POLITECNICO
DI TORINO

TU Delft

Master degree course in Electronic Engineering

Master Degree Thesis

An On-Chip Testing Platform for Characterizing Multi-Sensor Time-Mode Compressed Sensing Systems

Supervisor
prof. Gianluca Setti

Co-Advisors
dr. O. Can Akgun
dr. Fabio Pareschi

Candidate
Cristian Mastronardi

Academic year 2019-2020

Summary

This thesis work, made in collaboration with the Bioelectronics group of TU Delft (the Netherlands), is based on the topic of DFT (Design for Testability) applied to integrated circuits that exploit very small signals (e.g. bio-signals) that can easily drown in noise. For this type of systems, it is very complex to evaluate the correct functional operation of the circuit from the outside world, since due to the use of a big number of sensor lines, errors would be made due to the introduced disturbances. It is therefore useful to implement custom integrated testing platforms on the same chip, which are able to real-time check the correct functioning of the circuit introducing as little noise as possible.

In this project, a digital-to-analog conversion system has been developed in TSMC 40nm CMOS process, capable of translating digital test vectors stored in appropriate memories into very small amplitude analog signals to be supplied to the main system (i.e. a Multi-Sensor Time Mode Compressed Sensing IC), all trying to limit the impact of conversion errors and sampling noise (also called Johnson-Nyquist noise), which is the limiting factor in those systems where switched-capacitors are used to obtain the desired analog output signal.

In the first chapter, the object and the objectives of the thesis project have been briefly highlighted reporting some insights on the circuit to be tested.

In the second chapter, after an excursus about the state of the art of the converters made in CMOS technology, the most suitable topology has been chosen and improved in order to achieve the expected result, and, after that, the circuit has been designed in Cadence Virtuoso environment, operating with the available technological process.

Subsequently, in the third chapter, the circuit performances have been checked by means of computer aided simulations, with which good results have been obtained in terms of performance and noise for both 8 bits and 10 bits implementations, opening the way for future IC layout of the proposed testing system.

In conclusion, the work highlights the possible limitations of these converter topologies but, at the same time, opens the way to multiple reflections on their actual use in systems oriented towards integrated testing, especially in the bioelectronics field.

Acknowledgements

I would like to thank Delft University of Technology for the support they gave me in both technical and human terms, providing all the equipment and advice to work on this master thesis research project during the period spent abroad.

All the Department of Microelectronics and especially the Bioelectronics group chaired by Prof. Wouter A. Serdijn made me feel at home, allowing me to attend their group meetings where the most diverse and challenging research projects were presented and discussed by students, researchers and professors.

My thanks also to my external advisor and Italian supervisor for their support and availability.

Contents

List of Tables

List of Figures

1 Introduction

1.1 General Principles

Wearable low-cost sensing systems use is growing among common users as well as for medical monitoring, where small size, good performance, wireless transmission and low-power operation are factors that play a fundamental role.

Due to these constraints, these devices must be separated from the processing units because DSP algorithms require additional hardware with a much bigger cost in terms of power and area usage. Thus, several platforms like smartphones or computers-based monitoring stations can be used to aggregate and process wireless-transmitted data coming from the sensors [1].

In this scenario, new data-handling paradigms are coming out in order to solve limitations imposed by the traditional acquiring schemes.

One promising signal process technique is called **Compressed Sensing**.

1.2 Compressed sensing

In a traditional bio-medical sensor chain, the signals coming from the patient are sampled using the well know Nyquist theorem where the sampling frequency should be at least two times the maximum frequency of the sensed signal in order to be able to reconstruct it without alias.

Compressed sensing paradigm exploit a specific feature of bio-signals that is called **sparsity**, which means that most of their energy is localized in a given region of the signal space (e.g. in a specific frequency band) and due to this it can be expressed as the linear combination of only a few waveforms in a proper basis.

Exploiting the low-rate of significant events in these kind of signals, sub-Nyquist sampling is possible despite of the traditional method, reducing acquisition speed and power usage, being still able to reconstruct the information in a correct way. This is feasible multiplying the N-dimensional input vector x composed by N input samples for a particular matrix A called **sensing matrix** [2] obtaining an M-dimensional output vector called **measurement vector** y with M<N, as reported

in Equation 1.1.

$$y = \mathbf{A}x \tag{1.1}$$

The ratio $\frac{N}{M}$ is called **compression factor** CF and represents the reduction of the data transmitted and, subsequently, the reduction in power dissipation in the sensor node (by a similar quantity).

It comes by itself that sensing systems implementing this new scheme (from now on **CS**) must contain intelligent structures able to detect, digitize, compress and transmit obtained alias-free data without a big cost (dissipated power ideally negligible).

For the afore mentioned properties a CS sensor chain block representation should be the one reported in Figure 1.1.

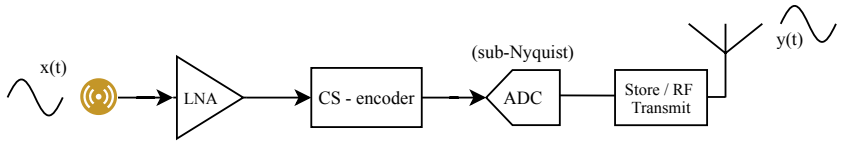

Figure 1.1. Analog CS transmission chain

Usually, specialized circuits (i.e. ASIC) are required in order to perform CS processing technique.

1.3 Time Mode CS

A way to improve the compressed sensing processing scheme consist in the use of the **Time Mode** signal processing paradigm instead of standard CMOS digital operation in order to achieve very low power consumption with an asynchronous operation [3].

In the system reported in Figure 1.2 the signals coming from 128 sensor lines are compressed exploiting an improved version of CS called *Rakeness-based* CS [4] where the measurement matrix is formed in a better way in order to reach greater compression factors and lower power consumption.

The m-dimensional output vector is converted into a 9-bit digital word using an Analog Time-to-Digital converter that is in charge to transform an analog signal coming from the sensor in a specific timed pulse that contains the signal

information. This pulse will be handled by some digital logic and transformed into a digital output signal.

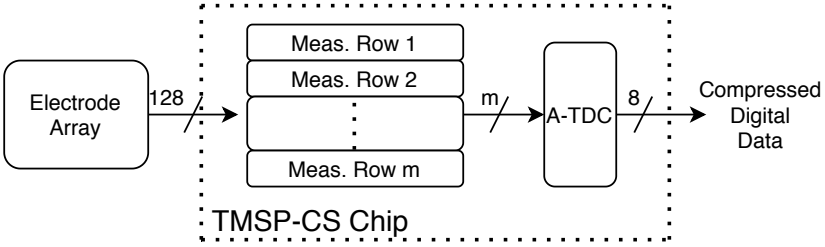

Figure 1.2. TMSP CS system high-level block diagram

1.4 Design For Testability

Bio-signals are characterized by bandwidths up to several kHz and when they are recorded (for example with an electrode probe) they show dynamic ranges > 35 dB and usually the required resolution in medical applications is about 8-10 bits.

The previously reported circuit in its IC realization needs to be tested but, with traditional testing methods, this is not possible because it is a 128 sensor system and the signals coming from them are very small (e.g. 10 mV_{pp} for ECG). Due to this, also stringent noise performance are required making the functional monitoring of chip performance a very hard challenge to manage.

With the scaling of technological nodes, CMOS circuits are becoming faster and smaller and the possibility to test functioning chips becomes harder due to several limitations regarding area, test instrumentation, etc.

The solution, with this constraints, must consist in the implementation of the so called **Built-In Self-Test** (BiST) which consists of an *On-Chip Testing Platform* able to check if the circuit is working in the desired way and in addition, it is possible to diagnose the circuit in case any problem emerges in the future.

The conversion structure, in this specific case, must be small and low power as previously reported, with an 8-bit minimum resolution and low noise operation.

In this thesis, we focused on the design of the DAC structure that is in charge to convert random digital input vectors generated by some digital circuitry and stored in specific memory blocks, making them available at the ATCs inputs as useful analog signals.

2 On-Chip Testing Architecture

After the considerations made in the previous chapter, a possible system implementation can be the one reported in Figure 2.1. The research carried out to develop this thesis work focused on the digital-to-analog conversion chain (highlighted in gray).

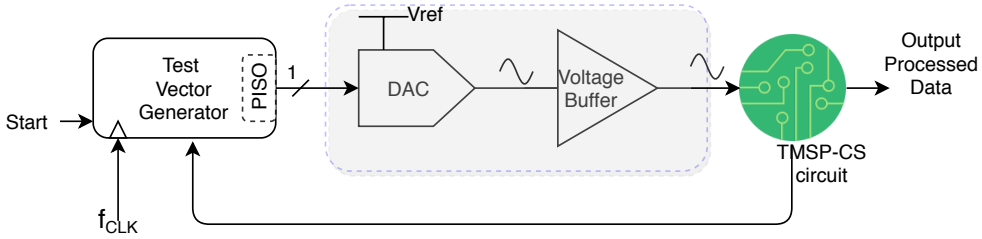

Figure 2.1. High-level system block diagram

In this system, $START$ and f_{CLK} signals are applied from the outside in order to load the test data into the system. The generator is synchronous but the signals coming from the block to be tested through the feedback, will be asynchronous. A sort of hybrid system needs to be implemented where a register which keeps track of which data to apply works asynchronously and the part that applies the correct data to the DAC works synchronously. A **PISO** (Parallel-in Serial-out) block is in charge to serialize the digital test vectors when they are generated.

Finally, based on limitations due to chip area, only two DACs will be employed to feed the ATCs conversion chains [3], one for the odd-numbered and one for the even numbered Analog-to-Time Converters.

For this reason, design efforts have been made in order to make the DAC much faster than the delay of the whole chain and quick enough to settle before applying new test data.

Initial specifications for the converter are reported in Table 2.1 and have been

followed for the topology selection and implementation during the design flow and extendend after obtaining the simulation results.

Property	Requirement
NOB	8-10
V_{FS}	10 mV
Sample frequency	2 $\frac{\text{MS}}{s}$
SNR	>50 dB
Total noise	$< \frac{LSB}{2}$

Table 2.1. General DAC requirements for TMSP CS chip

The noise constraint is fundamental just because in order to generate the correct analog value for the time-mode CS IC, the conversion chain must generate as little noise as possible.

At these levels of voltage (i.e. 10 mV_{pp}), the main source of problems is the sampling noise stored within the capacitors (cfr. Appendix A) and a very accurate choice based on circuits that use as few capacitors as possible can be a winner.

2.1 D/A Conversion Principles

A digital to analog converter, is a component able to map a digitally-coded data into an analog signal (current or voltage) corresponding to the input digital word. The principle of operation can be represented by Equation 2.1 where a sinc(t) function is exploited as mapping element between continuous and discrete domains.

$$Y(t) = A \cdot \sum_{n=-\infty}^{\infty} X(n) \cdot sinc(f_s(t - nT)) \tag{2.1}$$

where Y(t) is the reconstructed analog signal, A is a gain introduced by the DAC, $sinc(t) = \frac{\sin \pi x}{\pi x}$, f_s is the sampling frequency that must be selected according to Nyquist's Theorem, $T = \frac{1}{f_s}$ and X(n) is the input digital word that is represented

by Equation 2.2.

$$X(n) = \sum_{i=0}^{N-1} b_i(n) \cdot w_i \tag{2.2}$$

where $b_i(n)$ is the i-th bit within the N bits word and w_i is a weight that depends on what coding scheme is used (e.g. w_i=1 for all i in the thermometer-coded case). Due to the fact that the sinc function is non-causal, it can be substituted with another function that is equal to 1 within the sampling window T and equal to 0 in all other cases obtaining something similar to equation 2.2 but multiplied for the gain A introduced by the DAC. A problem, here, is represented by *Quantization Noise* that is present in this kind of structures because the varying input digital data is itself quantized and we can treat it as a white noise contribution that has a rectangle probability distribution across 0 and 1 and with a total power contribution $P_{QN} = \frac{1}{12}$, if the quantization step is unitary.

2.2 Literature analysis

First of all, in order to check the correct functioning of the core circuit (TMCS chip), the testing platform should be able to fetch the asynchronous conversion chain made of ATCs [3] with analog test signals coming from 8-10 bits digital words saved as binary vectors in specific memory blocks.

This produced analog signal, as reported after Table 2.1, have a peak-to-peak differential amplitude of 10 mV with a DC voltage of 300 mV which consists of half the supply voltage in that circuit. Thus, the circuit must be accurate and linear, generating very small errors from the conversion phase.

In addition, apart from area usage and noise that are very important design drivers, also power must be taken into account because nowadays electronic IC must be as portable as possible, indeed, their use in mobile applications and IoT is in constant growth and this wide range of applications requires very small equipment and therefore very small batteries that supply the hardware creating new technological challenges for circuit designers. In order to achieve this goal, several Digital-to-Analog conversion topologies were analyzed.

The most important features of different type of DACs emerged from the literature analysis are reported in Table 2.2 and Table 2.3.

