Converter data rate vs type of sensor

Figure 1.5 – (source [2]).

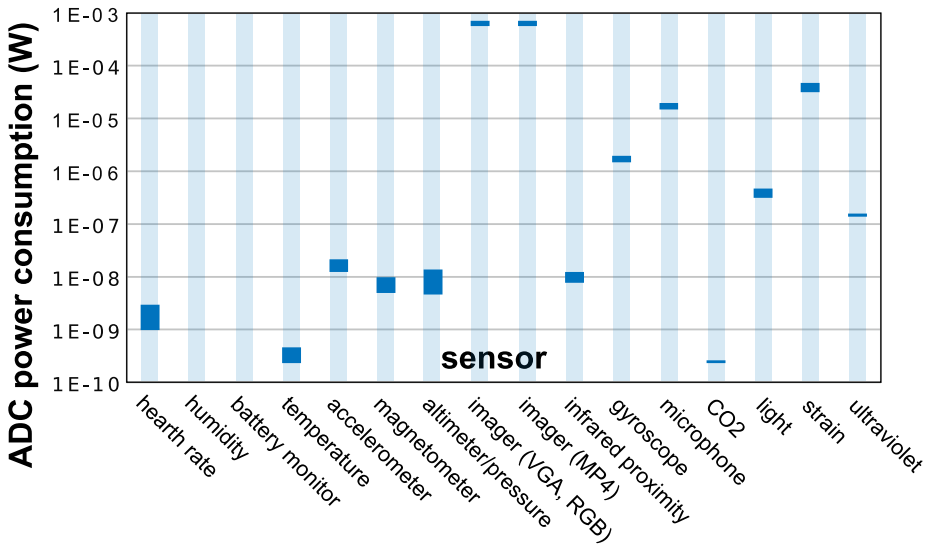


FIGURE 1.6 – Converter power vs sensor (source [2]).

1.0.2 ADC architectures overview

A brief review on most common architectures for analog to digital converters is presented:

Flash ADC It is potentially the fastest converter architecture, and it requires a large number of resistors and comparators to operate (2^N and 2^{N-1} respectively). The resistor ladder divides the supply reference voltage into 2^N equally spaced voltages which become the references to which input voltage is compared by each comparator (see Figure 1.7 on the next page). For a given input voltage a certain number of consecutive comparators will provide a logic '1' as output, and the remaining comparators will output a logic '0', translating input analog voltage in the so-called *thermometric code*. This thermometric code is then decoded into a binary word by a decoder.
Flash converters require very fast comparators, which translates into high power dissipation. Even tough conversion speeds can reach hundreds of MS/s, matching constraints on resistors limits converter resolution below 10 bits [6]. Common applications include data acquisition, satellite communication, radar processing, sampling oscilloscopes, and high-density disk drives [7].

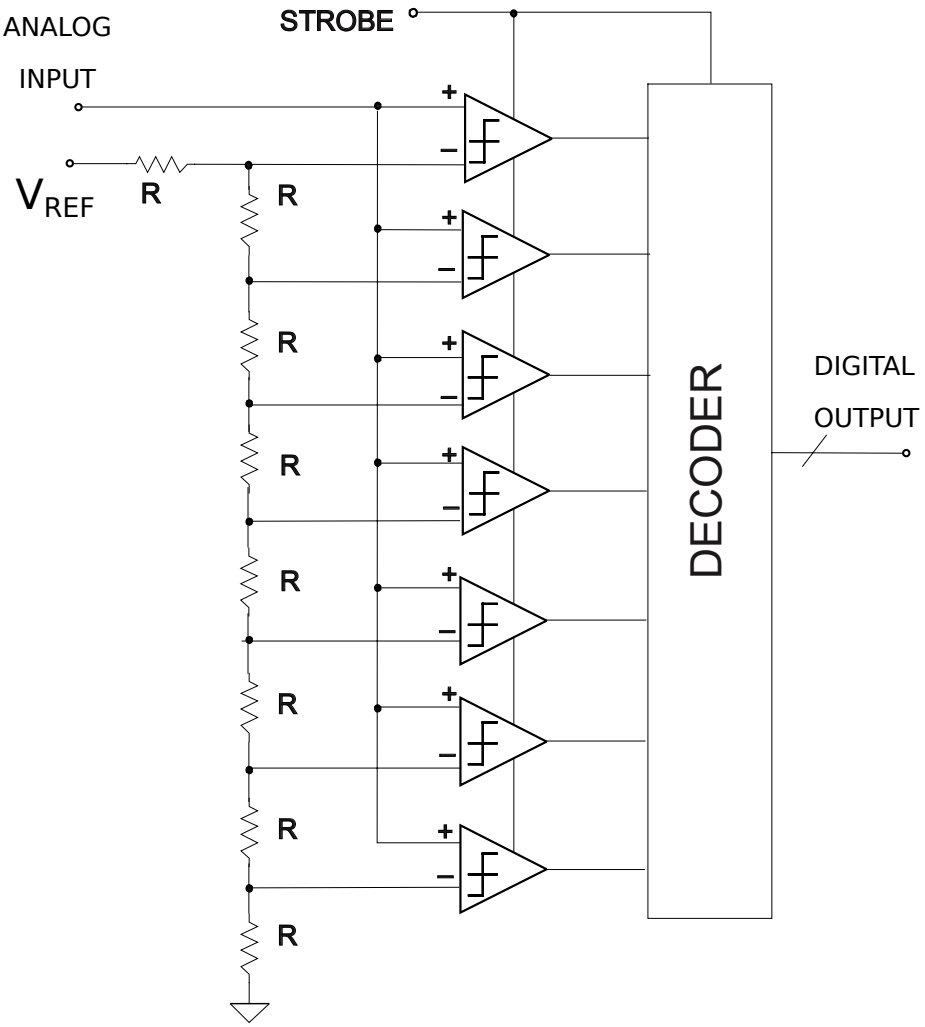


FIGURE 1.7 – Flash ADC, source [6].

SAR ADC Nowadays SAR ADC is the most popular architecture for data-acquisition applications, having resolutions from 8 to 18 bits, with sampling rates up to few MS/s [8].
They also prove to be the most performing ADCs in terms of energy per conversion(1), in which the DAC block plays a crucial part [2], making this topology largely employed in low power SoCs.
The general structure of a SAR ADC is shown in Figure 1.8. When con-

(1)Energy is expressed in terms of power P over equivalent Nyquist frequency f_{snyq}. This latter is the *equivalent Nyquist sampling frequency*, equal to half the bandwidth of the converter (independently on the actual sampling rate), useful to compare oversampled ADCs to Nyquist-rate ones

version starts, the Sample and Hold amplifier (SHA) is set into the *hold* mode, while the Successive Approximation Register (SAR) is reset to zero, except for the MSB which is set to '1'. Sampled input is compared to the SAR value, translated into an analog voltage level by the internal DAC. The result of comparison operation decides the new value for the MSB. By setting to '1' the second MSB the cycle restarts, until which all bits have been determined.

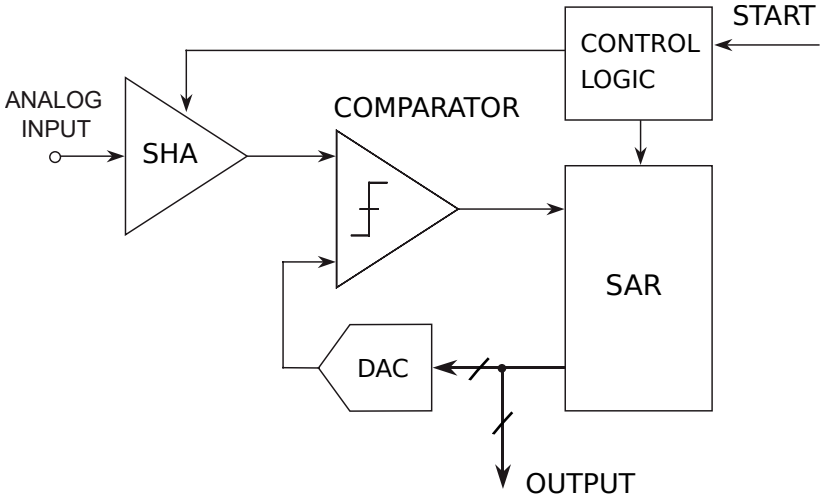


Figure 1.8 – SAR ADC.

In Figure 1.11 page 11, conversion energy is plotted as a function of SNDR, and SAR ADCs are in the typical IoT area.

Pipeline ADC Pipeline converter can achieve medium-low resolution (8 to 18 bits) and sample rates from few MS/s up to GS/s, replacing applications dominated in the past by Flash ADCs. Applications which require these high speed ADCs include instrumentation applications (digital oscilloscopes, spectrum analyzers, medical imaging) video, radar, communications (IF sampling, software radio, base stations, set-top boxes, etc.), and consumer electronics (digital cameras, display electronics, DVD, high-definition TV) [8].
The structure of a Pipeline ADC is shown in Figure 1.9 on the following page. When conversion begins, the first stage samples and holds the analog input, produces a k-bit digital estimate of the held input, converts this estimate to analog and subtracts it to the input voltage [9]. The result is amplified by a power of 2 and passed to the second stage, which performs the same operation of the first one, while the first stage samples

a new analog data. The first data is available after N/k cycles, then a whole converted word is available at each cycle.

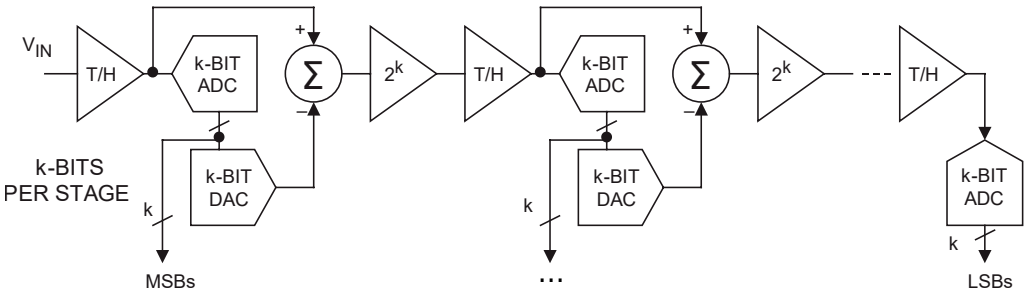


Figure 1.9 – Pipeline ADC, source [6].

$\Sigma\Delta$ **ADC** Sigma-Delta ADCs is employed whenever a low-bandwidth, low cost, high resolution converter is needed [10], and it is mostly employed nowadays in audio applications. Low resolution DACs are employed in $\Sigma\Delta$ ADC converters, where oversampling is employed to get high resolutions thanks to noise-shaping, avoiding stringent matching requirements. The general structure of a first order $\Sigma\Delta$ converter is shown in Figure 1.10, including a one-bit ADC and DAC, an integrator and an analog summing circuit. The principle of operation is based on oversampling the input signal so that quantization noise is spread over a bandwidth which is much larger than the one of the signal. Quantization noise is then shaped by the integrator so that the most of it is moved outside the band of interest and digitally filtered. Output data rate is finally reduced by a *decimator* circuit.

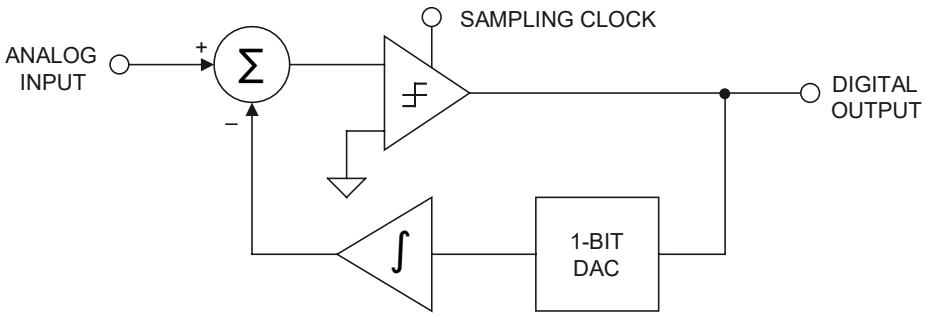


Figure 1.10 – $\Sigma\Delta$ ADC, source [6].

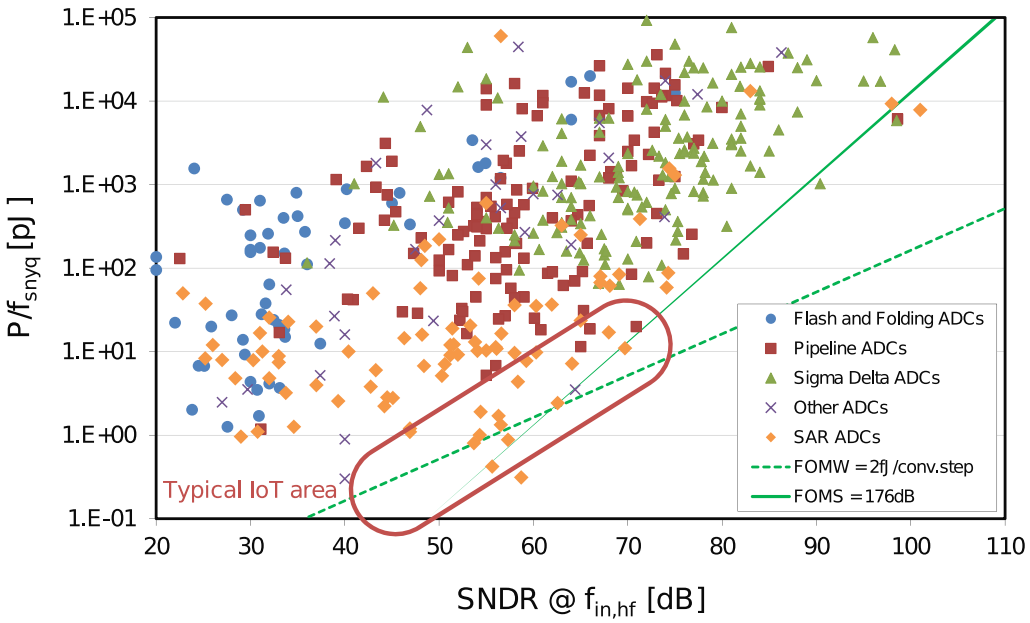


FIGURE 1.11 – Energy per conversion step for various ADC (source [2]).

1.0.3 DAC architectures overview

Resistor based DACs The most simple resistor-based DAC is the *Kelvin divider* (Figure 1.12 on the following page), a highly linear and intrinsically monotonic topology. The output is obtained by connecting it to a resistor tap, closing just one switch. Resolution is limited to few bits due to large amount of matched resistors needed, and conversion speed is basically limited by the output buffer needed to decouple the DAC code-dependent output impedance. It is rarely used as stand-alone converter, often implemented as a part of more complex converters [6].
To mitigate the large number of resistors a *binary weighted resistor network* is used, often implemented as R-2R (Figure 1.13 page 13).

Current steering DAC Current steering DACs are converters employed in high speed applications, from video to wireless communications, providing medium-low resolution of 8 to 12 bits and conversion rates up to few *G*S/s. The basic structure of a binary weighted current steering DAC is shown in Figure 1.14 page 14. Currents are steered to the physical ground or to a virtual ground to be summed-up depending on digital code. Summed currents are finally converted to a voltage with a transimpedance amplifier. The main issue of this topology is matching the

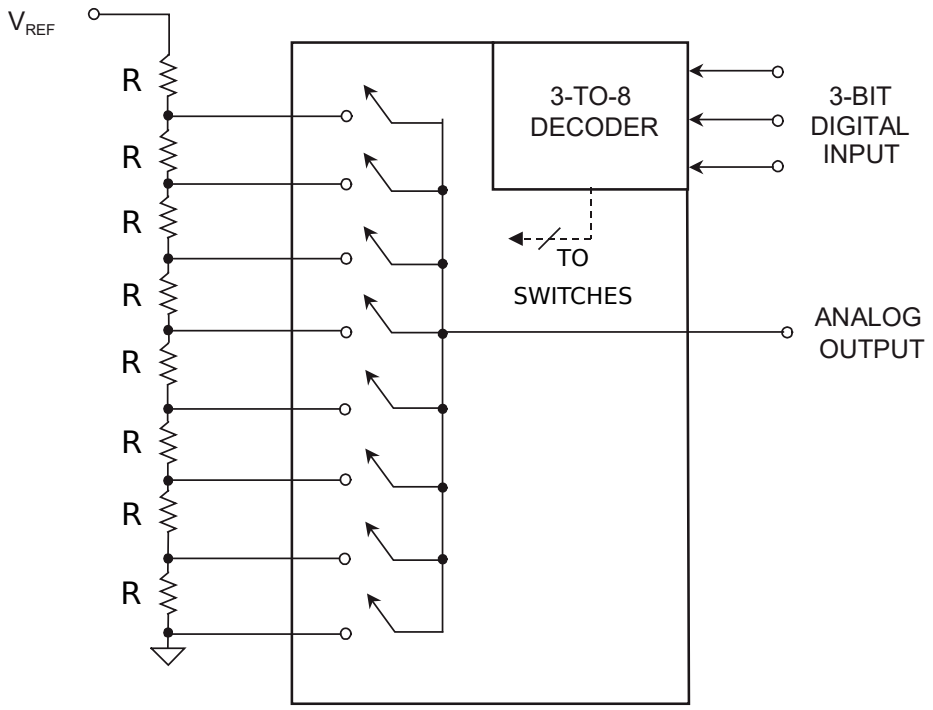


FIGURE 1.12 – Kelvin DAC, source [6].

binary-weighted current sources, which can be implemented as voltage-driven resistors connected to virtual ground or transistor current sources. The matching problem is partially solved using equal current sources driving a R-2R ladder (Figure 1.15 page 14). The topology is used in applications where energy is not a concern, suffering code-transition glitches and high static dissipation.

Capacitor DAC Capacitor based DACs are quite appreciated as low power converters, for their ideally null quiescent current, low sensitivity to opamp offset voltage, 1/f noise, and finite-amplifier gain [11]. A binary-weighted capacitor array DAC is shown in Figure 1.16 page 15. The conversion is performed by resetting all capacitors and subsequently applying the digital code to convert. Another advantage is that capacitor DAC itself behaves as a sample and hold [6]. Capacitor-based DACs having resolutions of 6 to 18 bits and conversion rates up to hundreds of MS/s are, as a matter of fact, the reason why SAR ADCs are easy-to-scale architectures, digital-in-nature and able to accommodate rail-to-rail signals without resorting to precision amplifiers.

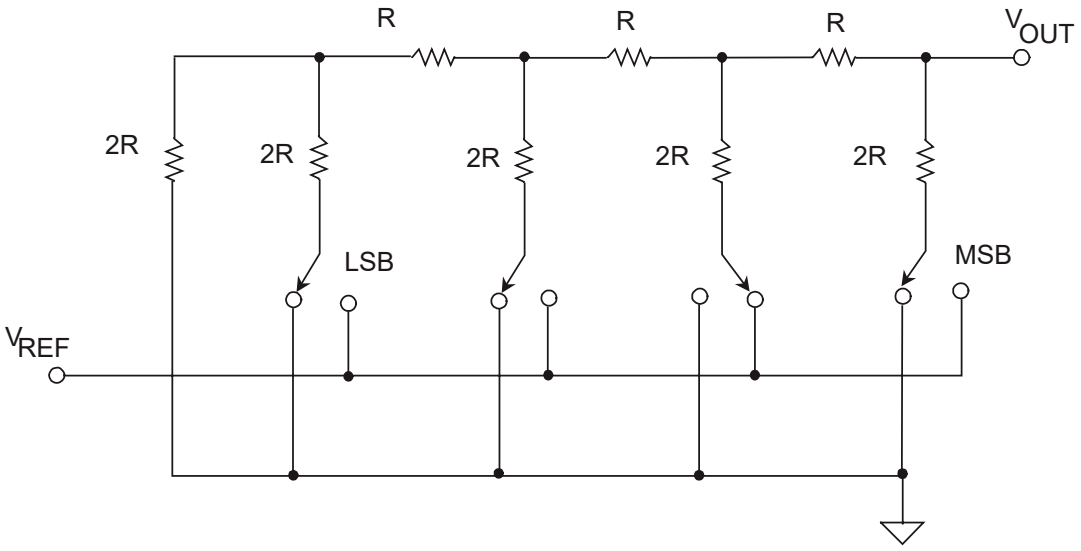


Figure 1.13 – R-2R voltage mode DAC, source [6].

$\Sigma\Delta$ **DAC** Sigma delta DAC, as well as their ADC counterpart, are used to trade sample frequency for resolution, but in the DAC case $\Sigma\Delta$ modulator is a digital one instead of an analog one [6].
The structure of a $\Sigma\Delta$ DAC is depicted in Figure 1.17 page 15. The low frequency digital data is loaded, upsampled and interpolated by a digital interpolation filter. The digital $\Sigma\Delta$ modulator moves quantization noise at higher frequencies and data is translated into a bitstream by a 1-bit DAC, finally filtered by an analog LPF. Thanks to oversampling the complexity of analog LPF is much lower than in Nyquist-rate operation case [6].
$\Sigma\Delta$ converters have sampling rates in the order of kS/s and resolutions up to 24 bits, are employed in audio applications and, in general, where a wide dynamic range is required.

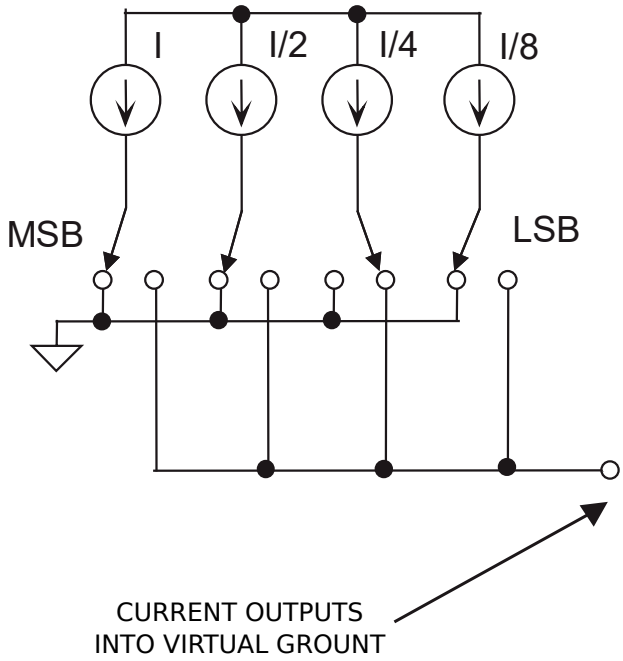


Figure 1.14 – Current steering DAC, source [6].

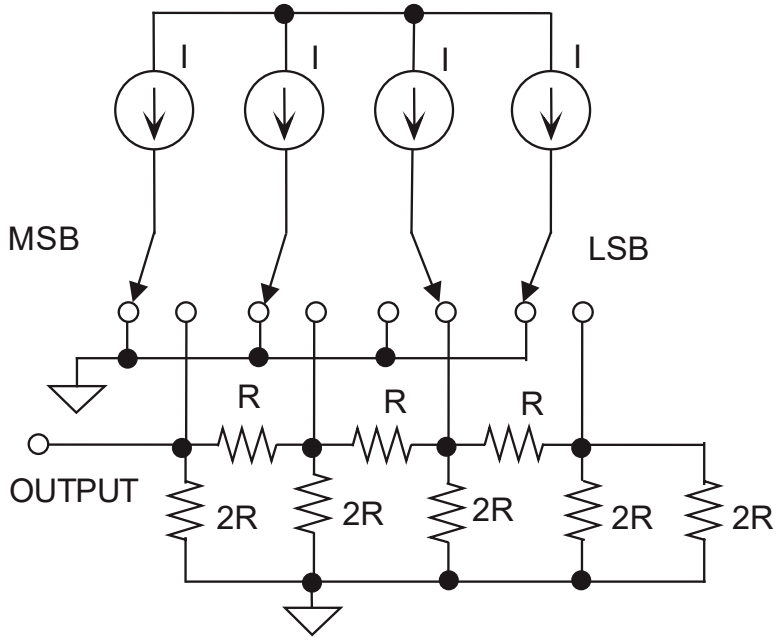


Figure 1.15 – Current steering DAC using R-2R ladder, source [6].

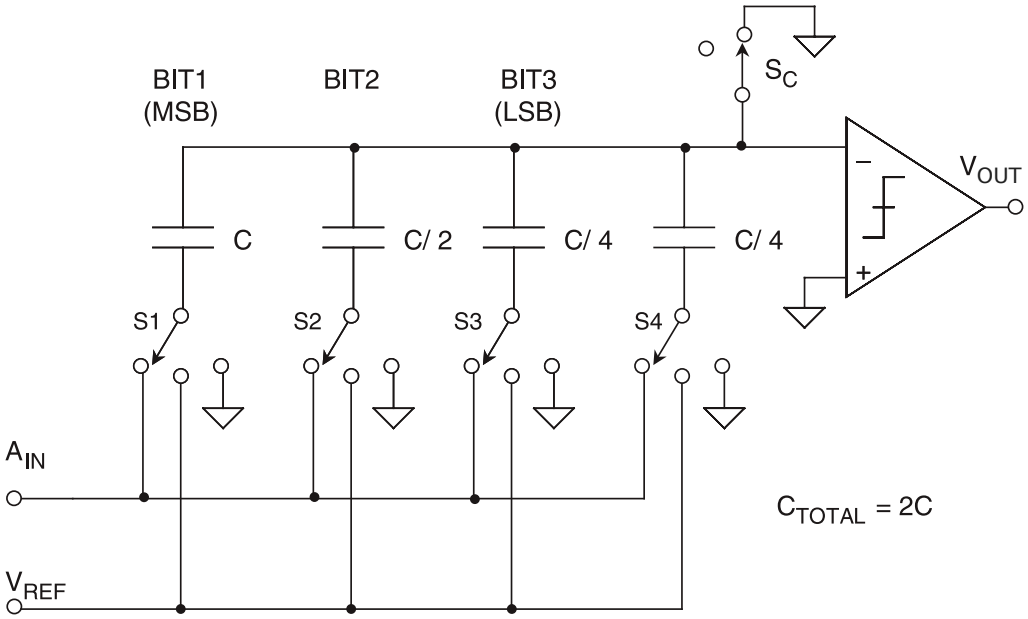


Figure 1.16 – Capacitor DAC, source [6].

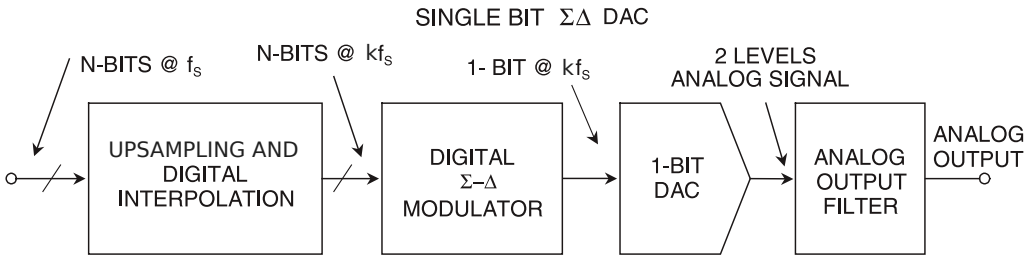


Figure 1.17 – One bit ΣΔ DAC, source [6].

1.1 This work

In this framework, the thesis addresses the design and the integration in a 40nm CMOS technology process of a Digital-to-Analog Converter (DAC) suitable to the challenges of IoT applications. The converter is based on the novel-in-concept *relaxation digital-to-analog conversion technique*, which exploits the impulse response of a first order RC network to generate and sum up binary weighted voltages for D/A conversion.

1.2 Thesis organization

In the Introduction Chapter, Internet of Things (IoT) trends and IoT-related data converters have been presented.

Second Chapter offers an overview of Digital-to-Analog Converter (DAC) static and dynamic specifications.

In Chapter Three, the working principle of the Relaxation DAC is exposed, fundamental equations are derived, and models to predict the effect of clock frequency error, jitter, charge injection, component parasitics and thermal noise on converter linearity are analysed. The theoretical estimation of Relaxation DAC energy per conversion is also performed.

In Chapter Four a design procedure for the Relaxation DAC is presented, from passive components sizing to active devices, following guidelines derived in the previous chapter, defining a complete parasitics-compensation technique and a set of test-case analog designs to explore the space of design solutions.
Cadence simulations have been performed on different converter blocks at the behavioural, schematic and post-layout level, in typical conditions and for process and temperature variations. The evaluation of converter performance allowed to choose a set of best-performing test-cases to carry on final simulations.

In Chapter Five, the digital design of blocks previously modelled as behavioural ones is shown, and design choices explained. Layout of digital blocks, described in Verilog and synthesized, has been generated. After complete post-layout simulations, the whole Relaxation DAC is characterized, and conclusions on converter performance are drawn-up, along with the possible future developments.

Chapter 2

DAC performance specification

This chapter presents a review of static and dynamic performance specifications of digital-to-analog converters, followed by an overview of applications where low power and low area DAC is required.

2.1 DAC specifications

A digital-to-analog converter converts a digital N-bit word to 2^N corresponding discrete voltage levels between 0 and full scale voltage V_{FS} (Figure 2.1 on the following page). There exists a variety of different input code schemes, but the most common is the natural binary one, for which a simple linear expression relates output voltage to bit values:

$$V_{\mathrm{o}} = V_{\mathrm{FS}} \cdot \sum_{i=0}^{N-1} \frac{b_i}{2^{N-i}} \tag{2.1}$$

where b_{N-1} represents the most significant bit (MSB) while b_0 is the least significant bit (LSB). While full scale value is not reachable digitally, since the maximum discrete voltage is $V = (1 - 1/2^N)V_{\mathrm{FS}}$, it is the reference with respect to which analog levels are normalized.
The transfer response of a D/A converter is defined to be the set of analog levels that occur for each of the digital input words [11] and it is represented for $N = 3$bit in Figure 2.1 on the next page.
Digital-to-Analog Converters are designed, characterized and compared to each other according to a common set of performance benchmarks.
The most important ones for DACs are resolution, accuracy and sample rate [10].

Resolution is the number of voltage levels the DAC is able to produce (e.g. 2^N for an N-bit converter). The resolution of data converters may be expressed in several different ways: as weight of the LSB, parts per million of full-scale (ppm FS), millivolts (mV), etc [6]. Resolution is not necessarily an indication of the accuracy of converters, due to converter nonlinear errors.

Absolute Accuracy The absolute accuracy of a converter is defined to be the absolute value of the worst-case difference between the actual output value and the nominal output value for any input code, without gain and offset correction [12] (see Figure 2.2 on the facing page). The so-called *nonlinearity errors* define the accuracy after the offset and gain errors have been removed.
Accuracy can be expressed equivalently as a percentage error of full-scale, as the effective number of bits, or as a fraction of LSB.

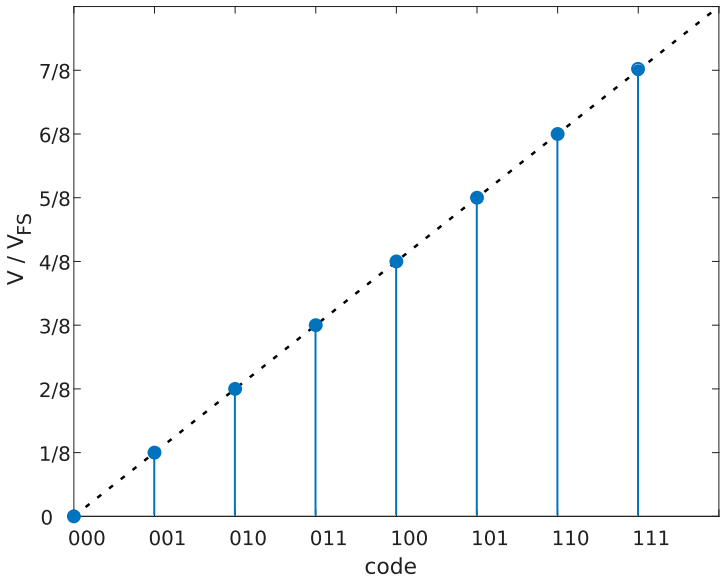


Figure 2.1 – DAC trans-characteristic for N=3-bit.

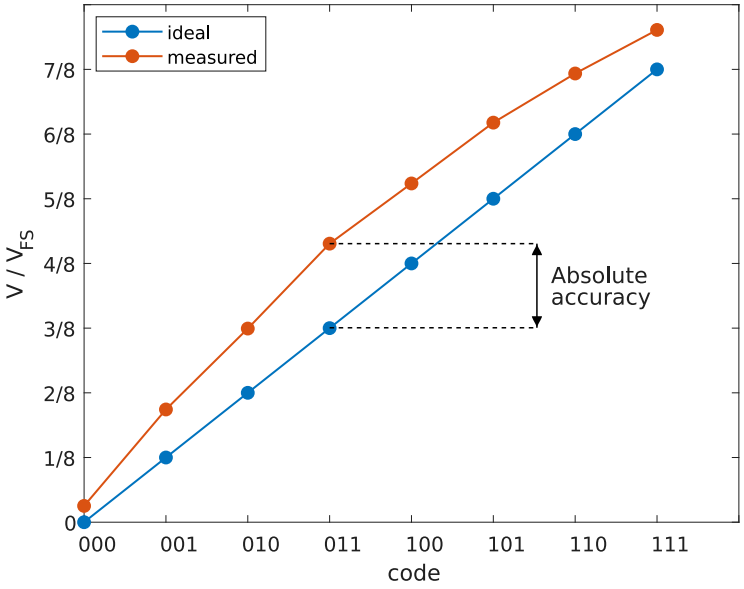


FIGURE 2.2 – Absolute accuracy representation.

2.1.1 Static Errors

Static converter errors of gain, offset, INL, DNL measure the converter static performance using as reference a best-fit regression line through all measured points.

Gain and Offset: In the input-output characteristic the offset error V_{OFF} is measured as the converter output voltage deviation where the ideal characteristic passes through zero [13], as shown in Figure 2.3 on the next page.
For a D/A-converter the gain error G is defined as follows [12] (see Figure 2.4 on the following page):

$$G = \frac{U[k] - V_{OFF} - \epsilon[k]}{Q \cdot k + U_0} \tag{2.2}$$

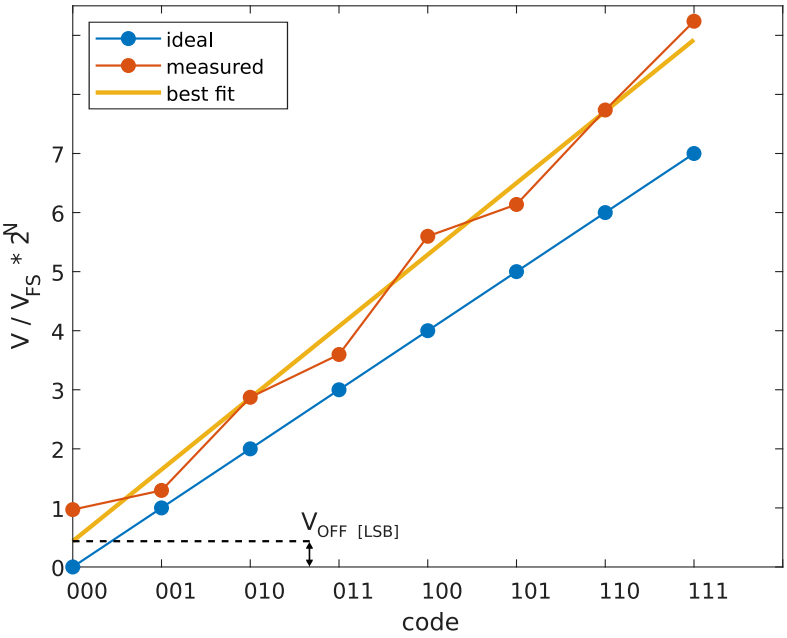


FIGURE 2.3 – Offset error representation.

