Politecnico di Torino

Doctoral dissertation

Doctoral Program in Electrical, Electronics and
Communications Engineering (cycle XXXIII)

Curriculum in “Electronic Devices” in convention with
the National Institute for Nuclear Physics (INFN)

CMOS distributed signal processing systems for radiation sensors

Andrea Di Salvo

Supervisors:
Prof. Angelo Rivetti, Supervisor (INFN)
Prof. Goano Michele, Co-supervisor (Politecnico di Torino)
Dr. Manuel Dionisio Da Rocha Rolo, Co-supervisor (INFN)

Doctoral Examination Committee:
Laura Gonella, Referee (University of Birmingham, UK)
Valerio Re, Referee (University of Bergamo, Italy)

2021

Abstract

The purpose of this thesis is to address aspects of digital signal processing for multi-channel ASICs employed to read out radiation sensors. These detectors are characterized by a high degree of segmentation and they are coupled to integrated front-end electronics embedding many channels operating in parallel. The number of channels found on a typical front-end ASICs ranges from 16 to 128, but some applications require up to thousands of processing block integrated on the same die. In these systems, low power consumption is often at a premium. In tracking detectors used in particle physics, for instance, the mechanical infrastructure needed to cool down the system adds a significant material on the particle path that can blurry the reconstructed tracks. In satellite-based applications, power consumption is always a primary concern for obvious reasons.

Modern deep-submicron CMOS technologies allow to achieve densely packed systems, but the non-recurring engineering costs can be problematic. It is therefore preferable to develop flexible circuits that can be re-used for different sensors, thus serving multiple experimental setups. Consequently, multi-channel mixed-signal ASICs where fast ADCs are mated to digital processors become increasingly interesting in the radiation instrumentation community. In the following, this architecture will be referred to as a *full sampling system* to distinguish it from other more specialized topologies. Although very common in mainstream electronics, the full sampling approach has not been widely used so far in radiation instrumentation because the power consumption of fast ADCs prevented their use in most multi-channel systems, where the power consumption has to be kept well below 10 mW/channel. The typical resolution required to these converters in the applications of interest in this thesis is between 8 and 12 bits, while the sampling frequencies ranges from 10 MS/s to more than 1 GS/s. A sampling frequency in the 10 - 100 MS/s range is however already adequate in many cases. The digital processor must be able to extract signal features such as the energy of the impinging particle and the time of occurrence of the event. Compared to commercial products, the design of these systems is challenging. For instance, on the analog side the distribution of a clean reference voltage for the ADCs is very critical, as many ADCs have to operate in parallel, thus heavily loading the reference voltage. On the digital side a very low-power consumption and fast processing must be simultaneously achieved. Additionally, radiation hardness to both total ionizing dose and single event effects are required to guarantee the circuit functionality. Hence, all these aspects have to be taken into account, requiring the development of custom solutions.

The aim of this work was to investigate digital signal processing solutions for low-power radiation detector systems. The work focused in particular on the application of ADC digital calibration algorithms optimized for the radiation detection environment and on the design of low power processors for feature extraction. A custom high-level simulation environment that allows to compare different options was also deployed.

The thesis is organized as follows. Chapter 1 introduces key concepts about radiation sensors which are relevant to discuss key issues addressed in the rest of the work. The most frequently used topologies in the implementation of multi-channel front-end ASICs for radiation detectors are briefly reviewed, with an emphasis on the state of the art of full sampling circuits.

Chapter 2 starts with the discussion of ideal analog-to-digital converter characteristics and the causes of the error conversion such as the quantization error, thermal noise, jitter and capacitance mismatch. The discussion focuses in particular on SAR ADCs, which are of particular interest for this work. A high-level code that allows to model the relevant ADC errors was developed on purpose to provide inputs to test different error correction strategies.

In Chapter 3 key techniques used in digital calibration of ADCs are reviewed. A section describes in detail the chosen digital algorithm, named Offset Double Conversion, selected to mitigate the non-linearities of a converter. A following section explains how the calibration processor was implemented, describing the finite state machine, all its functionalities and the developed solutions for an on-chip implementation.

Chapter 4 discusses the digital signal processor which in a full sampling system follows the ADC. A dedicated section considers the difference between the FIR and IIR filters implementation. An overview on the digital signal processor is presented with the description of the circuit at block level. All the features implemented in this unit are described, starting from the anti-glitch system based on self-adjustable thresholds. The bisection algorithm for the square root computation is discussed. This algorithm is used to prepare to the baseline restoration which benefits of the same dynamic thresholds. A pile-up unit to manage the rejection of events too close in time has also been developed. The mathematical description of a digital $CR{-}RC^4$ pulse shaper is derived and its signal-to-noise ratio performance is discussed. The trapezoidal filter used belongs to the deconvolution filter class and it is referred to as mobile window deconvolution. Its explanation is reported as well as the circuit employed for the energy extraction of the signal. A snippet code is provided to show in more detail the strategy adopted in the calculation of this feature. The chapter ends with a descriptions of the digital filters implemented to calculate the time of occurrence of the event.