	Zhang [5]	Zhang [6]	Lofti [7]	Cho [8]
Tech (nm)	130	65	130	180
V_{DD} (V)	1	0.7	1	1
NOB	10	10	10	8
LSB	–	–	–	–
ENOB	–	9.1	10	–
SF (kS/s)	1	1	–	10
Power (nW)	33	0.75	80	60
kTC noise (uV)	–	–	–	–

Table 2.2. DAC comparison table 1

Lee [9]	Bh [10]	Maio [11]	Itzik [12]	Reaz [13]
180	90	800	350	130
1.5	1.2	3	–	–
10	16	10	10	10
–	12.3 uA	0.5 uA	2.5 mV	100 nA
8	–	–	–	–
–	50	500	100	–
1.74	–	–	–	–
400	–	–	–	–

Table 2.3. DAC comparison table 2

As can be seen, most of the proposed circuits in literature work at 8-10 bits of signal resolution in different CMOS technological nodes. A great number of implementations are current-steering based, where MOST transistors are sized and used emulating a binary weighted code with their ON currents. Here, the main drawback is that device mismatch introduces a conversion error that, in order to fulfill the objective of this work, can be a problem and in addition this kind of converters require logic circuitry in order to work in the proper way [14].

Other structures are based on the use of multiple weighted capacitors where area usage can be a very important limiting factor along with bigger sampling noise (cfr. Appendix A) produced by bigger capacitors.

A possible candidate that matches the requirements imposed by the TMSP CS chip is the **Successive Approximation Serial DAC** [15] where small capacitors, low power consumption, full CMOS compatibility and simple circuit operation can lead to a good result in this kind of application.

The topology and circuit working algorithm will be analyzed in the following sections.

2.3 Circuit Design

2.3.1 Successive Approximation Serial DAC

Traditional conversion chains usually require a lot of hardware for sampling, process and store acquired signals, ending up with large area occupation and power usage. After the literature study based on the problem to solve, a pseudo-passive successive approximation DAC topology was chosen [15] and implemented in 40 nm CMOS technology. In this structure, switched-capacitor circuits are used in order to achieve good accuracy and low power operation at the expenses of lower conversion rates.

In addition, the error introduced by capacitor mismatch that is one of the main limiting factors in this kind of structures, is independent from the input signal and introduces a very small contribute on the signal spectrum (this property has been widely analyzed with FFT simulations).

The principle of operation is based on two ideally identical capacitors that work dividing a charge stored according to the digital word that should be converted as showed in Figure 2.2.

Initially, the two capacitors are discharged in a pre-conversion reset phase connecting them to ground with a specific reset signal that turns ON *SW2* and *SW4*, then, starting from the LSB, $b_1(n)$, capacitor C_1 is precharged to V_{REF} if $b_1(n) = 1$, closing *SW1* or connected to ground if $b_1(n) = 0$, while, the capacitor C_2 is maintained isolated because the MOST-based switch *SW3* is OFF.
When the charging phase on C_1 is complete, the charge is redistributed between the two capacitors closing the switch *SW3* and letting the stored charge to flow towards C_2.

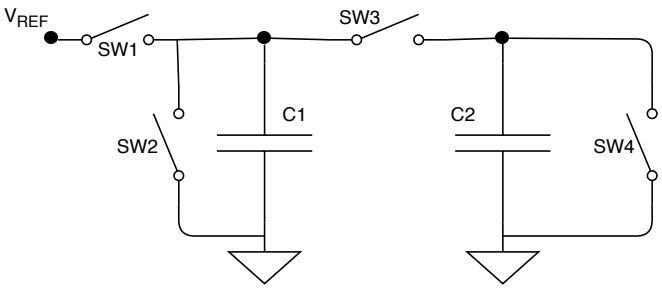

Figure 2.2. Serial CR-DAC scheme of principle

The resulting voltages, for the first conversion step, are reported in Equation 2.3.

$$V_{C_1} = V_{C_2} = \frac{b_1(n)}{2} \cdot V_{REF} \tag{2.3}$$

The conversion ends when the N-th bit (MSB) of the digital word n is evaluated. Equation 2.4 reports the voltage relationship between two successive steps.

$$V(n,i) = \frac{C_1}{C_1 + C_2} b_i(n) V_{REF} + \frac{C_2}{C_1 + C_2} V(n, i-1) \tag{2.4}$$

In addition, the final analog output is represented by Equation 2.5.

$$V_{C_{1_f}} = V_{C_{2_f}} = V_{REF} \cdot \sum_{i=1}^{N} \frac{2^i b_i(n)}{2^{N+1}} \tag{2.5}$$

According to this behavior, the conversion phase is slow compared with other DAC topologies and leakage becomes a big problem when the sampling rate is increased over a certain limit but can be good enough for the TMSP CS chip test also because it is monotonic, low-power and area-efficent.

In addition, if a variable clock tree is available, it is possible to change the resolution modifying only the control signals without touching the physical structure. This can be an interesting feature.

2.3.2 Circuit implementation

A custom version of the serial SA-DAC has been designed in Cadence Virtuoso using the TSMC 40nm complementary metal oxide semiconductor (CMOS) technology.

First of all, an improved switching scheme has been implemented in order to reduce the number of simultaneously active switches, limiting power consumption and area usage. Then, some other novelties were introduced and are reported in Figure 2.3.

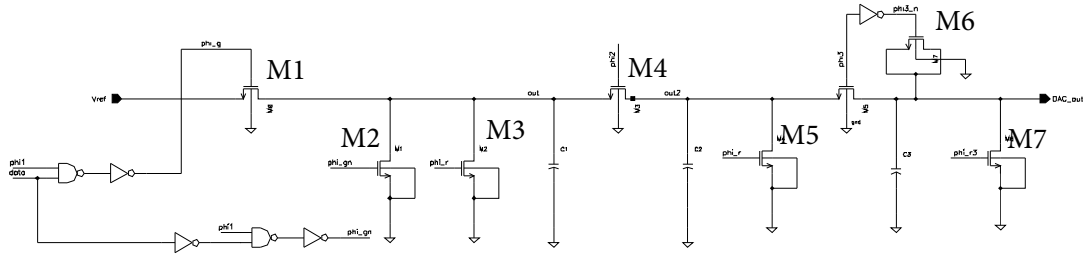

Figure 2.3. Serial CR-DAC Implementation

This DAC is an evolution of the one presented in [16] where some other components have been added in order to fulfill the TMSP-CS system specifications. In addition to the standard 2-cap redistribution structure, a second stage was designed in order to reach a proper voltage division factor regarding to the selected voltage reference level, set in order to assure a good operation, limiting also charge injection and clock feedthrough that are the main sources of conversion errors along with capacitors mismatch.

From table reported in Appendix A, in order to produce a thermal noise smaller than LSB/2, a total capacitance near to 10 pF was implemented, dividing this value between the 2 charge division stages.

For higher reference voltages (e.g. $V_{DD}/2$), the higher conversion step in the first stage implies different values for the capacitors as the division factor is different. An explanatory example is shown in Table 2.4.

In fact, with such ratio, starting from V_{REF} it is possible to end the conversion with a 10mV V_{FS} that is the amplitude required for the signals of interest.

According to Johnson-Nyquist theory (cfr. Appendix A), the total implemented equivalent capacitance should allow to maintain the sampling thermal noise below the upper limit.

In addition, a dummy MOST-based switch was inserted in order to reduce conversion errors limiting charge injection and a combination of several **INV** and

Capacitor	160 mV	550 mV
C_1	600 fF	200 fF
C_2	600 fF	200 fF
C_3	9 pF	10.8 pF

Table 2.4. Charge redistribution capacitors for different Vref

NAND2 standard logic gates works together with the serial data signal in order to perform the conversion algorithm in the correct way.

The structure follows the operation reported in section 2.3.1 but works in a 2-step fashion, in fact, each digital word is previously totally converted and stored on C_2 and then, in the 2nd phase, this voltage is redistributed between C_2 and C_3 obtaining the final value.

The operation is allowed by different clock signals that drive the various switch inside the structure. Two non-overlapping square waves Φ_1 and Φ_2 drive the charge of C_1 and C_2, according with Equation 2.3. Serial data coming from the test vector generator as in Figure 2.1 is valid for a period of time equal to 62.5 ns for the 8 bits resolution and 50 ns for the 10 bits resolution and is gated with Φ_1 through the *NAND2* in order to understand if the i-th bit is 0 or 1, then the charge of C_1 is performed turning on the specific switches if $b_i(n) = 1$ otherwise nothing happens. The first part of the system is highlighted in Figure 2.4.

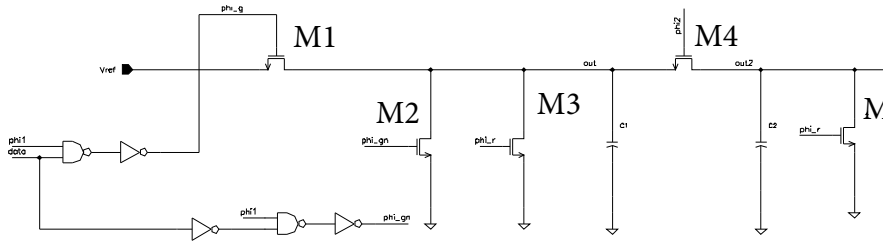

Figure 2.4. DAC first redistribution stage

When the process is complete, Φ_1 goes low and Φ_2 high, allowing the charge stored on the first capacitor to redistribute between the two, as reported in Equation 2.4. This second part of the system is highlighted in Figure 2.5.

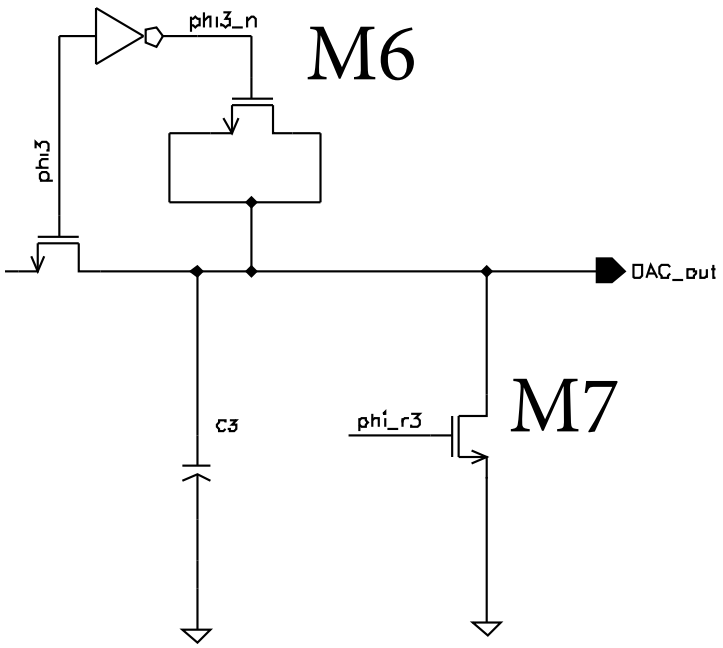

Figure 2.5. DAC second redistribution stage

This process is repeated N+1 times (a dummy step is mandatory and will be explained later), until the final voltage value corresponding to the input digital word is present on C_2. At this point, new input data can start to be processed and in the meantime Φ_3 goes high for a bit, performing the final redistribution between C_2 and C_3, where the final desired voltage value is obtained.

After this phase, Φ_3 goes low and the capacitors C_1 and C_2 can be discharged with Φ_r that goes high on the rising edge of the next input data first dummy bit. Output voltage can be read-out during the reset phase (the 2nd stage is insolated from the 1st because Φ_3 is low) using an additional MOST-based switch and then, also C_3 can be discharged by means of Φ_{3r}.

The signals that drive the whole structure are reported in Figure 2.6.

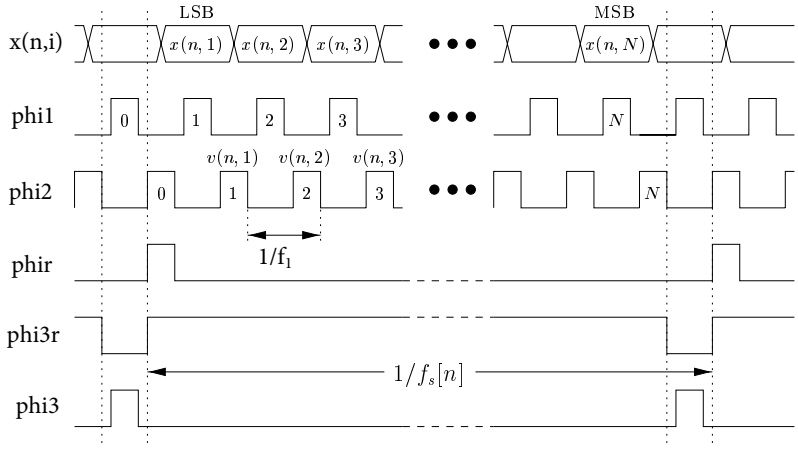

Figure 2.6. DAC timing signals

MOST-based switches

The implemented SA-DAC architecture requires fast switches in order to charge and discharge capacitors allowing the circuit to redistribute charge and perform the D/A conversion. In this work, different kind of MOST-based switches were designed and tested. Among these we find:

- **Single MOST**
- **Single low V_{TH} MOST**
- **Transmission Gate**

The design flow started implementing the whole structure with standard NMOST switches (for further optimization refer to section 2.3.3) that are very good for small signal transmission, just to test the correct operation and the actual limitations. This is possible only with MOSFETs because bipolar transistors typically need complex circuits to perform sampling. Measuring by means of simulations the time required by the switch when turned ON for charging the capacitor to the maximum input level (V_{REF}) starting from 0 V, a complete speed characterization was possible.