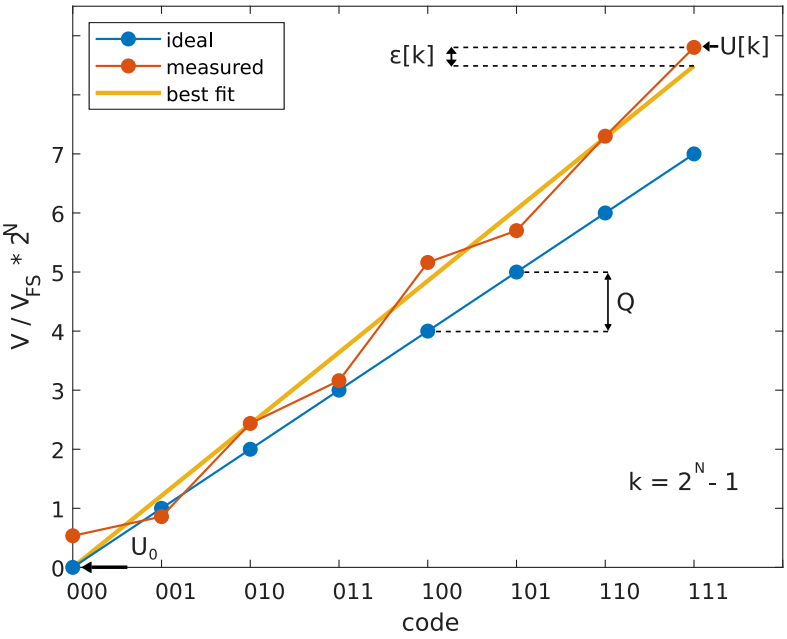


FIGURE 2.4 – Gain error representation (offset error has been compensated).

- k is the highest input code ($2^N - 1$).
- $U[k]$ is the k^{th} output value when the k^{th} input code is applied to the input.
- U_0 is the ideal value corresponding to $U[0]$.
- Q is the ideal width of a code, i.e. the full scale range divided by the number of codes ($V_{FS}/2^N$).
- $\epsilon[k]$ is the residual error corresponding to the k^{th} code: $\epsilon[k] = U[k] - G \cdot (Qk + U_0) - V_{OFF}$

Given the gain defined as in 2.2, gain error (GE) is the difference between the actual gain and the ideal gain.

$$GE = G - 1 \tag{2.3}$$

Offset and gain can be easily compensated by changing the reference voltage or current of the converter and then adding a right amount of offset voltage/current to the output.

Differential Nonlinearity (DNL) is defined for each code transition (bit $k-1$ to bit k), as the difference between the measured step height and the ideal LSB step (see Figure 2.5 on the next page) after gain and offset have been compensated [12].

$$DNL[k] = \frac{G \cdot (U[k+1]) - U[k]) - Q}{Q} \qquad k = 0...2^N - 2 \tag{2.4}$$

A sufficient condition for monotonicity is that $DNL(k)$ greater than -1 LSB for each k. A DNL higher than $+1$ LSB does not imply non-monotonicity, but it is still undesirable [6].

Integral Nonlinearity (INL) is the deviation of the k^{th} measured output from the best fit straight line, usually expressed in LSB or percentage of full scale range [12]:

$$INL[k] = \frac{\epsilon[k]}{2^N \cdot Q \cdot G} \qquad k = 0...2^N - 1 \tag{2.5}$$

A sufficient condition for monotonicity of a converter is that INL is bounded by $\pm 1/2$ LSB [13].

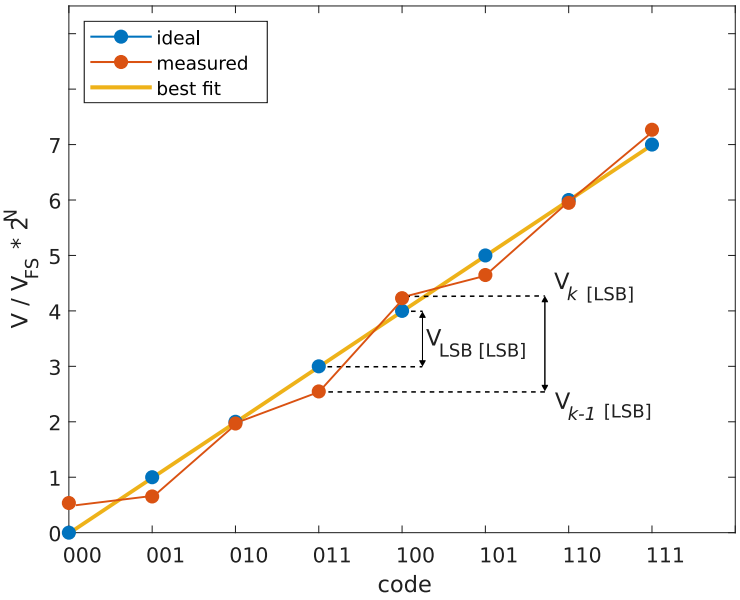


FIGURE 2.5 – Differential error representation (both gain and offset have been compensated, so that the best fit passes trough the ideal curve points).

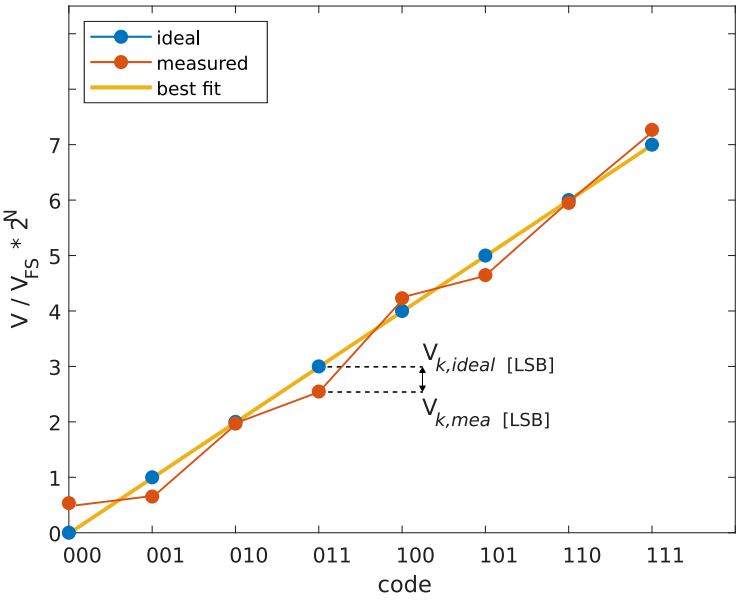


FIGURE 2.6 – Integral error representation (after gain and offset compensation).

2.1.2 Dynamic Errors

Spurious Free Dynamic Range (SFDR) Among DACs figures of merit is Spurious Free Dynamic Range (SFDR), obtained by synthesizing a tone and measuring on output spectrum the spreading between the fundamental amplitude and the highest non-fundamental component, expressed in decibels [12] . Often SFDR is expressed in relation to the carrier amplitude (dBc) or the converter full-scale (dBFS) [10]. Example measurement of SFDR is shown in Figure 2.7.

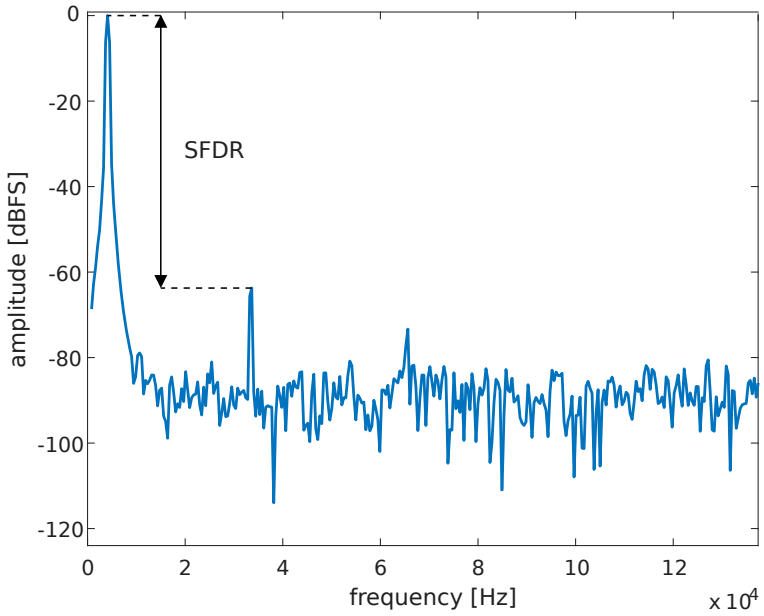


Figure 2.7 – Spurious Free Dynamic Range representation.

Signal to Noise and Distortion Ratio (SNDR) Signal to Noise and Distortion Ratio (SNDR) is the ratio of output sine-wave signal power to the noise and distortion [12]. Unless otherwise specified it is evaluated as the ratio between root mean square (rms) value of the single tone synthesized to the rms value of all other spectral components (i.e. noise and distortion) except dc-component [12][10]. The amplitude and frequency of the synthesized tone must be specified, as well as the measuring bandwidth, depending on application.

Signal to Noise Ratio (SNR) It is the ratio between the rms value of the synthesized tone to the rms value of the sine-fit residual error, excluding harmonics from the 2^{nd} to the 10^{th}.

Total Harmonic Distortion (THD) By synthesizing an integer number of cycles for a single tone, THD is defined as the ratio of the rms value of a defined set of harmonics of the fundamental (usually from the 2^{nd} to the 10^{th}) over the fundamental rms value. Often it is expressed in dBc.

2.2 Low power DAC applications

Reasons to investigate DAC solutions are multiple, both in the framework of mostly-digital and digital-assisted integrated circuits and to accomplish low power/area designs:

As part of ADCs The most performing ADCs in terms of energy per conversion, nowadays, are those based on SAR architecture. This is above all due to capacitor DACs, their ideally null quiescent current, no need for hold circuit and the numerous switching strategies which have been developed to reduce conversion energy.
Low resolution DACs are also employed in $\Sigma\Delta$ converters where high sampling frequency is traded with resolution, avoiding stringent matching requirements.

Calibration DAC are also quite useful in analog circuits as calibration elements, where low area and almost null quiescent current is required. Example applications are auxiliary DACs to calibrate current sources in current steering DACs, and to measure and compensate capacitor mismatch in switched capacitors DACs [9].

Dynamic voltage-frequency scaling In the framework of low power microprocessor systems one common technique is duty-cycling the circuit operation by powering off the device when no action is needed. If duty-cycled operation is still not compliant with energy requirements, a solution is to dynamically vary supply voltage and clock frequency to provide high performance only when needed [14]. To this purpose, a variable reference voltage for the regulator is needed, and the task is well accomplished by a DAC supplied by a fixed reference voltage. The reference voltage and DAC function can be even performed by the same circuit as in [15].

Chapter 3

Relaxation DAC: theoretical analysis

In this chapter the working principle of the Relaxation DAC is presented, fundamental equations are derived, and the sources of static errors are analysed, from clock frequency systematic errors and jitter to charge injection, parasitic capacitances and thermal noise. Finally a theoretical estimation of the Relaxation DAC energy per conversion is provided.

Technology scaling and voltage-reduction trend in System on a Chip manufacturing is posing tough limits to analog circuit design, whose performance are deteriorated by electronic noise, matching requirements and lowered intrinsic transistor gain, unlike their digital counterpart [16]. Matching, area and power requirements can be loosen by moving to the digital world most functionalities, designing Mostly Digital (MD) circuits and allowing simpler analog circuits having lower performance, to be compensated with digital processing (Digitally Assissted (DA) circuits). Towards these goals the Relaxation Digital-to-Analog Converter topology has been proposed, and its operation is described.

3.1 Operating Principle

A Relaxation digital-to-analog converter exploits the impulse response of an RC network ideally connected to a three state buffer, driven by a stream of N rectangular pulses of the same duration T and amplitude

equal to power supply voltage (V_{DD}) or 0V (gnd).
A similar behavior is easily accomplished by using a N-bits shift register

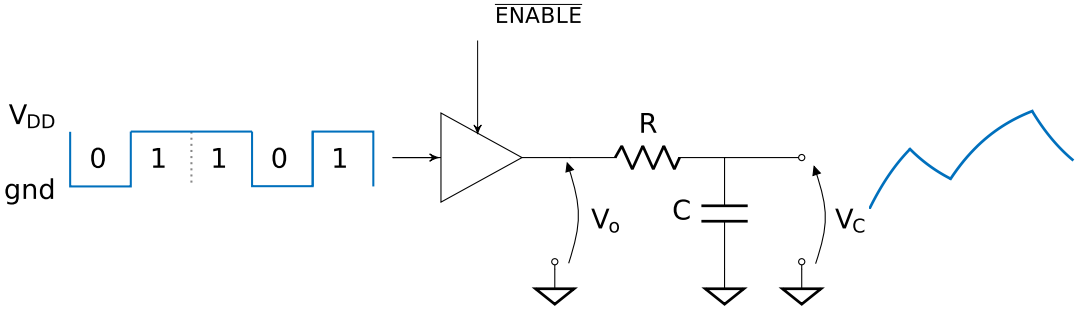


FIGURE 3.1 – Operating principle schematic.

which drives a three-state buffer, so that the bits of the binary word decide if the driving voltage must be high or low for the clock cycle T.
After all bits of the word to be converted have been shifted, the enabling signal goes high, thus disabling the three-state output and preventing the capacitor to discharge, so the voltage across it remains constant until a new conversion begins.
A scheme for the shift-register implementation is shown in Figure 3.3 page 28.
To analyze circuit behavior we can refer to the model depicted in Figure 3.2 on the facing page.
Time evolution of voltage across the capacitor during the i^{th} time interval, corresponding to bit b_i, is completely described by the initial condition across the capacitor in that period $v_C(iT)$, the steady state voltage $v_{C,i}(\infty)$ and time constant $\tau = RC$, according to the following:

$$v_{C,i}(t) = v_{C,i}(\infty)\left[1 - e^{-\frac{t-iT}{\tau}}\right] + v_C(iT)e^{-\frac{t-iT}{\tau}} \qquad \begin{matrix} i = 0...(N-1) \\ t \in [iT, (i+1)T] \end{matrix} \qquad (3.1)$$

We can express the steady state value as $v_{C,i}(\infty) = V_{DD}b_i$, according to the bitstream hypothesis made above. Moreover, assuming $v_C(0) = 0$ as reset

condition for the capacitor, we can iterate (3.2) as follows:

$$\begin{aligned}
v_{C,0}(t) &= V_{DD} b_0 \left[1 - e^{-\frac{t}{\tau}}\right] \\
v_C(T) &= v_{C,0}(T) = V_{DD} b_0 \left[1 - e^{-\frac{T}{\tau}}\right] \\
v_{C,1}(t) &= V_{DD} b_1 \left[1 - e^{-\frac{t-T}{\tau}}\right] + v_C(T) e^{-\frac{t-T}{\tau}} \\
v_C(2T) &= v_{C,1}(2T) = V_{DD} b_1 \left[1 - e^{-\frac{T}{\tau}}\right] + V_{DD} b_0 \left[1 - e^{-\frac{T}{\tau}}\right] e^{-\frac{T}{\tau}}
\end{aligned} \tag{3.2}$$

And, after generic N clock periods:

$$v_C(NT) = V_{DD}\left[1 - e^{-\frac{T}{\tau}}\right] \cdot \sum_{0}^{N-1} b_i e^{-\frac{(N-i-1)T}{\tau}} \tag{3.3}$$

We now choose time constant τ such that the following is satisfied:

$$e^{-\frac{T}{\tau}} = \frac{1}{2} \qquad \Longrightarrow \qquad \tau = \frac{T}{\log(2)} \tag{3.4}$$

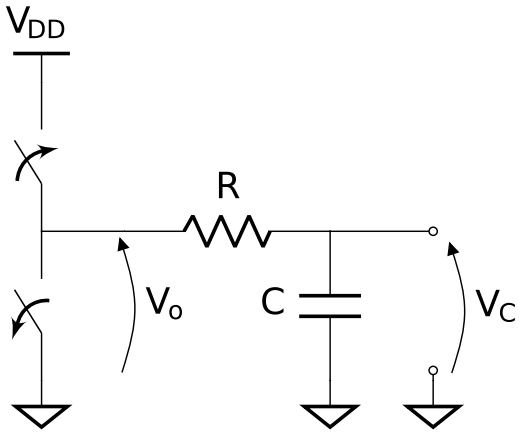


Figure 3.2 – Simplified circuit model of the DAC.

By substituting (3.4) in (3.3) it turns out that:

$$v_C(NT) = V_{DD} \cdot \sum_{0}^{N-1} \frac{b_i}{2^{N-i}} \tag{3.5}$$

(3.5) coincides with the one presented in (2.1), which describes the behavior of a generic N-bit DAC converter.

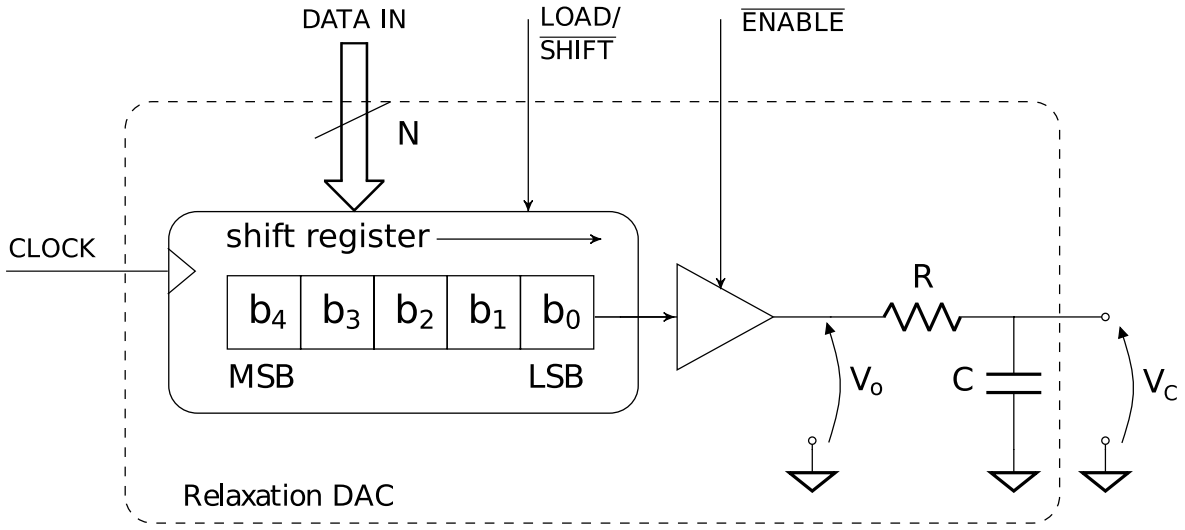


Figure 3.3 – Block diagram of a N=5bit Relaxation DAC.

On figure 3.5 waveforms related to 4 possible words of a 3-bit converter are shown.
Equation (3.2) has been implemented in Matlab software to plot all the possible waveforms generated by an 8-bit relaxation converter. The result is shown in Figure 3.4 on the next page.

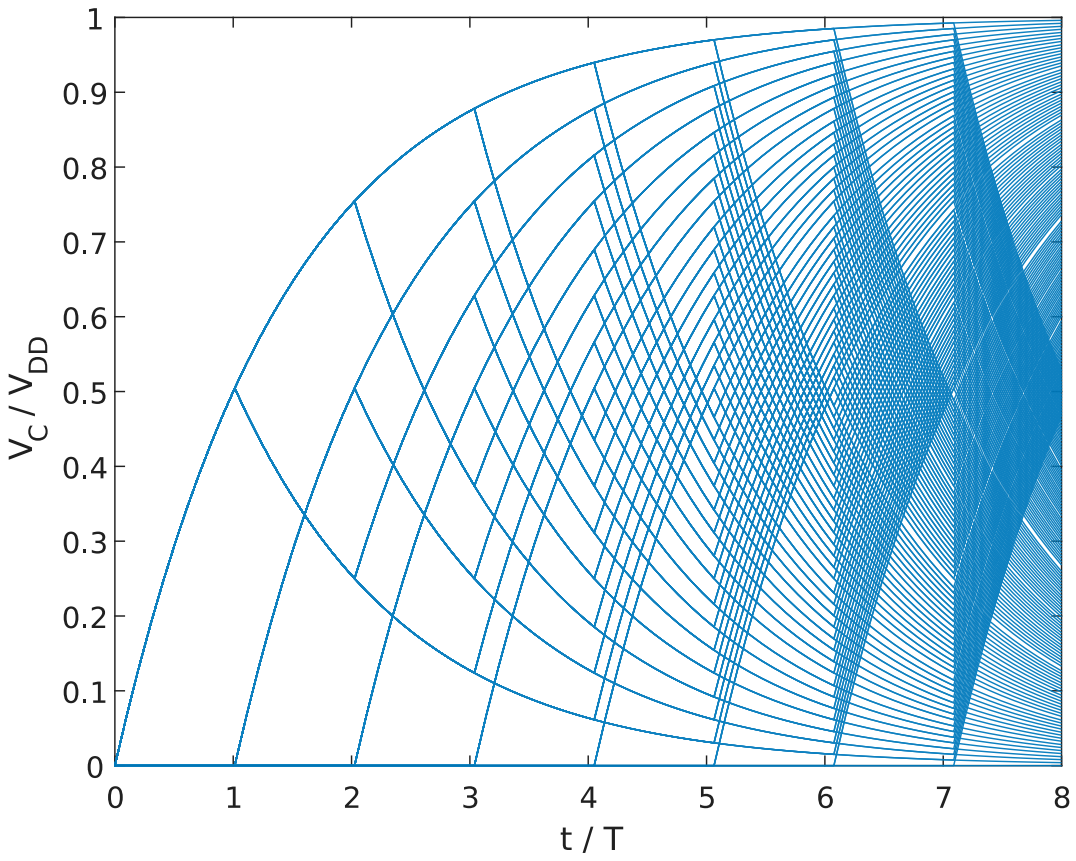
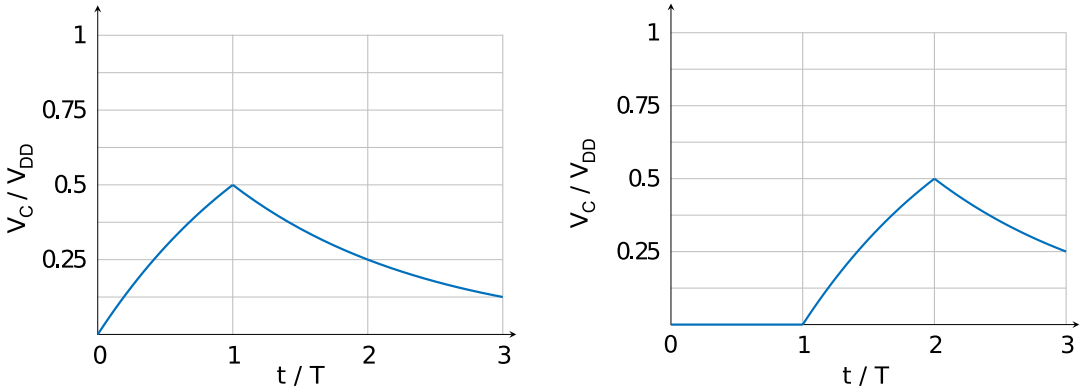
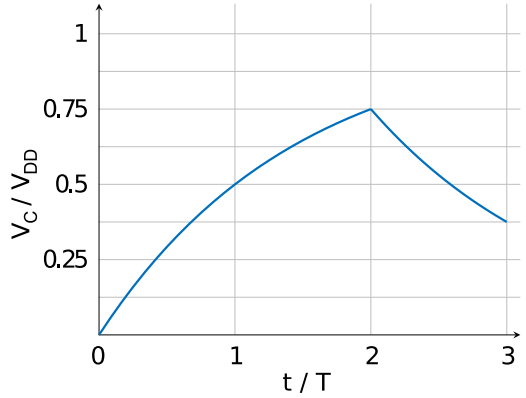


FIGURE 3.4 – Superimposed conversion waveforms for a N =8-bit converter.

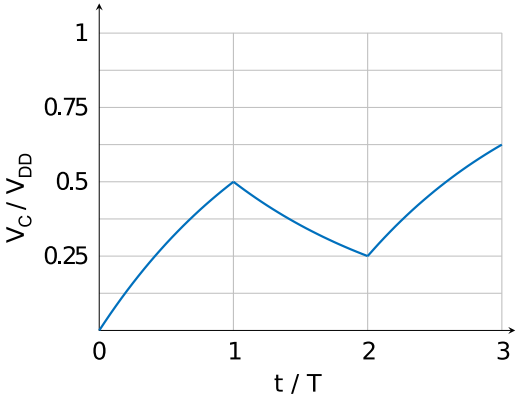


(a) $b_2b_1b_0 = 001$

(b) $b_2b_1b_0 = 010$



(c) $b_2b_1b_0 = 011$



(d) $b_2b_1b_0 = 101$

FIGURE 3.5 – Some conversion waveforms for a N =3-bit converter.

3.1.1 Ideal Converter Behaviour

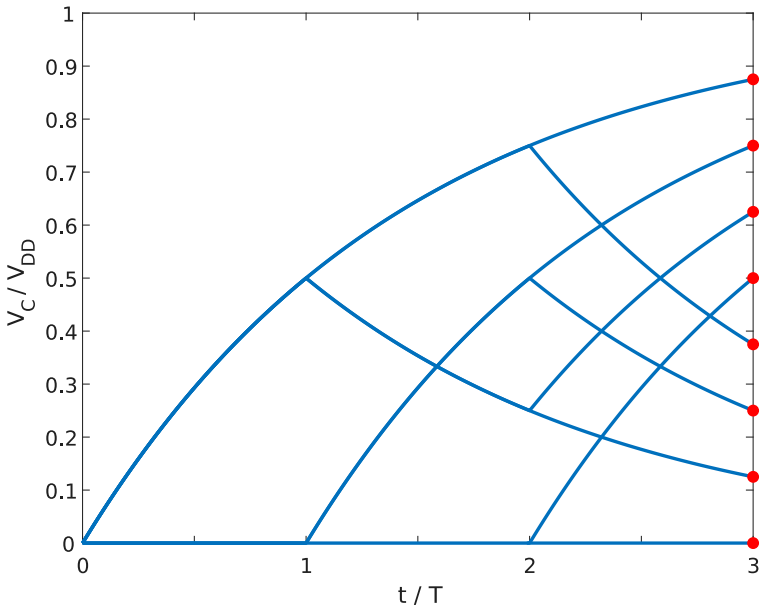
If no error is introduced the converter presents:

- No **offset and gain errors**, since the capacitor keeps being discharged to ground voltage when the null word is applied.
- No **INL** or **DNL**, since we saw that relation 3.5 coincides with the ideal DAC trans-characteristic.
- The property of **monotonicity** is guaranteed, and **accuracy** coincides with **resolution** for the above facts.

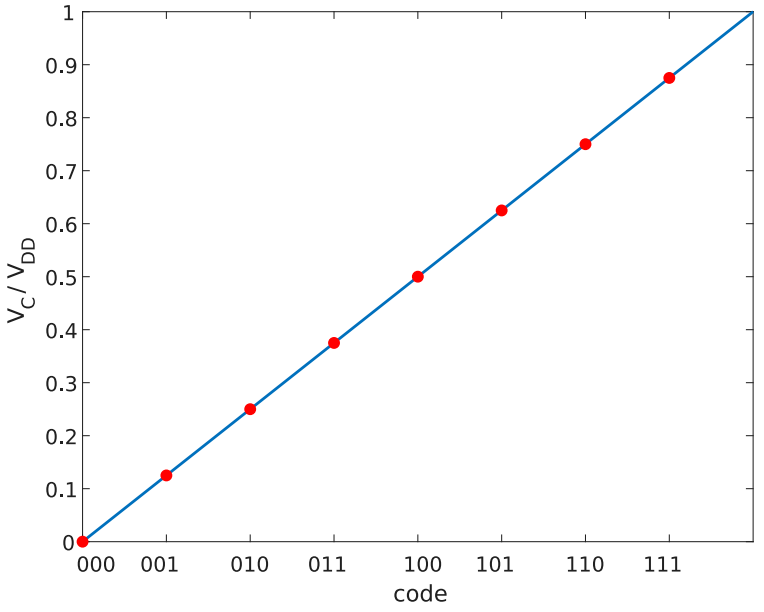
Waveforms and transcharacteristic generated by a three bit ideal converter are depicted in Figure 3.6(a) on the following page and Figure 3.6(b) on the next page respectively.
In a real converter, tough, different sources of error are possible:

- **Clock errors**, including clock frequency systematic error and clock phase noise (jitter).
- **Passive components** can produce nonlinear errors due to non precise time constant, resistor and capacitor nonlinearity, parasitic capacitances towards substrate.
- **Driving circuit** transistors can induce errors due to non-null and non linear triode resistance, injected charge during commutation and current leakage.

Effects and relevance of these nonidealities to the converter accuracy are analysed in the following sections.



(a) Superimposed converter waveforms when no error introduced for a $N = 3$-bit converter. End-conversion values are highlighted in red.



(b) Converter end values when no error is introduced (in red) superimposed to the ideal transcharacteristic (in blue).

FIGURE 3.6 – Waveforms and transcharacteristic for an ideal 3-bit converter.

3.2 Clock Error Effect on Accuracy

A first possible error considered for the converter is a systematic error in the clock period.
We are supposing to have a clock period $T' = T + \Delta T$ different from the ideal period T by an error quantity ΔT. If we substitute T' to T in the previously derived (3.3) expression, it follows that:

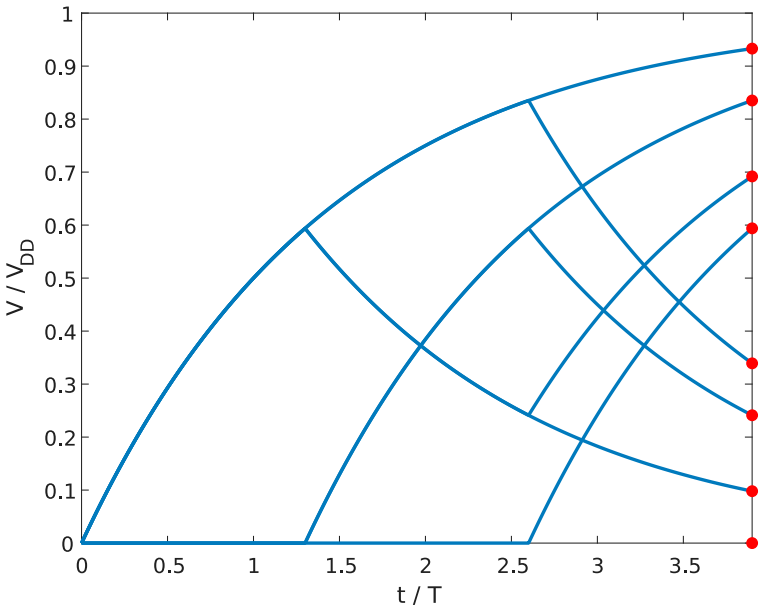
$$v_C(NT') = V_{DD}\Big[1 - e^{-\frac{T+\Delta T}{\tau}}\Big] \cdot \sum_{0}^{N-1} b_i e^{-\frac{(N-i-1)(T+\Delta T)}{\tau}}$$

$$v_C(NT') = V_{DD}\Big[1 - e^{-\frac{T}{\tau}} e^{-\frac{\Delta T}{\tau}}\Big] \cdot \sum_{0}^{N-1} b_i \Big(e^{-\frac{T}{\tau}} e^{-\frac{\Delta T}{\tau}}\Big)^{(N-i-1)} \tag{3.6}$$

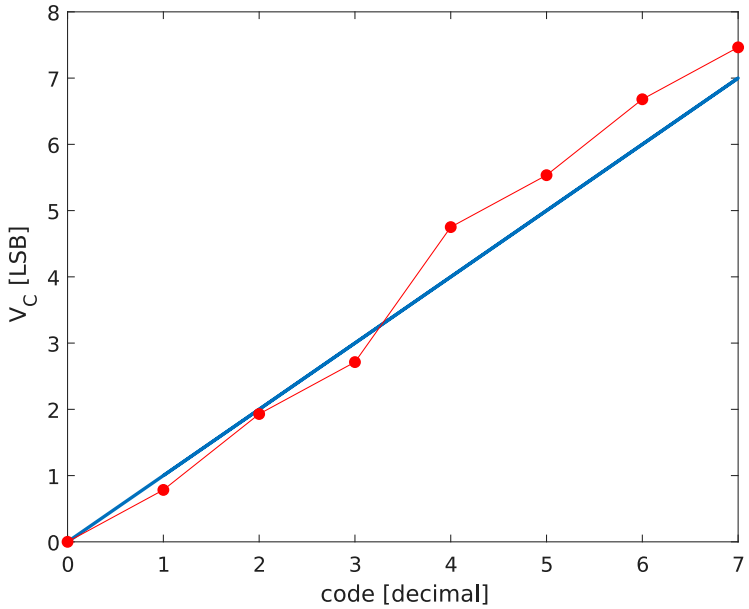
Recalling the imposed (3.4) and substituting it in (3.6):

$$v_C(NT') = V_{DD}\left[1 - \frac{e^{-\frac{\Delta T}{\tau}}}{2}\right] \cdot \sum_{0}^{N-1} b_i \left(\frac{e^{-\frac{\Delta T}{\tau}}}{2}\right)^{(N-i-1)} \tag{3.7}$$

Waveforms and transcharacteristic of a converter having a constant positive relative error of $\frac{\Delta T}{T} = 30\%$ on shift period have been plotted by implementing (3.7) in Matlab code, and are visible in Figure 3.7(a) on the following page and Figure 3.7(b) on the next page respectively.
If a negative error $\frac{\Delta T}{T} = -30\%$ is introduced instead, time evolution and converted values are the ones plotted in Figure 3.8(a) page 35 and Figure 3.8(b) page 35.
The integral error (INE) (difference between the converter transcharacteristic and ideal one, without offset and gain compensation) shows a fractal behaviour with increasing number of bits. It can be noticed that INE value converges to INL with increasing number of bits (compare Figure 3.9(a) to Figure 3.9(c)).
Best-fit INL is shown in Figure 3.9(f) page 36 and Figure 3.9(h) page 36.

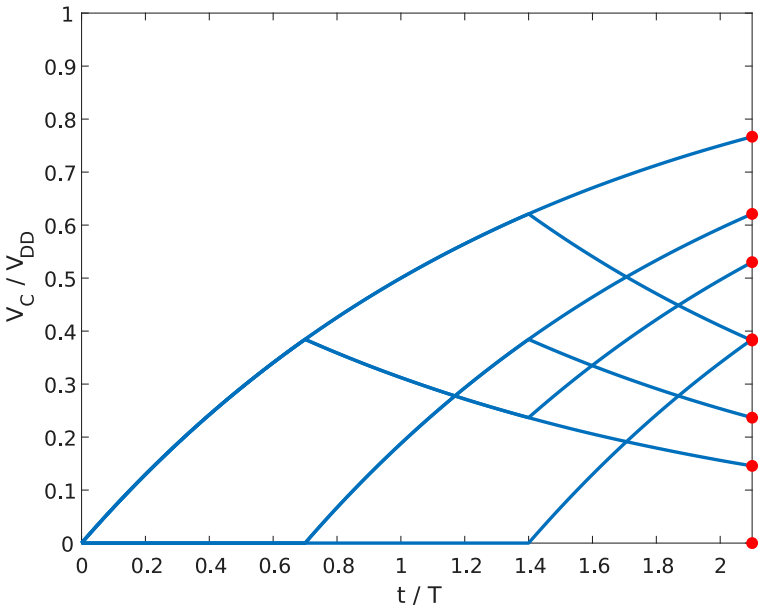


(a) Converter waveforms (in blue) when positive 30% period error is introduced for a $N = 3$-bit converter. End-conversion values are highlighted in red.

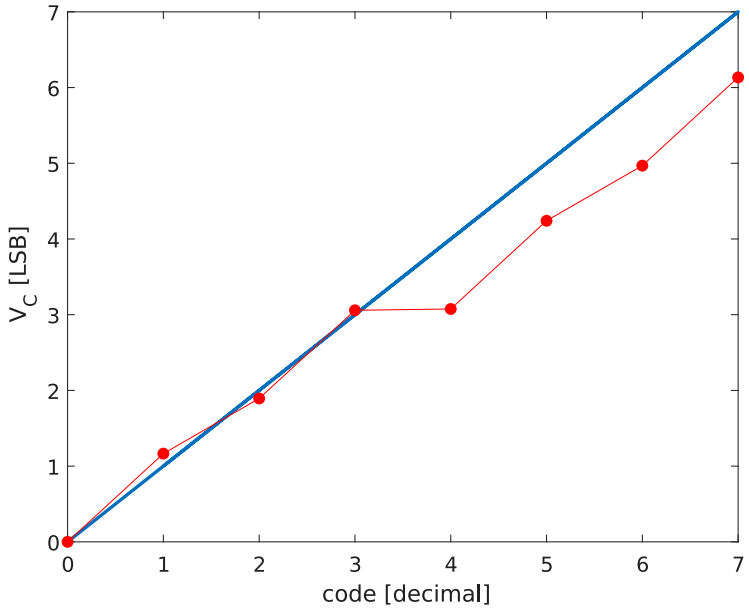


(b) End-conversion values (in red) superimposed to the ideal transcharacteristic (in blue) when positive 30% period error is introduced.

FIGURE 3.7 – Waveforms and transcharacteristic of a 3-bit converter when a positive 30% error on shifting period is introduced.