Chapter 5 is divided in two parts to separately show the physical implementation of the calibration block and the digital signal processor. For both circuits the post P&R simulation carried out in the typical corner are reported. The power analysis was only executed for the 65 nm CMOS technology, highlighting the contribution of each sub-block in the total power budget. For what concerns the calibration circuit, a prototype has been realized in 110 nm CMOS process. This chip includes a segmented SAR ADC with a nominal resolution of 12 bits, the correction block, two serializers and LVDS banks. A section reports the experimental results and discusses their analysis. Finally, a protection against single event upset and multiple bit upsets is presented.

Chapter 6 is reserved to data compression topics and three different techniques are discussed. These methods were used in a preliminary study about data compression for RD53A which is a pixel readout integrated circuit designed for the CMS experiment upgrade at CERN and fabricated on silicon. The last chapter summarizes the key outcome of the work, presenting conclusion and outlooks.

Ai miei genitori,
senza i cui sforzi tutto questo
non sarebbe stato possibile,
non così almeno.

A tutta la mia famiglia.

Acknowledgments

Writing a PhD thesis is a hard work which is essentially a patchwork of knowledge, tips and people.

Thus, I would like to thank the INFN staff of Torino that have helped me over these years. I express my first thank to Angelo Rivetti for this experience in a research context and also for scientific, technical and informal conversations, both in person and at a distance. I am deeply grateful to Floarea Dumitrache for the bonding of the prototype chip, Francesco Rotondo for the design of the test board and Richard Wheadon for his essential work with the DAQ system during the tests as well as our small talks. Without yours contributions, I would not have been able to collect the results reported in the current dissertation. I acknowledge the important support of Giulio Dellacasa for the power analysis of my digital processors.

I would also like to thank Manuel Da Rocha Rolo, Marco Mignone and Barbara Pini while I was joining the Arcadia project.

Special thanks go to Pisana Placidi and Giuseppe Baruffa for the discussions on data compression and the days I spent in Perugia.

Thanks also to my PhD colleagues and the Postdoctoral Researchers for their help. In addition, I really appreciated the nice time together because it is important as the technical support.

Lastly, I am grateful to my family, my friends and who shared this period with me. Many times even just yours listening and a couple of words have lightened my efforts. Thank you, with love.

Andrea

Contents

Chapter 1

ASIC architectures for radiation sensor read-out

A radiation detector is a device designed to capture the interaction of an impinging particle, converting part of its energy into a suitable electrical signal. This event is characterized by some features such as signal amplitude and time of occurrence that can be extracted with appropriate data manipulation. To get the idea we can think of a common digital camera. The light source coming from the environment is collected, then it is converted into an image that freezes its information on a memory bank. The image contains a representation of space features related to the status of the surrounding world at a given time. Radiation detectors are widely used in many fields, from astrophysics to high-energy physics or medical applications. In order to record and analyze the interactions, an appropriate front-end electronics coupled with the sensor is necessary. To achieve high performance in terms of energy discrimination, spatial accuracy and timing resolution, the sensor area is partitioned into many independent channels. This implies a massive parallelism in the processing of the generated data. Such technological challenge can be fulfilled through the implementation of Application Specific Integrated Circuits (ASIC), in Very Large Scale of Integration (VLSI) technologies.
Figure 1.1 depicts a typical acquisition channel that is primary formed by the sensor which is the sensitive area where the interaction occurs. In the case represented in the picture, the sensors are organized in a pixel matrix where each unit is independent from the others. When the signal is formed, its amplitude could be too small to be directly manipulated by the following electronics. Thus, a first stage amplifies the original signal to be fed to the next blocks. In the image this role is accomplished by a Charge Sensitive Amplifier (CSA). In a straightforward implementation, the CSA output is filtered and digitized by an Analog-to-Digital converter (ADC). Therefore, the data can be further elaborated on-chip with a Digital Signal Processor (DSP) to

extract the features of interest . Eventually, also data compression schemes can be applied on the processed values to satisfy specific data transmission requirements. Alternatevely, the converted samples can be directly sent out off chip and acquired by a Field Programmable Gate Arrays (FPGA).