The single NMOST switch is shown in Figure 2.7.

The transient response needs ideally infinite time to reach the final value and due to this a certain error band must be used within which the output is considered settled and the rise time can be measured.

ON-resistance and the value of the sampling capacitor are the factors that modulate the sampling speed, thus, in order to increase the speed, a large aspect

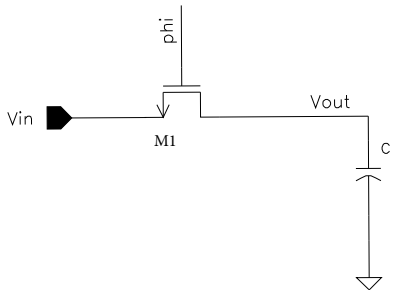

Figure 2.7. Basic NMOST sampling circuit

ratio and a small capacitor must be used and for this reason the devices were sized with non-minimum **Aspect Ratio**.

In addition, for NMOS switches, the higher is the input voltage the bigger is the ON-resistance leading to a longer time constant worsening the switch operation but at the same time limiting the clock feedthrough effect because the parasitic capacitance between gate and drain or gate and source cannot inject charge inside the channel.

The resistance increase can be better analyzed referring to Equation 2.6.

$$R_{ON} = \frac{1}{\mu_n C_{ox} \frac{W}{L}(V_{DD} - V_{in} - V_{TH})} \tag{2.6}$$

Then, the transmission gate (TG) structure was implemented, just to check possible differences in terms of performance.

The TG is shown in Figure 2.8 and its use as sampling circuit in Figure 2.9.

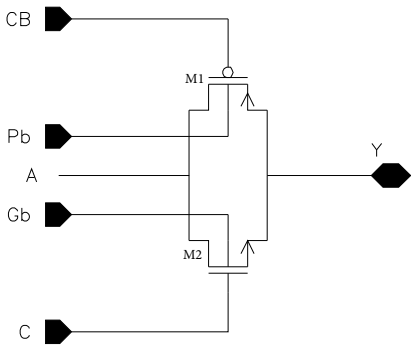

Figure 2.8. Complementary Switch

Then, the TG-based sampling circuit is reported in Figure 2.9.

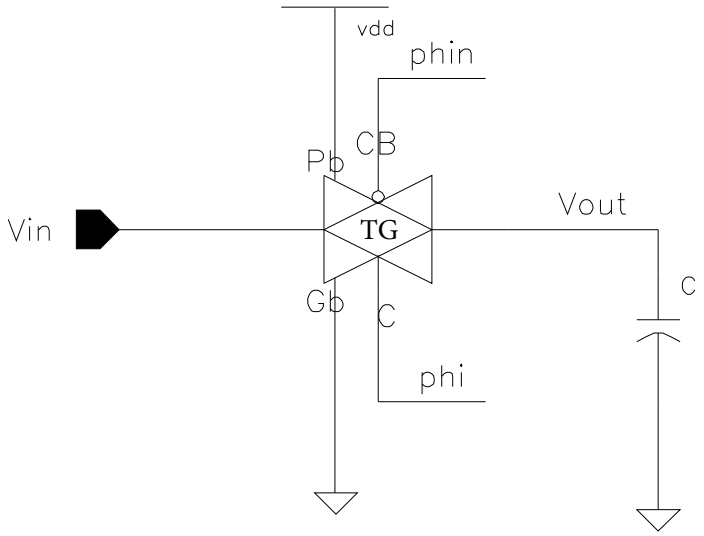

Figure 2.9. Complementary Switch sampling circuit

This structure is also called *complementary switch* because, as can be seen from the upper figure, both kind of transistor devices are employed in order to increase the switch performance, enlarging the operative voltage swings, in fact, this kind of structure shows an R_{ON} almost independent from the input signal [17] because for the smaller ones the transmission is performed by the NMOST and viceversa.

For this reason, if the designer decide to use a higher V_{REF} for the DAC because of system constraints, the use of TGs is a mandatory choice, in fact, in this scenario, their resistance is lower than the simple NMOS devices.

2.3.3 Circuit optimization

Transistor Sizing

When high performance of MOST-based switches are required, one of the main factors limiting the precision (the sampling error) is called *Charge Injection* (cfr. Appendix B) and depend strongly on the input level.

This phenomenon causes sampling errors by means of some parasitic charge that remains trapped inside the device channel when the switch is turned OFF and

which can be quantified by Equation 2.7.

$$Q_{ch} = WLC_{ox}(V_{DD} - V_{in} - V_{TH}) \tag{2.7}$$

where W and L are the device physical dimensions, C_{ox} the gate oxide thickness, V_{DD} the supply voltage, V_{in} the input signal and V_{TH} the threshold voltage.

In order to reduce it, a second dummy switch **M6** that works in counter-phase with the effective switch must be introduced and sized with a minimum channel length L and a width $W = 2 \cdot W_{SW}$ in order to remove the charge injected by the previous one, when forming its conductive channel.

In addition, another problematic phenomenon occurs and is called *Clock Feedthrough* (cfr. Appendix C) where the switch couples the clock signal to the sampling capacitor through its gate-drain or gate-source overlap parasitic capacitances but with the sizing choice made on the dummy switch, also this problem is suppressed.

The NMOST were sized larger than minimum in order to drive more current and be fast in charging and discharging the switched capacitors but with minimum channel length in order to avoid an increase of channel resistance worsening the switches performance. In fact, for process variation and reducing the threshold voltage V_{TH} a little bit, it is possible to increase the channel length to 60-70 nm, but no more than that.

The choice on W introduces extra parasitic capacitance and as consequence a bigger noise contribution, that must be checked in order to not exceed the noise budget and for speed up the sampling capacitors charge and/or discharge some devices were implemented with low threshold voltage transistors that show an increase of leakage currents but are faster than the standard counterpart.

The MOS devices aspect ratios are reported in table Table 2.5.

2.4 CMOS Voltage Buffer

In order to decouple the DAC from the ATCs chain, an **Operational Transconductance Amplifier** OTA was designed [18]. This analog block is in charge of increasing the DAC output signal power in order to drive correctly the load capacitance and is made by two consecutive stages.

The first one is a differential stage that takes care of rejecting the common mode unwanted signals when the structure operates in differential mode and the second

MOST	W	L	Type
M1	2 µm	60 nm	Low-VT
M2	1 µm	60 nm	Standard-VT
M3	1 µm	60 nm	Standard-VT
M4	2 µm	60 nm	Low-VT
M5	1 µm	60 nm	Standard-VT
M6	500 nm	60 nm	Standard-VT
M7	2 µm	60 nm	Standard-VT

Table 2.5. DAC transistors aspect ratios

stage is a gain-booster, where the output resistance is increased with the use of *Cascode* structures.

The standard CMOS amplifier is reported in Figure 2.10.

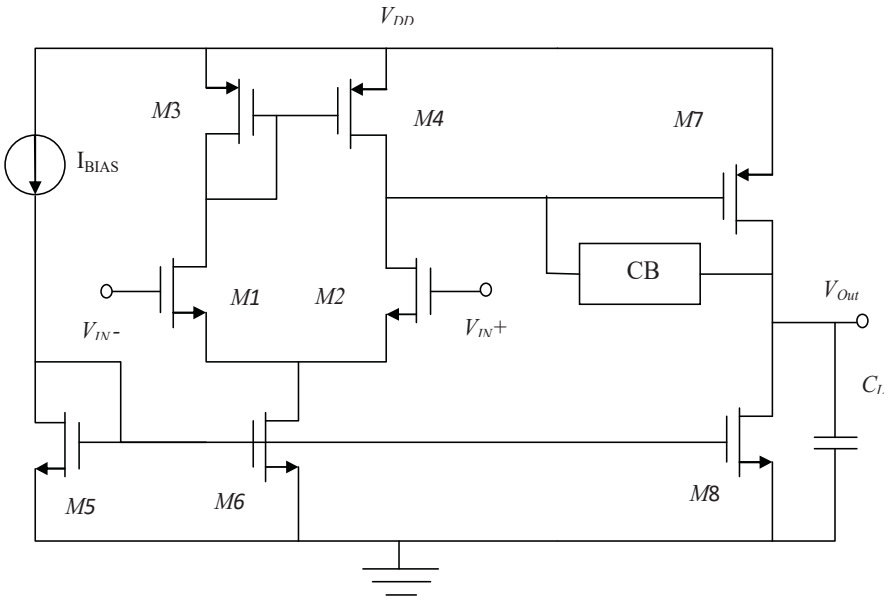

Figure 2.10. Two-stage compensated CMOS OTA

This kind of topology, in order to drive C_L in the correct way needs to be compensated using, for example, the so called *Miller Compensation* that as drawback, introduces a zero into the amplifier open loop frequency response worsening the bandwidth and as a consequence the phase margin, making the behavior less stable.

In order to solve this problem, an RC feedback instead of pure C compensation block must be designed limiting the zero effect and consequently the phase shift and the value of the compensation capacitance C_c and resistance R must be selected referring to the load capacitance.

In addition, the input differential couple must be implemented with big aspect ratio PMOS devices instead of NMOS obtaining a better matching while limiting the low frequency $\frac{1}{f}$ noise contribution because the noise modelling factor of the former is about one order of magnitude smaller than the latter [19].

The equations that represent the MOST operation in strong inversion region are reported in Equation 2.8 and Equation 2.9.

$$I_D = \frac{\mu_n C_{ox}}{2}(\frac{W}{L})(V_{GS} - V_{TH})^2 = \frac{\mu_n C_{ox}}{2}(\frac{W}{L})V_{eff}^2 \tag{2.8}$$

$$g_m = \sqrt{2\mu_n C_{ox}(\frac{W}{L})I_D} = \frac{2I_D}{V_{eff}} \tag{2.9}$$

For the OTA in Figure 2.10 design equations are available and allow the designer to operate in order to obtain an amplifier that operates according to specifications.

For the gain and bandwidth, Equation 2.10 represents the amplifier transfer function in the Laplace domain and from which, some parameters can be extrapolated (in the case of an RC compensation scheme, it is slightly different of course).

$$A_s = \frac{w_{GBW}}{s} \cdot \frac{1 + \frac{C_c + C_{gs8}}{g_{m8}}s}{s^2(\frac{C_L C_{gs7}}{C_C g_{m7}}\frac{C_C + C_{gs8}}{g_{m8}}) + s(\frac{C_L C_{gs7}}{C_C g_{m7}}\frac{C_{gs8}}{g_{m8}}) + 1} \tag{2.10}$$

with $w_{GBW} = A_{DC} w_{p1} = \frac{g_{m1}}{C_c}$.

The DC gain, can be computed with the following Equation 2.11.

$$A_{DC} = g_{m1} g_{m7} (r_{ds2} || r_{ds4})(r_{ds7} || r_{ds8}) \tag{2.11}$$

And the first dominant pole can be found with Equation 2.12.

$$w_{p1} = \frac{1}{g_{m7}(r_{ds2}||r_{ds4})(r_{ds7}||r_{ds8})C_C} \tag{2.12}$$

Moreover, the position on the s-plane of the other poles and nulling zero can be extrapolated from these equations.

For the input CMR and output swing, no tight constraints are present because the input signal, apart from its DC voltage, is a very small signal and the risk of stage saturation is not present.

Slew Rate is important, but a value of at least 5 V/us can be enough in order to assure that the amplifier output is settled before applying new voltage at its input.

Of course, this OTA works in *Unity Gain Buffer* configuration with the additional purpose of decoupling the output ATCs chain from the conversion part.

Some design specification are reported in Table 2.6

Specification	Result
V_{DD}	1.1 V
I_{Bias}	not given uA
Gain Band Width	10 MHz
OL Gain	65 dB
Phase Margin	60 deg
SR +	5 V/us
SR -	5 V/us

Table 2.6. Unity Gain Buffer OTA specifications

The final topology is shown in Figure 2.11 where all the transistors were sized in order to assure a very big open loop gain and fast operation.

First of all, the first differential stage was sized starting from the PMOS input couple that was implemented with a non-minimum channel length and some other structures were added maintaining a number of stacked devices less than 5 because of limited supply voltage of this technology.

A custom current mirror was exploited in order to generate the required currents within the two OTA stages (i.e. small current for the differential stage and big current for the output stage producing a gain boost).