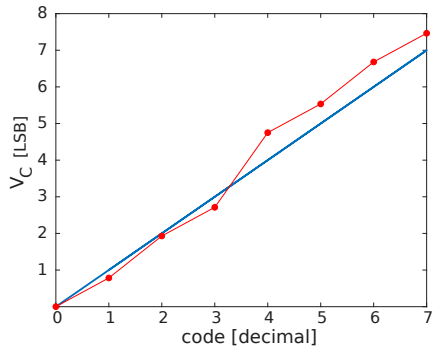


(a) Converter waveforms (in blue) when negative 30% period error is introduced for a $N = 3$-bit converter. End-conversion values are highlighted in red.

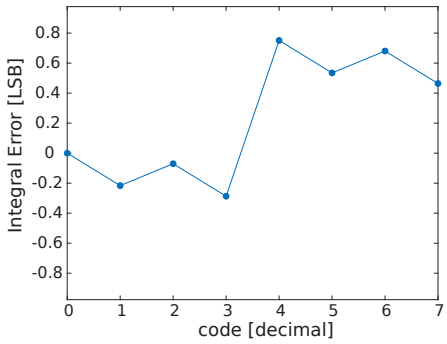


(b) End-conversion values (in red) superimposed to the ideal transcharacteristic (in blue) when negative 30% period error is introduced.

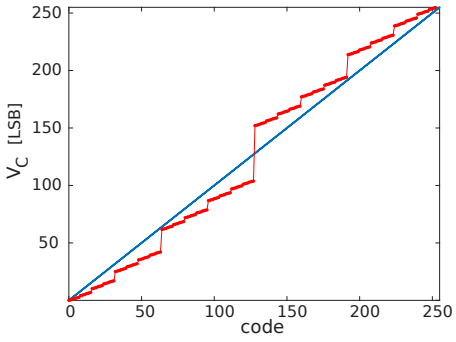
FIGURE 3.8 – Waveforms and transcharacteristic of a 3-bit converter when a negative 30% error on shifting period is introduced.



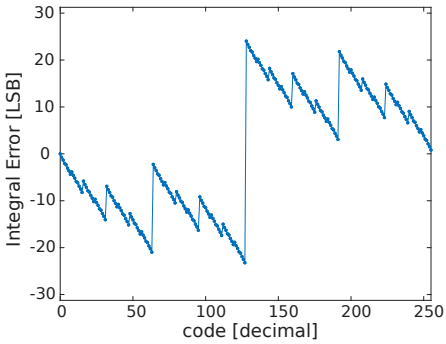
(a) Transcharacteristic for N = 3-bit and $\Delta T/T = 0.3$.



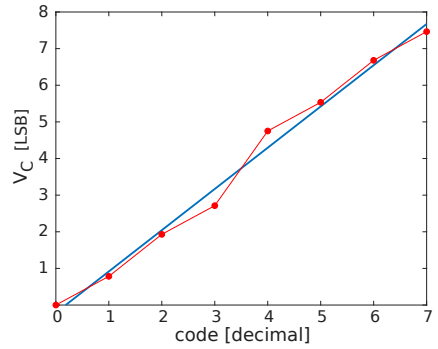
(b) Integral error relative to figure (a) .



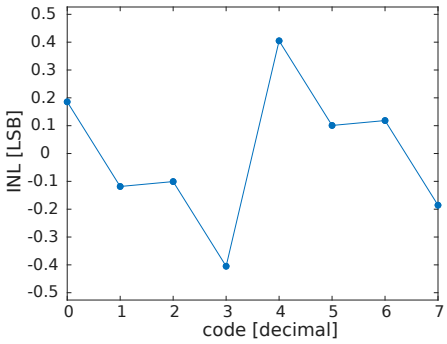
(c) Transcharacteristic for N = 8-bit and $\Delta T/T = 0.3$.



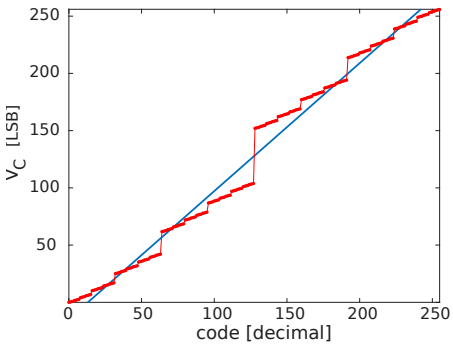
(d) Integral error relative to figure (c) .



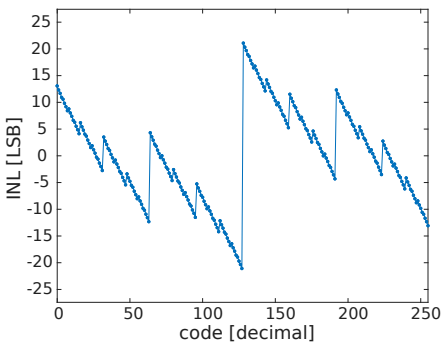
(e) Transcharacteristic for N = 8-bit and $\Delta T/T = 0.3$, best fit reference curve.



(f) INL relative to figure (e) .



(g) Transcharacteristic for N = 8-bit and $\Delta T/T = 0.3$, best fit reference curve.



(h) INL relative to figure (c) .

FIGURE 3.9 – Transcharacteristics, INE and INL on 3 and 8 bits for 30% delay.

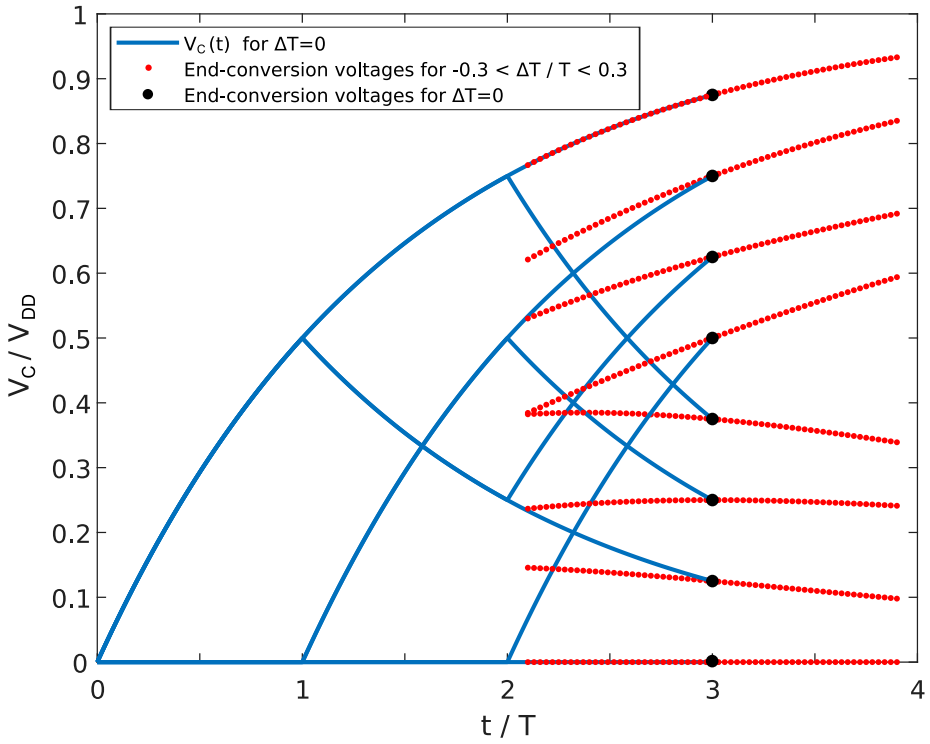


FIGURE 3.10 – End-conversion values for $\Delta T/T$ span in range -0.3 to 0.3.

Few things can be noticed:

- **Swing** is compressed for negative clock errors and expanded for positive ones (see Figure 3.10).
- **Offset** error keeps being zero for end-point reference curve (see Figure 3.9(a) on the preceding page and Figure 3.9(c) on the facing page). For the best-fit case offset error is proportional to ΔT (see Figure 3.11 on the next page).
- **Gain** error (in the end-points reference case) saturates to $1/2^N$, for positive delays (due to supply voltage limit); it increases for negative ΔT errors since swing is compressed.
  In the best-fit curve case, gain error is proportional to ΔT for both positive and negative delay errors (see Figure 3.11 on the following page).
- Maximum **INL** and **DNL** appears always in the middle of swing where exponential derivative is maximum (Figure 3.10) for both positive and negative time error (Figure 3.15(a) page 43 and Figure 3.15(b) page 43).
- **Monotonicity** is guaranteed for every positive ΔT error, provided

that time error keeps being constant trough all the swing. For smaller periods, monotonicity is guaranteed only if error is limited in absolute value by a constant depending on the number of bits, and it keeps constant across the whole swing.

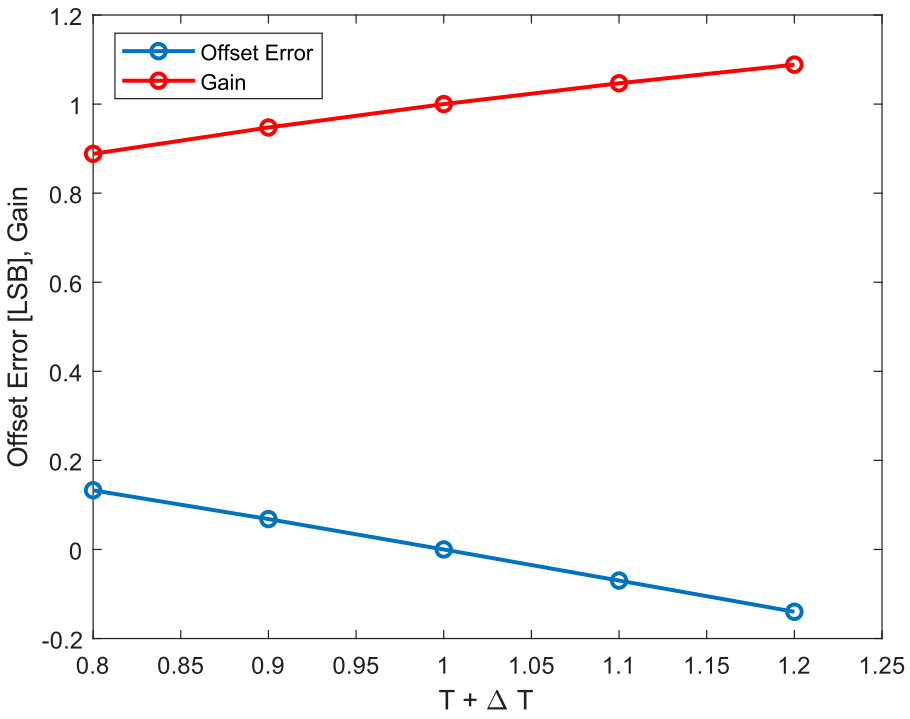


FIGURE 3.11 – Best-fit offset error and gain with $-0.3 < \Delta T/T < 0.3$.

3.2.1 Clock Period Calibration

By referring to 3.13, the bigger step in the integral error can be observed at half swing, and the step is positive or negative depending on the sign of the ΔT error on clock period T. A compensation strategy to minimize integral error would be to compare converted voltages corresponding to the two middle-swing codes (i.e. $V_{C,'01\ldots1'}$ and $V_{C,'10\ldots0'}$), and to apply a feedback to reduce/increase clock period based on the error voltage $\delta = V_{C,'10\ldots0'} - V_{C,'10\ldots0'}$. The calibration process is shown in Figure 3.12 on the facing page.

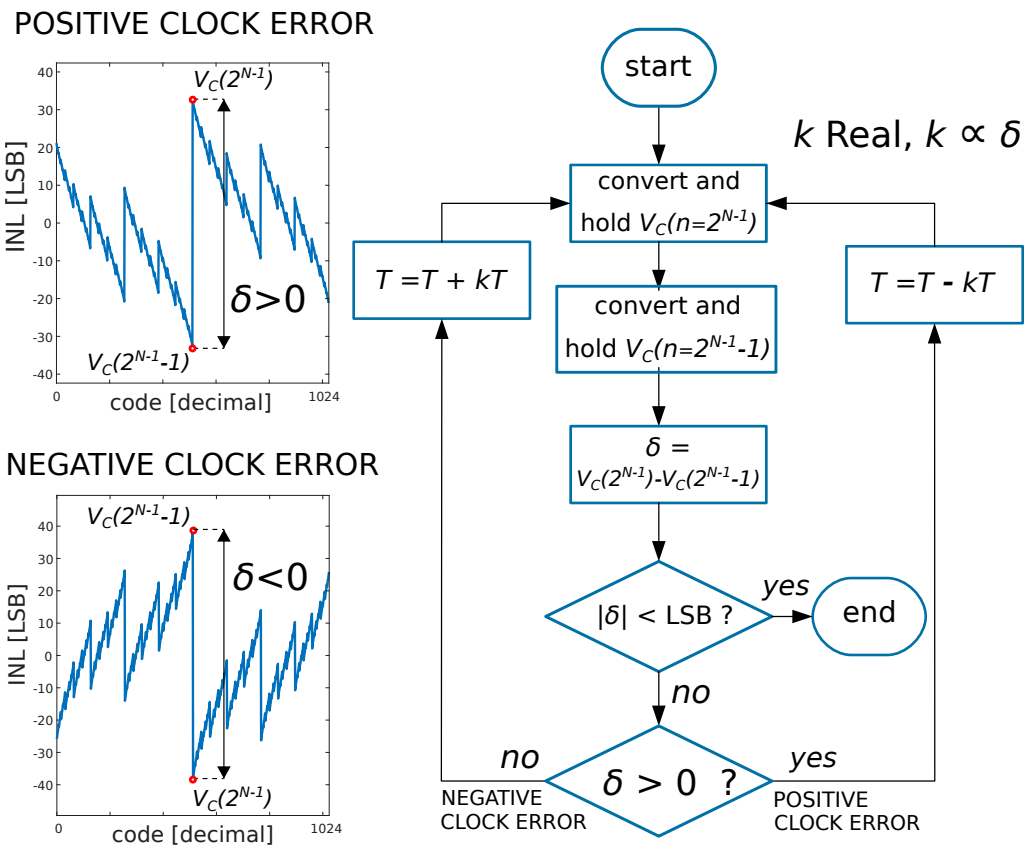
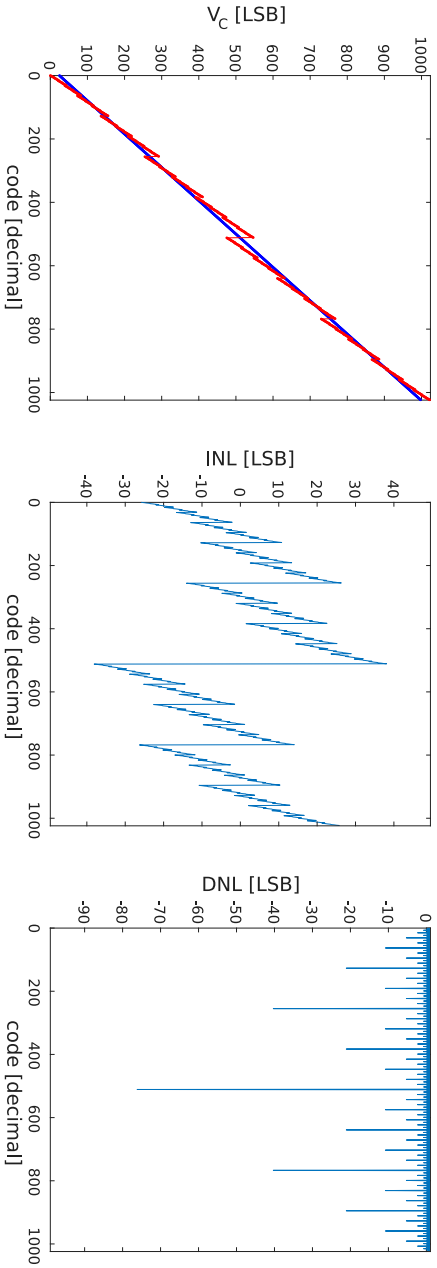


Figure 3.12 – Clock calibration strategy.



(a) Transcharacteristic, INL, DNL, for 10-bit converter, $\Delta T/T = +0.1$.

(b) Transcharacteristic, INL, DNL, for 10-bit converter, $\Delta T/T = -0.1$.

FIGURE 3.13 – Transcharacteristics, INL, DNL, for 10-bit converter.

	$(T'/T)_{min}$		$(T'/T)_{min}$
$N = 3$	0.69424	$N = 4$	0.87914
$N = 5$	0.94677	$N = 6$	0.97522
$N = 7$	0.98810	$N = 8$	0.99419
$N = 9$	0.99713	$N = 10$	0.99857
$N = 11$	0.999291	$N = 12$	0.999646

TABLE 3.1 – Minimum normalised period to fulfil monotonicity constraint for a given number of bits.

3.2.2 Negative Error on Period and Monotonicity

Given a negative constant time error ΔT, the minimum normalised period

$$\left(\frac{T'}{T}\right)_{min} = \left(\frac{T + \Delta T}{T}\right)_{min} \qquad \Delta T < 0 \tag{3.8}$$

which guarantees monotonicity, has been found for different number of bits, through numerical analysis, using models previously found. Obtained values are reported in Table 3.1.
Values in table show, as expected, an exponential behaviour with increasing number of bits, due to exponentially increasing resolution.
Values in table have been fit as depicted in Figure 3.14 on the following page in order to retrieve an analytical expression providing the minimum period which guarantees monotonicity when error is not varying through the dynamic. Exponential curve is the following:

$$\left(\frac{T'}{T}\right)_{min} = 1 + a \cdot e^{bN} + c \cdot e^{dN} \tag{3.9}$$

$$a = -11.089 \quad b = -1.553 \quad c = -1.6893 \quad d = -0.7102$$

It is worth noting that up to a 19-bit converter (not shown in table) the relative error on period is in the order of 10^{-6}, thus theoretically compatible with quartz oscillators accuracy.

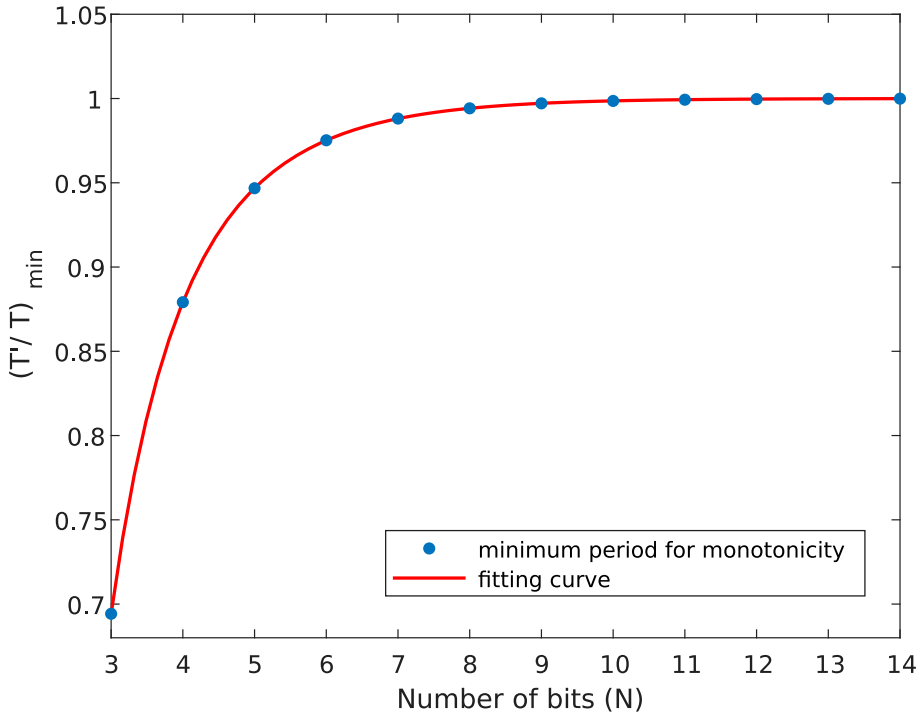


FIGURE 3.14 – Minimum normalised period to obtain monotonic transcharacteristic for negative constant error on hsift period.

Summarizing: If error on period keeps constant through all the swing, and ideally no phase noise is present, it appears that positive error is to be preferred to negative one since

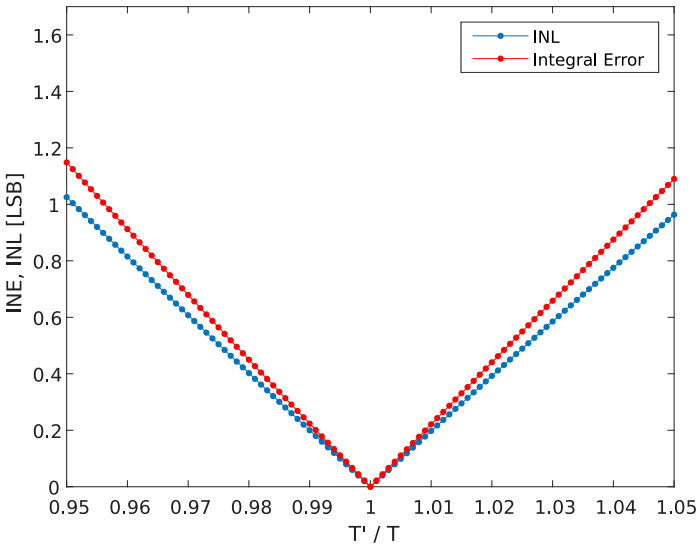
- Gain error is upper bounded to $1/2^N$ for positive delay, while keeps increasing for negative ones.
- Monotonicity is guaranteed for any positive error which keeps constant through all the swing.

Last point must not be misinterpreted as a way to obtain monotonic curve in any case. We'll see this is not valid in presence of clock jitter.

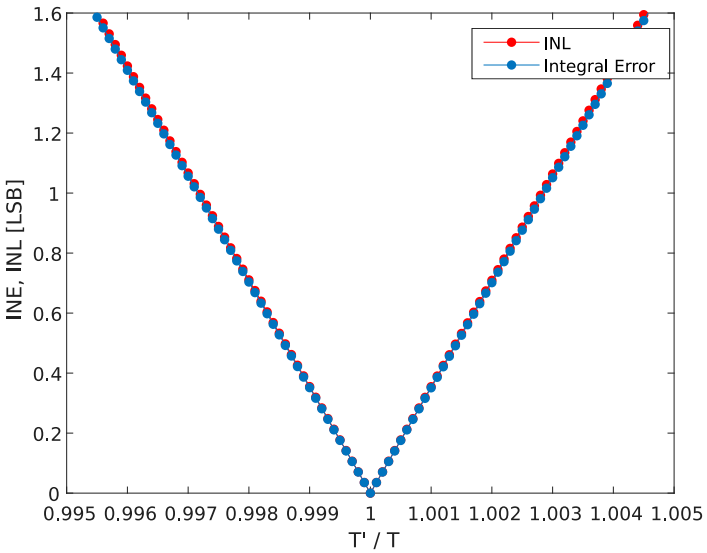
3.2.3 INL, INE and Accuracy

By plotting the maximum integral error and INL with varying normalised period, it can be observed that they vary linearly. Moreover, integral

error is approaching INL when the number of bits increases, as shown in Figure 3.15(a) and Figure 3.15(b).



(a) INE, INL versus normalised period variation, 6-bit converter.



(b) INE, INL versus normalised period variation, 10-bit converter.

FIGURE 3.15 – INE, INL, vs. normalised period.

INE and INL accuracies can be expressed as equivalent number of bits , according to the following:

INE Accuracy

$$\frac{V_{DD}}{V_{INE}} = 2^{N_{INE}} \tag{3.10}$$

INL Accuracy

$$\frac{V_{DD}}{V_{INL}} = 2^{N_{INL}} \tag{3.11}$$

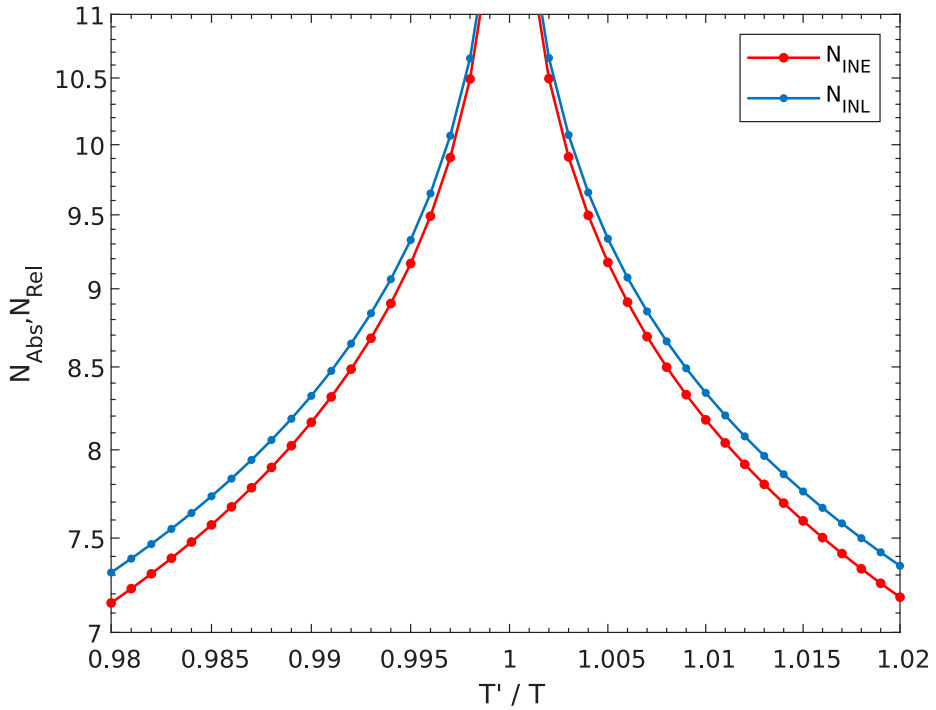


FIGURE 3.16 – INE and INL accuracy as effective bits versus normalised shift period, 6-bit converter.

With a Matlab implementation of equation 3.7 INE and INL accuracy have been evaluated when a constant time error affects shift period through all the conversion swing. Plots of INE and INL accuracy as a function of normalised period, related to a 6-bit and 10-bit converter,

are reported in Figure 3.16 on the preceding page and Figure 3.17 respectively. It can be noticed an hyperbolic behaviour of both curves, and their convergence to the same value due to the reduced gain error when number of bits increases (visible by comparing Figure 3.9(b) page 36 and Figure 3.9(d) page 36).

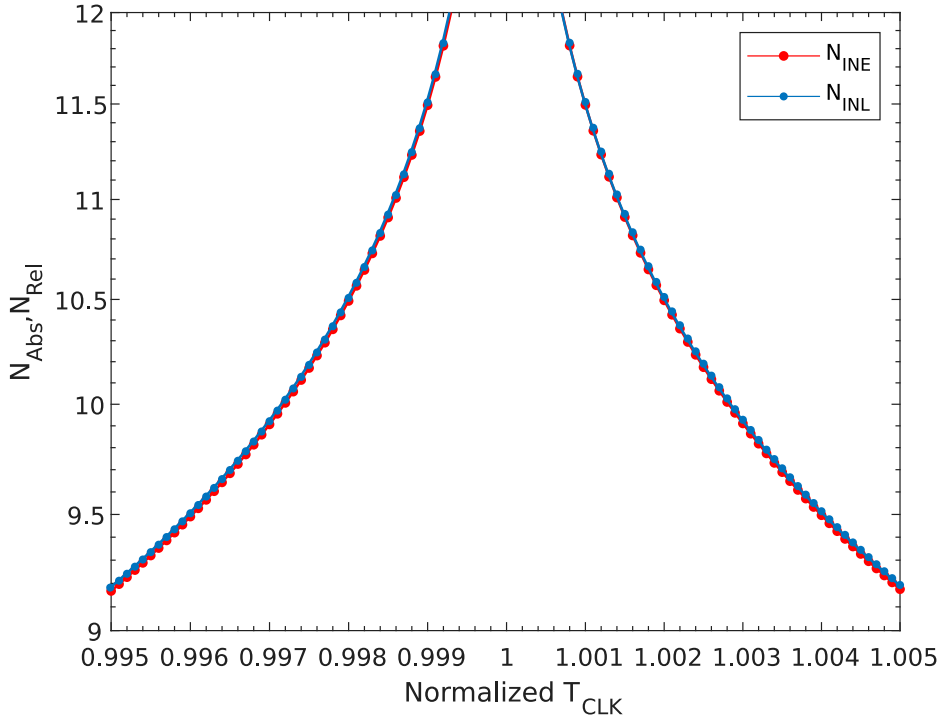


FIGURE 3.17 – Absolute and relative accuracy as effective bits versus normalised shift period, 10-bit converter.

3.3 Clock Jitter Effect on Static Errors

Constant error hypothesis on relaxation DAC clock period has been expanded by taking into account the possible effects of a clock phase noise/jitter (thus a different time duration for each period).
In a similar way as introduced in (3.2), the value across the capacitor $v_{C,i+1}$ at the end of i^{th} period can be expressed as a function of initial condition $v_{C,i}$ in that period, the steady state voltage $v_{C\infty,i}$ and the time

constant τ as follows:

$$v_{C,i+1} = v_{C\infty,i}\left(1 - e^{-\frac{T_i}{\tau}}\right) + v_{C,i} \cdot e^{-\frac{T_i}{\tau}} \qquad i = 0...(N-1) \tag{3.12}$$

where:

$$\begin{aligned} T_i &= T + \Delta T_i \\ T &= \tau \cdot \log(2) \end{aligned} \tag{3.13}$$

and ΔT_i is the time error on i^{th} period, different in general, for each cycle. Given (3.13), (3.12) becomes

$$\begin{aligned} v_{C,i+1} &= v_{C\infty,i}\left(1 - e^{-\frac{T+\Delta T_i}{\tau}}\right) + v_{C,i} \cdot e^{-\frac{T+\Delta T_i}{\tau}} \\ v_{C,i+1} &= v_{C\infty,i}\left(1 - \frac{e^{-\frac{\Delta T_i}{\tau}}}{2}\right) + v_{C,i} \cdot \frac{e^{-\frac{\Delta T_i}{\tau}}}{2} \qquad i = 0...(N-1) \end{aligned} \tag{3.14}$$

Iterating (3.14) starting from $i = 0$ we get:

$$v_{C,1} = V_{DD} b_0 \left(1 - \frac{e^{-\frac{\Delta T_0}{\tau}}}{2}\right) + 0$$

$$v_{C,2} = V_{DD} b_1 \left(1 - \frac{e^{-\frac{\Delta T_1}{\tau}}}{2}\right) + V_{DD} b_0 \left(1 - \frac{e^{-\frac{\Delta T_0}{\tau}}}{2}\right) \frac{e^{-\frac{\Delta T_1}{\tau}}}{2}$$

$$v_{C,3} = V_{DD} b_2 \left(1 - \frac{e^{-\frac{\Delta T_2}{\tau}}}{2}\right) + \left[V_{DD} b_1 \left(1 - \frac{e^{-\frac{\Delta T_1}{\tau}}}{2}\right) + V_{DD} b_0 \left(1 - \frac{e^{-\frac{\Delta T_0}{\tau}}}{2}\right) \frac{e^{-\frac{\Delta T_1}{\tau}}}{2}\right] \frac{e^{-\frac{\Delta T_2}{\tau}}}{2}$$

...

and so on, thus the expression for a general N number of bits can be written:

$$v_{C,N} = \left\{ b_{N-1}\left[1 - \frac{e^{-\frac{\Delta T_{N-1}}{\tau}}}{2}\right] + \sum_{j=0}^{N-2} b_j \left[1 - \frac{e^{-\frac{\Delta T_j}{\tau}}}{2}\right] \prod_{k=j+1}^{N-1} \frac{e^{-\frac{\Delta T_k}{\tau}}}{2} \right\} \tag{3.15}$$

Equation (3.15) has been used to implement a Matlab script which simulates 10^4 conversions for each code of the swing of a 3-bit converter. The conversion is affected by a jitter error on clock period uniformly distributed around $\pm 10\%$ the nominal period. Conversion waveforms along with converted voltages (represented as clouds of red dots) are shown in Figure 3.18.

For a better insight about converted voltages deviation from nominal

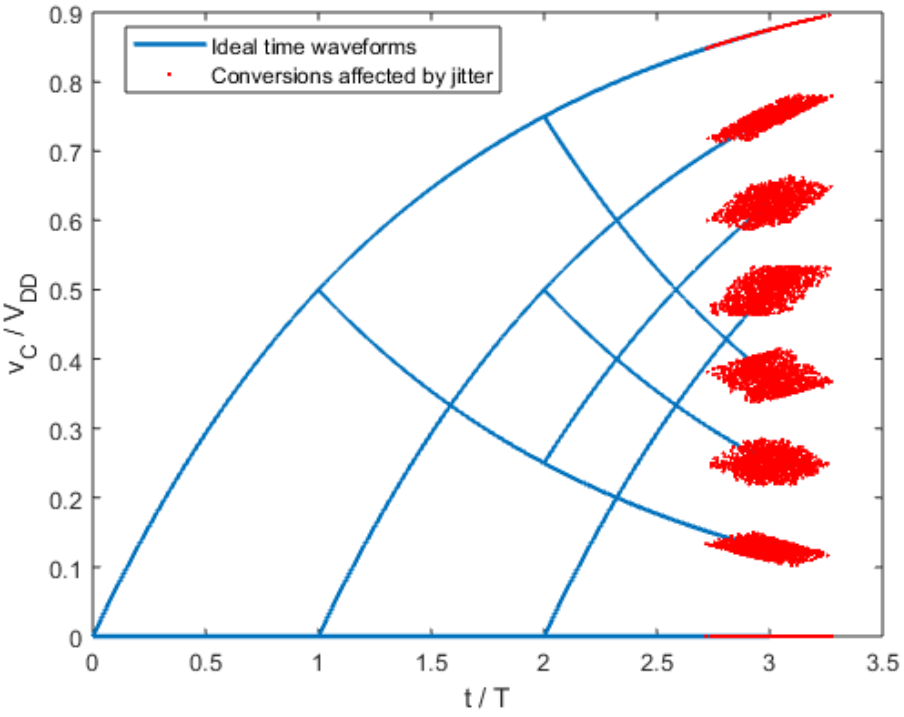


FIGURE 3.18 – Ideal waveforms and 10^4 conversion of the whole dynamic with clock jitter uniformly distributed in $\pm 10\%$ of clock period.

values, dots distributions have been fitted along the vertical dimension by Gaussian distributions, as shown in Figure 3.20(a) page 50;

From figure it can be observed that the maximum error appears in the middle of the dynamic, where the distribution keeps being almost-uniform, while other codes distributions are narrowed, approaching triangular ones.

To evaluate which is the maximum phase noise one can have to accomplish monotonicity given a certain number of bits, we start from the observation

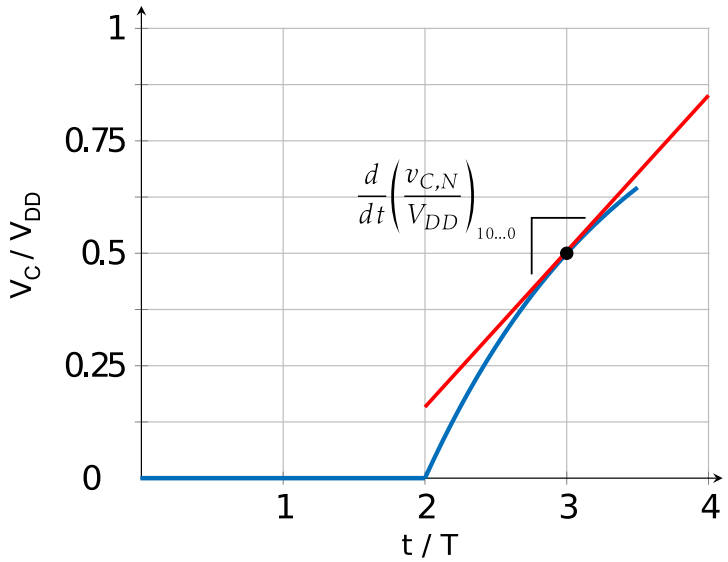


FIGURE 3.19 – Representation of the Taylor expansion in the end of the last converted bit.

that:

- Maximum voltage deviation for a given jitter happens in the middle of the dynamic (digital code '10...0').
- Voltage error on that code is related only on the MSB clock jitter, being all other bits set to zero (this is why the parallelogram-shaped distribution of red dots seen in Figure 3.18 on the previous page).