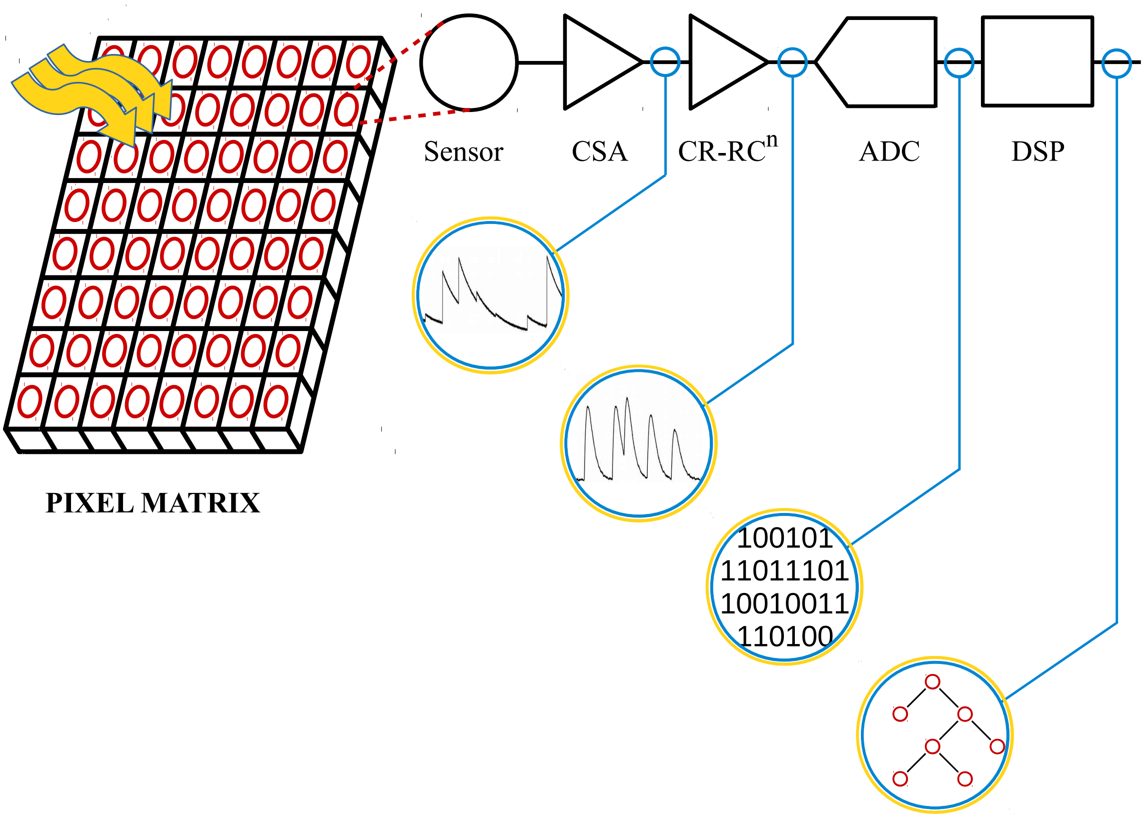

Figure 1.1: Block representation of a typical read-out channel composed by a sensor, a preamplifier stage (CSA), a pulse shaper (CR-RC), an analog-to-digital converter (ADC) and a digital signal processor (DSP).

1.1 Radiation sensors

Many different types of radiation sensors are employed, depending on the application and its requirements. Due to the complexity and broadness of the topic, only key aspects relevant to front-end electronics design are discussed here. More detailed explanations about these devices can be found in the references quoted in the bibliography [1] [2] [3]. An interesting description of the history of particle detectors is also available in [4]. The basic understanding of the sensor properties as well as of the specific application needs is a critical step to properly optimize the design of the front-end. In Figure 1.2 a representation of a ionization chamber is reported, where two conductive plates (the grey darker strips) confine a sensitive material (the grey lighter zone). In order to maintain an electric field within this volume, a voltage V_{bias} is applied between the two electrodes. The volume which defines the sensitive part of detector can be filled by gasses or their mixture. However, semiconductor materials, and silicon in particular, became more and more popular in the recent decades. Insulator materials are not suitable to be employed because of the polarization phenomenon. When a charged particle or a photon crosses the sensor, its interaction with the atoms of the device generates electron-ion pairs if the volume is formed by a gas and electron-hole pairs in case of a semiconductor. If a photon is energetic enough, it can set an electron free. This electron loses in turn its energy by ionization in the medium, thus creating a detectable signal. It is fundamental to highlight that the detected signal is due to the induction of the charge carriers that move inside the sensitive volume and it is not formed by their collection at the electrodes. Indeed, when the conductive plates gather all the charges, the signal is no longer observable by the front-end electronics. For a comprehensive analysis of this process the reader is referred to [5]. If the sensor is not equipped with an internal amplification system, the signal is generated by the primary charges originated after the interaction. The charge released can be considerably small. For instance, in a silicon detector, the charge can be $5 \cdot 10^{-17}C$ in case of 1 keV x-ray and $4 \cdot 10^{-15}C$ for a minimum ionizing particle traversing a 300 μm thick device. When the sensor is composed by a gas, the signal is not strong enough to be detected therefore a further ionization is required. The process that strengthens the signal is the avalanche one which takes place when the weak primary charge is driven into a region of a high electric field. A third class of sensor is represented by multi-stage systems. An example is given by a scintillator crystal which absorbs a gamma ray, partially converting its energy into visible or near UV photons. The latters are thus collected by a photon detector exploiting the well-know photo-electric effect.