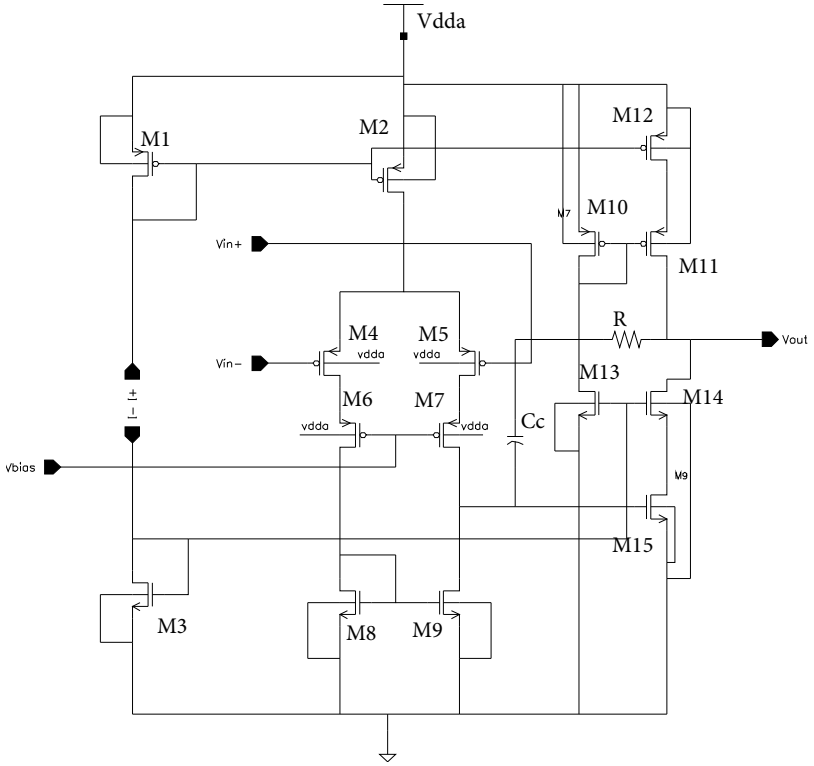

Figure 2.11. Implemented CMOS OTA

Then the second gain stage was introduced in order to increase the amplifier output resistance and as consequence, gain.

Finally, the compensation block was implemented following the design tips reported in [18] and [20] but most of the work has been carried out by means of EDA tools (i.e. Cadence Spectre simulator).

The MOST transistors size are reported in Table 2.7, in addition, the C_C and R values were set to 220 fF and 22 kΩ respectively, referring on the C_{gs15} for finding the correct value for the compensation capacitor but also using the following Equation 2.13.

$$GBW = \frac{g_{m1}}{2\pi C_C} \tag{2.13}$$

with C_C that cannot be smaller that 3 times the equivalent capacitance on the gate of M15 to ground and grater than 2-3 times C_L [20].

The device channel lengths were set to a dimension bigger than minimum depending on the gain required.

MOST	W	L
M1	4 µm	1 µm
M2	2 µm	1 µm
M3	480 nm	2.6 µm
M4-M5	14 µm	400 nm
M6-M7	8 µm	800 nm
M8-M9	380 nm	1 µm
M10-M11	10 µm	1 µm
M12	10 µm	1 µm
M13	5 µm	1 µm
M14	12.5 µm	1 µm
M15	12.1 µm	1 µm

Table 2.7. MOS transistors aspect ratios for OTA

3 System Simulations

3.1 DAC performance measurements

In order to characterize a digital to analog converter in terms of performance, some measures must be performed both in the code and frequency domain. For the static performance, the main measures are:

- **Static Gain and Offset**
- **Differential Non Linearity DNL**
- **Integral Non Linearity INL**

And for the spectral domain the following can be defined:

- **Signal-to-noise Ratio, SNR**
- **Signal-to-(noise + distortion) Ratio, SNDR**
- **Sporious Free Dynamic Range, SFDR**
- **Total harmonic distortion, THD**

3.1.1 Static Gain and Offset

An ideal DAC should convert each digital word starting from all-0 N bit vector into an analog value that belongs to a straight line.

By the way, in a real circuit, this is not true and the converted values are quite different from the ideal ones and reported in Figure 3.1 where the minimum voltage increment possible (between two adjacent digital words) is called **Least Significant Bit**, $LSB = \frac{V_{REF}}{2^N}$ [20].

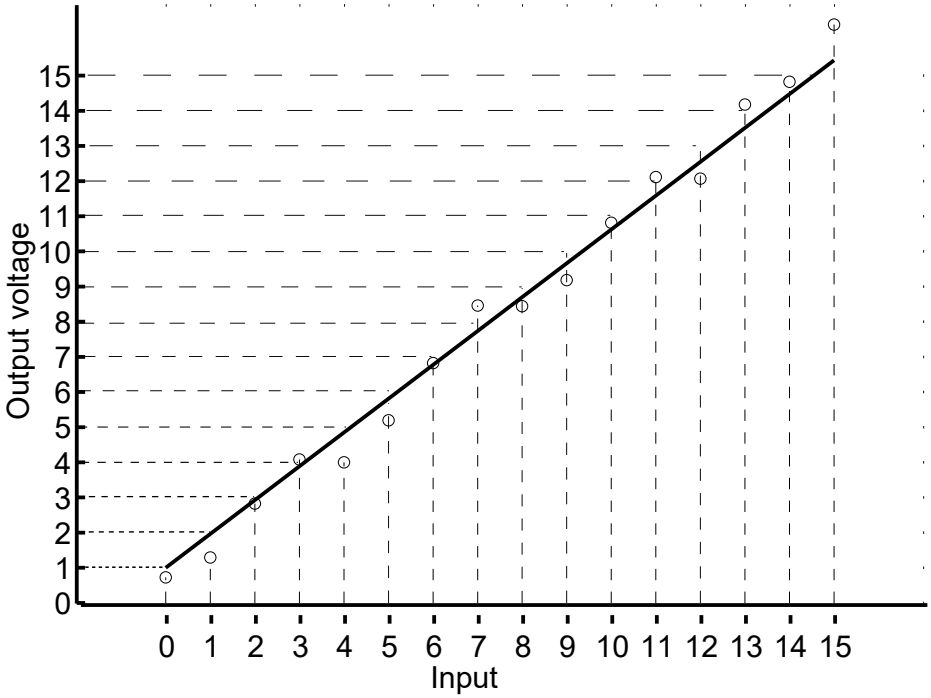

Figure 3.1. Static transfer characteristic of a non-ideal 4-bit DAC

In addition, the resulting transfer characteristic shows two static non-ideality that are **Offset** and **Gain** as showed in Figure 3.2.

For this reason, the output can be represented with Equation 3.1.

$$Y(X(n)) = k \cdot X(n) + Y_{OS} \tag{3.1}$$

where k is the gain error and Y_{OS} is the introduced offset voltage.

3.1.2 DNL

The differential nonlinearity describes the deviation from the ideal value between two adjacent steps (that differ just for one LSB) and is reported in Equation 3.2.

$$DNL(X(n)) = \frac{Y(X(n)) - Y(X(n-1)) - k}{k} \tag{3.2}$$

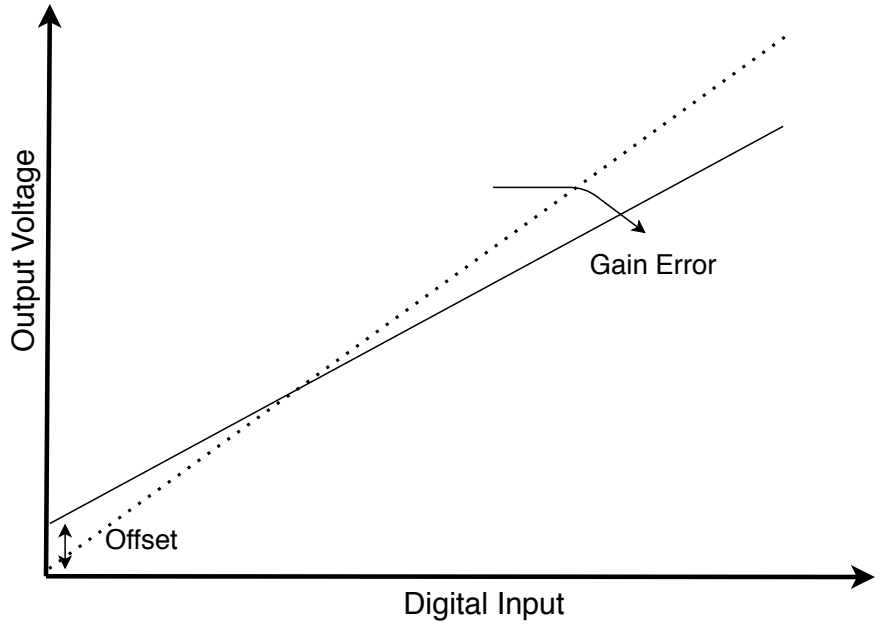

Figure 3.2. DAC Offset and Gain errors

3.1.3 INL

The integral nonlinearity describes the deviation from the ideal curve[1] Y(X(n)) [20] and is expressed by Equation 3.2.

$$INL(X(n)) = \frac{Y_{ideal}(X(n)) - Y(X(n))}{k} \tag{3.3}$$

On the other hand, in order to test the DAC into the frequency domain, a single-tone sinusoidal signal is quantized and then sent to the converter.

The nonlinear behavior will introduce some harmonic distortion on the output that should be analyzed and taken into account.

3.1.4 SNR

Signal-to-noise ratio defines the difference between the power of signal and the power of noise, that usually is independent from the input signal (e.g. thermal

[1]Usually if INL is known, DNL is not considered.

noise) and appears with a flat spectral shape. In order to compute it we refer to Equation 3.4.

$$SNR = \frac{P_S}{P_n} \tag{3.4}$$

3.1.5 SNDR

This measure is similar to SNR but also distortion is taken into account as reported in Equation 3.5.

$$SINAD = SNDR = \frac{P_S}{P_n + P_D} \tag{3.5}$$

In addition, rearranging the terms and computing in dB, we get the result reported in Equation 3.6.

$$SINAD = 6.02 \cdot N + 1.76\text{dB} \tag{3.6}$$

where N is the number of bits.

3.1.6 Sporious Free Dynamic Range, SFDR

This measure defines how much linear is the converter because is related to the ratio between the signal power and the power of the bigger unwanted harmonic as reported in Equation 3.7.

$$SFDR = \frac{P_S}{P_{H_{max}}} \tag{3.7}$$

3.1.7 Total harmonic distortion, THD

This measure takes into account the normalized weight of all the spurious components, starting from the 2-nd harmonic as reported in Equation 3.8.

$$THD = \frac{\sum_{i=2}^{\infty} P_{H_i}}{P_S} \tag{3.8}$$

In the case of the successive approximation DAC, has been demonstrated with simulations that the THD is approximately equal to $(-5 + 20log_{10}(\delta))$ dB, where $\delta = \frac{C_1 - C_2}{C_1 + C_2}$ is the capacitor mismatch parameter.

This involves a reduction in linearity which is unacceptable in high performance applications but can be good enough in the bio-medical field.

3.2 DAC Characterization

3.2.1 Test Bench

In order to test the DAC properly, a test bench platform was designed in *Cadence Virtuoso* and simulated in SPECTRE.

The MATLAB script **Random Bit Generation** reported in Appendix D.1, computes the digital vectors for testing the DAC static performance.

For the dynamic part, a SIMULINK project was created where a sinusoidal signal was quantized using a quantization block.

The binary data in both cases were transformed in text files, used as source for a pwlf generator within the test bench, allowing the structure to work in the desired way.

In addition, considering the layout routing, a 10fF input capacitance for each ATC was considered, getting a total equivalent load capacitance in the order of 500fF that was used during simulations also for the OTA design stage.

The DAC testing system is reported in Figure 3.3.

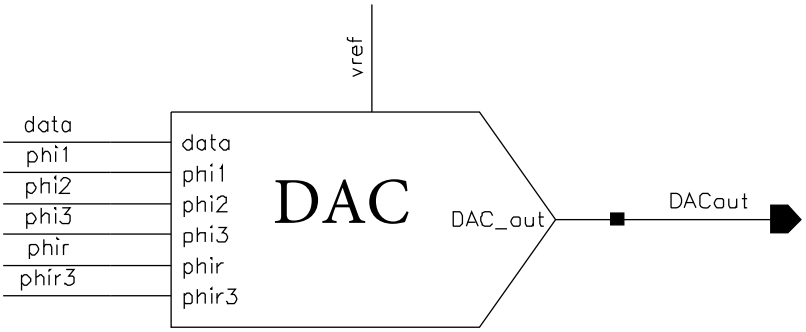

Figure 3.3. DAC test bench

The several control/clocking waves were generated using standard voltage sources working as reported in Figure 3.4 where also the output analog waves can be noted.