We can thus make a first order Taylor expansion around the end of the last shift period, as represented in Figure 3.19 for a three bit converter:

$$\frac{d}{dt}\left(\frac{v_{C,N}}{V_{DD}}\right)_{10...0} = \frac{d}{dt}\left(1 - e^{-\frac{t}{\tau}}\right)\Big|_{t=T} = \frac{e^{-\frac{T}{\tau}}}{\tau} \tag{3.16}$$

given relation 3.4:

$$\frac{d}{dt}\left(\frac{v_{C,N}}{V_{DD}}\right)_{10...0} = \frac{1}{2\tau} \tag{3.17}$$

If the clock phase error ΔT is small enough we can write the Taylor expansion as:

$$\frac{\Delta v_{C,N}}{V_{DD}} = \Delta T \cdot \frac{d}{dt}\left(\frac{v_{C,N}}{V_{DD}}\right)_{10...0} + \quad o(\Delta T) \tag{3.18}$$

Neglecting $o(\Delta T)$ and imposing conversion error to be less than half LSB:

$$\left.\frac{\Delta v_{C,N}}{V_{DD}}\right|_{max} = \Delta T \cdot \frac{1}{2\tau} < \frac{1}{2^{N+1}}$$
$$\Delta T < \frac{\tau}{2^N}$$

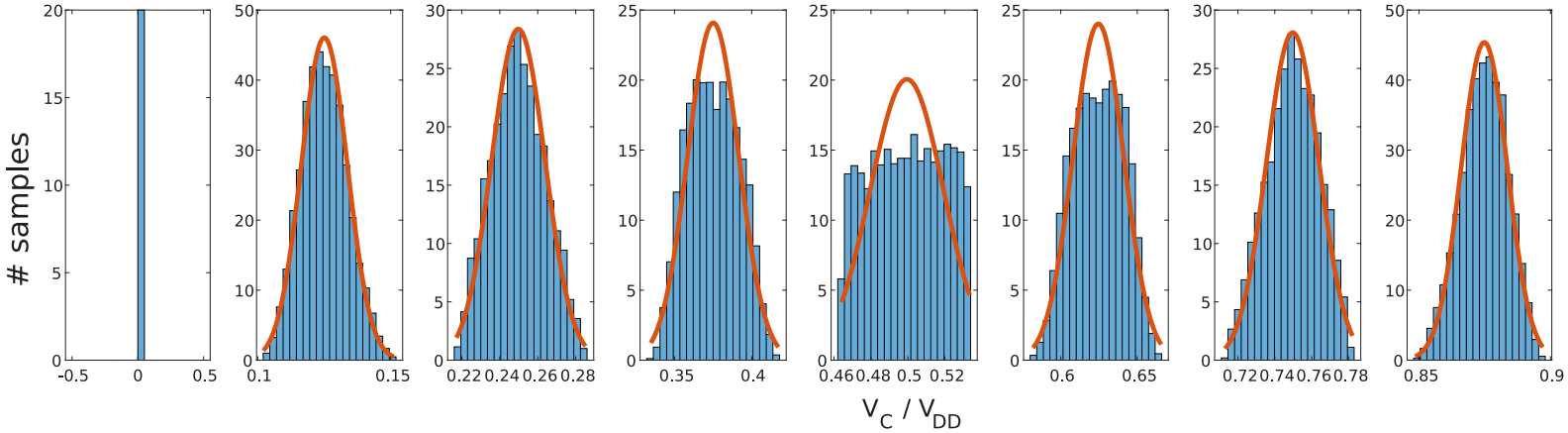
Recalling that $T = \tau \cdot log(2)$ (equation 3.4) we get:

$$\left.\frac{\Delta T}{T}\right|_{max} = \frac{1}{2^N \cdot log(2)} \tag{3.19}$$

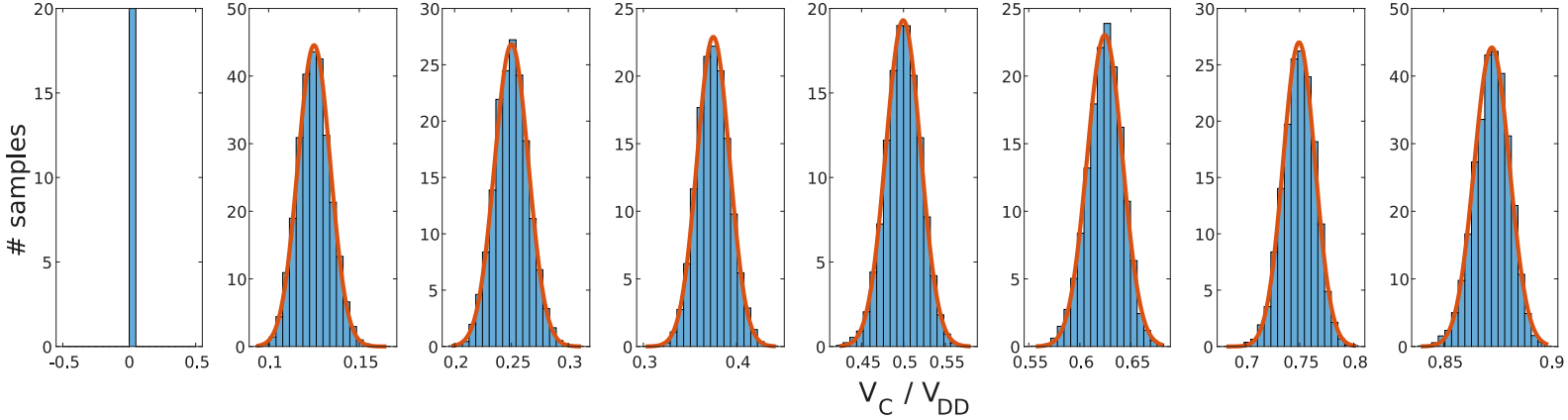
Evaluating 3.19 for $N = 3 - bit$, we obtain a maximum time error constraint of $\Delta T/T = \pm 18\%$, which satisfies monotonicity in most of the cases. This approximation is expected to improve with increasing number of bits. It has to be noticed that random uniform jitter inside $\pm 18\%$ of the clock period is less than the 31% fixed error found in Table 3.1 page 41 to get monotonicity in a 3-bit converter, owing to the randomly alternating positive and negative errors which make the constraint tighter.
A more realistic approach is to consider phase noise a normally distributed instead of uniformly, having standard deviation σ_T.
Replacing ΔT with $3\sigma_T$ in equation 3.19 :

$$\left.\frac{\sigma_T}{T}\right|_{max} = \frac{1}{3 \cdot 2^N \cdot log(2)} \approx \frac{1}{2^{N+1}} \tag{3.20}$$

Voltage waveforms and end-conversion voltages for a 3-bit converter relative to 10^4 simulations of whole dynamic, with a normally distributed jitter on clock having deviation $3\sigma_T/T = 0.18$ (found with 3.20), is visible in Figure 3.21 page 51
Again, fitting the dot distributions with Gaussian ones, we obtain the flattened view in Figure 3.20(b) on the next page, in which it can be seen the monotonicity constraint being fulfilled.
Repeating the analysis for a 10-bit converter, maximum clock jitter is evaluated having $3\sigma_T = 1.4 \cdot 10^{-3}$. Output voltage distributions are fitted with Gaussian ones, and their $3\sigma_v$ deviation is plotted versus digital codes (see Figure 3.22 page 51). From the image it can be noticed that monotonicity is fulfilled (3σ deviation is always below half LSB) and equation 3.20 is a pretty good approximation for high number of bits.



(a) Gaussian interpolation of end-conversion values relative to 10^4 conversions of each code of a 3-bit converter with clock jitter uniformly distributed in $\pm 10\%$ of the clock period.



(b) Gaussian interpolation of end-conversion values relative to 10^4 conversions of each code of a 3-bit converter with clock jitter normally distributed with $3\sigma_T/T = 0.18$.

FIGURE 3.20 – Gaussian interpolation of converted voltage distributions.

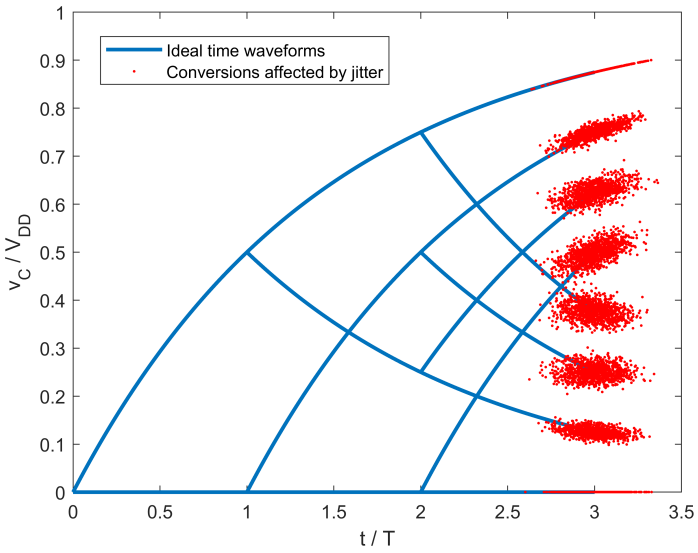


FIGURE 3.21 – Ideal waveforms and 10^4 conversion of the whole dynamic with clock jitter normally distributed with $3\sigma_T/T = 0.18$.

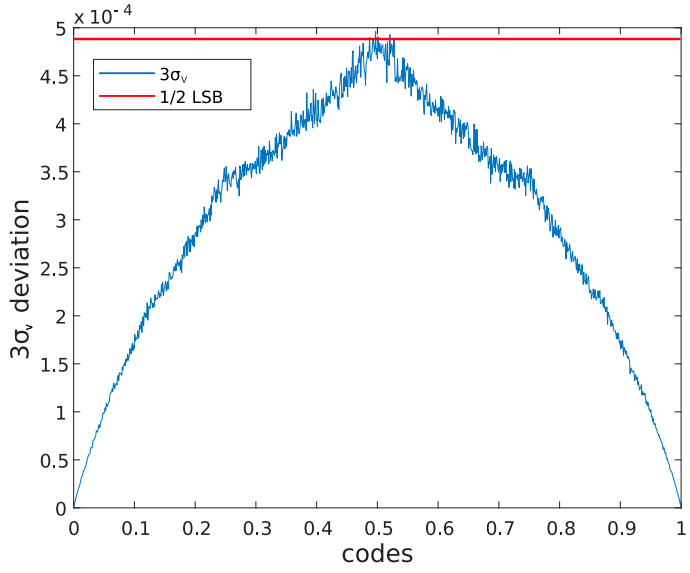


FIGURE 3.22 – $3\sigma_V$ deviation for all converted codes of a 10-bit converter, with $3\sigma_T/T = 1.4 \cdot 10^{-3}$, showing monotonicity ($3\sigma_V$ deviation less than $1/2LSB$ for practically all codes.

3.4 Buffer Charge Injection

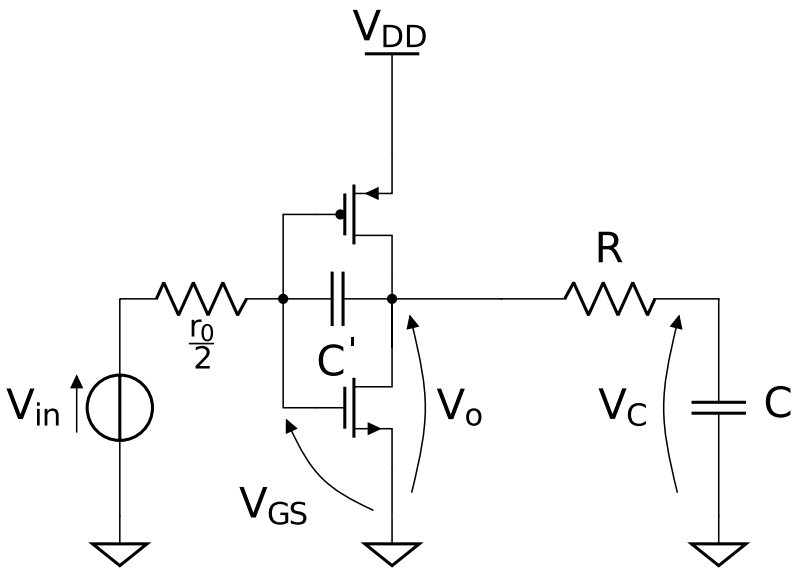


Figure 3.23 – Buffer RC model.

In order to take into account the effect of driving buffer to the static nonlinearity of the converter, the model depictedd in Figure 3.23 is employed.
Few assumptions have been initially made:

- Resistor $R >> r_0$, where r_0 is the average output resistance of the transistor channel during commutation, considered equal for both n and p transistors.
- Charge injection when transistors are in triode is neglected due to the lower channel resistance.
- The shown inverter is supposed to be driven by an identical stage, modelled with the output source impedance of $r_0/2$.
- A_v is the voltage gain of the inverter.
- Transition time is dominated by the Miller time constant τ_M

Simplified time waveforms during commutation are shown in Figure 3.24 on the facing page

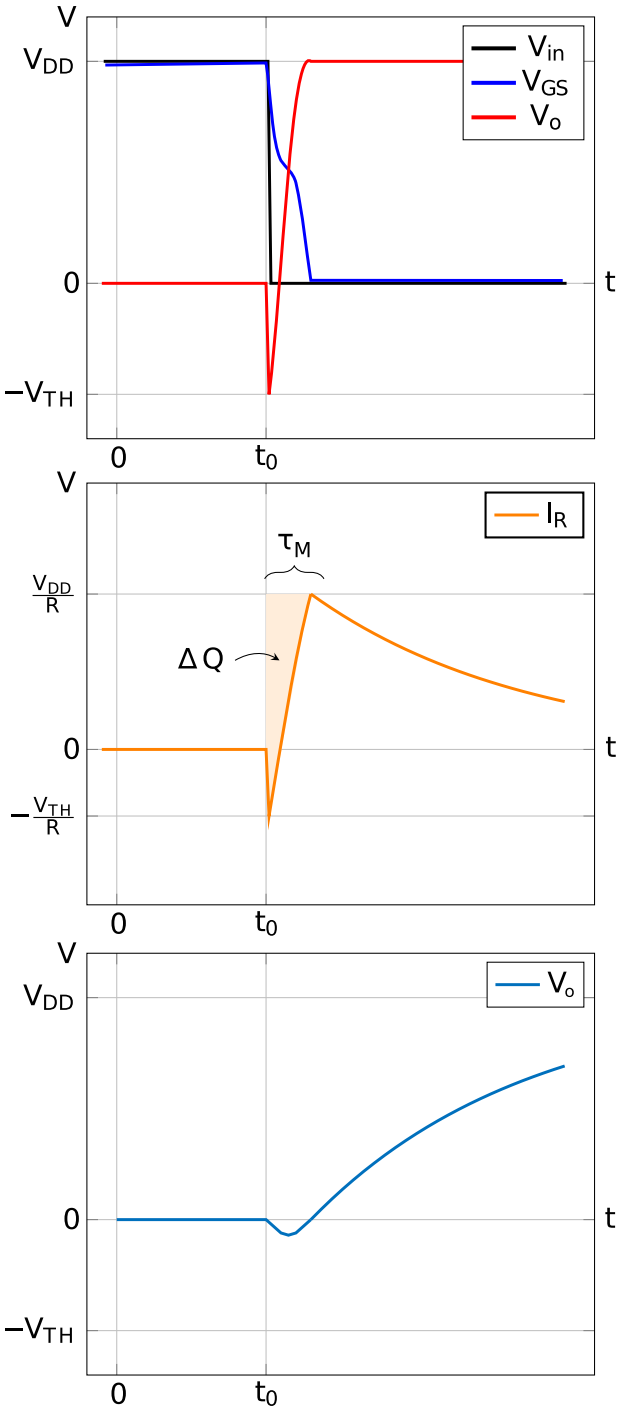

V
V_DD
0
−V_TH
0
t_0
t
V_in
V_GS
V_o
V
I_R
τ_M
V_DD/R
ΔQ
0
−V_TH/R
0
t_0
t
V
V_DD
0
−V_TH
0
t_0
t
V_o

Figure 3.24 – .

Since an error in output voltage is due to by a charge error ΔQ (emphasized as the orange area delimited by current in Figure 3.24 on the previous page), we can estimate the output voltage error by approximating the current transient with a triangular shape and evaluating the charge error:

$$\Delta Q \approx \frac{V_{DD} + V_{TH}}{R} \cdot \frac{\tau_M}{2} \tag{3.21}$$

If we want this charge to produce an output voltage change which is less than 1/2LSB we impose:

$$\begin{aligned} &\Delta V_C = \frac{\Delta Q}{C} < \frac{V_{DD}}{2^{N+1}} \qquad N = \text{number of bits} \\ &\frac{V_{DD} + V_{TH}}{RC} \cdot \frac{\tau_M}{2} < \frac{V_{DD}}{2^{N+1}} \\ &\frac{\tau_M}{\tau} < \frac{V_{DD}}{V_{DD} + V_{TH}} \cdot \frac{1}{2^N} \end{aligned} \tag{3.22}$$

If we consider τ_M being a delay introduced by the buffer, it is coherent the fact that evaluating the ratio τ_M/τ it is of the same order of magnitude as $\Delta T/T$ in 3.19. We should remember that the conclusion obtained is valid under the assumptions above, in particular the one that switching non-ideality is dominated by saturation/Miller effect. As we will see in the following paragraph this is not the case.

3.5 Buffer Impedance and Resistor Distributed Capacitance

Different models have been developed to predict time behaviour of transistors as RC-tree networks, from Elmore's delay model [17] to Penfield-Rubinstein [18] and Horowitz [19].
The RC delay model has been mainly developed to model in a computationally efficient way delay of logic gates. It models transistors as an averaged resistance (R_{eff}) and capacitance (C_{eff}) over the switching range of the gate [20].
In the framework of our design, the following assumptions are made:

- Channel effective resistance for a unit transistor is considered to be orders of magnitude below with respect to the nominal RC resistor.
- Gate and diffusion capacitances (considered equal and equivalent to C_{eff}) along with stage gain are considered to be small enough so that Miller effect can be neglected too.

Moreover, n and p transistors are sized to have the same channel resistance:

$$\left(\frac{W}{L}\right)_p \approx 2\left(\frac{W}{L}\right)_n \quad \Longrightarrow \quad R_p \approx R_n = R_{\text{eff}}$$
$$C_n = C_{\text{eff}} \qquad C_p \approx 2C_n \tag{3.23}$$

RC-delay model can thus be estimated, and the transistor model of Figure 3.25(a) is transformed into the RC-delay one of Figure 3.25(b).

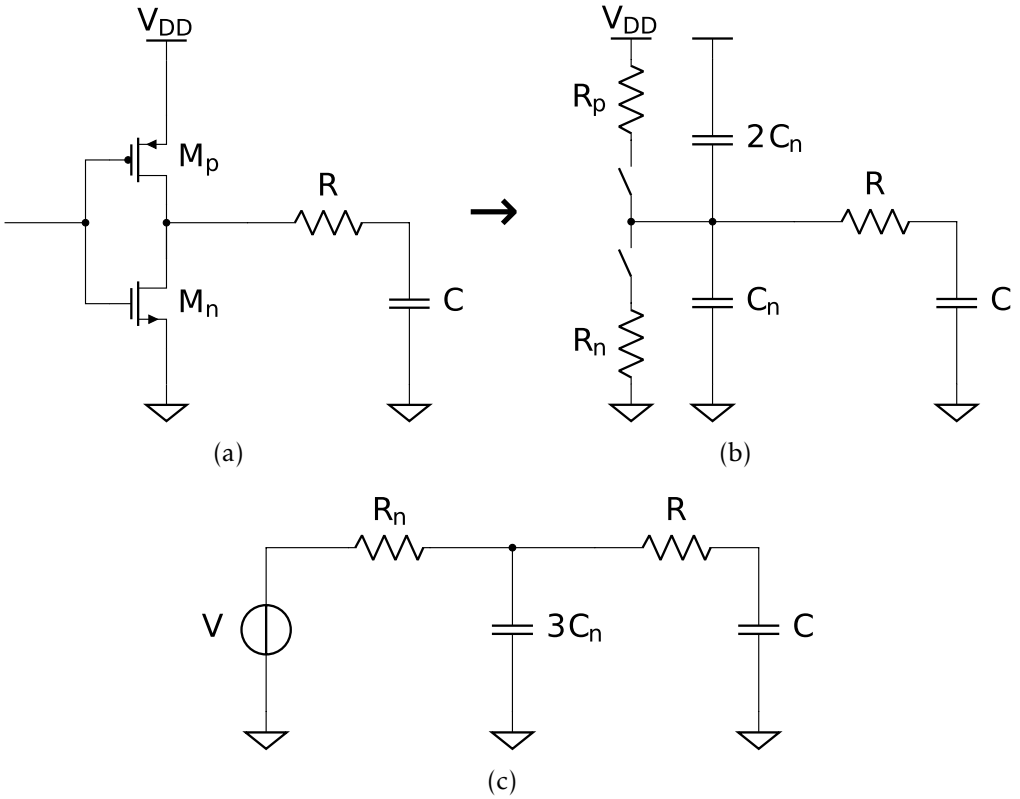


Figure 3.25 – Transistor model transformation into RC delay model.

For transients modelling, the circuit in Figure 3.25(b) on the preceding page can be further simplified into the one of Figure 3.25(c) on the previous page, where the voltage forced by the ideal voltage source can be 0V or V_{DD}.
Considering the above assumptions (dominance of triode resistance effect over Miller/transmission zero), and what we learn from Horowitz [19]:

- RC trees dominated by a pole which is at very low frequency with respect to other poles and zeros, is well approximated by a single-resistor single-capacitor circuit.
- There are only two classes of linear network that cannot be approximated by a single pole response: circuits with coincident poles and circuits with a low frequency pole-zero pair, which causes the output to have a two time constant behaviour.

if we design the circuit to have transient response dominated by R and C, we can approximate the system behaviour with a single time constant τ_{Di}:

$$\tau_{Di} = \sum_k R_{ki} \cdot C_k \tag{3.24}$$

where τ_{Di} is the *first order moment of the circuit impulse response*, using Elmore's notation [17]; τ_{Di} defines the delay of the signal from the input to node i of the RC-tree ; C_k is the capacitance at the k^{th} node of the RC-tree and R_{ki} is the intersection between the resistance path from input to node i and from input to node k.
Referring to Figure 3.25(c) on the preceding page our RC-tree has just two nodes and we can evaluate the first order moment to the output capacitance as:

$$\tau_{Di} = R_n \cdot 3C_n + (R_n + R) \cdot C \tag{3.25}$$

If we choose for our design reasonable values:

- R_n in the order of one $k\Omega$
- R above a hundred of $k\Omega$
- C_n in the fF range
- C to be hundreds of fF

we can neglect the first term of the sum, which becomes five order of magnitude less than the second one, meaning that buffer parasitic capacitances do not play significant role with respect to triode resistance R_n:

$$\tau_{Di} \approx (\Delta R + R) \cdot C = \Delta\tau + \tau \qquad \Delta R = R_n \tag{3.26}$$

Since maximum error happens at half the dynamic we can impose that the voltage difference between the ideal conversion, obtained with ideal time constant τ, and the one obtained with the first order moment τ_{Di} is less than half LSB:

$$\begin{cases} V_{C,Di_{10..0}} = V_{DD}(1 - e^{-\frac{T}{\tau_{Di}}}) \\ V_{C_{10..0}} = V_{DD}(1 - e^{-\frac{T}{\tau}}) \\ V_{C_{10..0}} - V_{C,Di_{10..0}} < \frac{1}{2^{N+1}} \end{cases}$$

By substituting the first two relations in the third we get:

$$-e^{-\frac{T}{\tau}} + e^{-\frac{T}{\tau_{Di}}} < \frac{1}{2^{N+1}}$$
$$-\frac{T}{\tau_{Di}} < log\left(\frac{1+2^N}{2^{N+1}}\right)$$

which means that the first order moment τ_{Di} must be less than:

$$\tau_{Di} < -\frac{T}{log\left(\frac{1+2^N}{2^{N+1}}\right)} \qquad \tau_{Di} \approx RC + \Delta R \cdot C = \tau + \Delta\tau \tag{3.27}$$
$$\tau + \Delta\tau < -\frac{T}{log\left(\frac{1+2^N}{2^{N+1}}\right)}$$

The error on time constant introduced by the driving buffer is then bounded accordingly:

$$\Delta\tau < -\frac{log(2)}{log\left(\frac{1+2^N}{2^{N+1}}\right)}\tau - \tau$$

$$\frac{\Delta\tau}{\tau} < -\left(\frac{log(2)}{log(\frac{1+2^N}{2^{N+1}})} + 1\right) \tag{3.28}$$

Remembering (3.27), (3.28) means also that:

$$\frac{\Delta R}{R} < -\left(\frac{log(2)}{log(\frac{1+2^N}{2^{N+1}})} + 1\right) \tag{3.29}$$

By evaluating equation 3.28 for a 10 bit converter we get:

$$\frac{\Delta\tau}{\tau} = 1.41 \cdot 10^{-3} \tag{3.30}$$

This result is almost coincident to the one obtained in Table 3.1 page 41 for negative constant delay, and can be explained by thinking that a delay in the switching buffer is equivalent to a reduction of clock period. This can be seen by comparing exponential terms in the two cases:

Buffer delay of $\Delta\tau$:

$$e^{-\frac{T}{\tau+\Delta\tau}} = e^{-\frac{T}{\tau(1+\frac{\Delta\tau}{\tau})}} = e^{-\frac{T}{\tau}(1-\frac{\Delta\tau}{\tau})} \cdot e^{o(\frac{\Delta\tau}{\tau})} \tag{3.31}$$

Clock reduction of ΔT:

$$e^{-\frac{T-\Delta T}{\tau}} = e^{\frac{T}{\tau}(1-\frac{\Delta T}{T})} \tag{3.32}$$

Neglecting the term $o(\frac{\Delta\tau}{\tau})$ (3.31) and (3.32) have the same form.
This means that buffer delay can almost completely be compensated introducing a controlled increase in clock period equal in magnitude to the buffer delay:

$$\frac{\Delta T}{T} \approx \frac{\Delta\tau}{\tau} \tag{3.33}$$

From (3.29), we also know that:

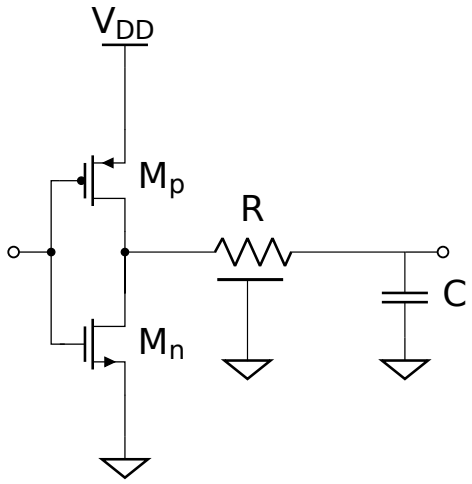
$$\frac{\Delta T}{T} \approx \frac{\Delta R}{R} \tag{3.34}$$

3.5.1 Resistor distributed capacitance

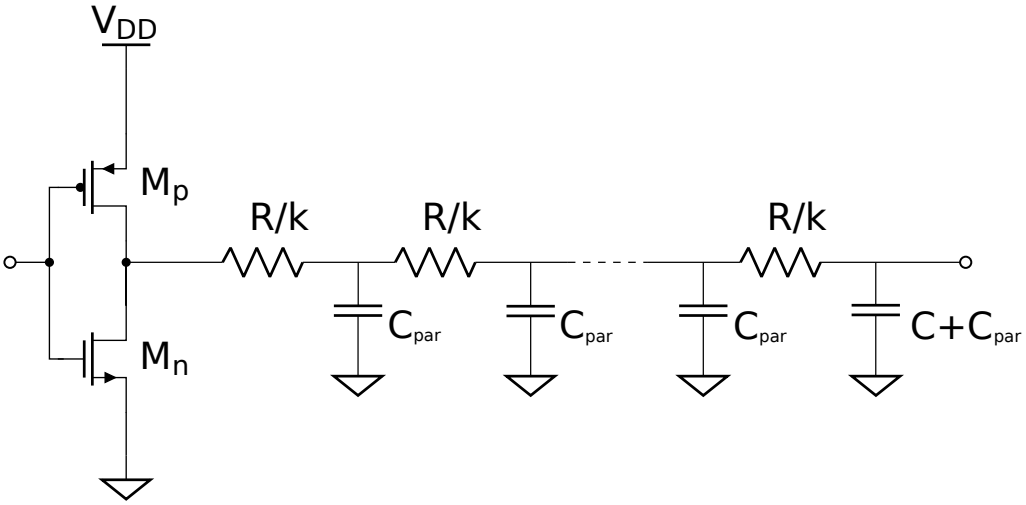
We saw that converter accuracy is strictly dependent on time base accuracy. This means that, for a given sample rate requirement and capacitor C designed to have low energy dissipation (thus very small), resistor R must be sized to satisfy (3.4). This can lead to quite large polysilicon integrated resistors (kΩ-MΩ range) which leads to the drawback of parasitic distributed capacitance towards substrate, modelled as in Figure 3.26 on the following page .
Again, theory about RC-trees comes to help, since we know that [19]:

- Voltage in the end of a long polysilicon wire (modelled as a series of small RC sections) is also nicely approximated by an exponential behaviour.
- Adding a capacitive load in the end of an RC line lowers more the frequency of the dominant pole, which means the single-pole estimate is an even better approximation.

Parasitic capacitance is therefore changing the nominal time constant value (which can anyway be compensated by a correction on clock period) while keeping the linear exponential behaviour.
For the sake of our converter operation the presence of higher order poles is anyway not completely negligible, and their effect must be properly compensated, as we will see in the following chapter.



(a) Resistor with parasitic distributed capacitance towards substrate



(b) Distributed capacitance model

FIGURE 3.26 – Resistor and its distributed capacitance model.

3.6 Energy per Conversion estimation

In order to estimate the DAC power consumption, it has been supposed that all contributions are negligible but the energy to charge the output capacitor.

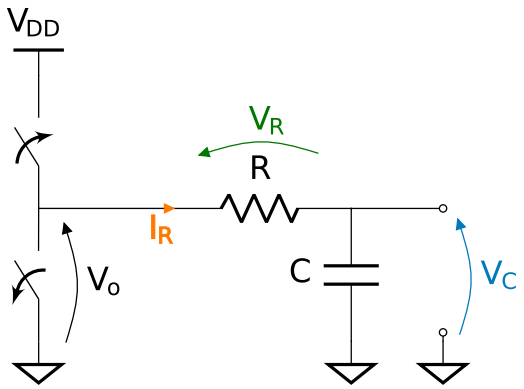


Figure 3.27 – Simplified circuit model of the DAC.

Energy provided by power supply depends on the code which is being converted, and in particular on the time intervals in which the bit is '1', so that capacitor is charged trough V_{DD}.
If we consider, as example, the energy dissipated by a 3-bit converter when all bits are set to '1' we have waveforms as in Figure 3.28 page 63:

$$\begin{aligned} V_C(t) &= V_{DD}(1 - e^{-\frac{t}{\tau}}) \\ V_R(t) &= V_{DD} - V_C(t) = V_{DD}e^{-\frac{t}{\tau}} \\ I_R(t) &= \frac{V_R}{R} = \frac{V_{DD}}{R}e^{-\frac{t}{\tau}} \end{aligned} \tag{3.35}$$

the power provided by supply during conversion is:

$$P_{gen} = V_{DD} \cdot I_R(t) = \frac{V_{DD}^2}{R}e^{-\frac{t}{\tau}} \tag{3.36}$$

and the energy dissipated is:

$$\begin{aligned} E_{gen,111} &= \int_0^{3T} P_{gen} dt \\ &= \int_0^{3T} \frac{V_{DD}^2}{R} e^{-\frac{t}{\tau}} dt \\ &= \left[-\tau \frac{V_{DD}^2}{R} e^{-\frac{t}{\tau}} \right]_0^{3T} \\ &= -\frac{RC}{R} V_{DD}^2 \left[e^{-\frac{3T}{\tau}} - 1 \right] \\ &= -CV_{DD}^2 \left[\frac{1}{2^3} - 1 \right] = \frac{7}{8} CV_{DD}^2 \end{aligned} \tag{3.37}$$

Energy can be also expressed as a function of voltage increments on the capacitor $\Delta V_C(i) = V_C((i+1)T) - V_C(iT)$, where i indicates each i^{th} period corresponding to a '1' bit. We notice that the generic voltage increment $\Delta V_C(i)$ is also equal to the voltage across the resistor in the end of the same period, i.e.

$$\Delta V_C(i) = V_R((i+1)T) \tag{3.38}$$

Conversion energy in (3.37) can thus be expressed as:

$$\begin{aligned} \frac{7}{8} CV_{DD}^2 &= \frac{\Delta V_C(0) + \Delta V_C(1) + \Delta V_C(2)}{V_{DD}} CV_{DD}^2 \\ &= \frac{V_R(T) + V_R(2T) + V_R(3T)}{V_{DD}} CV_{DD}^2 \end{aligned} \tag{3.39}$$

Conversion energy for code '101' is again related to time waveforms (see Figure 3.29 on the facing page):

$$\begin{aligned} E_{gen,101} &= \int_0^{T} \frac{V_{DD}^2}{R} e^{-\frac{t}{\tau}} dt + \int_{2T}^{3T} \frac{6}{8} \frac{V_{DD}^2}{R} e^{-\frac{t-2T}{\tau}} dt \\ &= \left[-\tau \frac{V_{DD}^2}{R} e^{-\frac{t}{\tau}} \right]_0^{T} + \frac{3}{4} \left[-\tau \frac{V_{DD}^2}{R} e^{-\frac{t-2T}{\tau}} \right]_{2T}^{3T} = \frac{7}{8} CV_{DD}^2 \end{aligned} \tag{3.40}$$

(3.40) can be also expressed as:

$$\frac{7}{8} CV_{DD}^2 = \frac{V_R(T) + V_R(3T)}{V_{DD}} CV_{DD}^2 \tag{3.41}$$

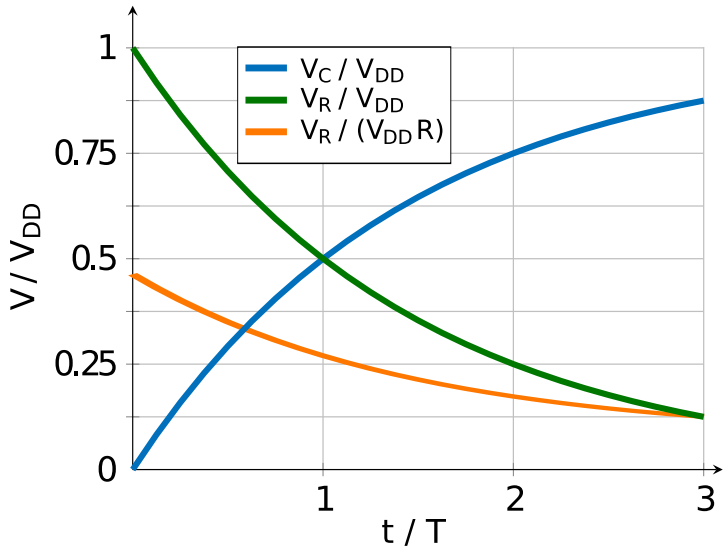


FIGURE 3.28 – Time waveforms relative to conversion of code '111'.

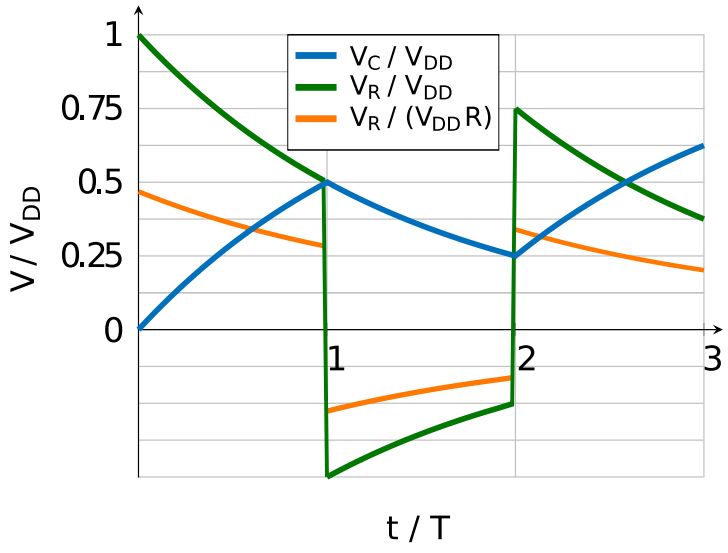


FIGURE 3.29 – Time waveforms relative to conversion of code '101'.

By generalising (3.39) and (3.41), conversion energy relative to the generic digital code '$b_{N-1}...b_0$' is:

$$E_{gen,[b_{N-1}:b_0]} = CV_{DD}^2 \sum_{0}^{N-1} \frac{V_R((i+1)T)}{V_{DD}} \cdot b_i \tag{3.42}$$

Simplifying the notation and normalising energy with respect to CV_{DD}^2 we obtain normalised conversion energy:

$$E_n = \sum_{0}^{N-1} \frac{V_{R,i+1}}{V_{DD}} \cdot b_i \tag{3.43}$$

(3.43) has been used to evaluate conversion energy for each digital code of a 10-bit converter. Energy trend and its average is shown in Figure 3.30.

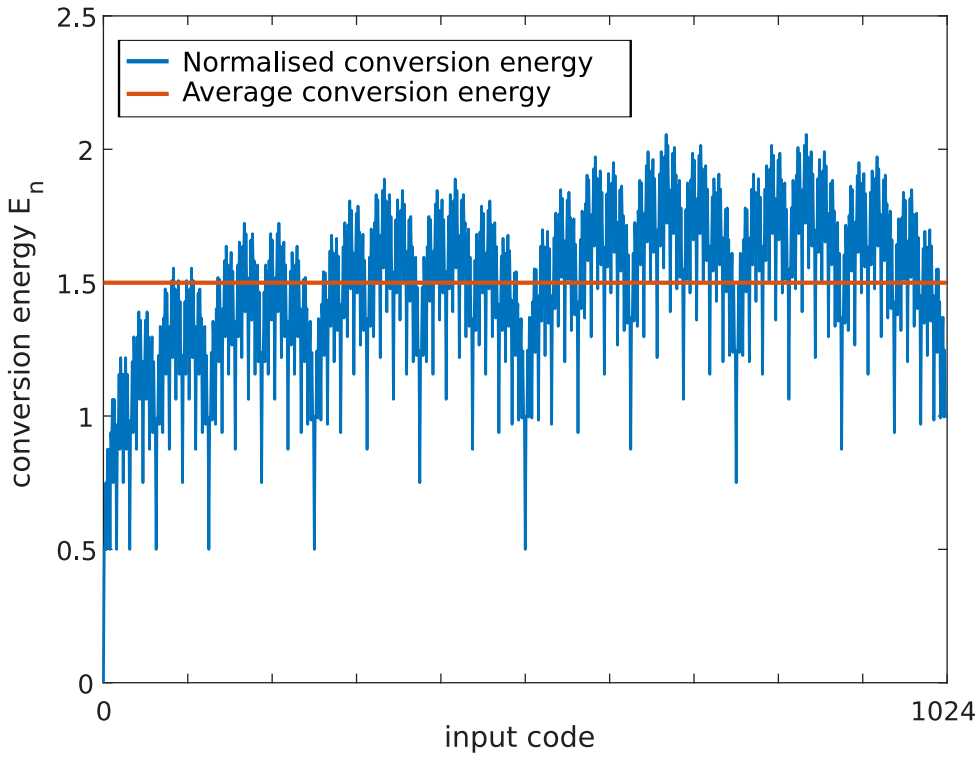


Figure 3.30 – Conversion energy and its average of a 10-bit converter.

Average conversion energy has been evaluated for different word lengths, and it has been verified its linear increase with the number of bits, as shown in Figure 3.31.

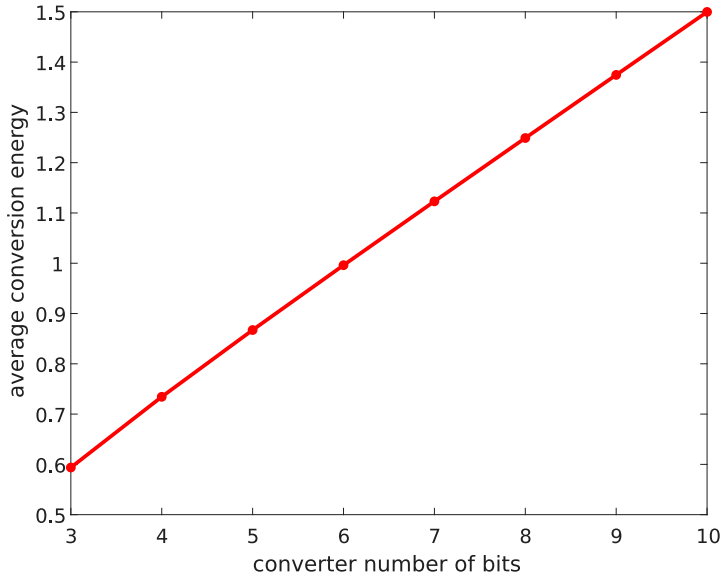


FIGURE 3.31 – Average conversion energy versus converter number of bits.

From Figure 3.30 on the preceding page we observe that conversion energy is higher for the upper half swing. This suggests to use a different conversion strategy, i.e. resetting the capacitor voltage to V_{DD} instead of ground when the code to convert is in the upper half swing (see Figure 3.32 on the following page).
Differently from the reset-to-ground operation, tough, resetting the capacitor to V_{DD} requires some amount of energy, equal to $1/2 \cdot CV_{DD}^2$ or less. Hypothesis made is that converted signal is varying much slower than the conversion frequency, thus the energy required to reset capacitor to V_{DD} is the same needed to charge the capacitor from the final converted voltage to V_{DD} (see red curve in Figure 3.32 on the next page).

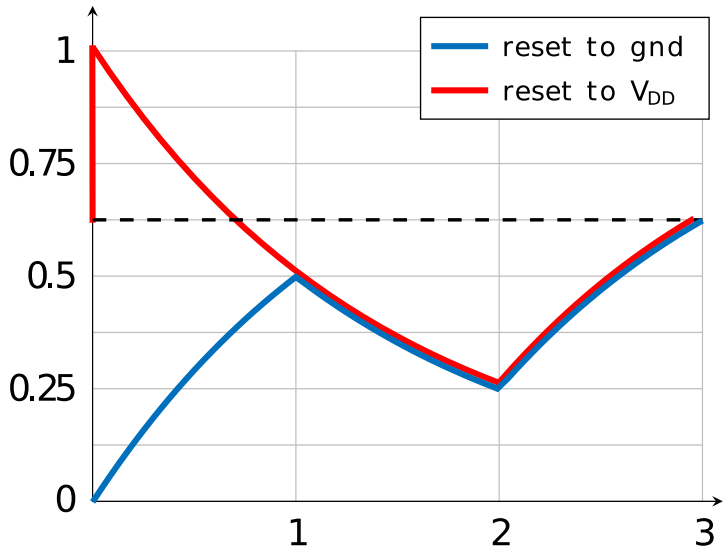


FIGURE 3.32 – Same code conversion with different capacitor-reset strategies.

In order for the DAC to convert the binary code $[b_{N-1}...b_0]$, after resetting the capacitor to V_{DD}, the code has to be transformed into a new binary code, by applying the two's complement and one's complement in sequence:

$$[b_{N-1}...b_0] \rightarrow \mathcal{C}2's \rightarrow \mathcal{C}1's \rightarrow [\tilde{b}_{N-1}...\tilde{b}_0] \tag{3.44}$$

Operation 3.44 is equivalent to subtracting 1 to the original code. The obtained digital word $[\tilde{b}_{N-1}...\tilde{b}_0]$ can then be loaded directly in the shift register after resetting capacitor to V_{DD}.
Given the above-stated assumptions, analytical expression for conversion energy can be easily derived following the same approach, obtaining:

$$E_n = \frac{(1-b_{N-1})}{V_{DD}} \cdot \left[\sum_{0}^{N-1} V_{R,i+1} \cdot b_i\right] + \frac{b_{N-1}}{V_{DD}} \cdot \left[V_{R,N-1} + \sum_{0}^{N-1} V_{R,i+1} \cdot \tilde{b}_i\right] \tag{3.45}$$

where the term b_{N-1} is used to discriminate between the lower and upper half swing. Normalised conversion energies for both conversion strategies are plotted on Figure 3.33 on the facing page with their respective average energy.

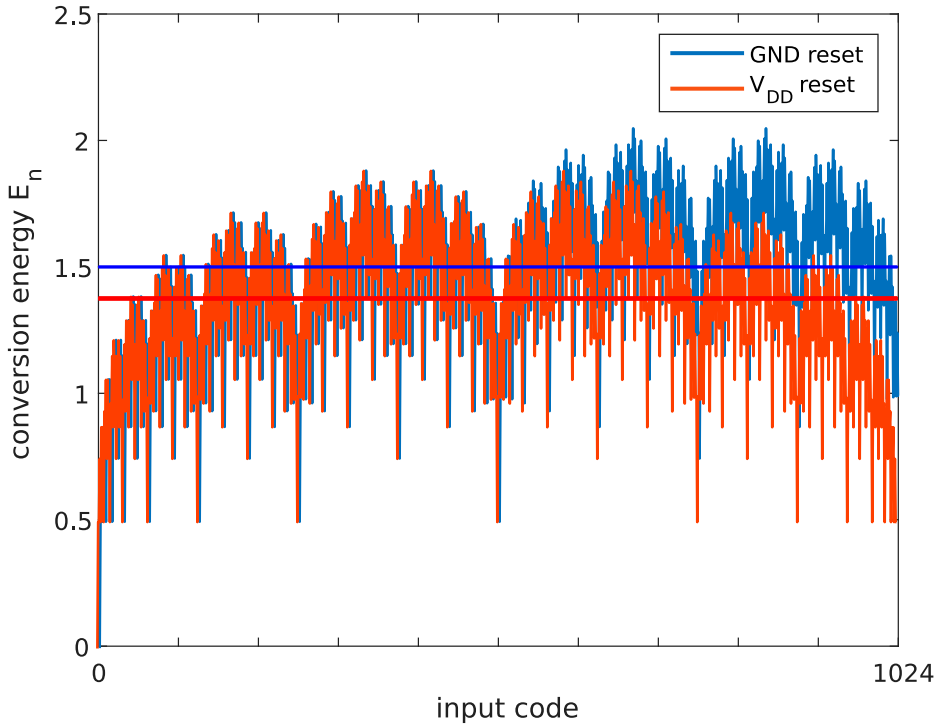


FIGURE 3.33 – Normalised conversion energies and their average for a 10-bit converter.

3.7 Output capacitor and thermal noise limit

Designing our RC network, we would like to have a capacitor as small as possible, being conversion energy directly proportional to capacitance. The lower bound to capacitance is not given by technology limit, rather by thermal noise requirements.
Output thermal noise can be evaluated [21] by considering the thermal model for the resistor depicted in Figure 3.34 on the next page, where v_{nR}^2 is the thermal noise spectral density of resistor R, $\Bbbk$ the Boltzmann constant and T the absolute temperature:

$$v_{nR}^2 = 4\Bbbk T R \tag{3.46}$$

Being the transfer function between v_R and v_C:

$$\frac{V_C}{V_R}(s) = \frac{1}{sRC+1} \tag{3.47}$$

Input noise will be shaped by the transfer function according to the known relation:

$$\begin{aligned} v_{nC}^2 &= v_{nR}^2 \cdot \left| \frac{V_C}{V_R}(f) \right|^2 \\ &= 4\Bbbk T R \frac{1}{(2\pi RCf)^2 + 1} \end{aligned} \tag{3.48}$$

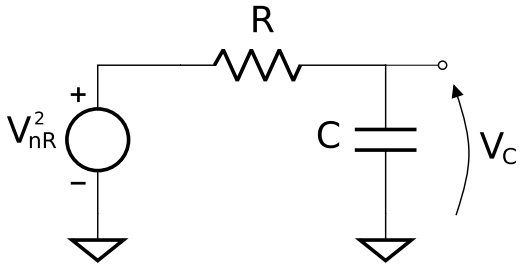


Figure 3.34 – RC network thermal model.

The overall noise output power will be:

$$\begin{aligned} P_{nC} = \int_0^\infty v_{nC}^2 df &= \int_0^\infty \frac{4\Bbbk T R}{(2\pi RCf)^2 + 1} df \\ &= 4\Bbbk T R \cdot \frac{\operatorname{atan}(2\pi RCf)}{2\pi RC}\Big|_0^\infty \\ &= \frac{2\Bbbk T}{\pi C}\left(\frac{\pi}{2} - 0\right) \\ &= \frac{\Bbbk T}{C} = \overline{v_{nC}^2} \end{aligned}$$

from which we can derive the total noise *rms* voltage at the output:

$$v_{nC,rms} = \sqrt{(\overline{v_{nC}^2})} = \sqrt{\frac{\Bbbk T}{C}} \tag{3.49}$$

We can now evaluate the minimum capacitor $C_{\min}$ for a N bit converter such that thermal voltage gives an error contribution of less than $1/2$ LSB on output:

$$3v_{nC,rms} = 3\sqrt{\frac{\Bbbk T}{C}} < \frac{1}{2^{N+1}} \tag{3.50}$$

$$C > \frac{9\Bbbk T}{V_{DD}^2} 2^{2(N+1)} = C_{\min} \tag{3.51}$$

Evaluating (3.51) at a temperature of 300K, the minimum capacitance required to meet thermal noise constraint is plotted as a function of N in Figure 3.35, for different V_{DD}. Based on this requirement, the capacitance of the relaxation DAC has been designed.

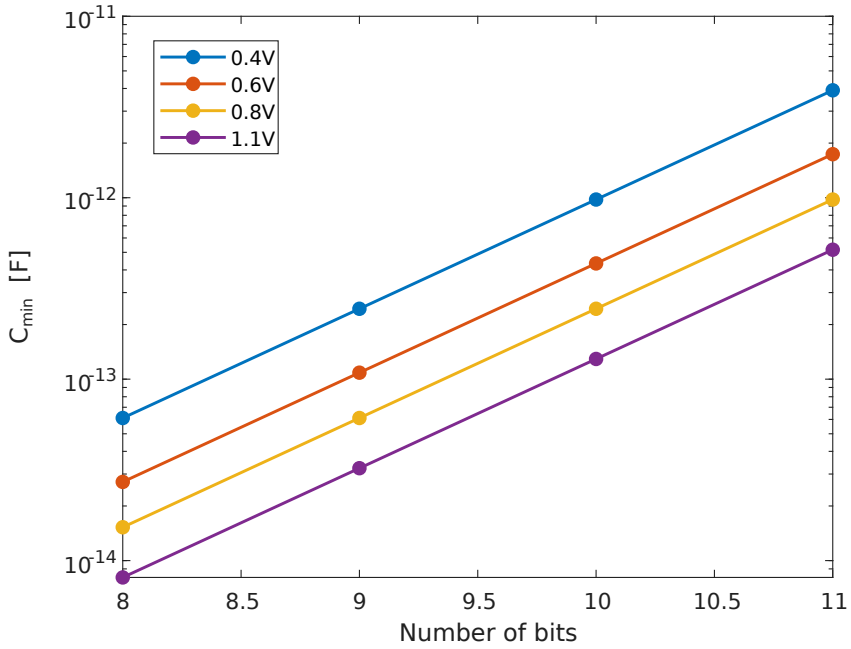


Figure 3.35 – Minimum capacitance for thermal noise constraint versus converter number of bits for different supply voltages.

Chapter 4

Design and Simulations

This chapter presents a design procedure for the Relaxation DAC, from passive components sizing to active devices, following the guidelines derived in the previous chapter. A technique to compensate non-ideal transient behaviour is defined and a set of test-cases are fixed to explore the space of design solutions.
Cadence simulations are performed on test-cases blocks at the behavioural, schematic and post-layout level, using nominal conditions and in the presence of mismatch and temperature variations.

4.1 Simulation Setup

Different simulations have been carried out to validate the converter performance:

- Preliminary schematic level simulation of the RC network driven by ideal voltage pulses, to verify time domain circuit response (simulation setup shown in Figure 4.3 page 74).
- Schematic level simulations of the RC network properly driven by behavioural Verilog-A blocks, in order to perform full-swing conversion and evaluate static converter errors (simulation setup shown in Figure 4.4 page 75).
- Post layout simulations of the RC network properly driven by behavioural Verilog-A blocks, in order to take into account layout

parasitics effect on static errors (simulation setup in Figure 4.4 page 75).
- Post layout simulations of both the RC network and tristate buffer driver, to evaluate the effects of finite driving impedance, leakage and charge injection on converter linearity (simulation setup in Figure 4.5 page 76).

In order to perform the above simulations, different blocks needed to be implemented, described as follows.

RC block The core converter block is the RC network, which symbol is shown in Figure 4.2(b) page 74. It is composed by a n-well isolated polysilicon resistor and a p-well isolated metal capacitor. The block can be simulated at schematic or extracted-layout level.

ADC_10 It is a behavioural Verilog-A block in charge of generating digital words to be converted, emulating an ideal 10-bit ADC, which samples a generic analog signal *'in'* on the clock *'CK'* edge and converts it to the 10-bit digital word *'D<9:0>'*. In the same time it holds the sampled analog voltage on the output 'V_{out}'. All voltages are referred to the pin *'ref'*. The block is visible in Figure 4.2(a) page 74. Verilog-A code is reported in Appendix A page 119.

P_SHIFT It is a Verilog-A block which emulates the behaviour of a parallel-in serial-out shift register, loading a digital word *'D<9:0>'* or shifting a bit of the loaded word on pin *'out'*, depending on the control voltage level of pin *'LD'*, on the rising clock *'CK'* edge. Pins *'dd'* and *'ref'* define output voltage levels corresponding to a '1' or a '0' (V_{DD} and GND respectively). The block is visible in Figure 4.2(c) page 74. Verilog-A code is reported in Appendix A page 121.

OutputComp It is a Verilog-A block which compares ideal converted voltage level *'adc'* to the output DAC voltage *'dac'*, sampled on the rising edge of *'samp'* signal. The block outputs the difference between converted voltage and ideal analog voltage (i.e. integral error) on pin *'dac_adc'*, while holding DAC converted voltage on *'dac_hold'* pin. The block is visible in Figure 4.2(d) page 74. By changing supply voltage between *'ref'* and *'dd'* pin it is possible to compensate for DAC gain error when evaluating INE. Verilog-A code is reported in Appendix A page 123.

Tristate Buffer In simulations where tristate buffer has been simulated as well, two additional blocks have been added, which are final buffer stage **tristate_out** and its driving block **tristate_out**, provided with enabling signal *'EN'*. The two blocks are visible in Figure 4.1.

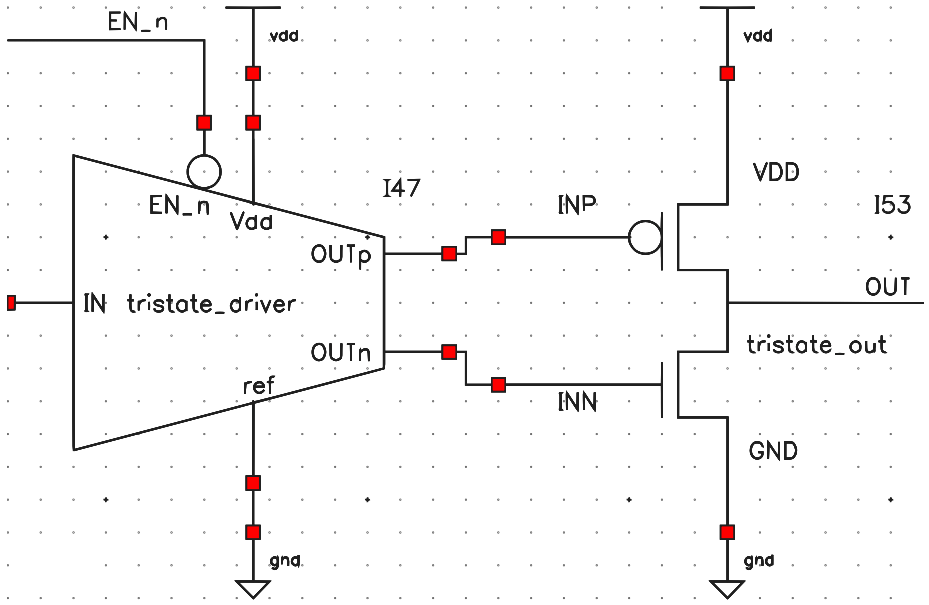


Figure 4.1 – Cadence schematic blocks for the tristate buffer.

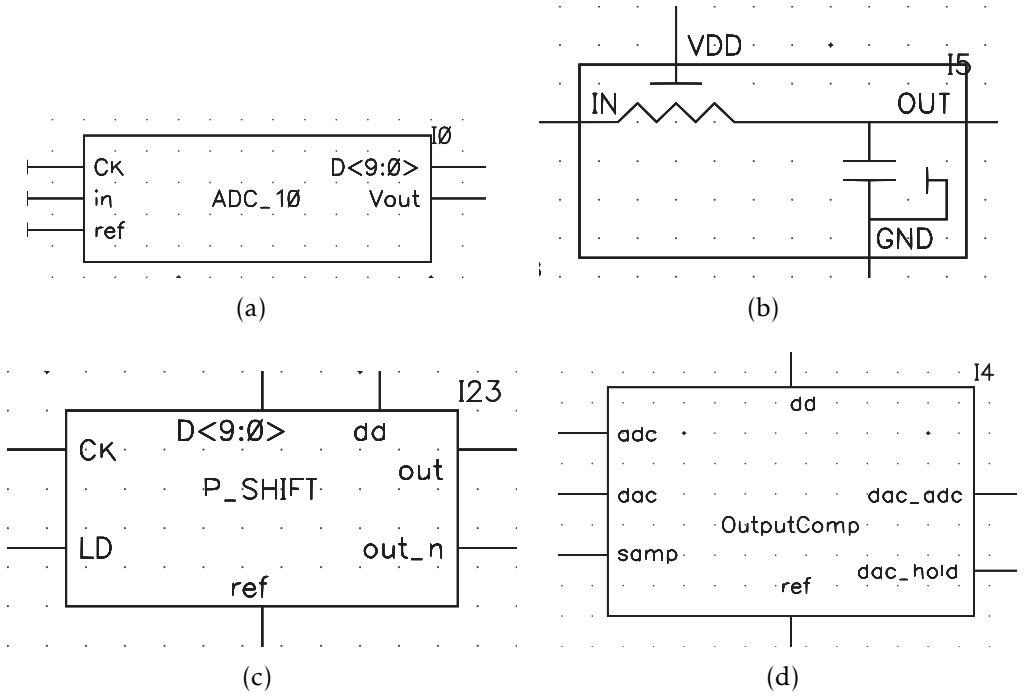


Figure 4.2 – Blocks for the initial simulation setup.

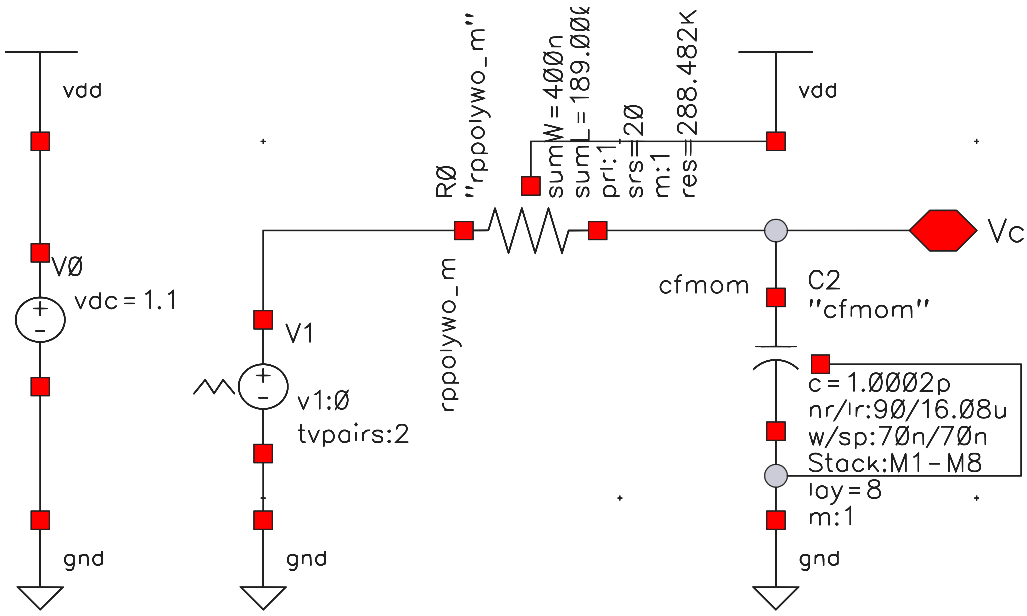


Figure 4.3 – Test circuit for transient response evaluation.

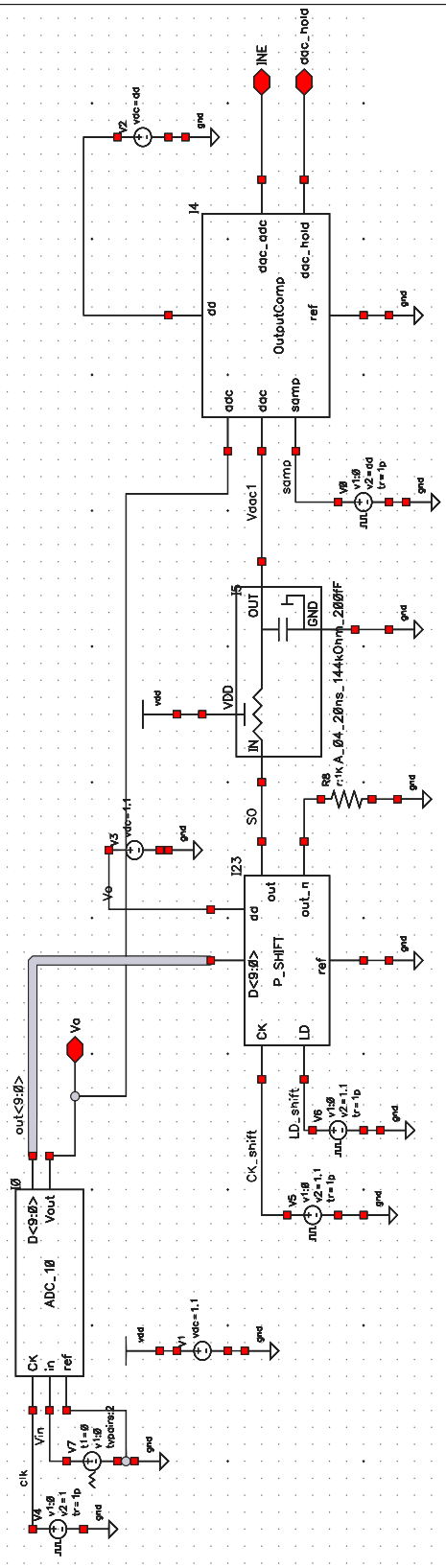


FIGURE 4.4 – Simulation setup for swing characterization.

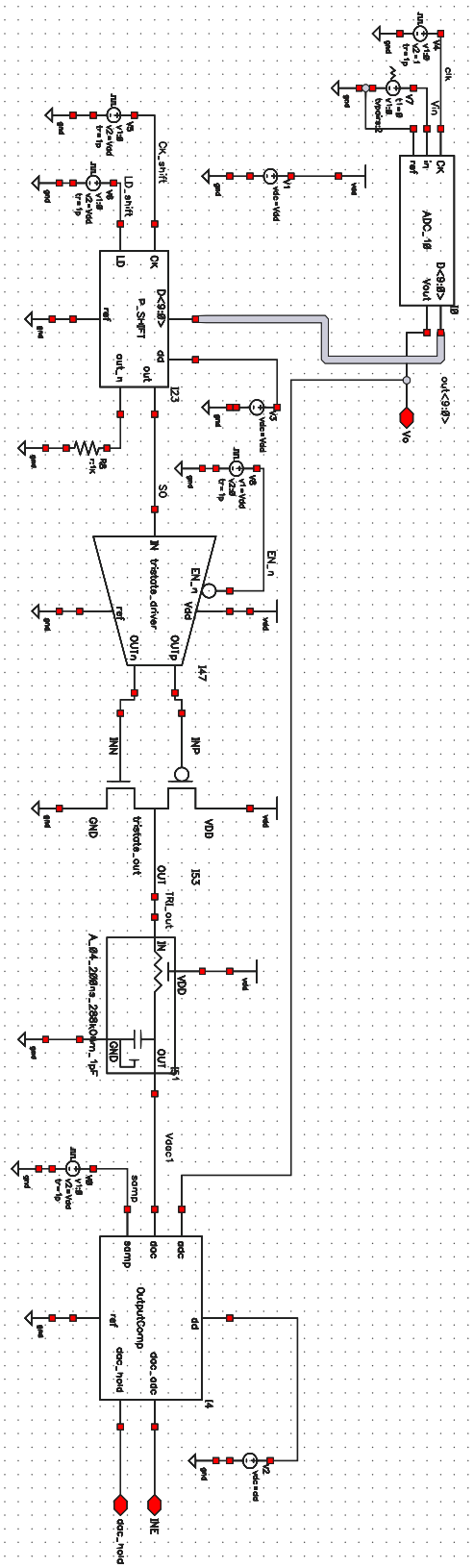


FIGURE 4.5 – Simulation setup for swing characterization, including driving buffer.

4.2 Design

The general guidelines, which have been followed to design the RC network and the buffer are reported in what follows.

4.2.1 RC design and time response cosiderations

In order to match low-power constraints, which is the main idea of our topology, the design of the passive network began from capacitor sizing. No particular matching constraint is present, so the capacitance is limited just by thermal noise. Since an initial accuracy of 10 bits has been targeted, if we refer to Figure 3.35 page 69, when supply voltage ranges from 0.4V to 1.1V, a thermal noise constraint below half LSB is fulfilled for capacitances above $130fF$ ($V_{DD} = 1.1$V, $N = 10$bits, $T = 300$K):

$$C_{\min} = \frac{9kT}{V_{DD}^2} 2^{2(N+1)} = 129.2fF$$

In order to analyse performance for different combinations of C and R, a set of capacitances ranging from $200f$F to $1p$F have been chosen. An extra configuration having a much larger capacitor of $7.13p$F has been chosen so that to have a test case in which distributed resistor capacitance is for sure negligible with respect to the nominal one.
Then clock periods have been sized along with resistors, by setting a range for conversion rate between $100k$S/s and $4M$S/s, supposing that clock can be externally provided with the required accuracy, leading to clock periods in the range $20n$s $\rightarrow 1\mu$s (if we consider 10 conversion periods and 2 hold ones) and resistors compliant with (3.4):

$$e^{-\frac{T}{\tau}} = \frac{1}{2} \quad \Longrightarrow \quad \tau = \frac{T}{log(2)} \quad \Longrightarrow \quad R = \frac{T}{C \cdot log(2)}$$

A different supply voltage has been assigned to each case, ranging from 0.6V to 1.1V.
Precise matching of resistor R is again not particularly relevant since (3.4) can be re-established through calibration of clock period T. The initial set of chosen configurations is listed in the rows of Table 4.2 page 85, where clock period, R, C and supply voltage are indicated for each test case.

Components in Table 4.2 page 85 have been implemented in Cadence design environment using $40n$m technology PDK. The highest resistivity

polysilicon resistor type has been used, of minimum width, to reduce parasitic capacitance towards substrate. The maximum number of available metal layers has been used for capacitors, in order to minimize overall area.
To evaluate time domain RC response, test circuit in Figure 4.3 page 74 has been initially driven by rectangular pulses of variable width, so that commutations happen when capacitor voltage reaches $1/2 \cdot V_{DD}$, $1/4 \cdot V_{DD}$, $3/8 \cdot V_{DD}$, and so on, as shown in Figure 4.6.
Hypothesis about the shown behaviour is that it is due to multiple-time-constant response dominated by the nominal RC pole and higher order poles mainly addressable to parasitic distributed capacitance of polysilicon resistor.
Effort has been made to approximate the simulated response with an ideal delayed exponential (see red curve in Figure 4.7 on the facing page).

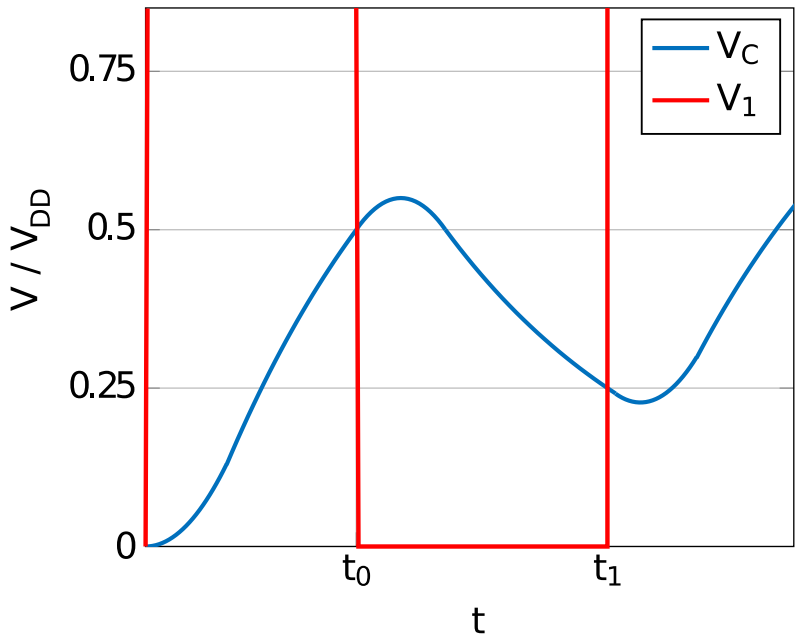


Figure 4.6 – Representation of the driving pulses and simulated behaviour.

From simulations it appears that ideal exponential waveform approximates the simulated one quite well, after the transient due to higher order poles is extinguished. This means that dominant-pole hypothesis is verified.
Being the polysilicon resistor nonlinearities negligible, we can make con-

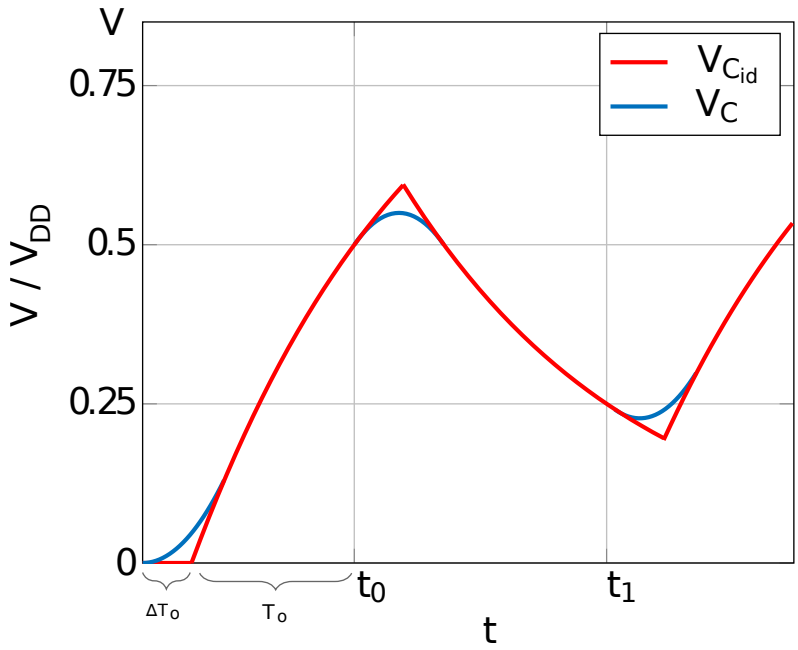


FIGURE 4.7 – Simulated waveform and ideal exponential-fit representation for input code '101'.

siderations on impulse response to get an accurate conversion even if the RC network does not behave as a first order system.
Impulse response of our simulated RC network can be expressed as a multiple pole impulse response:

$$h(t) = a_1 \cdot e^{-t/\tau} + a_2 \cdot e^{-t/\tau_2} + ... + a_n \cdot e^{-t/\tau_n} \tag{4.1}$$

where:

$$h_1(t) = a_1 \cdot e^{-t/\tau}$$

is the ideal RC impulse response (having time constant τ) and:

$$h_\epsilon(t) = a_2 \cdot e^{-t/\tau_2} + ... + a_n \cdot e^{-t/\tau_n}$$

represents the contribution of higher order poles to circuit impulse response. If time constants relative to $h_\epsilon(t)$ are smaller enough with respect to τ, after some delay the simulated impulse response converges to the ideal one (see Figure 4.8 on the next page).