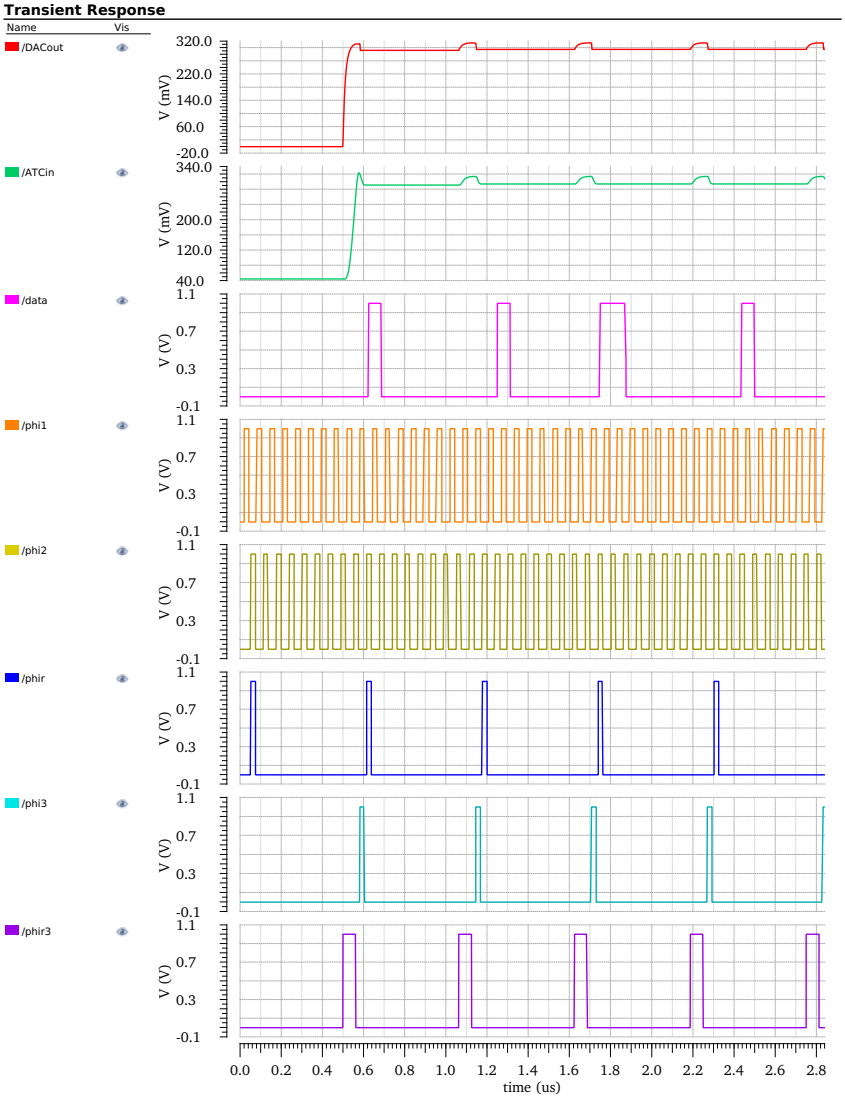

Figure 3.4. DAC clocking waves

3.2.2 Simulation Results

In order to compute the DAC output voltage, the Virtuoso Built-in calculator was used producing .csv file of data sampled on the obtained curves for each kind of simulation.

After this process, the samples where post-processed in MATLAB with the various programs reported in Appendix. D.

Gain error and Offset

Figure 3.5 shows the DAC measured output voltage with respect to the ideal one (highlighted in blue).

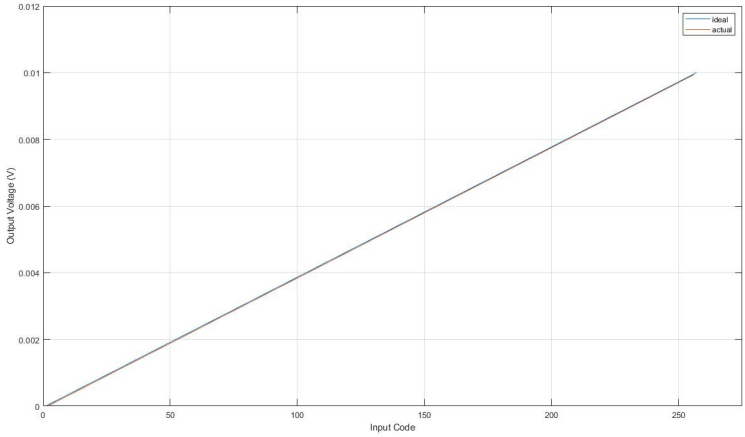

Figure 3.5. 8 bit DAC transfer characteristic

The introduced offset is equal to -28 uV and gain error is very small, in fact, the real curve is almost parallel to the real one. In addition, no glitches are present because of the single serial bit operation.

DNL

The result for 8 bit DNL simulation, using the generated incremental digital code, is reported in Figure 3.6 where a quasi-uniform distribution of the differential non linearity can be noted.

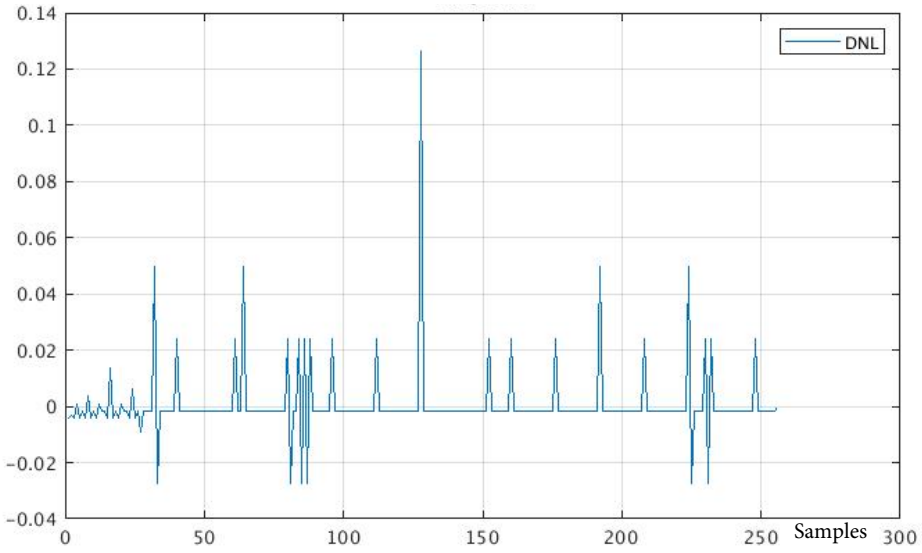

Figure 3.6. DAC DNL

INL

The result for INL simulation, using the generated incremental digital code, is reported in Figure 3.7 and, as expected the result is something that accumulates as the code goes towards the MSB.

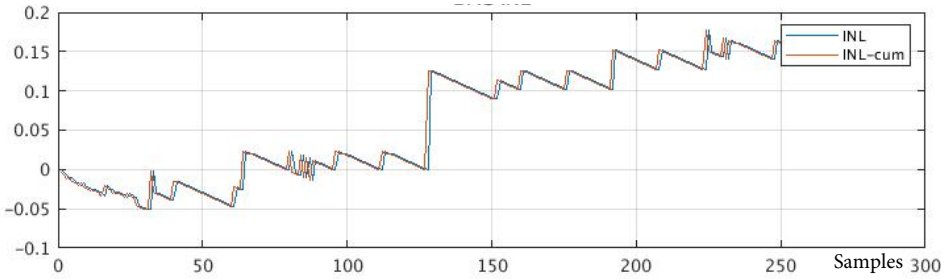

Figure 3.7. DAC INL

has can be noted, the biggest error is present when the MSB switches from 0 to 1 causing an up-level shift of the voltage stored by the redistribution capacitors.

Transient Analysis

A simple transient simulation with 1 MHz quantized sinusoidal input wave was performed obtaining the result reported in Figure 3.8, with a sampling rate slightly bigger than 2 MHz in order to respect the Nyquist theorem with a coherent sampling window.

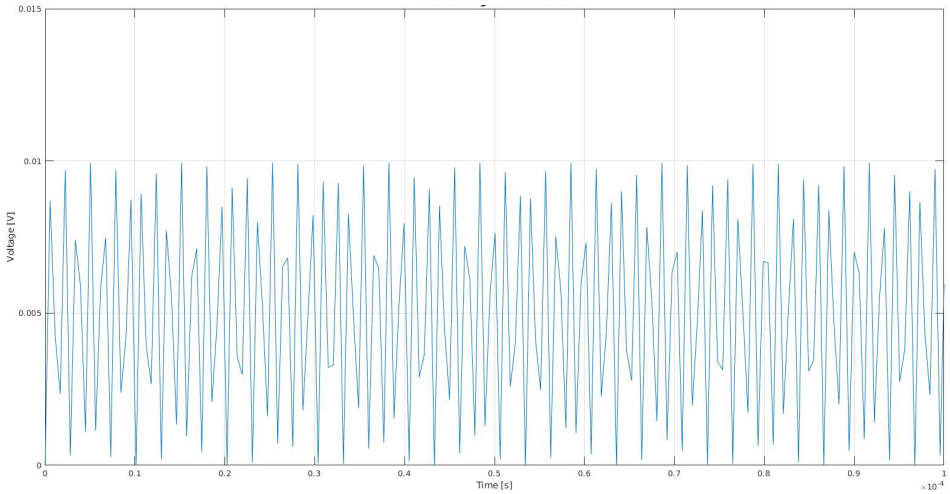

Figure 3.8. DAC transient simulation

The sinusoidal wave shows the correct amplitude equal to 10 mV_{pp}.

Transient Noise analysis and FFT

Due to the fact that is very important to know how much noise the DAC introduces, **TranNoise** analysis were run in *ADE L* Virtuoso environment. Running this kind of analysis it is possible to test the real behavior of the circuit at the expenses of longer simulation times.

All the MOST within the structure were selected as noise sources and the noise F_{max} set equal to 500 MHz for both 8 bit and 10 bit implementations with standard capacitors value and reduced capacitors value (reduction factor equal to 3) checked in order to prove if a capacitance reduction may or may not introduce unacceptable noise . Performing the 8192 points FFT simulation using coherent sampling and the script reported in Appendix D.4 the results for the 8 bit resolution are reported in Figure 3.9 and Figure 3.10

The obtained results for the 10 bit resolution are reported in Figure 3.11 and Figure 3.12.