Since we are mainly interested on the capacitor voltage in the end of conversion, we can evaluate the convolution integral between the input

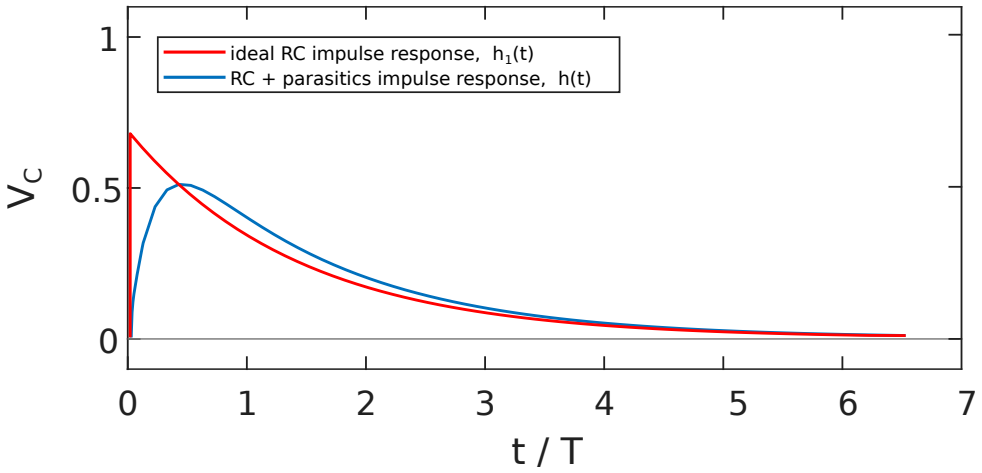


FIGURE 4.8 – Single and multiple poles impulse response representation in Matlab.

train of rectangular pulses $x(t)$ and the impulse response $h(t)$:

$$y(t) = (x * h)(t) = \int_{-\infty}^{+\infty} x(\lambda) \cdot h(t-\lambda) d\lambda \tag{4.2}$$

where $y(t)$ is the voltage waveform across the capacitor and input $x(t)$ is expressed as:

$$x(t) = V_{DD} \cdot [b_0 \cdot r(t) + b_1 \cdot r(t-T) + ... + b_{N-1} \cdot r(t-(N-1)T]$$

$$r(t) = \begin{cases} 0 & t < 0 \\ 1 & 0 \leq t \leq T \\ 0 & t > T \end{cases}$$

where $r(t)$ is the rectangular function of width T.
If we now evaluate the voltage across capacitor at the end of conversion, i.e. $y(t)$ for $t = NT$ we get:

$$\begin{aligned} y(NT) &= \int_{-\infty}^{+\infty} x(\lambda) \cdot h(NT-\lambda) d\lambda \\ &= \int_{-\infty}^{+\infty} x(\lambda) \cdot [h_1(NT-\lambda) + h_\epsilon(NT-\lambda)] d\lambda \\ &= y_{id}(NT) + \int_{-\infty}^{+\infty} x(\lambda) \cdot h_\epsilon(NT-\lambda) d\lambda \end{aligned} \tag{4.3}$$

$y_{id}(NT)$ is the ideal converted voltage, while integral term is the error due to higher frequency poles. Provided that all time constants inside $h_\epsilon(t)$ are smaller enough with respect to τ, (4.3) can be evaluated a time T_{del} after the MSB period, with T_{del} large enough so that all unwanted exponential terms are practically expired:

$$\begin{aligned} y(NT + T_{del}) &= y_{id}(NT + T_{del}) + \int_{-\infty}^{+\infty} x(\lambda) \cdot h_\epsilon(NT + T_{del} - \lambda) d\lambda \\ &\approx y_{id}(NT + T_{del}) \end{aligned} \tag{4.4}$$

A representation of the error term in the convolution integral is visible in Figure 4.9. This conversion strategy allows to compensate for the non ideal RC behaviour, with a tiny attenuation of $y_{id}(NT + T_{del})$ with respect to the ideal converted voltage $y_{id}(NT)$.
In practice, the compensation works only if input voltage is null for $t > NT$, i.e. if an extra zero bit is shifted after the MSB and capacitor voltage is sampled with a delay T_{del}.

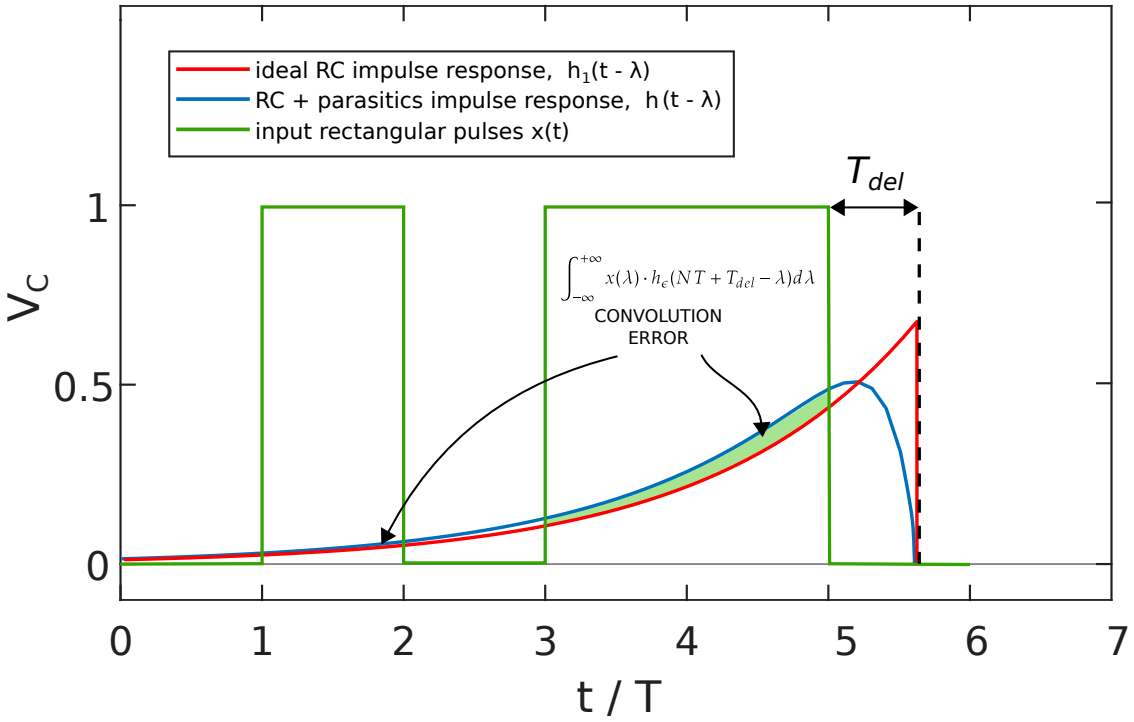


Figure 4.9 – Convolution error representation.

Transistor	(W / L)
M_1 (LVT, SVT)	$930nm/40nm$
M_2 (LVT, SVT)	$2.46\mu m/40nm$
M_1 (HVT)	$1.55\mu m/40nm$
M_2 (HVT)	$4.1\mu m/40nm$
M_3	$410nm/40nm$
M_4	$410nm/40nm$
M_5	$310nm/40nm$
M_6	$410nm/40nm$
M_7	$310nm/40nm$
M_8	$310nm/40nm$

Table 4.1 – Buffer transistors aspect ratio.

4.2.2 Buffer design

As we saw in the DAC working principle presentation, the three-state buffer (or an equivalent functioning block) is used to drive the RC network and keep constant capacitor voltage during hold phase. To design the tristate output buffer attention has been paid to three main aspects which can affect converter nonlinearity: transistor triode resistance, charge injection and leakage.
To this purpose a three-state buffer with no stacked transistors as final stage has been searched, and the configuration in Figure 4.10 on the next page has been initially chosen. This choice enables to separate the two final transistors from the rest of buffer (as it has be done in simulation setup, see Figure 4.1 page 73) reducing channel resistance when on, and allows to take into account for the most nonidealities by extracting layout parasitics for just the two final transistors. This choice is in view of directly driving them from digital logic. All transistors except for the final stage have a form factor equal to the minimum digital inverter size. The complete schematic setup employed in simulations where buffer is present has been shown in Figure 4.5 page 76. As we mentioned, RC-driving transistors have been designed to reduce the effects of nonidealities:

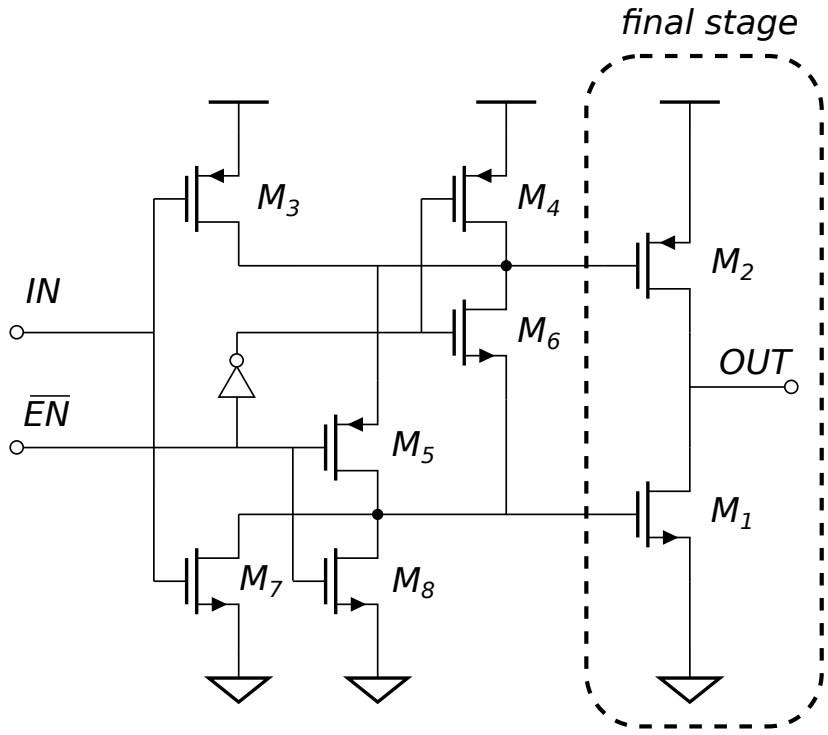


FIGURE 4.10 – Tristate buffer schematic.

Transistor triode resistance Low threshold voltage (LVT) transistors have been initially chosen and sized so that the nonlinear behaviour introduced by triode resistance is negligible, given the target converter accuracy, with respect to nominal RC resistor value, according to what we found in (3.29):

$$\frac{\Delta R}{R} < -\left(\frac{log(2)}{log(\frac{1+2^N}{2^{N+1}})} + 1\right)$$

Based on this, transistors have been sized of minimum length ($40n$m) to minimize capacitances and a width such that to have a triode resistance of less than $0.5k\Omega$ (see Table 4.1 on the facing page) keeping n and p transistor triode resistances symmetric as possible.

Transistor leakage To take into account for the leakage error during hold phase, final stage has been sized with standard threshold voltage (SVT) and high threshold voltage (HVT) transistors as well, following the guidelines described above.

SVT and HVT final stages have been respectively substituted to LVT on the need, when simulated leakage current during hold phase caused an error greater than half LSB related to the target accuracy. Type of transistors suitable for each configuration are reported in Table 4.7 page 96, while layout for the LVT final stage transistors is visible in Figure 4.16 page 94.

Charge injection As supposed in Section 3.4 page 52, for the small transistor sizes involved, gate-drain capacitances and diffusion ones are quite small. Thanks also to the filtering of the RC network, charge injection effect on output capacitor results to be negligible after simulations.

# config.	T	R	C	V_{DD}
1	$20ns$	$144k\Omega$	$200f$F	1.1V
2	$40ns$	$128k\Omega$	$450f$F	0.6V
3	$70ns$	$224k\Omega$	$450f$F	1.1V
4	$100ns$	$480k\Omega$	$300f$F	0.8V
5	$200ns$	$288k\Omega$	$1p$F	0.6V
6	$500ns$	$1M\Omega$	$700f$F	0.6V
7	$1us$	$202k\Omega$	$7.13p$F	1.1V

Table 4.2 – Passives configurations.

4.3 Test Cases and Simulation Results

As described in the design section, different combinations of R, C and T values have been chosen for Cadence simulations, having different supply voltages. Initial test cases are reported in Table 4.2.
For typical process, mismatch and temperature conditions different analysis have been carried on:

1. Integral nonlinearity of the whole swing, using schematic model of the RC network driven by an ideal voltage source (schematic setup of Figure 4.4 page 75).
2. Integral nonlinearity of the whole swing, using extracted parasitics layout model of the RC network driven by an ideal voltage source (schematic setup of Figure 4.4 page 75).
3. Integral nonlinearity of the whole swing, using extracted parasitics layout model for both RC network and its driving buffer (schematic setup of Figure 4.5 page 76).

Montecarlo analyses have been performed as well, by sweeping temperature and re-calibrating one sampled case to re-establish accuracy comparable to the nominal case.

4.3.1 Typical Conditions

4.3.1.1 Schematic level simulations

Fixed T compensation As a first compensation choice we can set a clock period which minimizes integral error in the middle of the swing, since, as shown in Section 3.2 page 37, that is the point of swing where integral error should be maximum.
The period which satisfies the null error at half swing condition, turns to be the one we called $T = T_o + \Delta T_o$ in Figure 4.7 page 79, obtained from schematic-level simulations of the passive networks. The period values which minimize integral error at half the swing are listed in Table 4.4 page 95 for each configuration.
The minimum integral error is reached only for config. #7, in which output capacitor is much bigger than distributed resistor capacitance. For all other cases integral error does not result to be minimized with this compensation strategy (as it would be for an ideal case) and follows a trend like the one in Figure 4.11 on the next page.
The shown pattern repeats twice the one predicted analytically for positive time error on clock period (see Figure 3.9(d) page 36) in the ideal case.
The positive error pattern is explained by thinking about the multiple time constant behaviour of our RC network. Due to delay introduced by parasitics, a longer period $T_o + \Delta T_o$ is needed for the multiple time constant response to reach exactly half-swing with respect to the ideal exponential response period T_o.
The duration of ΔT_o (delay ideal exponential must have to follow simulated behaviour) has been retrieved from schematic-level simulations for each configuration and it is reported in Table 4.3 on the next page. ΔT_o depends mainly on distributed capacitance of resistors, and thus increases quadratically with resistance length (having all resistors the same minimum width). The quadratic behaviour is emphasized in Figure 4.12 page 88.

# config.	T	R	ΔT_o
1	$20ns$	$144k\Omega$	$0.3ns$
2	$40ns$	$128k\Omega$	$0.24ns$
3	$70ns$	$224k\Omega$	$0.7ns$
4	$100ns$	$480k\Omega$	$3ns$
5	$200ns$	$288k\Omega$	$1.15ns$
6	$500ns$	$1M\Omega$	$14.4ns$
7	$1us$	$202k\Omega$	$0.5ns$

TABLE 4.3 – ΔT_o value for each configuration.

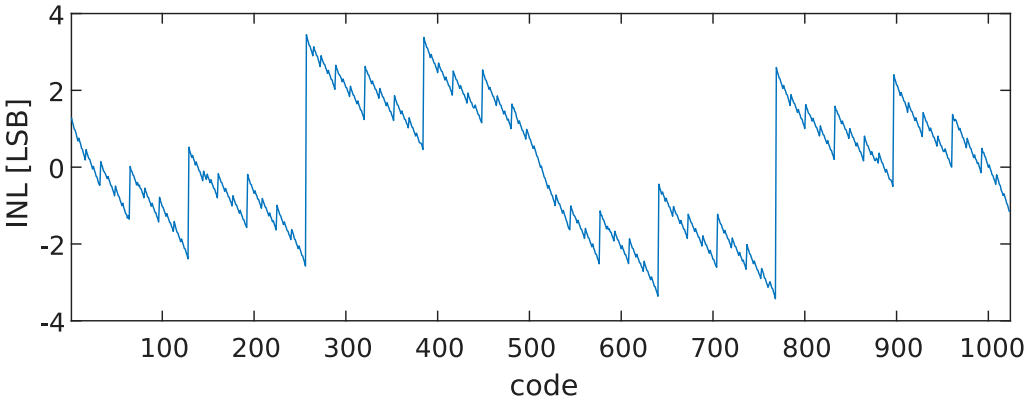


FIGURE 4.11 – INL for config. #1 when T is sized to null integral error at half the swing. ($T = T_o + \Delta T_o = 18.75ns$).

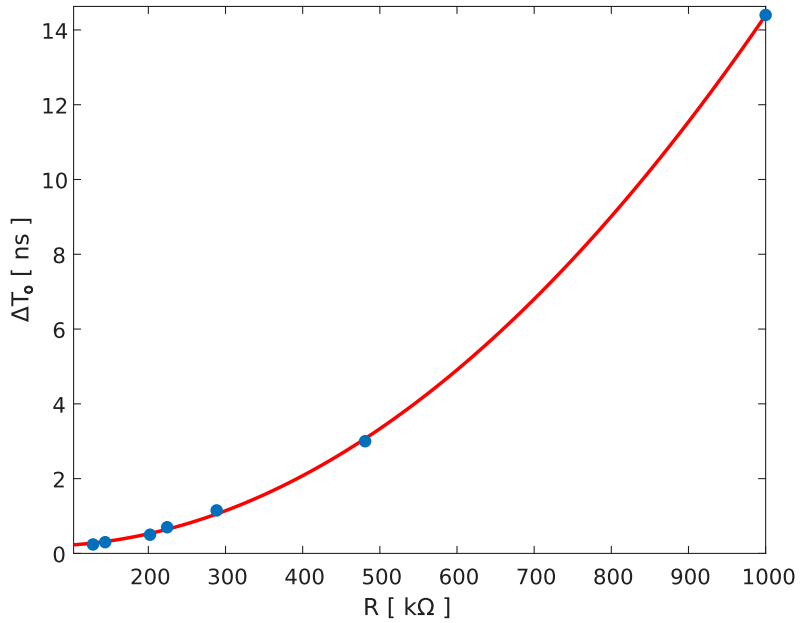
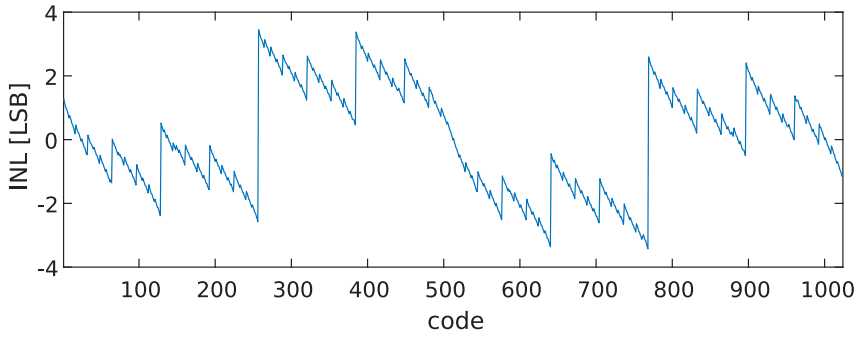


FIGURE 4.12 – Qadratic dependence of ΔT_o delay on resistance.

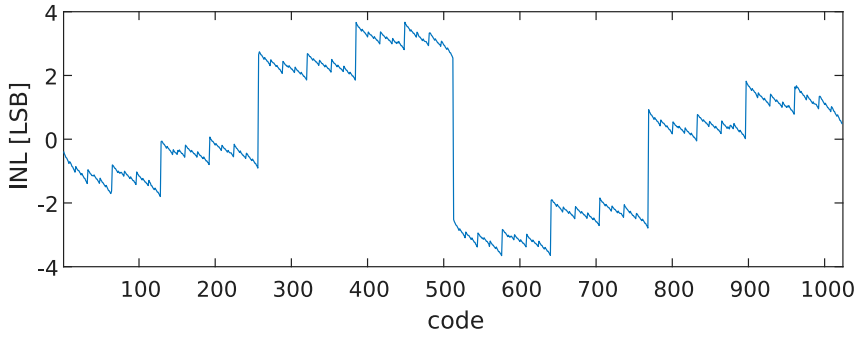
T plus T_{del} compensation If we want to minimize integral error on the whole swing for configurations having non-negligible parasitics, the compensation technique detailed in Section 4.2.1 page 81 must be used.
Clock period has to be reduced to $T = T_o$, i.e. the one related to dominant time constant τ. This process is shown in Figure 4.13 page 90 and it is equivalent of minimizing integral error at $1/4$ and $3/4$ of the full scale range. The negative half-swing step visible in the integral error of Figure 4.13(d) is due to the difference between the multiple time constant impulse response $h(t)$ and the ideal one $h_1(t)$, which is negative and quite relevant near the origin (see Figure 4.8 page 80).
Therefore, for the upper-half swing, having 1 as MSB implies a huge error on convolution integral with respect to the ideal one, while it is negligible for the lower-half swing (0 MSB implies that input $x(t)$ is zero at least by a time T_o when voltage is sampled). Last zero bit is therefore shifted and the signal is sampled a time T_{del} after. Error attenuation due to T_{del} is shown in Figure 4.14 page 91.
T plus T_{del} compensation has been performed for all the chosen configurations, and obtained accuracies are reported in Table 4.5 page 95

Few observations can be made on these results:

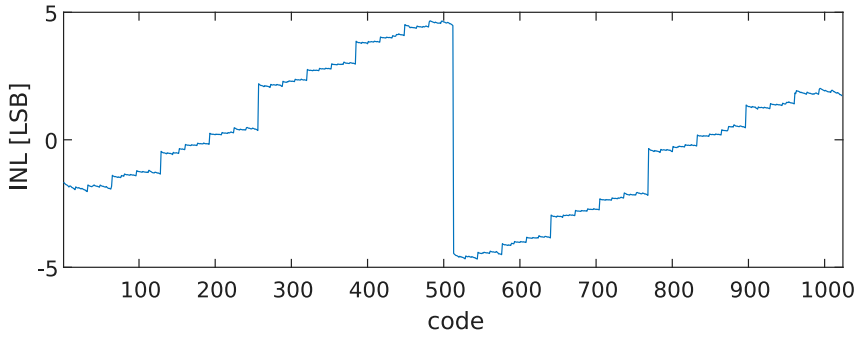
- Accuracies obtained by *fixed T compensation* in the middle of the swing are quite well predicted by reverting relation (3.19), replacing $\Delta T = \Delta T_o$, where values of ΔT_o are the ones reported in Table 4.3 page 87.
- Accuracies strongly improve (3 bits on average) with *T plus* T_{del} *compensation*. The only drawback of this approach is that attenuation due to the extra 0 bit reduces converter swing by some amount ptoportional to T_{del}, which must be as small as possible.



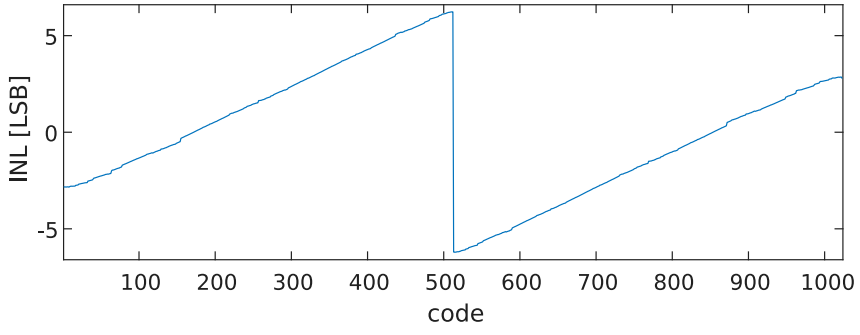
(a) INL for config. #1 when $T = 18.77ns$



(b) INL for config. #1 when $T = 18.65ns$

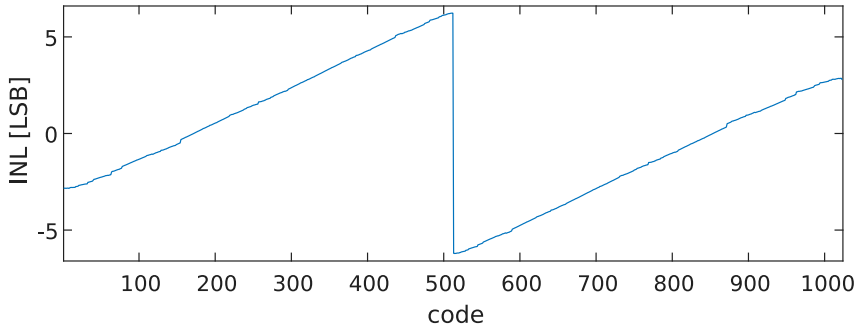


(c) INL for config. #1 when $T = 18.55ns$

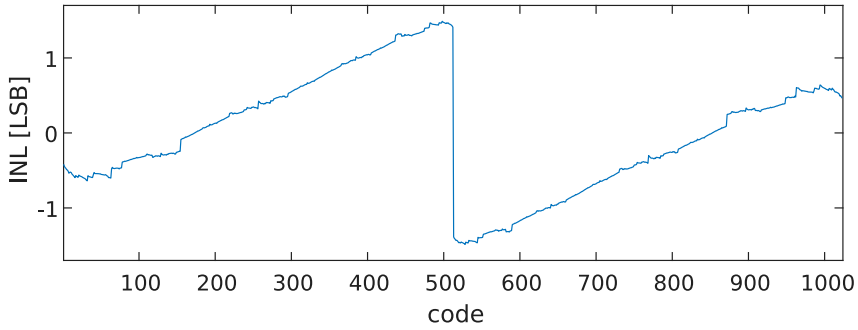


(d) INL for config. #1 when $T = 18.46ns$

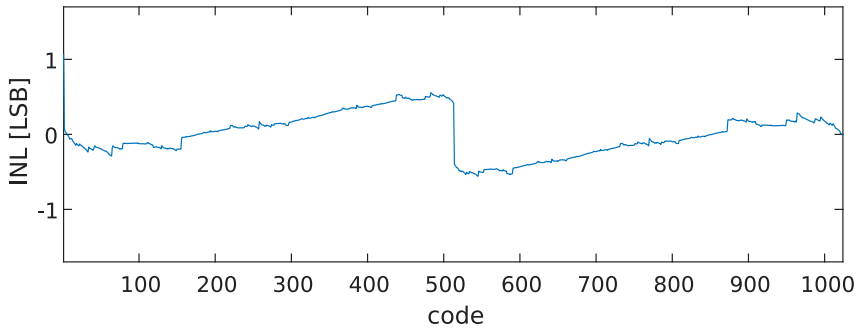
Figure 4.13 – Varying INL for config. #1 to reach minimum nonlinearity on half swing.



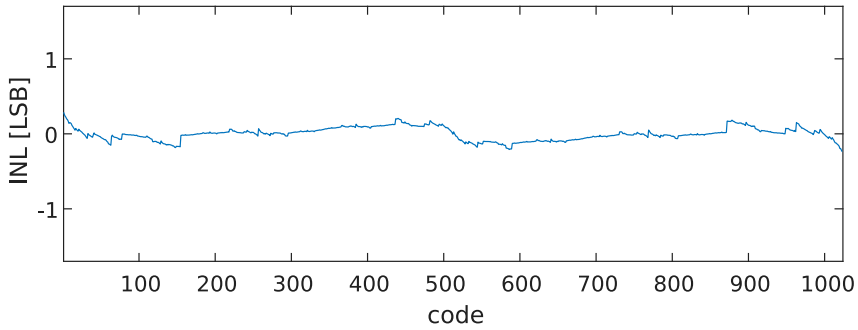
(a) INL for config. #1 when $T = 18.46ns$ and $T_{del} = 0ns$.



(b) INL for config. #1 when $T = 18.46ns$ and $T_{del} = 0.3ns$



(c) INL for config. #1 when $T = 18.46ns$ and $T_{del} = 0.5ns$



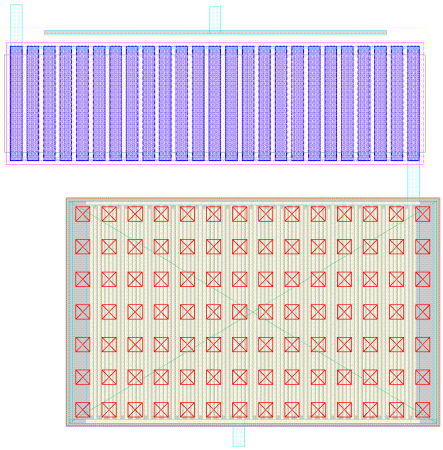
(d) INL for config. #1 when $T = 18.46ns$ and $T_{del} = 2.24ns$

FIGURE 4.14 – Increasing delay T_{del} in config. #1 to reach minimum nonlinearity by attenuating half-swing step.

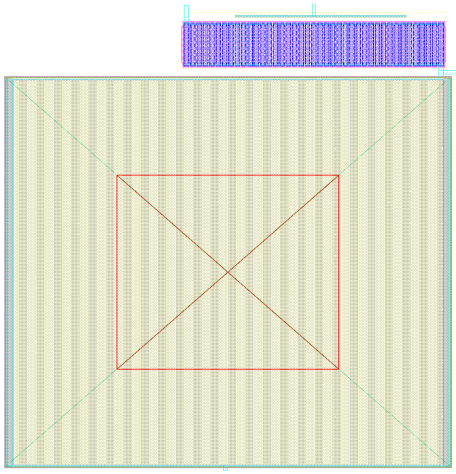
4.3.1.2 Layout simulations

To better take into account for parasitics, layout has been drawn for all RC configurations. Some of them are reported in Figure 4.15 on the next page. The parasitics have been extracted from all drawn layouts, and simulation have been performed with setup in Figure 4.4 page 75 to retrieve plots of integral errors. No significant change in accuracy can be appreciated for the *fixed T compensation* case. The slight accuracy changes in the *T plus* T_{del} compensation case are reported in Table 4.6 page 96, where maximum INL and equivalent number of bits is reported. Parasitics related to layout result to have thus little influence on converter accuracy.
Four configurations among the ones with higher accuracy have been chosen, to be representative of different conversion rates and power dissipation.

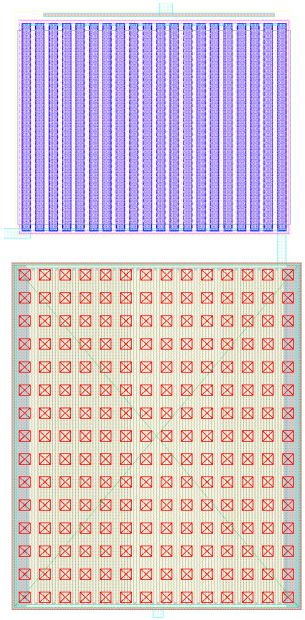
Chosen configurations are the ones having period of $40ns$, $200ns$, $500ns$ and $1\mu s$. Layout has been drawn up for LVT, SVT and HVT buffer final stages, and the LVT one is shown in Figure 4.16 page 94. Cadence simulations have been performed to retrieve static accuracy for the chosen configurations, using extracted layout models of both RCs and final stages in test schematic of Figure 4.5 page 76.
Obtained accuracies for the chosen configurations are reported in Table 4.7 page 96. From simulation results it can can be observed that accuracy loss due to buffer insertion (compared to the ideally driven RC network) is proportional to the amount of voltage error produced by leakage, which plays a dominant role over charge injection and channel resistance nonlinearity.



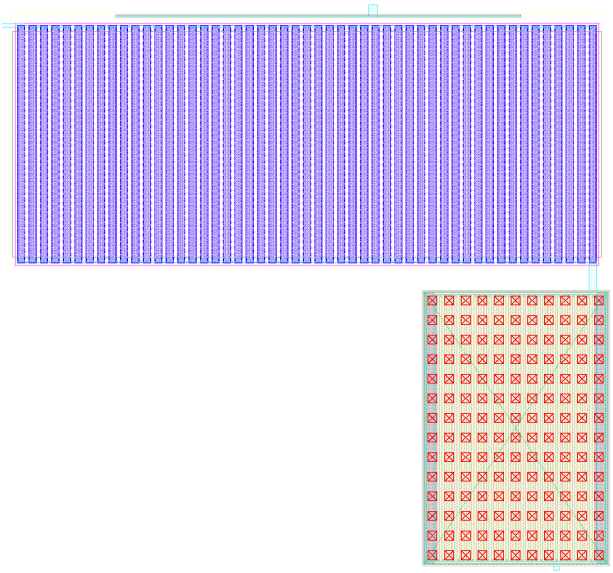
(a) RC layout for config. #2. Layout size: $16\mu\text{m} \cdot 14\mu\text{m}$.



(b) RC layout for config. #7. Layout size: $42\mu\text{m} \cdot 41\mu\text{m}$.



(c) RC layout for config. #5. Layout size: $14\mu\text{m} \cdot 29\mu\text{m}$.



(d) RC layout for config. #6. Layout size: $34\mu\text{m} \cdot 14\mu\text{m} + 10\mu\text{m} \cdot 16\mu\text{m}$.

FIGURE 4.15 – Layouts for configurations #2, #5 and #7.

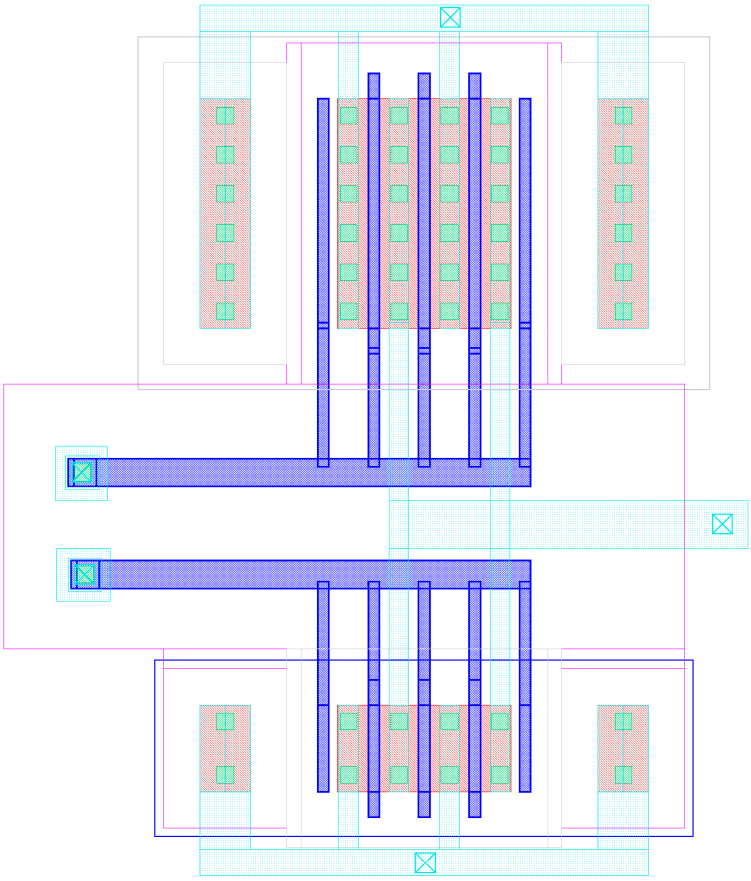


Figure 4.16 – Layout of the buffer final stage, LVT version.

# config.	T_{nom}	T	INL_{max}	N_{eff}
1	$20ns$	$18.75ns$	2.96 LSB	7bit
2	$40ns$	$39.83ns$	1.76 LSB	8bit
3	$70ns$	$70.42ns$	1.6 LSB	8bit
4	$100ns$	$99.73ns$	5.5 LSB	6bit
5	$200ns$	$202ns$	1.58 LSB	8bit
6	$500ns$	$497ns$	3.96 LSB	7bit
7	$1us$	$998ns$	0.24 LSB	11bit

TABLE 4.4 – Fixed T compensation: nominal period, period duration which nulls integral error at half swing and corresponding accuracy expressed as INL on 10 bits and effective bits.

# config.	T_{nom}	T	T_{del}	INL_{max}	N_{eff}
1	$20ns$	$18.46ns$	$2.24ns$	0.26 LSB	11bit
2	$40ns$	$39.61ns$	$0.5ns$	0.34 LSB	10bit
3	$70ns$	$69.67ns$	$11.8ns$	0.36 LSB	10bit
4	$100ns$	$96.67ns$	$8.33ns$	0.49 LSB	10bit
5	$200ns$	$200.7ns$	$2ns$	0.37 LSB	10bit
6	$500ns$	$485.3ns$	$85.7ns$	0.37 LSB	10bit
7	$1us$	$998ns$	N/A	0.24 LSB	11bit

TABLE 4.5 – T plus T_{del} compensation for schematic-level simulations of ideally driven RC network: accuracy expressed as INL on 10 bits and effective bits.

# config.	T_{nom}	T	T_{del}	INL_{max}	N_{eff}
1	$20ns$	$18.62ns$	$1.3ns$	0.3 LSB	10bit
2	$40ns$	$40.17ns$	$2ns$	0.31 LSB	10bit
3	$70ns$	$70.21ns$	$2ns$	0.29 LSB	10bit
4	$100ns$	$97ns$	$9ns$	0.74 LSB	9bit
5	$200ns$	$201.45ns$	$3ns$	0.38 LSB	10bit
6	$500ns$	$487.4ns$	$37ns$	0.29 LSB	10bit
7	$1us$	$1007.9ns$	N/A	0.24 LSB	11bit

TABLE 4.6 – *T plus T_{del} compensation* results for each configuration after layout parasitics extraction of the ideally-driven RC network.

# config.	T	Transistor type	Leakage error	T_{del}	INL_{max}	N_{eff}	V_{DD}
2	$41.52ns$	SVT	$762\mu V$	$4ns$	1.3 LSB	9bit	0.6V
5	$203.25ns$	LVT	$195\mu V$	$4ns$	0.3 LSB	10bit	0.6V
6	$487.7ns$	HVT	$46\mu V$	$34ns$	0.2 LSB	11bit	1.1V
7	$1us$	LVT	$177\mu V$	N/A	0.8 LSB	9bit	0.6V

TABLE 4.7 – *T plus T_{del} compensation* parameters for each configuration after layout parasitics extraction of both RC and buffer.

4.3.2 Mismatch and Temperature Variations

To evaluate the effect of mismatch variations on converter nonlinearity, Monte Carlo (MC) analysis has been performed on config. #5 with 200 samples at 27^oC, after parasitic extraction from both buffer and RC blocks. Famlily plot related to INL is visible in Figure 4.17.
It can be noticed that, since mismatch variations mainly result in a change of the time constant τ, nominal accuracy can always be recovered by re-calibrating T, re-enforcing condition (3.4).
This mismatch-insensitive behaviour has been emphasized by comparing MC plot to the INL family plot obtained by uniform distribution of the constant error $\Delta T/T$ inside $\pm 0.5\%$, using corner analysis with typical models (Figure 4.17 and Figure 4.18 on the next page).

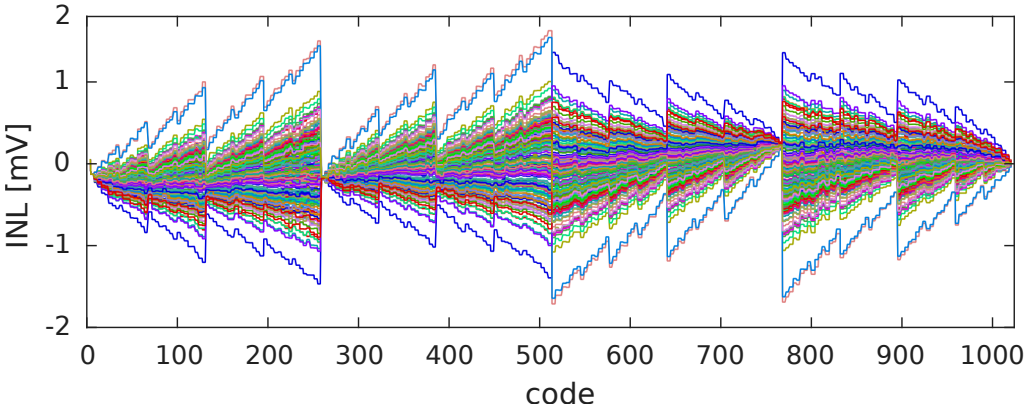


Figure 4.17 – INL relative to 200 samples Monte Carlo analysis, mismatch variations, performed on config. #5. .

For each Monte Carlo INL curve, it can be found a corresponding one from the typical corner simulation whose fixed delay ΔT well approximates the generic MC sample. Mismatch variations have thus the same effect on INL as a Gaussian distributed error on clock period.
The same MC analysis has been repeated by sweeping temperature from -45^oC to 200^oC, in order to evaluate possible effects on static errors. Maximum INL values and corresponding standard deviations resulting from MC simulations are reported as a function of temperature in Figure 4.20 page 99. From the analysis, it can be concluded that an almost-constant accuracy around 10 bits is achieved across a wide temperature range of 10^oC to 125^oC, beyond which the model has not been considered still reliable.
By picking a sample from Monte Carlo analysis performed at a random

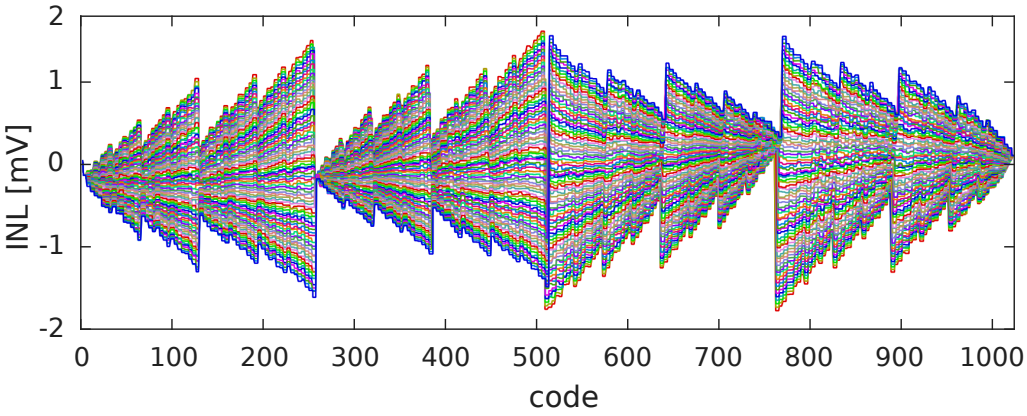


FIGURE 4.18 – INL parametric curves relative to config. #5 with $\Delta T/T$ uniformly distributed in $\pm 0.5\%$.

temperature (85^oC in this case) shown in Figure Figure 4.19, if we minimize integral error at half swing while keeping the nominal T_{del} value, nominal accuracy can be obtained again. The re-calibration process is shown, for the random chosen sample, in Figure 4.21 page 100, where an integral error of less than half millivolt is finally obtained, providing an accuracy of more than 10 bits.

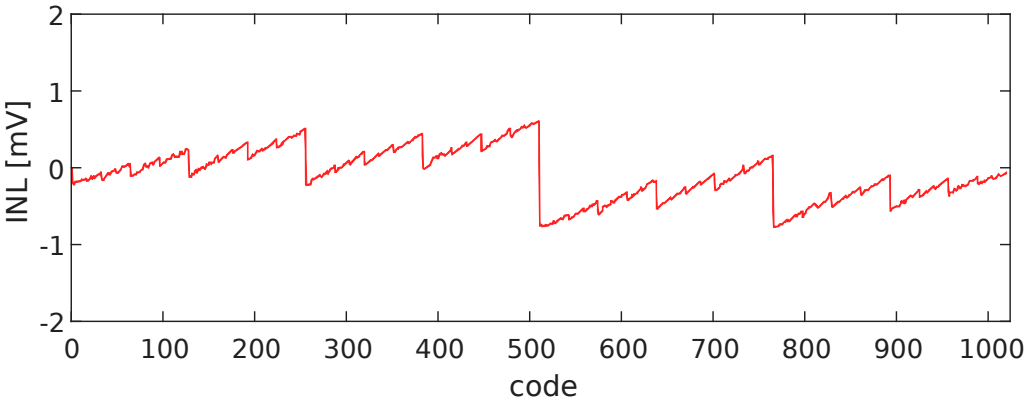


FIGURE 4.19 – INL of a random MC sample on config. #5, nominal period $T = 203.25n$.

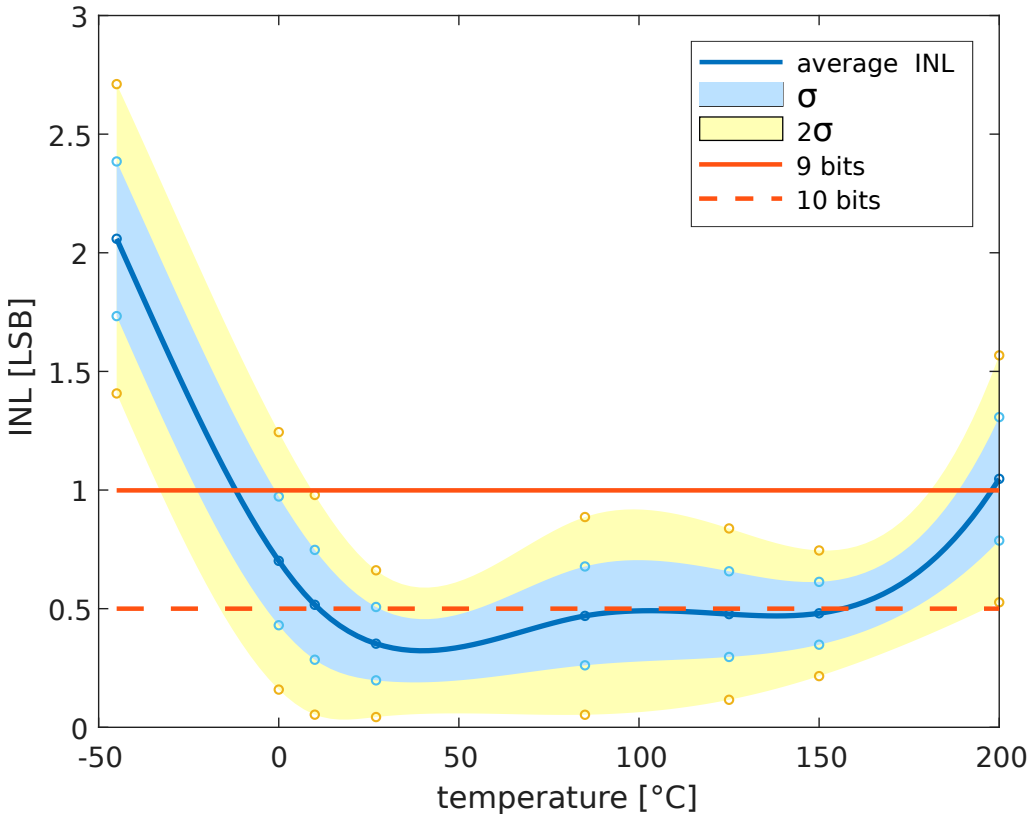
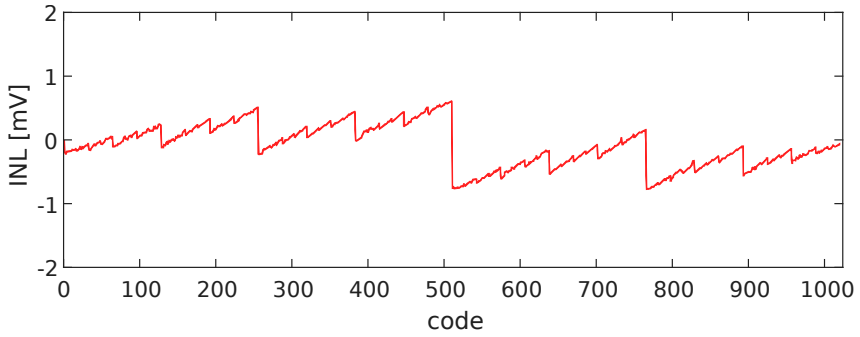
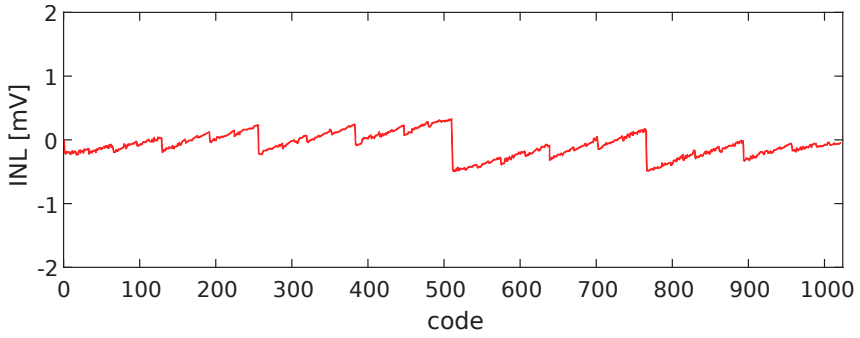


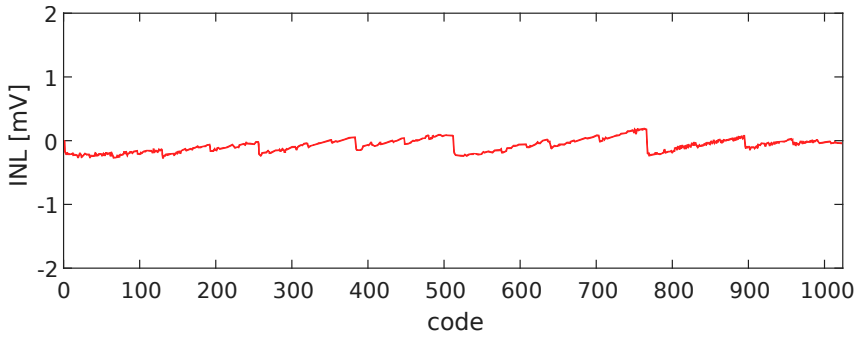
Figure 4.20 – Monte Carlo analysis: average INL and standard deviation as a function of temperature (spline interpolation on 8 points) obtained by simulating config. #5.



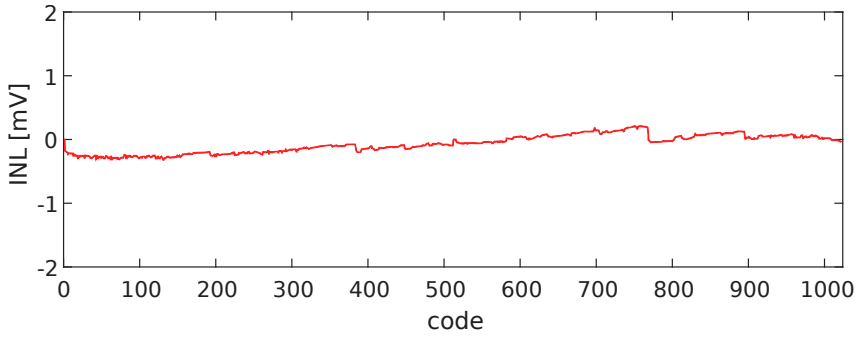
(a) INL of a random MC sample on config. #5, nominal period $T = 203.25n$



(b) INL of the random MC sample of config. #5, period $T = 203.42n$



(c) INL of the random MC sample of config. #5, period $T = 203.57n$



(d) INL of the random MC sample of config. #5, period $T = 203.68n$

Figure 4.21 – Re-calibration of a random chosen Monte Carlo sample.

Chapter 5

Digital Design and Post Layout Verification

In this chapter, the complete design of the Relaxation DAC shift register with its digital control block is described. Transistor level circuit is synthesized from Verilog-described digital blocks and layout is generated. After complete post-layout simulations, the whole Relaxation DAC is characterized, and conclusions on converter performance are drawn-up, along with possible future developments.

5.1 Digital design

Simulations carried in Chapter 4 have been carried on by taking advantage of behavioural Verilog-A blocks (see Appendix A page 119). To provide more realistic results about converter operation, approaching in the same time the circuit fabrication stage, the need to design and synthesize digital blocks allowing DAC functionality arises.
Two core blocks have been implemented which allow the DAC to operate: a modified shift register with enable and parallel load and a control unit providing timing signals. These blocks are shown in Figure 5.1 on the following page.

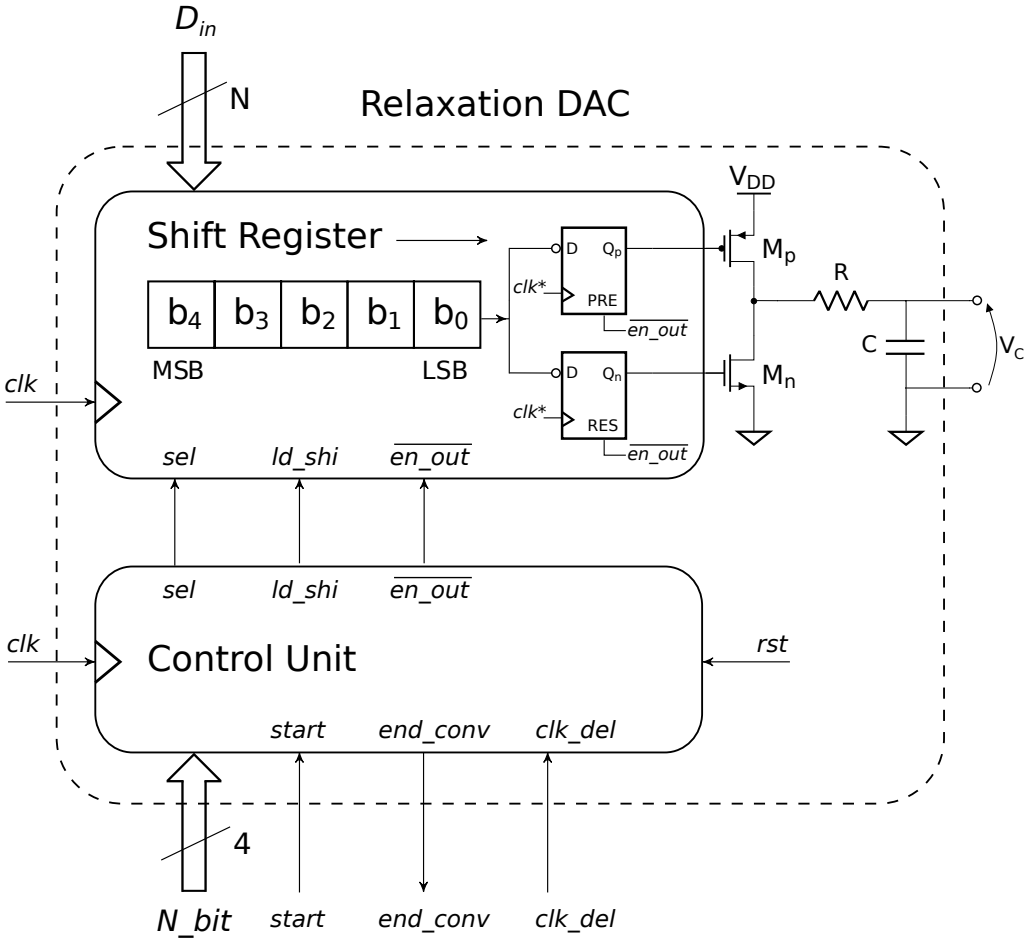


FIGURE 5.1 – Diagram of the digital/analog DAC core blocks.

5.1.1 Shift Register block

As mentioned, previous simulations which included tristate buffer have been performed by separating the two final transistors from the rest of buffer, for a precise purpose. We saw the importance of precise timing in our converter, with special attention to the MSB period duration and delay-to-hold T_{del}, to which converter accuracy is most sensitive.
Given that, in order to avoid unwanted delays and glitches due to asymmetric analog paths, design choice has been to implement a modified shift register version, with two output flip flops, designed to directly drive both n and p final-stage transistors during shift phase, and driving them off in the hold phase, thanks to a preset function triggered by the enabling signal ' $\overline{en\_out}$ '.
Shift register block is also provided with clock gating function for energy saving purposes, so that shift register sensitivity to clock variations is

masked when converter is in hold phase, depending on selection signal *'sel'* (see Figure 5.2).
Shift register structure includes a parallel input *'D_{in}'*, clock signal *'clk'* and signal *'ld_shi'*, which imposes whether data should be loaded into register or shifted.

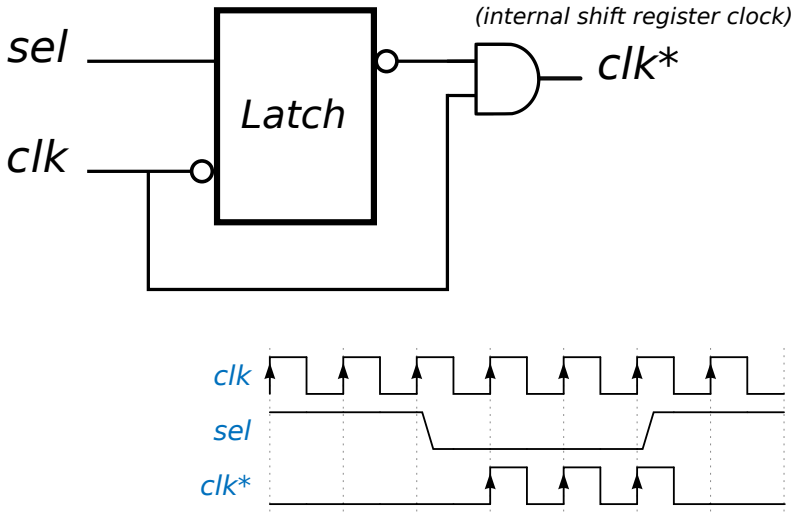


Figure 5.2 – Clock gating implementation.

5.1.2 Control Unit block

Control unit is charged with the task of providing signals with correct timing to the shift register, interfacing meanwhile with the source which provides input digital data flow.
To this purpose a *'start'* signal tells the control unit when first data is available to *'D_{in}'* port to be converted, while an end of conversion *'end_conv'* signal tells the external world, a clock period in advance, when the new digital word will be sampled for conversion. A four bit word *'N_bit'* shows how many bits it is long the input data to be converted, i.e. how many shift operation have to be performed. This choice, introduced for test purposes, also gives the possibility of dynamically change converter resolution from word to word, trading accuracy with conversion speed and then power dissipation (average conversion energy decreases indeed with the number of bits).
Output signals named *'ld_shi'* and *'sel'* lead the load-and-shift operations of the shift register (as previously described) and the clock gating functionality respectively.

Last but not least, ' $\overline{en\_out}$ ' is the signal which goes high a time delay T_{del} after the extra '0' MSB has been shifted out, triggering the hold state. This signal could be driven synchronously by providing the control unit with a high frequency clock and a proper clock divider, to reach the wanted T_{del} duration. However, the high frequency clock solution goes in the opposite design direction with respect to low power targets, so different solutions should be investigated.
Being T_{del} a small fraction of the bit-period T, the idea is to feed the control unit with the main clock *'clk'* at bit-frequency, and a second clock signal *'clk_del'*, ideally obtained from *'clk'* with a tunable delay network, which provides a phase shift proportional to T_{del}.
This *'clk_del'* signal asynchronously presets into hold state the shift register output flip-flops, only when the control unit is transparent for the *'clk_del'* to trigger preset.
Flowgraph and timing diagram for the Control Unit block are shown in Figure 5.3 and Figure 5.4 on the next page respectively.

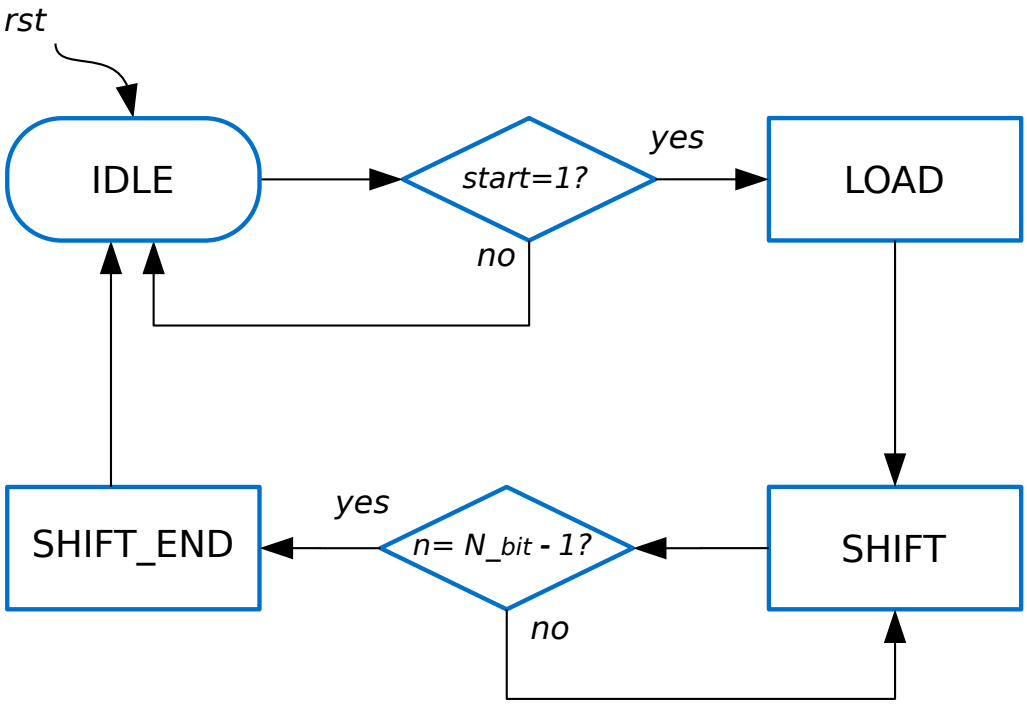


Figure 5.3 – Control Unit flow-graph.

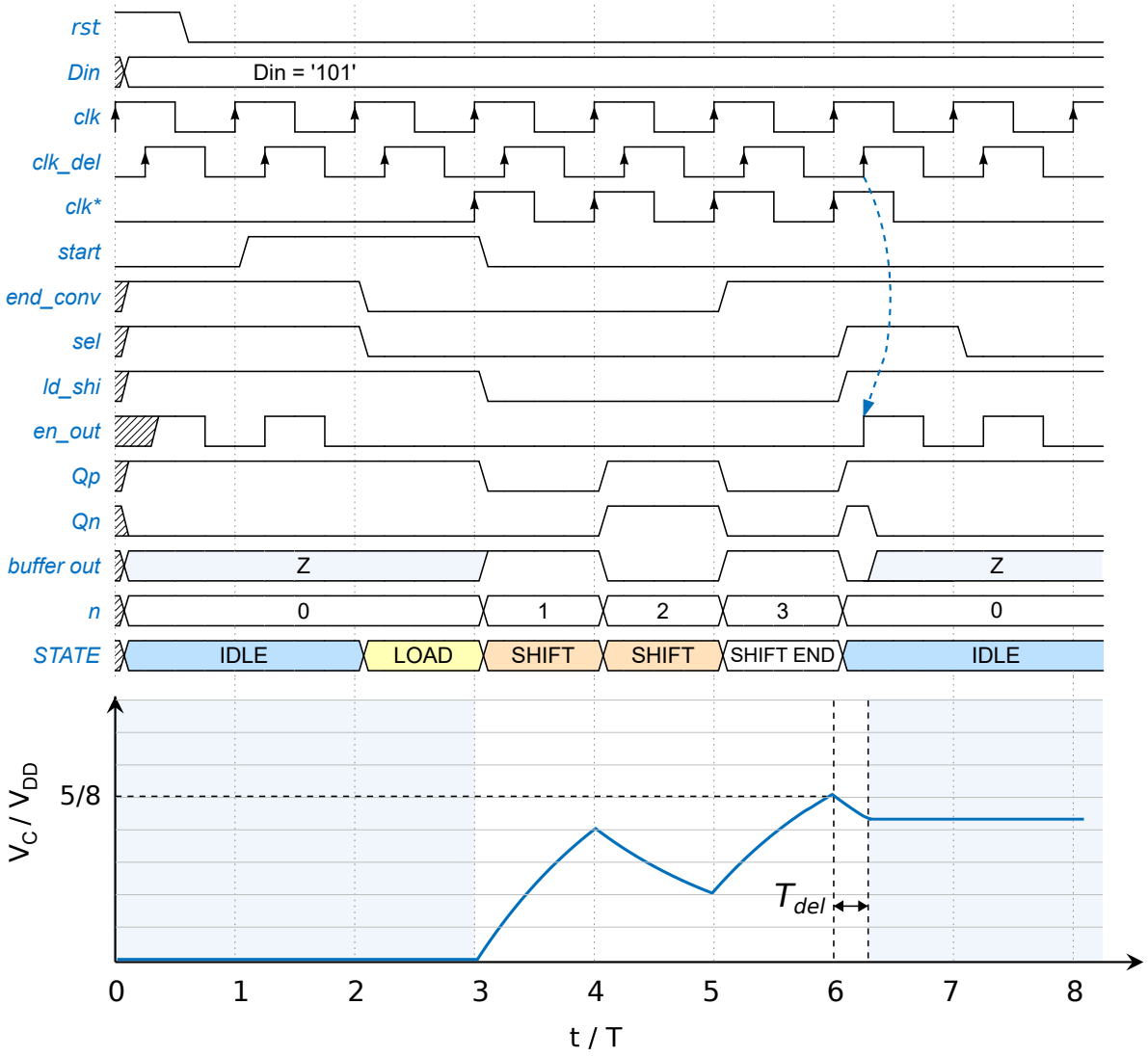


Figure 5.4 – Control Unit timing diagram: conversion of digital code '101'.

5.2 Synthesis and Post Layout simulations

To perform post layout simulations configurations with periods $T = 40ns$ and $T = 200ns$ have been chosen, being the ones with better accuracy at higher sample rates. Digital blocks have been synthesized in $40nm$ technology standard cells, and layouts have been generated through place and route (PnR) flow. Analog and digital blocks have been put all together and parasitics have been extracted to perform post layout simulations, using setup in figure Figure 5.5 page 108. The full layout is visible for the $T = 200ns$ configuration in Figure 5.6 page 109, where the two square blocks are Shift Register and Control Unit (top and bottom layouts respectively).
In order to perform simulations over the whole swing, a Verilog-A block generating digital words corresponding to a ramp has been implemented (*'rampVa'* block in Figure 5.5 page 108).
Static performance related to the chosen configurations are reported in Table 5.1 page 114 while plots relative to INL and DNL. are shown in Figure 5.7 page 110. The curved INL trend endorses the hypothesis that, after calibration, leakage contribution is the most relevant to residual nonlinearity, since it depends on the difference in capacitor voltage with respect to power supplies, and seems consistent with leakage amounts reported in Table 4.7 page 96.
The overall energy per conversion and the partial contributions of analog and digital blocks are shown in Figure 5.8 page 111 and Figure 5.9 page 112.
The analog conversion energy shown in figure reveals a lower average energy with respect to the theoretical one (Figure 3.30 page 64), and more similar to the case in which voltage is reset to V_{DD} for the upper-half swing (see Figure 3.33 page 67). This is due to the fact that, differently to what it has been supposed in theoretical analysis, here voltage is not reset to zero after conversion, but the new conversion directly begins. In a similar way, charge was not wasted in the end of conversion in the V_{DD}-reset strategy. This energy saving implies no side effect on nonlinearity, since initial voltage error is attenuated by a factor 2^{N+1} on the final converted voltage.
For what concerns energy dissipation in digital blocks, it is higher than the analog one for both test cases, and it has been verified that leakage energy is negligible with respect to the dynamic one.
Spurious Free Dynamic Range (SFDR) has been evaluated by re-introducing the Verilog-A *'ADC_10'* block and feeding the DAC with a digitized version of a $4k$Hz and $20k$Hz single tone signals for config. #2 and config.

#5 respectively, covering 90% of the swing. Resulting SFDR for both configurations is visible in Figure 5.10 page 113.
Using the same test configuration Signal to Noise and Distortion Ratio (SNDR), Effective Number of Bits (ENOB) and Total Harmonic Distortion (THD) have been evaluated from the spectrum as well.
Energy figure of merit (FOM), expressed as energy per conversion over 2^{ENOB}, is evaluated for the two configurations as well, and reported along with dynamic measurements in Table 5.2 page 114.

State of the Art comparison Post-layout simulated performance of the two configurations have been compared with those of the FPGA Relaxation DAC (RDAC) prototype measurements, already published in [22], and the capacitive DAC of the SAR ADC in [23], which performance have been estimated from the ADC characterization.
As expected, the post-layout simulated RDAC achieves much higher conversion rates and better performance for both static and dynamic accuracy. Compared to capacitor DAC in [23], the two integrated RDAC configurations occupy a normalised area with respect to technology feature size of less than $0.5 \cdot 10^6$, which is 10 times lower than the $5.03 \cdot 10^6$ of [23], and an energy FOM of $1.1fJ/c.step$, 2.3 times less than the $2.4fJ/c.step$ of [23], which is supposed to be even improved by optimizing digital blocks for low power consumption.
Comparison of the four converters is highlighted in Table 5.3 page 115.

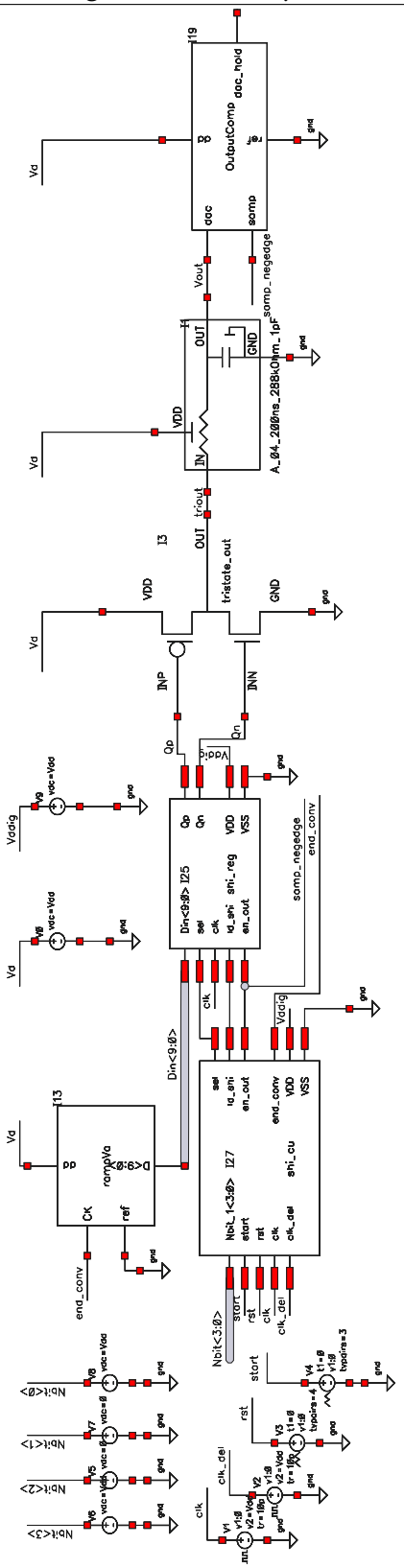


FIGURE 5.5 – Simulation setup for swing characterization, in post layout simulations.