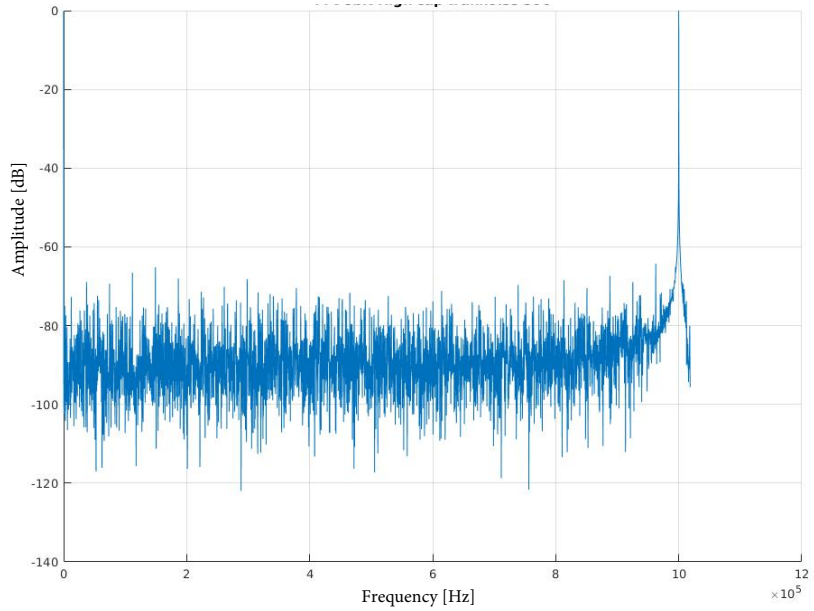

Figure 3.9. Transient Noise Simulation for 8 bit DAC with standard capacitors

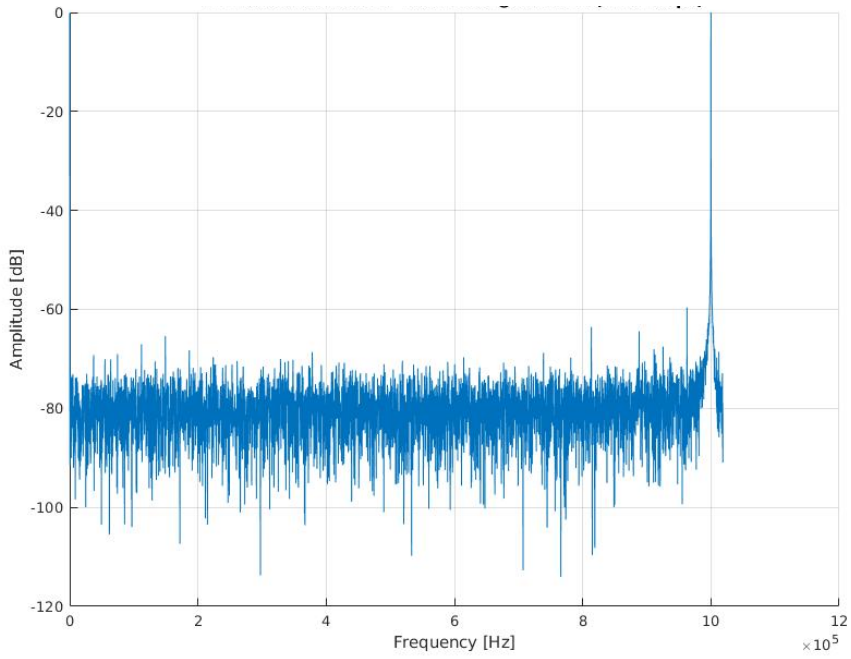

Figure 3.10. Transient Noise Simulation for 8 bit DAC with reduced capacitors

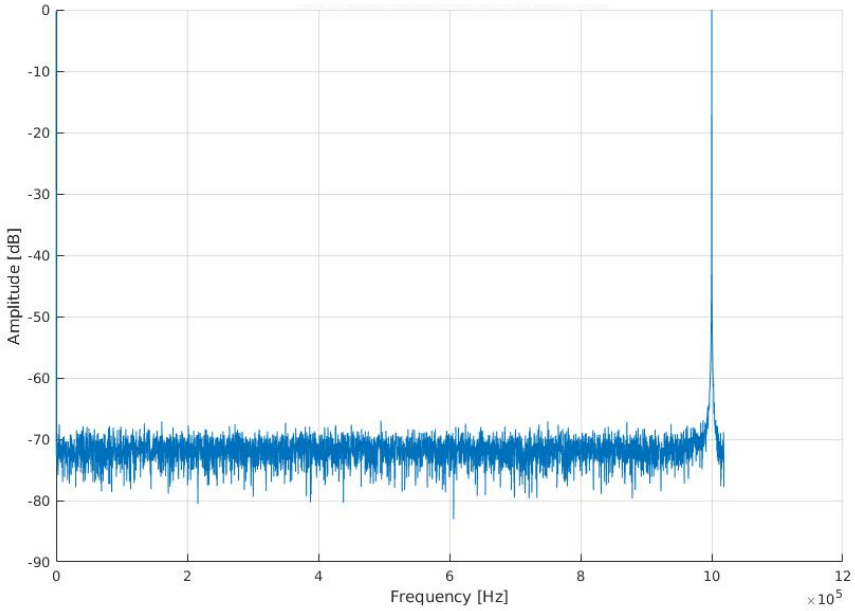

Figure 3.11. Transient Noise Simulation for 10 bit DAC with standard capacitors

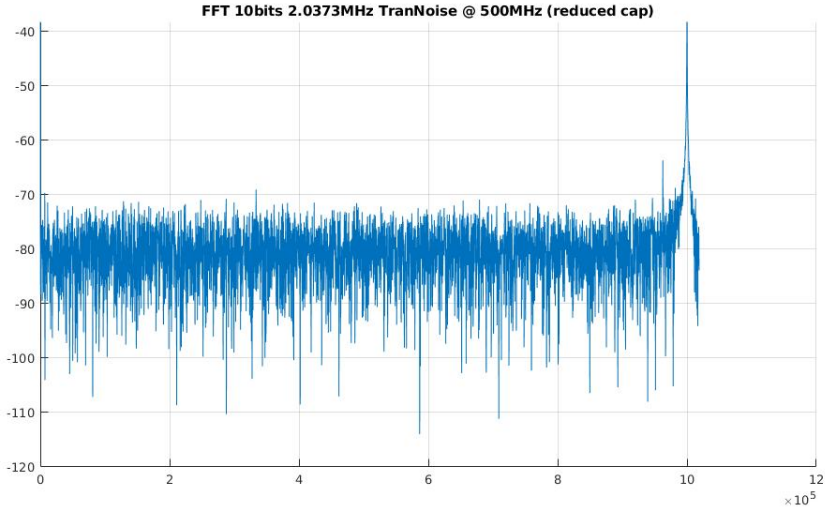

Figure 3.12. Transient Noise Simulation for 10 bit DAC with reduced capacitors

Then, performing the FFT on a TranNoise simulation where the input signal was a sine sampled at 9.978 MHz in order to visualize the spectrum on a bigger frequency range, the plot reported in Figure 3.13 was obtained.

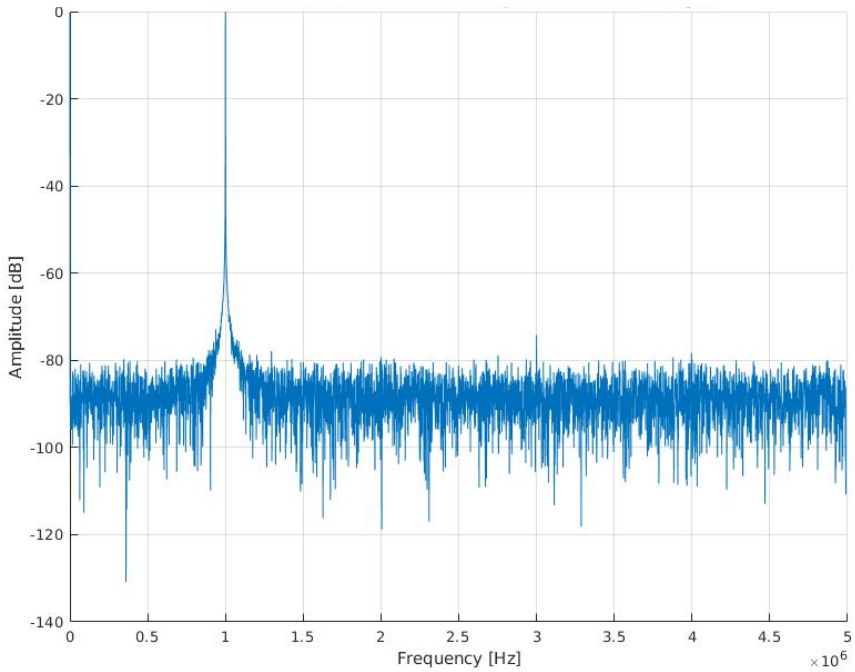

Figure 3.13. FFT performed on DAC output

As can be noted from the showed FFT plots, if the redistribution capacitor size goes down, the noise contribution becomes unacceptable according to Johnson-Nyquist theory, reducing SINAD and accuracy, resulting also with a reduced **Equivalent Number of Bits** ENOB.
The DAC results are reported in Table 3.1.

Specification	Result
V_{ref}	160 mV
NOB	8-10
Offset	-20 uV
DNL	0.12 (LSB)
INL	0.18 (LSB)
SNR	65 dB
Noise	<LSB/2

Table 3.1. Summary of simulated results for proposed DAC

3.3 OTA Characterization

3.3.1 Test Bench

In order to test the OTA properly, a test bench platform was designed in *Cadence Virtuoso* and simulated in SPECTRE. The design flow started with the specs definition and then with the transistor sized in agreement with values reported in Table 2.7.

The device operation in saturation has been checked using a Virtuoso built-in function called *Circuit Condition*.

Then, after this first simulation stage, the Miller compensation block was introduced checking the pole-splitting effect on the frequency response in order to satisfy the project specifications.
The implemented test bench for the buffer OTA is shown in Figure 3.14 and obtained results are presented in the following sections.

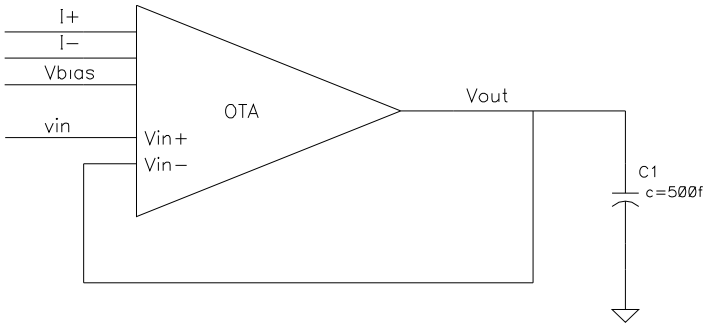

Figure 3.14. Unity Gain Buffer test bench

3.3.2 Simulation Results

DC Analysis

For the DC analysis, one of the two differential inputs has been fixed and the other swept in order to check the transfer static characteristic of the amplifier.

From the figure, can be noted that in the range of interest (around 300 mV) the OTA behaves as required by the project specifications.

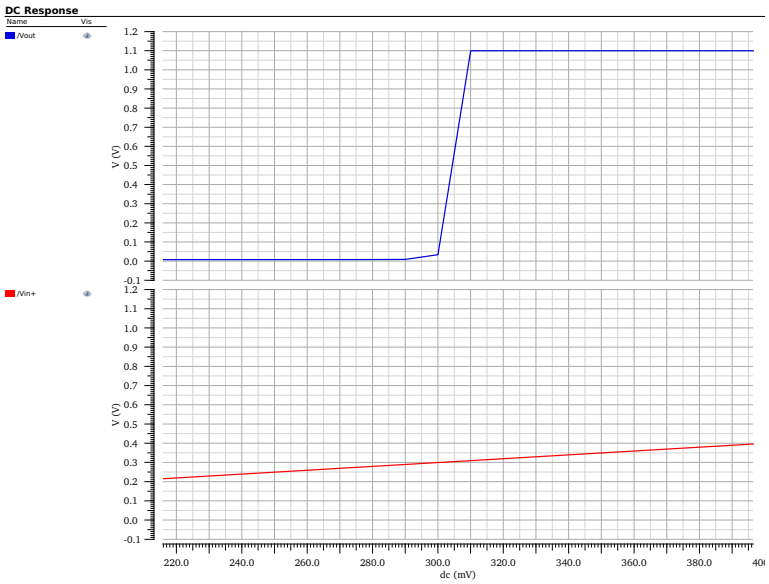

Figure 3.15. OTA DC analysis

AC Magnitude and Phase

In this stage, the circuit was linearized by the tool and a 1V wave was applied in order to check its frequency behavior.

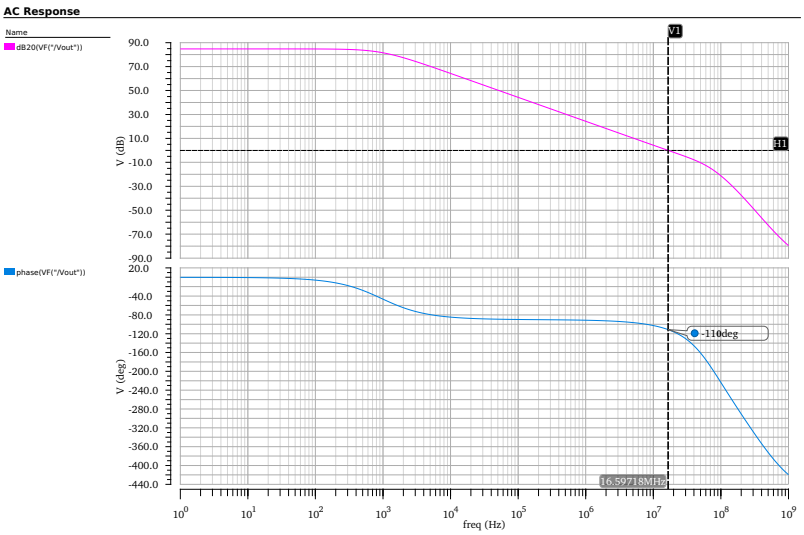

Figure 3.16. OTA gain and phase

The OTA shows a very high gain and a good phase margin with values that match and go fairly beyond the project specifications.

Transient Analysis

For the transient analysis, an input sinusoidal wave was used and selected in order to match the project specifications ($300mV_{DC}$, $10mV_{pp}AC, 1MHz$).

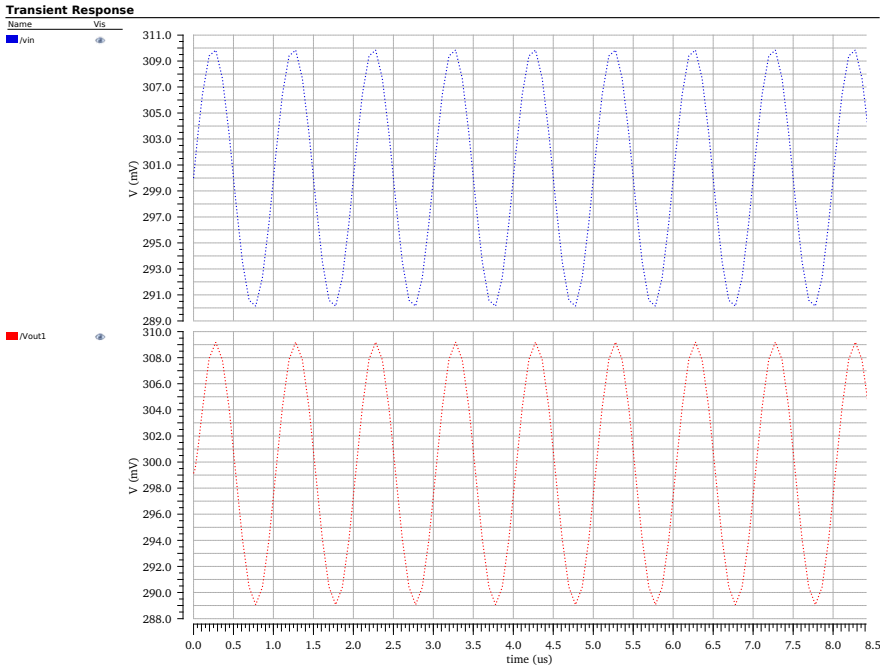

Figure 3.17. OTA transient behavior

From the figure, a good operation can be noted, where, the OTA in unity gain buffer operation perfectly copies the input signal.

Slew Rate

For the slew rate specification, no specific constraints were required by the project but this parameter has been tested in order to verify that the amplifier was fast enough to manage samples coming from the DAC. A pulse with very fast rise and fall times was applied at the input and the amplifier output was measured in order to obtain the slew rate parameter.