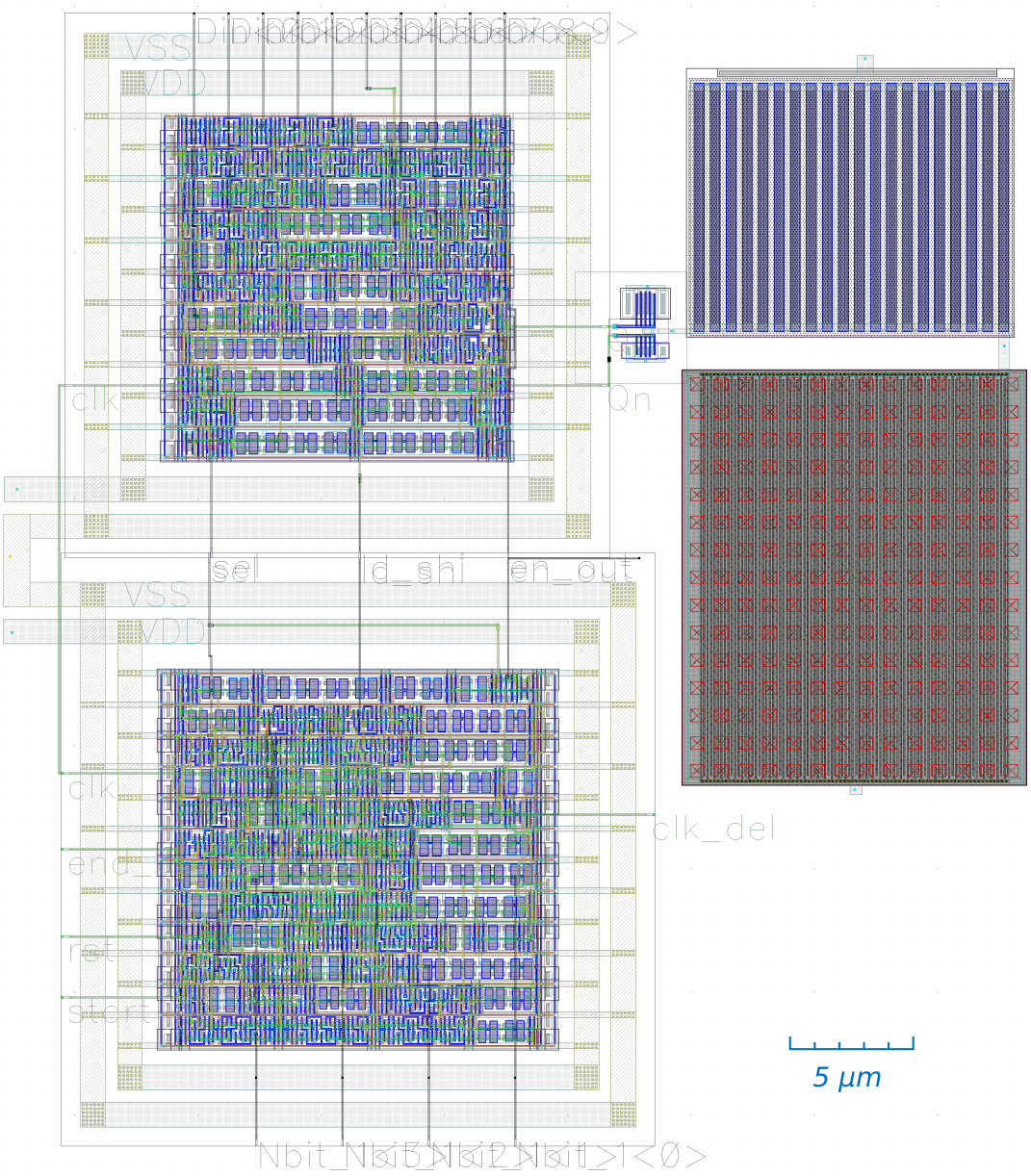
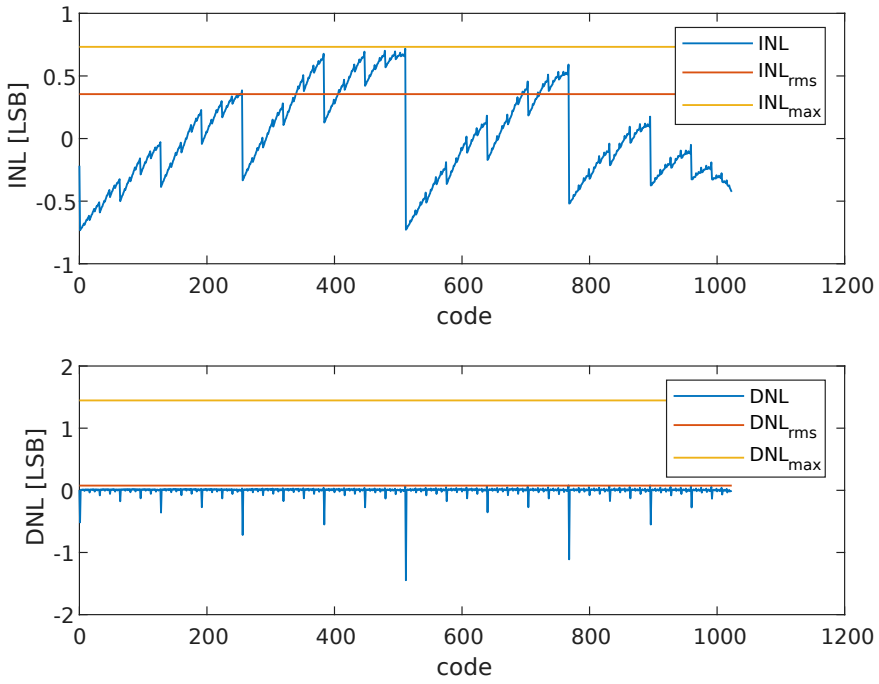
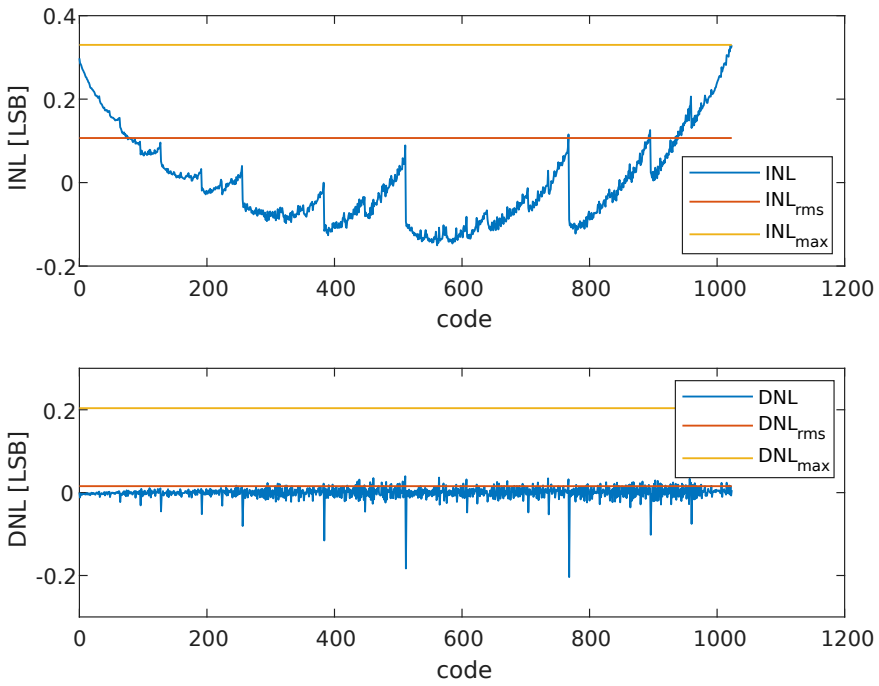


Figure 5.6 – Complete Relaxation DAC layout for the $T = 200ns$ configuration. Overall area is $910\mu m^2$.



(a) Post layout, post calibration nonlinear errors for $T = 40ns$ configuration.



(b) Post layout, post calibration nonlinear errors for $T = 200ns$ configuration.

FIGURE 5.7 – Static errors for the two chosen configurations.

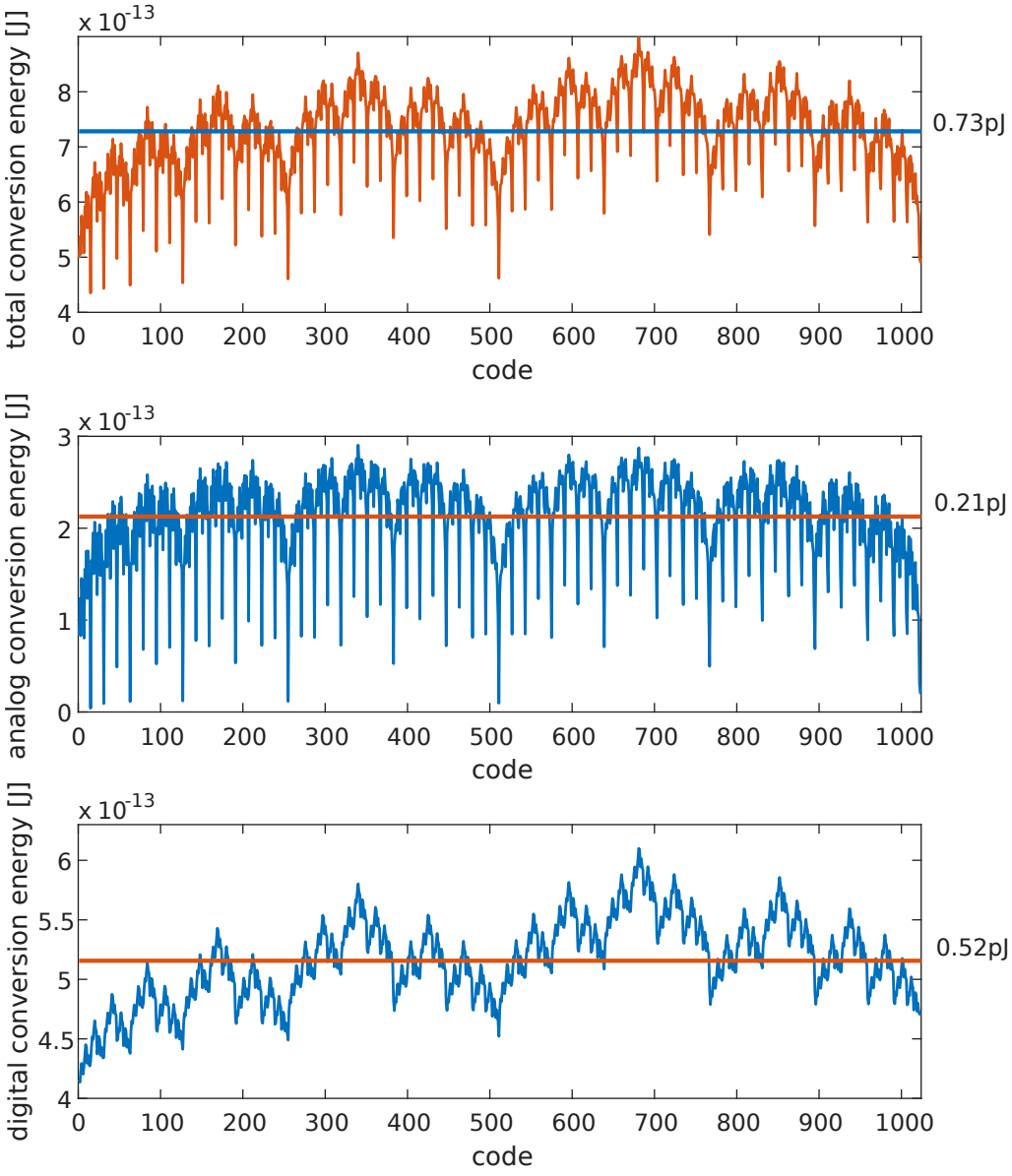


FIGURE 5.8 – Total and partial conversion energies for $T = 40ns$ configuration in post layout simulations.

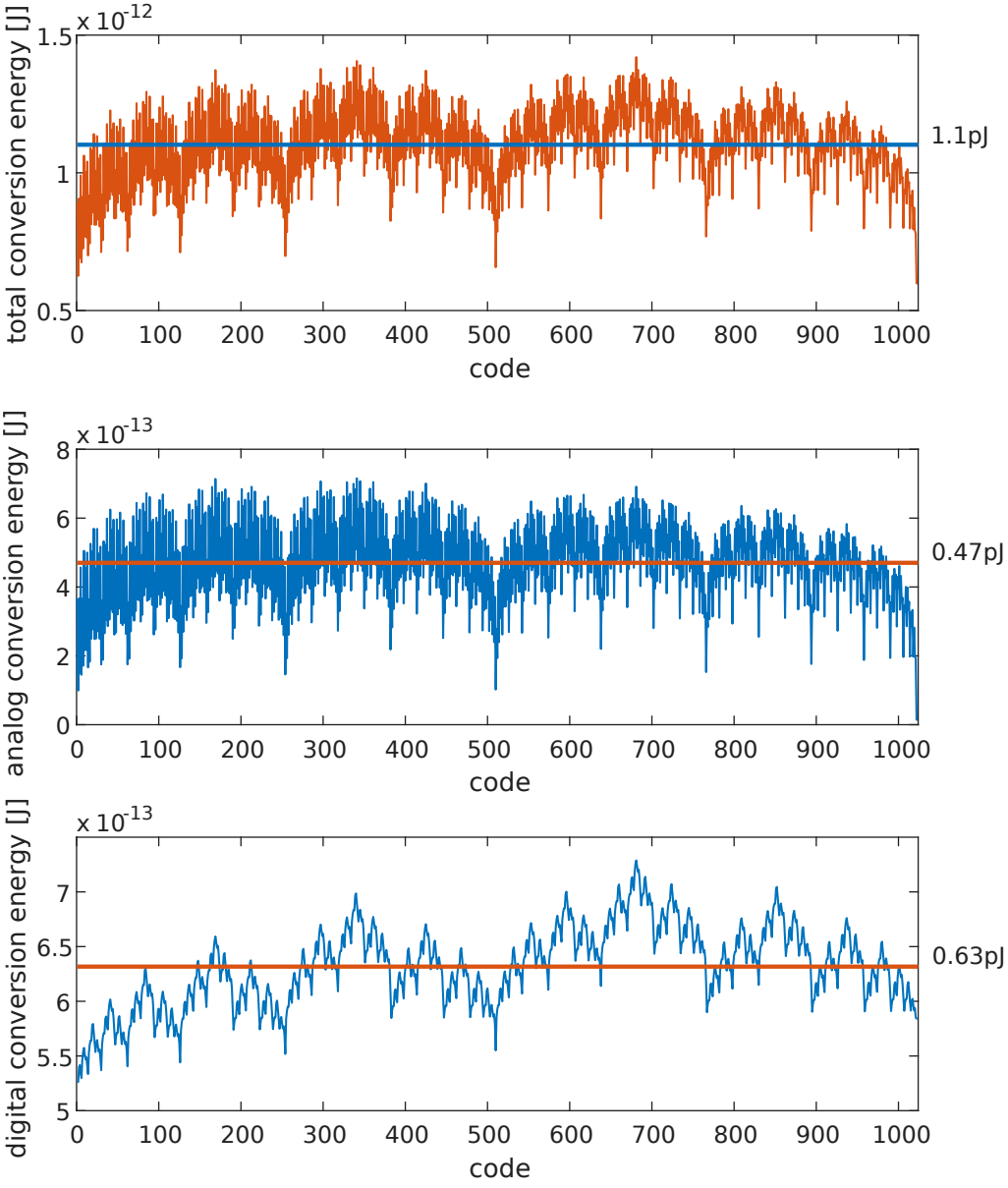
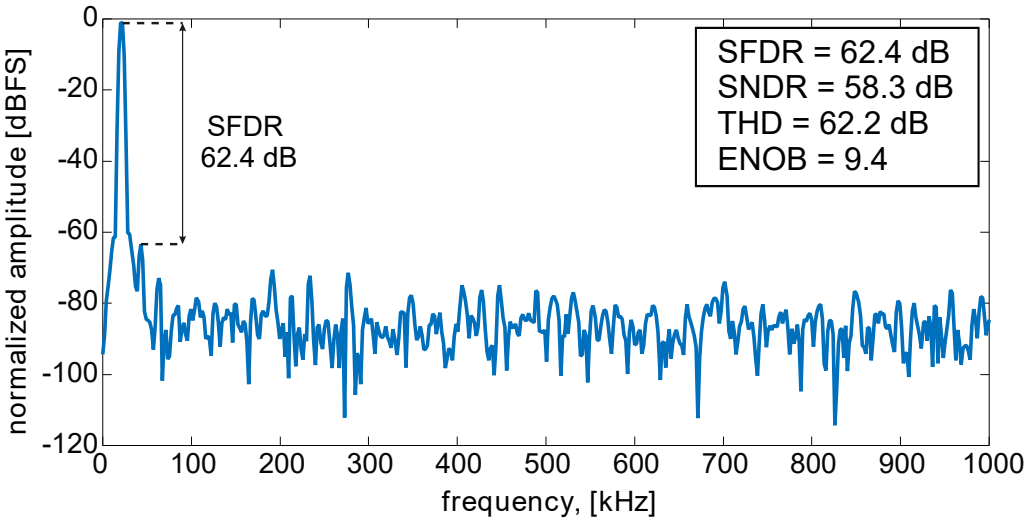
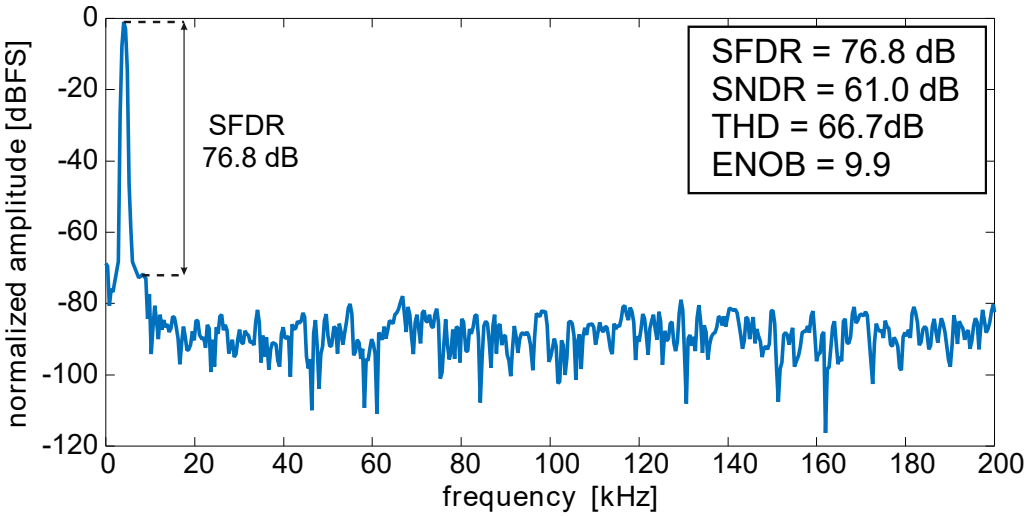


FIGURE 5.9 – Total and partial conversion energies for $T = 200ns$ configuration in post layout simulations.



(a) Post layout, post calibration SFDR for $T = 40ns$ configuration, $4kHz$ input tone.



(b) Post layout, post calibration SFDR for $T = 200ns$ configuration, $20kHz$ input tone.

FIGURE 5.10 – SFDR for the two chosen configurations.

# config.	T	T_{del}	INL_{max}	INL_{rms}	DNL_{max}	DNL_{rms}	N_{eff}	V_{DD}
2	$41.52ns$	$4ns$	0.72 LSB	0.34 LSB	1.27 LSB	0.07 LSB	9bit	0.6V
5	$202.72ns$	$4ns$	0.33 LSB	0.10 LSB	0.2 LSB	0.01 LSB	10bit	0.6V

TABLE 5.1 – Full post-layout static performance for the two configurations.

# config.	Samp. rate	SFDR	SNDR	ENOB	THD	En.Conv.	FOM
2	$2MS/s$	62.4dB	58.3dB	9.4bit	62.2dB	$0.73pJ$	$1.08fJ/(c.s.)$
5	$0.4MS/s$	76.8dB	61dB	9.9bit	66.7dB	$1.1pJ$	$1.15fJ/(c.s.)$

TABLE 5.2 – Full post-layout dynamic and energy performance for the two configurations.

	[24]	[23]	config. #5	config. #2
Type	CDAC	Rel.DAC	Rel.DAC	Rel.DAC
Validation	meas.	meas.	sim.	sim.
Technology [nm]	180	FPGA	40	40
Supply [V]	0.6	0.8	0.6	0.6
R [$k\Omega$]	N/A	100	288	128
C [pF]	9.2	2200	1	0.45
Area [μm^2]	163000	N/A	910	677
Norm.Area [$10^6 \cdot$F^2]	5.03	N/A	0.5	0.3
Resolution [bit]	10	10	10	10
Sample rate [kS/s]	20	0.3	400	2000
INL_{max} [LSB]	0.46	2.4	0.33	0.72
INL_{rms} [LSB]	N/A	0.9	0.10	0.34
DNL_{max} [LSB]	0.44	3.3	0.2	1.27
DNL_{rms} [LSB]	N/A	0.62	0.01	0.07
SNDR [dB]	58.3	43.27	61.0	58.3
SFDR [LSB]	67.7	51.36	76.8	62.4
THD [dB]	N/A	47.52	66.7	62.2
ENOB [bit]	9.4	7.13	9.9	9.4
En.Conv. [pJ]	1.7	N/A	1.1	0.73
FOM [fJ/(c.s.))]	2.49	N/A	1.1	1.08

Table 5.3 – Comparison with State-of-the-Art.

Conclusions

This work presented the design of a Relaxation DAC (RDAC) along with a systematic design flow and some possible implementation in $40n$m technology.
The analysis begun with trends in the IoT field, focusing on IoT nodes architecture and data converter performance specifications.
The working principle of the Relaxation DAC has been presented and verified by deriving analytical expressions.
Systematic errors on clock frequency and jitter have been modelled to predict their effects on converter accuracy and conditions to guarantee monotonicity, highlighting that timing errors can be nulled calibrating clock period by the use of dichotomic search.
The effect of parasitics related to passive components has been evaluated, as well as the ones introduced by the active driver, determining design guidelines to minimize their impact on converter accuracy. In particular, the most relevant nonideality is addressable to the parasitic capacitance of the RC resistor towards substrate, which introduces higher order poles to the RC response. This multiple time constants response has been compensated by introducing an extra zero MSB to the digital word, letting the transient due to higher order poles to be damped.
Errors due to driver charge injection and finite triode resistance can instead be neglected by proper components sizing. In the end it results that absolute components values are not relevant since nominal performance can be re-established just by clock re-calibration.
Being accuracy mainly dependent on linearity and distributed capacitance of the RC resistor, it is supposed to be enhanced in technologies having high resistivity linear resistors.
RDAC energy per conversion has been estimated, along with minimum capacitance required by thermal noise constraints.
The implementation of RDAC converter in $40n$m technology has been addressed, employing both previously derived guidelines and circuit simulations.

A possible implementation of the digital blocks allowing the DAC operation has been designed in Verilog and synthesized, implementing the strategy to compensate resistor higher order poles.
Converter performance has been evaluated based on post-layout simulations of two representative test cases, having conversion rates of $400k$S/s and $2M$S/s on 10bit resolution.
A proven significant result is that performance can always be restored to the nominal ones, after mismatch and temperature variations, by clock period recalibration, being accuracy dependent on the single parameter T/τ.
Residual nonlinearity is mainly addressable to buffer leakage, which is not compensated by the calibration strategy and slightly dependent on temperature variations, allowing to keep accuracy practically constant in a wide temperature range ($10^o\text{C} \rightarrow 125^o\text{C}$).
Energy related to digital blocks accounts for more than half of the overall dissipation, and it is dominated by dynamic power, so that digital flow optimization would even enhance energy FOM.
Presented topology turns out to be a mismatch-insensitive, low power architecture, with a figure of merit (FOM) of $1.1f$J/c.step.
Occupying a very small area (less than $1000\mu\text{m}^2$) and being almost completely digitally synthesizable, it proves to be an easily scalable and reconfigurable architecture.
The very low area, equally shared between analog and digital blocks, makes converter particularly suitable to calibration applications and designs with stringent area constraints.
Part of the results presented in this work have been already published in [23] and [22], where the significant advantage of the integrated RDAC implementation in terms of area, energy per conversion and sample rate has been highlighted with respect to [23] and [24].
Future developments include digital design optimization to further enhance power dissipation and chip fabrication to perform validating measurements. The block can be later impremented as a part of more complex architectures like DAC based ADCs, calibration blocks or programmable oscillators.

Appendix A

Verilog-A Code

ADC_10 block

```
// ADC_10, VerilogA

`include "constants.vams"
`include "disciplines.vams"

module ADC_10(in,ref,CK,D,Vout);
input in, ref,CK;
output [10-1:0]D;
output Vout;
electrical in,ref,CK,Vout;
electrical [10-1:0]D;


real Vcomp,Vsamp,VD;
real Vcomm[10-1:0];
integer i;
genvar ii;

// SAR-like conversion
analog begin
  @(initial_step) Vcomp=0.5;
  @(cross(V(CK,ref)-0.5,1))
  begin
    Vcomp=0.5;
    Vsamp = V(in,ref);
    for(i=10-1; i>=0; i=i-1) begin
      if (Vsamp < Vcomp)
        begin
```

```
29              Vcomm[i] = 0; // bit commutation to 0
                Vcomp = Vcomp-1/pow(2,10+1-i);
31            end
          else
33            begin
                Vcomm[i] = 1; // bit commutation to 1
35              Vcomp=Vcomp+1/pow(2,10+1-i);
              end
37      end
    end
39 // Analog hold voltage transition ///////////////
    VD=0;
41  for(i=10-1; i>=0; i=i-1) begin
      VD = VD + Vcomm[i]*pow(2,i);
43  end
    V(Vout) <+ transition(VD/pow(2,10),0,1p);
45 ////////////////////////////////////////////////

47 // Bits transitions ////////////////////////////
    for(ii=10-1; ii>=0; ii=ii-1) begin
49    V(D[ii]) <+ transition(Vcomm[ii],0,1p);
    end
51 ////////////////////////////////////////////////
  end
53

55 endmodule
```

P_SHIFT block

```
// P_SHIFT, VerilogA

`include "constants.vams"
`include "disciplines.vams"

module P_SHIFT(CK,ref,LD,D,out,out_n,dd);
input CK,ref,LD,dd;
input [9:0]D;
output out,out_n;
electrical CK,ref,LD,out,out_n,dd;
electrical [9:0]D;

real Vout,Vout_n,Vdd,thre;
real Dint[9:0],Dint_n[9:0];
integer i;
genvar ii;


analog begin
  @(initial_step) begin
    for(i=9; i>=0; i=i-1)begin
      Dint[i]=0;
      Dint_n[i]=0;
    end
    Vout = 0;
    Vout_n = 1;
    Vdd = V(dd,ref);
    thre=Vdd/2;
  end

  @(cross(V(CK,ref)-thre,1))
  begin
    if(V(LD,ref)>thre) // if load
      begin
        for(ii=9; ii>=0; ii=ii-1) // for each bit
        begin
            Dint[ii] = V(D[ii],ref)*Vdd; // load that bit into
    internal array
            Dint_n[ii] = Vdd-Dint[ii];  // load same bit into
    negative-reset array

        end
        Vout = Dint[0];      // D0 is shown directly on output
        Vout_n = Dint_n[0];   // D0 negated is shown direvtly
    on output
      end
     else // else the data is shifted
```

```
45        begin
            for(i=0; i<9; i=i+1)
47          begin
                Dint[i] = Dint[i+1]; // shift positive array
49              Dint_n[i] = Dint_n[i+1]; // shift negated array
            end
51          Dint[9] = 0; // zero value is shifted for zero-reset
      array
            Dint_n[9] = Vdd; // one value is shifted for negated-
      reset
53          Vout = Dint[0];    // positive output updated
            Vout_n = Dint_n[0]; // negated output updated
55        end
    end
57
  V(out) <+ transition(Vout,0,100p);     // transitions...
59 V(out_n) <+ transition(Vout_n,0,100p);
  end
61 endmodule
```

OutputComp block

```
// OutputComp, VerilogA

`include "constants.vams"
`include "disciplines.vams"

module OutputComp(adc,dac,ref,samp, dac_adc, dd, dac_hold);
input adc,dac,ref,samp,dd;
output dac_adc,dac_hold;
electrical adc,dac,ref,samp,dac_adc,dd,dac_hold;
electrical [10-1:0]D;

real thre;
real x_time, Vdac_Vadc, Vdynamic,dac_h;
integer i,fp;

analog begin
  @(initial_step)begin
    //fp = $fopen("/workareas/rrubino/Output.txt","w");
    i = 0;
    Vdac_Vadc = 0;
    Vdynamic = V(dd,ref);
    thre = Vdynamic/2;
    dac_h=0;
  end
  @(cross(V(samp)-thre,-1))
  begin
    x_time = $abstime;
    //$fwrite(fp,"%d \t %e \t %e \t %e \t\n",i,x_time, V(dac),V
    (dac_adc));
    i = i + 1;
    Vdac_Vadc = (V(dac)-Vdynamic*V(adc))/Vdynamic;
    dac_h = V(dac,ref);
  end
  @(final_step)begin
  //  $fclose(fp);
  end

  V(dac_adc) <+ transition(Vdac_Vadc,0,1p);
  V(dac_hold) <+ transition(dac_h,0,1p);
end
endmodule
```

RampGen block

```
// RampGen, VerilogA

`include "constants.vams"
`include "disciplines.vams"

module RampGen(dd,d,CK,ref,D,out);
input dd,d,ref,CK;
output out;
output [10-1:0]D;

electrical dd,d,ref,out,CK;
electrical [10-1:0]D;

real Vdd,Vstep,Vcomp,Vsamp;
real Dint[10-1:0];
integer count;
integer i;
genvar ii;

analog begin
  @(initial_step)
  begin
    Vdd=V(dd,ref); // initialize Vdd
    Vstep = Vdd/V(d,ref); // initialize step voltage
    count =0;   // initialize counter
    for (i=10-1; i>=0; i=i-1)
    begin
      Dint[i] = 0;
    end
  end
  @(cross(V(CK,ref)-Vdd/2,1))
  begin
    if (count!=V(d,ref))
    begin
      Vcomp=Vdd/2;
      Vsamp = count*Vstep;
      for(i=10-1; i>=0; i=i-1) begin
        if (Vsamp < Vcomp)
          begin
            Dint[i] = 0; // bit commutation to 0
            Vcomp = Vcomp-Vdd/pow(2,10+1-i);

          end
        else
          begin
            Dint[i] = 1; // bit commutation to 1
            Vcomp=Vcomp+Vdd/pow(2,10+1-i);
```

```
48              end
            end
50          count=count+1;
        end
52    end
// Transitions ////////////////////////////////////
54    for(ii=10-1; ii>=0; ii=ii-1) begin
        V(D[ii]) <+ transition(Dint[ii],0,1p);
56      // $display("D%d = %f ",ii,Dint[ii]);
    end
58
    V(out) <+ transition(Vsamp/Vdd,0,1p);
60 ////////////////////////////////////////////////////
  end
62 endmodule
```

Symbols and Abbreviations

ADC Analog-to-Digital Converter. 5, 7–13, 24, 72, 107, 118

D/A digital-to-analog. v, 17, 19
DA Digitally Assissted. 25
DAC Digital-to-Analog Converter. v, vi, viii, 5, 8–18, 20, 22–25, 27, 28, 31, 45, 61, 66, 71, 72, 82, 101, 102, 106, 107, 109, 118
DNL differential non-linearity. 19, 21, 31, 37, 40, 106
DU Digital Universe. 1

ENOB Effective Number of Bits. 107, 114

FOM figure of merit. v, vi, 107, 118

GE gain error. 21

HDL Hardware Description Language. v
HVT high threshold voltage. 83, 84, 92, 96

ICs integrated circuits. v
ICT Information and Communication Technology. 1, 2
INE integral error. 33, 36, 42–44, 72
INL integral non-linearity. 19, 21, 31, 33, 36, 37, 40, 42–44, 87, 90–92, 95, 97–100, 106
IoT Internet of Things. v, 1–5, 9, 16, 117

LPF Low Pass Filter. 13
LSB least significant bit. 17, 18, 21, 49, 51, 54, 57, 69, 77, 84
LVT low threshold voltage. 83, 84, 92, 94, 96

MC Monte Carlo. 97, 98, 100
MCU Microcontroller Unit. 5
MD Mostly Digital. 25
MSB most significant bit. 9, 17, 48, 81, 88, 102, 104, 117

PDK process development kit. v, 77
PnR place and route. 106

RC resistor - capacitor. v, 25, 52, 55, 59, 71, 72, 77–79, 81–86, 92, 93, 95–97, 117
RDAC Relaxation DAC. 107, 117, 118
rms root mean square. 23, 24

SAR Successive Approximation Register. 8, 9, 12, 24, 107
SFDR Spurious Free Dynamic Range. 23, 106, 107, 113, 114
SNDR Signal to Noise and Distortion Ratio. 9, 23, 107, 114
SNR Signal to Noise Ratio. 24
SoC System on a Chip. 5, 8, 25
SVT standard threshold voltage. 83, 84, 92, 96

THD Total Harmonic Distortion. 24, 107, 114

$\mathbf{V_{DD}}$ power supply voltage. 26

Bibliography

[1] IDC report. *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*. URL: https://www.emc.com/leadership/digital-universe/2014iview/internet-of-things.htm (visited on 06/05/2019).

[2] Massimo Alioto (eds.) *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. 1st ed. Springer International Publishing, 2017. ISBN: 978-3-319-51480-2,978-3-319-51482-6.

[3] Analog Devices Ian Beavers. *Intelligence at the Edge Part 1: The Edge Node*. URL: https://www.analog.com/en/technical-articles/intelligence-at-the-edge-part-1-the-edge-node.html (visited on 06/05/2019).

[4] W. Lv et al. "A General Architecture of IoT System". In: *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. Vol. 1. July 2017, pp. 659–664. DOI: 10.1109/CSE-EUC.2017.124.

[5] L. Fick et al. "A 346 µm 2 VCO-Based, Reference-Free, Self-Timed Sensor Interface for Cubic-Millimeter Sensor Nodes in 28 nm CMOS". In: *IEEE Journal of Solid-State Circuits* 49.11 (Nov. 2014), pp. 2462–2473. ISSN: 0018-9200. DOI: 10.1109/JSSC.2014.2358589.

[6] Analog Devices Inc. Engineeri. *Data Conversion Handbook*. Analog Devices series. Elsevier; Newnes, 2005. ISBN: 9780750678414,0750678410.

[7] Maxim Integrated. *Understanding Flash ADCs*. URL: https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/810.html (visited on 09/26/2019).

[8] Analog Devices. *Which ADC Architecture Is Right for Your Application?* URL: https://www.analog.com/en/analog-dialogue/articles/the-right-adc-architecture.html (visited on 09/26/2019).

[9] Behzad Razavi. *Principles of data conversion system design / Behzad Razavi*. eng. New York: The Institute of Electrical Engineers, 1995.

[10] Inc. Analog Devices. *Analog-digital Conversion Handbook; 3rd Ed.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986. ISBN: 0-13-032848-0.

[11] Tony Chan Carusone; David Johns; Kenneth W Martin. *Analog integrated circuit design*. 2nd ed. Wiley, 2011. ISBN: 9780470770108,0470770104.

[12] "IEEE Standard for Terminology and Test Methods of Digital-to-Analog Converter Devices". In: *IEEE Std 1658-2011* (Feb. 2012), pp. 1–126. DOI: 10.1109/IEEESTD.2012.6152113.

[13] Martin Clara (auth.) *High-Performance D/A-Converters: Application to Digital Transceivers*. 1st ed. Springer Series in Advanced Microelectronics 36. Springer-Verlag Berlin Heidelberg, 2013. ISBN: 978-3-642-31228-1,978-3-642-31229-8.

[14] T.D Burd et al. "A dynamic voltage scaled microprocessor system". eng. In: *IEEE Journal of Solid-State Circuits* 35.11 (2000), pp. 1571–1580. ISSN: 0018-9200.

[15] P. S. Crovetti. "A Digital-Based Virtual Voltage Reference". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.5 (May 2015), pp. 1315–1324. ISSN: 1549-8328. DOI: 10.1109/TCSI.2015.2402991.

[16] B Murmann. "Digitally Assisted Analog Circuits". eng. In: *IEEE Micro* 26.2 (2006), pp. 38–47. ISSN: 0272-1732.

[17] W. C. Elmore. "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers". eng. In: *Journal of Applied Physics* 19.1 (1948), pp. 55–63. ISSN: 0021-8979.

[18] J Rubinstein, P Penfield, and M.A Horowitz. "Signal Delay in RC Tree Networks". eng. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2.3 (1983), pp. 202–211. ISSN: 0278-0070.

[19] Mark Alan Horowitz. "Timing models for MOS circuits". PhD thesis. Stanford University, 1983.

[20] Neil H. E Weste. *CMOS VLSI design : a circuits and systems perspective / Neil H.E. Weste, David Money Harris*. eng. 4th ed. Boston: Addison-Wesley, 2010. ISBN: 9780321547743.

[21] Behzad Razavi. *Design of analog CMOS integrated circuits / Behzad Razavi*. eng. 2nd ed. New York: McGraw-Hill Education, 2017. ISBN: 9781259255090.

[22] Roberto Rubino, Paolo S. Crovetti, and Orazio Aiello. "Design of Relaxation Digital-to-Analog Converters for Internet of Things Applications in 40nm CMOS". eng. In: *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. October. IEEE, 2019.

[23] Yulin Zhang, Edoardo Bonizzoni, and Franco Maloberti. "Mismatch and parasitics limits in capacitors-based SAR ADCs". eng. In: *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2016, pp. 33–36. ISBN: 9781509061136.

[24] P. S. Crovetti, R. Rubino, and F. Musolino. "Relaxation digital-to-analogue converter". In: *Electronics Letters* 55.12 (2019), pp. 685–688. DOI: 10.1049/el.2019.0784.

[25] P. S. Crovetti. "A Digital-Based Analog Differential Circuit". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 60.12 (Dec. 2013), pp. 3107–3116. ISSN: 1549-8328. DOI: 10.1109/TCSI.2013.2255671.

[26] P. S. Crovetti. "All-Digital High Resolution D/A Conversion by Dyadic Digital Pulse Modulation". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.3 (Mar. 2017), pp. 573–584. ISSN: 1549-8328. DOI: 10.1109/TCSI.2016.2614231.

[27] O. Aiello, P. S. Crovetti, and M. Alioto. "Fully Synthesizable Low-Area Digital-to-Analog Converter With Graceful Degradation and Dynamic Power-Resolution Scaling". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* (2019), pp. 1–11. ISSN: 1549-8328. DOI: 10.1109/TCSI.2019.2903464.

[28] Jorge Fernandes (auth.) Taimur Rabuske. *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*. 1st ed. Analog Circuits and Signal Processing. Springer International Publishing, 2017. ISBN: 978-3-319-39624-8,978-3-319-39623-1.

[29] Jen-Wei Yang, Po-Tsang Huang, and Wei Hwang. "On-Chip DC-DC Converter with Frequency Detector for Dynamic Voltage Scaling Technology". eng. In: *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE, 2006, pp. 666–669. ISBN: 1424403863.