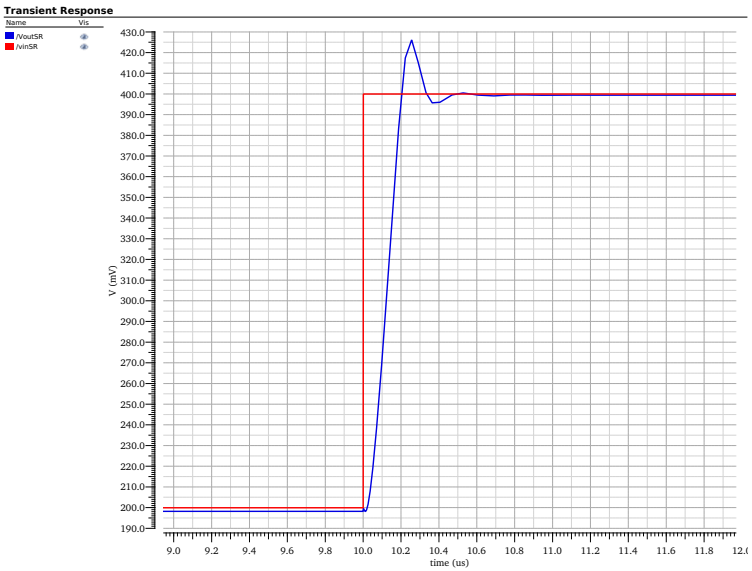

Figure 3.18. OTA SR+

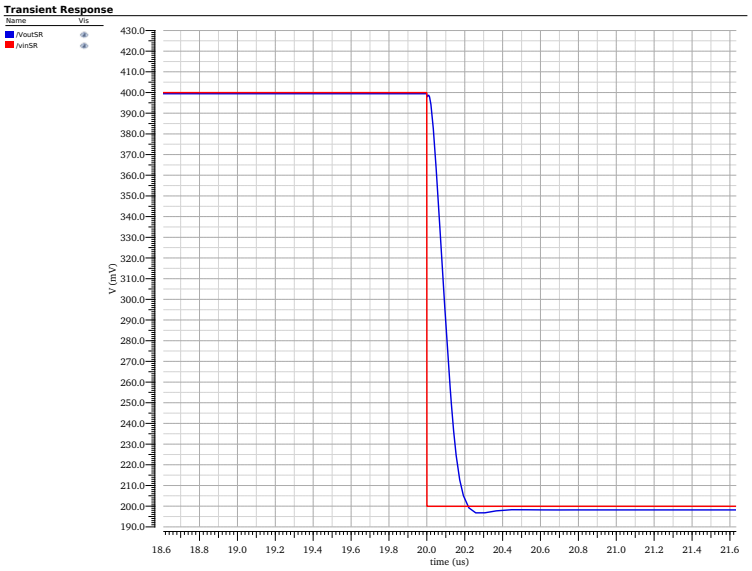

Figure 3.19. OTA SR-

The results for the designed amplifier are reported in Table 3.1.

Specification	Result
V_{DD}	1.1 V
I_{Bias}	4 uA
Gain Band Width	16.5 MHz
OL Gain	82 dB
Phase Margin	70 deg
SR +	6.7 V/us
SR -	6 V/us
Power Consumption	7.48 uW

Table 3.2. Summary of simulated results for proposed OTA

Conclusions

In this thesis project, a novel conversion architecture has been proposed and designed in order to real-time check the correct functional operation of a Multi-Sensor Time Mode Compressed Sensing integrated circuit converting 8 or 10 bits digital words produced by a test vector generator to 10 mV_{pp} full-scale analog signals ready to drive the asynchronous Analog-To-Time conversion channels. More precisely, the obtained results have proven that the proposed system can work in this kind of application (e.g. ECG bio-signals) resulting with good performance in terms of linearity and noise.

The circuit behavior shows that possible problems can occur during the post-layout verification because of capacitors mismatch that is the main limiting factor for the converter accuracy and that cannot be lower than 0.1% also trying to check process variation and to laying out the integrated circuit with the greatest possible care.

Therefore, due to the fact that this is a constant error source on the DAC static characteristics (DNL, INL, etc), the input data and operation can be modified using one of the error-shaping architectures published in literature that exploit the symmetry of the first part of the DAC, oversampling and averaging the results and consequently reducing errors.

In addition, a drawback is given by the circuit latency that is greater than other kind of conversion circuit topologies and can be a limiting factor in all those applications that require high speeds and resolutions.

In conclusion, the design and simulation results of a Unity Gain Buffer employing a Miller OTA used as DAC output stage, have been presented.
Specifically, the proposed amplifier can achieve a maximum gain of 82 dB, Phase Margin of 70 deg, UGB of 16.5 MHz at a power supply of 1.1 V with a small ICMR that can be increased introducing an high-swing stage and reducing the number of stacked transistors making this aspect a subject for possible future improvements.

A Johnson-Nyquist noise on capacitors

Ideally capacitors are lossless devices, but, when they operate in combination with a resistance in an RC chain as in the case of MOST-based sampling circuits, they produce a white thermal noise contribution that is called kTC noise. This contribution has a power spectral density (PSD) that can be estimated with Equation A.1 when the device operates in strong inversion and corresponds to a noise current [21].

$$S_{i_{TH}}(f) = \frac{8}{3} k T g_m \tag{A.1}$$

where k is the Boltzmann constant, $k = 1.38 \cdot 10^{-}23 \frac{J}{K}$, T the device absolute temperature in Kelvin and g_m the device transconductance.

Developing the model, the result is independent from the resistance R because the contribution of it in terms of frequency band reduction is counterbalanced by its noise contribution. The kTC noise, in both mean-square and RMS voltage [22] are reported in Equation A.2 and Equation A.3.

$$\bar{v_n^2} = \frac{4 k_B T R}{4RC} = \frac{k_B T}{C} \tag{A.2}$$

$$v_n = \sqrt{\frac{4 k_B T R}{4RC}} = \sqrt{\frac{k_B T}{C}} \tag{A.3}$$

Where k_B is the Boltzmann constant, T is the absolute temperature in Kelvin, C is the capacitance value and the noise bandwidth of the RC circuit is $\Delta f = \frac{1}{4RC}$

An extreme case is the zero bandwidth limit called the reset noise left on a capacitor by opening an ideal switch. The resistance is infinite, yet the formula still applies; however, now the RMS must be interpreted not as a time average, but as an average over many such reset events, since the voltage is constant when the bandwidth is zero. In this sense, the Johnson noise of an RC circuit can be seen to be inherent, an effect of the thermodynamic distribution of the number of electrons on the capacitor, even without the involvement of a resistor.

Thermodynamic fluctuations of charge stored within the capacitors produce this noise that remains on it also when the sampling circuit is open and its random value is characterised by a certain standard deviation.

Often this is a limiting factor for tight requirements circuits as reported in Table A.1.

Capacitance	$\sqrt{\frac{k_B T}{C}}$
1 fF	2 mV
10 fF	640 uV
100 fF	200 uV
1 pF	64 uV
10 pF	20 uV
100 pF	6.4 uV
1 nF	2 uV

Table A.1. Noise of capacitors at 300 K

B Charge Injection

When a MOST switch is turned OFF, a certain amount of charge is injected through the source and drain terminals as reported in Figure B.1 and this phenomenon is called **Charge Injection**.

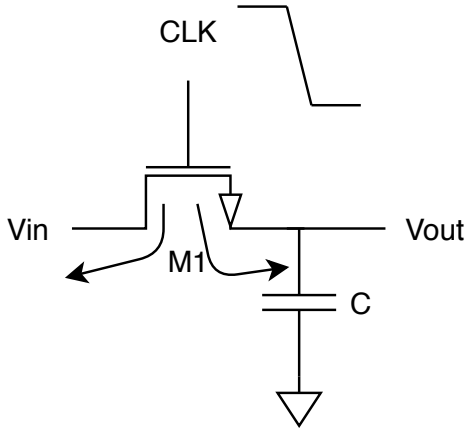

Figure B.1. Charge injection in a sampling circuit

The charge stored within the transistor channel can be evaluated with Equation B.1.

$$Q_{ch} = WLC_{ox}(V_{DD} - V_{in} - V_{TH}) \tag{B.1}$$

where W and L are the device physical dimensions, C_{ox} the gate oxide thickness, V_{DD} the supply voltage, V_{in} the input signal and V_{TH} the threshold voltage.

The amount of charge that goes towards the left terminal is absorbed by the source causing no effect but the part that goes toward the sampling capacitor remains stored inside it causing a voltage error on the output.

The amount of it can be evaluated with Equation B.2.

$$\Delta V = \frac{WLC_{ox}(V_{DD} - V_{in} - V_{TH})}{2C} \tag{B.2}$$

Observing the equation, it can be noted that the error is directly proportional to the product WLC_{ox} that is a physical parameter (WL is the transistor size chosen by the designer and C_{ox} is related to the specific CMOS technology) and inversely proportional to the sampling capacitor C.

Developing a model based on the previous equations and supposing that all the charge is injected inside the sampling capacitor, it can be proven that charge injection contributes with tree types of errors when sampling is performed with MOS transistors. The first one is related to an offset, the second one to a gain and the third one to a nonlinear effect which add up ending with a new output characteristic with respect to the ideal switch case.

C Clock feedthrough

When the switch control input is asserted, it causes the control circuit to apply a large voltage change (from V_{DD} to V_{SS}, or vice versa) at the gate of the CMOS switch. In this scenario, clock transitions are coupled to the sampling capacitor and the amount of injected voltage error signal depends on the magnitude of the gate-drain and gate-source overlap capacitances and is independent from the input level.

This effect is called **Clock Feedthrough** and is reported in Figure C.1.

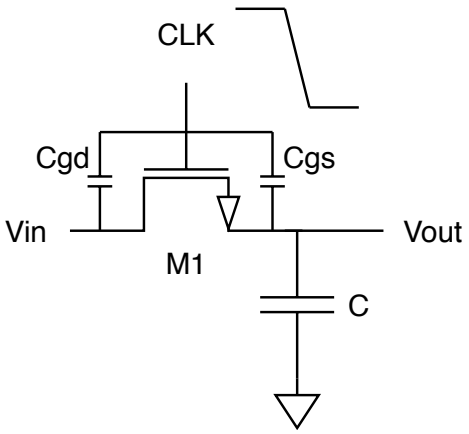

Figure C.1. Clock feedthrough in a sampling circuit

CF can be evaluated with Equation C.1.

$$\Delta V = V_{CLK} \frac{WC_{ov}}{WC_{ov} + C} \tag{C.1}$$

where V_{CLK} is the clock level and C_{ov} the total overlap capacitance per unit width.

A good way to reduce this error consist in the use of the so called dummy switch where the charge injection effect is suppressed limitating also the clock feedthrough because of the total equivalent capacitance.

D MATLAB Scripts

D.1 Random bit Generation

```
% Random bit generation (for 8bit DAC)
clear all

Fs = 16e+6; %sampling rate
T = 1/Fs; % Signa period
N=256;
t1=zeros(1,N);
t2=zeros(1,N);
i=4;
d=0;

t1(1)=0;
t2(1)=61.5e-9;
for i=2:N
t1(i)=t1(i-1)+62.5e-9;
t2(i)=t2(i-1)+62.5e-9;
end

%random bit vector
d = round(rand(1,length(t1)));

%write to file
fname = 'random_bit.txt';
fileID = fopen(fname,'w');
fprintf(fileID,'%d %d\n%d %d\n',[t1; d; t2; d]);
fclose(fileID);
```

D.2 DNL/INL Computation

```
% DNL and INL computation
i=0;
t_bit=62.5e-9;
NOB=8; %number of bits
Conversion_time=t_bit*NOB;
time_digital_words=linspace(0, Conversion_time*2^NOB, 2^NOB ); %2^NOB
output_ideal_voltage=linspace(0, 10e-3, (2^NOB)+1); %2^NOB

DNL=zeros(2^NOB,1);
INL=zeros(2^NOB,1); %correct with INL after changing its
value within the import data script
INL_test= zeros(2^NOB,1);
D=size(DAC_output_DNL);
INL_cum=zeros(2^NOB,1);

% Max DNL computation

for i=1:D-1

    DNL(i)=(((DAC_output_DNL(i+1)-DAC_output_DNL(i))-
    output_ideal_voltage(2))/output_ideal_voltage(2));
    % output_ideal_voltage(2) = 1 LSB
    INL(i)=(((DAC_output_DNL(i)+2.893e-5)-
    output_ideal_voltage(i))/output_ideal_voltage(2));
    %INL_test(i)=(((DAC_output_DNL(i)-y(i)))); % Subtracts the best-fit
    %straight line from the ideal transfer curve
    %INL(i)=(DAC_output_DNL(i+1)/39.0625e-6)-(i+1);
end

Max_DNL=max(DNL)
Max_INL=max(INL)
Max_INL_cum=max(INL_cum)
INL_cum=cumsum(DNL); % Cumulative sum of elements.


figure(1)
plot(output_ideal_voltage)
grid on
hold on
```

```
plot(DAC_output_DNL+2.893e-5) % offset error removed
hold off
legend('ideal','actual-8bit')

figure(2)
plot(DNL)
legend('DNL')
grid on

figure(3)
plot(INL(1:255))
hold on
plot(INL_cum)
%plot(INL_test)
hold off
grid on
legend('INL', 'INL-cum')
grid on
```

D.3 FFT Computation

```
%% Script to perform fft with coherent sampling
% clear all
close all
clc

%% Load the file(s) to read

% Check folder
%myFolder = '/users/agancedo/scripts/';
%myFolder = '/users/agancedo/insertchipname/SIM/PAPERSIMS/';
myFolder = '/users/cristianmastro/Desktop/Project/FFT/';
%myFolder = '/users/tmsp/agancedo/SIM/CELLTEST/';

fprintf('Source folder: %s\n',myFolder);
if ~isfolder(myFolder)
  errorMessage = sprintf('Error: The following folder does not
  exist:\n%s', myFolder)
  uiwait(warndlg(errorMessage));
  return;
```

```
end

% Run through every file with a specific name
%simulation = '100HzPULLDOWN'; % Change this name to match the simulation

%               simulation = 'name of the file to process without .csv';
simulation = 'FFT_10bit_highcap_500'
%files = dir([myFolder strcat('/formatted',simulation,'*.txt')]);
files = dir([myFolder strcat(simulation,'.txt')]);
numFiles = length(files);
fprintf('Number of files to process: %i\n', numFiles);
namestring=[];
ii = 0;

% Create the vector containing the file names
for ii = 1:numFiles
    if isempty(length(namestring))
        namestring = convertCharsToStrings(files(ii).name);
    else
        namestring = [namestring;convertCharsToStrings(files(ii).name)];
    end
end

%% Compute and plot fft and time signal
N = 2^13; % Number of samples
%skip = 5;
skip = 0
Ncyc = 4021; %number od cycles within the simulation window?
fin = 1e6;
% Ncyc = [3,3,3,11]; % Number of cycles
% fin = 0.01*10.^(2:5); % Input frequency (Hz)
fs = N*fin/Ncyc; % Sampling frequency (Hz)
linewidth = 2;
linewidthnoise = 1;
noisecolor = 'b';
fundamentalcolor = 'g';
harmonicscolor = 'r';
nHarmonics = 3; % Number of harmonics
ii = 0;
% numFiles=1; % DEBUGGING ONLY
```

```
for ii=1:numFiles
 fs = N*fin/Ncyc;
 %fs = N*fin(ii)/Ncyc(ii); % Recalculate for different input frequencies
 pulsewidth = importdata(strcat(myFolder,namestring(ii,:)));
 time = linspace(0,Ncyc/fin,N);
 %time = linspace(0,Ncyc(ii)/fin(ii),N); % Recalculate to match input
 % signal = sin(2*pi*fin*time); % Test signal DEBUGGING ONLY
 signal = pulsewidth(skip+1:end);
 signal = signal(1:N)
 if N ~= length(signal)
     disp('Number of samples do not match number of data points');
 else
Y = fft(signal,N);
%P2side = (abs(Y).^2);
P2side = abs(Y/N); % Double Side Band spectrum
P1side=P2side(1:(N/2)+1); P1side(2:end-1) = 2*P1side(2:end-1);
% Compute Single Side
% Band Spectrum
frequency=(0:(N/2))*fs/N; % Frequency points
fundamental=max(P1side(2:end)); % Find the value of the fundamental
index = find(P1side==fundamental); % Find the position of the fundamental
 % Noise and Distortion bins
NDbins = [P1side(2:(index-1));P1side((index+1):end)];
 % Signal to Noise and Distortion Ratio
SINAD = 20*log10(rms(fundamental)/rssq(NDbins));
% Distortion bins up to the 5th harmonic
Dbinsindex = (2:nHarmonics)*index-(1:nHarmonics-1);
Dbins = [P1side(Dbinsindex)]; % Values of the FFT at the harmonics

% Total Harmonic Distortion
THD = 20*log10(rssq(Dbins)/rms(fundamental));
ENOB = (SINAD-1.76)/6.02; % Effective Number of Bits
fprintf('Input frequency: %iHz\n', fin);
%fprintf('Input frequency: %iHz\n', fin(ii));
fprintf('Power of the main tone = %fdB\n', 20*log10(fundamental));
%fprintf('SINAD = %fdB\n', SINAD);
%fprintf('THD = %fdB\n', THD);
%fprintf('ENOB = %f bits\n\n', ENOB);
P1norm = P1side./fundamental;
P1normdb = 20*log10(P1norm);
```

```
%% Plot spectrum
figure(ii)
%subplot(2,1,1)
hold on
% Plot 1, 2 and 3 are the noise, fundamental and harmonics graphic
% properties
plot1 = plot(frequency,P1normdb);

%% COMMENT HERE TO DISABLE FUND+DIST+NOISE 3 COLORS
%{
plot1 = plot(frequency(1:index-1),P1normdb(1:index-1),'Color',...
    noisecolor,'LineWidth',linewidthnoise,'DisplayName','Noise');
plot2 = plot(frequency(index-1:index+1),...
    P1normdb(index-1:index+1),'LineWidth',linewidth,...
    'Color',fundamentalcolor,'DisplayName','Fundamental');
    %'Marker','*','MarkerIndices',index);
plot1 = plot(frequency(index+1:Dbinsindex(1)-1),...
    P1normdb(index+1:Dbinsindex(1)-1),'Color',noisecolor,...
    'LineWidth',linewidthnoise,'DisplayName','Noise');
for b = 1:length(Dbinsindex) % Plot harmonics and noise
plot3 = plot(frequency(Dbinsindex(b)-1:Dbinsindex(b)+1),...
    P1normdb(Dbinsindex(b)-1:Dbinsindex(b)+1),...
    'LineWidth',linewidth,'Color',harmonicscolor,...
    'DisplayName','Harmonics'); %'Marker','x','MarkerIndices',Dbinsindex(b)-b)
if b ~= length(Dbinsindex) % Plot quantization noise between harmonics
    plot1 = plot(frequency(Dbinsindex(b)+1:Dbinsindex(b+1)-1),...
        P1normdb(Dbinsindex(b)+1:Dbinsindex(b+1)-1),...
        'Color',noisecolor,'LineWidth',linewidthnoise,...
        'DisplayName','Noise');
    end
end
plot1 = plot(frequency(Dbinsindex(b)+1:length(frequency)),...
    P1normdb(Dbinsindex(b)+1:length(frequency)),...
    'Color',noisecolor,'LineWidth',linewidthnoise,...
    'DisplayName','Noise');
set(gca, 'XScale','log') % Set the X axis to log scale
grid on, grid minor % Grid on with finer grid
hold off
title(strcat('SINAD=',num2str(SINAD,'%2.2f'),'dB,',...
    ' THD=',num2str(THD,'%2.2f'),'dB,',...
    ' ENOB=',num2str(ENOB,'%2.2f'),'bits'))
```

```
%ylim([min(P1normdb) 0]); xlim([0 max(frequency)]); % Axis limits
ylim([-120 0]); xlim([0 max(frequency)]); % Axis limits
xlabel('Frequency [Hz]'); ylabel('Normalized Power [dB]');
legend([plot1 plot2 plot3]);
%}
%% UNTIL HERE

%{
%% Plot time domain signal
subplot(2,1,2)
hold on
plot(time,signal*1e6) % Plot time signal
hline = refline([0 mean(signal*1e6)]); % Plot mean of the signal
hline.Color = 'r'; % Color of the reference line
hold off
grid on
%title(['$sin(\overline {5*pi/4} ) = ',num2str(sin(5*pi/4)),'$'],
'interpreter','latex','FontSize',16)
title(strcat('Time domain. Mean Pulsewidth = ',...
    num2str(mean(signal*1e6),'%.4f'),'\mus.',...
    ' \DeltaPulsewidth = ',...
    num2str(max(signal*1e9)-min(signal*1e9),'%.2f'),' ns'))
axis tight
ylim([floor(min(signal*1e6)*1e3)/1e3 ...
    ceil(max(signal*1e6)*1e3)/1e3]);
xlim([0 max(time)]);
yticks(linspace(floor(min(signal*1e6)*1e3)/1e3,...
    ceil(max(signal*1e6)*1e3)/1e3, 5))
ytickformat('%.3f')
xlabel('Time [s]'); ylabel('Pulse width [\mus]');
%}

%% Save the figure as a file
%{
figure(ii);
fig = gcf; % Get current figure
fig.Units = 'centimeters'; % Set units of the figure
fig.Position(3) = 20; % Set figure width
fig.Position(4) = 16; % Set figure height
%opts.fontType = 'Times'; % Set font type
%opts.fontSize = 9; % Set font size
```

```
set(gca,'LooseInset',max(get(gca,'TightInset'), 0.02));
fig.PaperPositionMode   = 'auto';
saveas(fig,strcat(myFolder,'FIGURES/',erase(namestring(ii),...
    ["formatted",".txt"]),'.jpg')); % '-dpng', '-r600'
%}
 end
end
grid on
```

Bibliography

[1] Emil Jovanov, Aleksandar Milenkovic, Chris Otto, and Piet C De Groen. "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation". In: *Journal of NeuroEngineering and rehabilitation* 2.1 (2005), pp. 1–10.

[2] David L Donoho. "Compressed sensing". In: *IEEE Transactions on information theory* 52.4 (2006), pp. 1289–1306.

[3] Omer Can Akgun. "An asynchronous pipelined time-to-digital converter using time-domain subtraction". In: (2018), pp. 1–5.

[4] Mauro Mangia, Fabio Pareschi, Valerio Cambareri, Riccardo Rovatti, and Gianluca Setti. "Rakeness-based design of low-complexity compressed sensing". In: *IEEE transactions on circuits and systems I: Regular Papers* 64.5 (2017), pp. 1201–1213.

[5] Dai Zhang, Ameya Bhide, and Atila Alvandpour. "A 53-nW 9.1-ENOB 1-kS/s SAR ADC in 0.13-μ m CMOS for Medical Implant Devices". In: *IEEE Journal of Solid-State Circuits* 47.7 (2012), pp. 1585–1593.

[6] Dai Zhang and Atila Alvandpour. "A 3-nw 9.1-enob sar adc at 0.7 v and 1 ks/s". In: *2012 Proceedings of the ESSCIRC (ESSCIRC)*. IEEE. 2012, pp. 369–372.

[7] Reza Lotfi, Rabeeh Majidi, Mohammad Maymandi-Nejad, and Wouter A Serdijn. "An ultra-low-power 10-bit 100-kS/s successive-approximation analog-to-digital converter". In: *2009 IEEE International Symposium on Circuits and Systems*. IEEE. 2009, pp. 1117–1120.

[8] Sang-Hyun Cho, Chang-Kyo Lee, and Jong-In Song. "Design of a 1-Volt and $\mu-$power SARADC for Sensor Network Application". In: *2007 IEEE International Symposium on Circuits and Systems*. IEEE. 2007, pp. 3852–3855.

[9] Iffa Binti Sharuddin and L Lee. "An ultra-low power and area efficient 10 bit digital to analog converter architecture". In: *2014 IEEE International Conference on Semiconductor Electronics (ICSE2014)*. IEEE. 2014, pp. 305–308.

[10] Maruthi Chandrasekhar Bh and Sudeb Dasgupta. "A 1.2 volt, 90nm, 16-bit three way segmented digital to analog converter (DAC) for low power applications". In: *2009 10th International Symposium on Quality Electronic Design*. IEEE. 2009, pp. 447–450.

[11] Masanori Otsuka, Syuzo Ichiki, Toshiro Tsukada, Tatsuji Matsuura, and Kenji Maio. "Low-power, small-area 10 bit D/A converter for cell-based IC". In: *1995 IEEE Symposium on Low Power Electronics. Digest of Technical Papers*. IEEE. 1995, pp. 66–67.

[12] Dominik Przyborowski and Marek Idzik. "Development of a general purpose low-power small-area 10 bit current steering CMOS DAC". In: *2009 MIXDES-16th International Conference Mixed Design of Integrated Circuits & Systems*. IEEE. 2009, pp. 256–261.

[13] Mamun Bin Ibne Reaz and Torikul Islam Badal. "Design of a Low-Power 10-Bit DAC in 130 nm CMOS Technology". In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. IEEE. 2019, pp. 762–766.

[14] Behzad Razavi. "The current-steering DAC [a circuit for all seasons]". In: *IEEE Solid-State Circuits Magazine* 10.1 (2018), pp. 11–15.

[15] Ricardo E Suarez, Paul R Gray, and David A Hodges. "All-MOS charge-redistribution analog-to-digital conversion techniques. II". In: *IEEE Journal of Solid-State Circuits* 10.6 (1975), pp. 379–385.

[16] Jesper Steensgaard, U-K Moon, and Gabor C Temes. "Mismatch-shaping serial digital-to-analog converter". In: *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*. Vol. 2. IEEE. 1999, pp. 5–8.

[17] Behzad Razavi. *Design of analog CMOS integrated circuits*. Tata McGraw-Hill Education, 2002.

[18] G Palmisano, G Palumbo, and S Pennisi. "Design procedure for two-stage CMOS transconductance operational amplifiers: A tutorial". In: *Analog Integrated Circuits and Signal Processing* 27.3 (2001), pp. 179–189.

[19] Namkyu Park, Dong-Jun Yang, et al. "1/f noise of NMOS and PMOS transistors and their implications to design of voltage controlled oscillators". In: *2002 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium. Digest of Papers (Cat. No. 02CH37280)*. IEEE. 2002, pp. 59–62.

[20] Willy M Sansen. *Analog design essentials*. Vol. 859. Springer Science & Business Media, 2007.

[21] Richard Schreier, José Silva, Jesper Steensgaard, and Gabor C Temes. "Design-oriented estimation of thermal noise in switched-capacitor circuits". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 52.11 (2005), pp. 2358–2368.

[22] Rahul Sarpeshkar, Tobias Delbruck, and Carver A Mead. "White noise in MOS transistors and resistors". In: *IEEE Circuits and Devices Magazine* 9.6 (1993), pp. 23–29.

[23] R Jacob Baker. *CMOS: mixed-signal circuit design*. john Wiley & sons, 2008.

[24] R Jacob Baker. *CMOS: circuit design, layout, and simulation*. John Wiley & Sons, 2019.

[25] Adel S Sedra, Dean Emeritus Adel S Sedra, Kenneth Carless Smith, and Kenneth C Smith. *Microelectronic circuits*. New York: Oxford University Press, 1998.