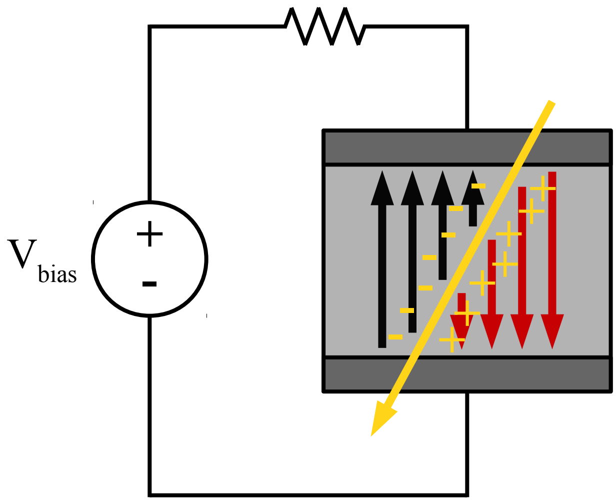

Figure 1.2: Representation of an ionization chamber.

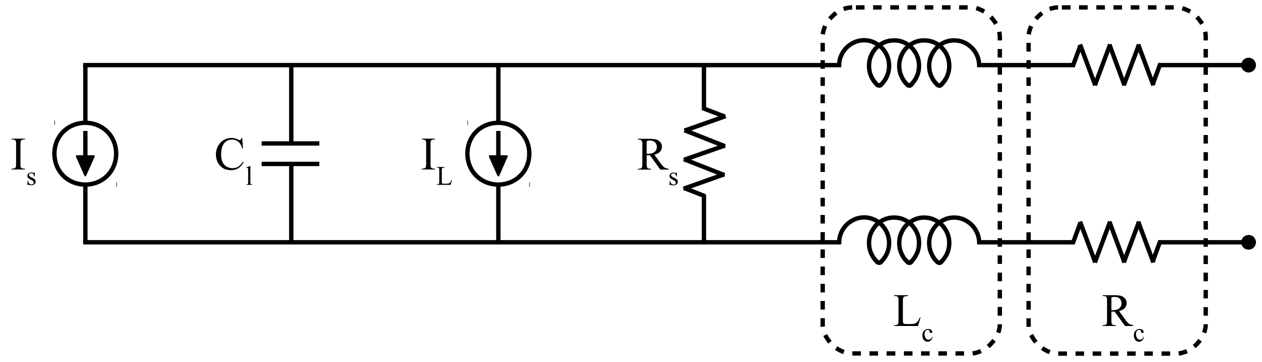

Figure 1.3: Modelling of a generic radiation sensor with an equivalent small signal circuit.

In general, a sensor can be modelled with the equivalent small signal circuit reported in Figure 1.3. Here, the current source I_s represents the sensor signal, while C_l is the capacitive load that the sensor presents to the front-end electronics. The electronic noise is closely related to this capacitance, thus sensors with smaller capacitance allow for better Signal-to-Noise ratio (SNR). I_L is another current source that models the leakage current. This quantity is also important to consider because in some application the sensor is DC coupled to the front-end amplifier. The leakage current is especially relevant for semiconductor devices since thermal process produces electron-hole pairs inside the depletion region that are later collected by the electrodes. In some cases, such as with Germanium detectors, a cooling system is mandatory to keep the leakage current low enough. The leakage current contributes also to the noise because the generation of the pairs is in itself a random process. Furthermore, it can even determine a sizeable shift of the DC operating points of the front-end amplifier. This issue can be fixed either by AC coupling or by adopting dedicated methods to compensate this undesired effect [6] [7] [8]. R_s is used to describe the intrinsic output impedance of the sensor or the physical resistor adopted to provide the bias voltage. Lastly, L_c takes into account the parasitic inductance and R_c the parasitic resistance of the connections. The model introduced so far is a generalization suitable for a large number of detectors, however the value of the parameters depends on the specific sensor design and its operating conditions. For instance, the I_s which models the timing evolution of the signal can be described by a Dirac-delta in the simplest case or can be defined as a composition of exponential terms.

In the following, three examples of radiation sensors will be discussed to show the validity of the presented model. The first one belongs to the composite class and as introduced before it is formed by two parts. Figure 1.4 illustrates the concept of a vacuum Photomultiplier Tube (PMT) which is still today one of the most popular devices to detect single photons.

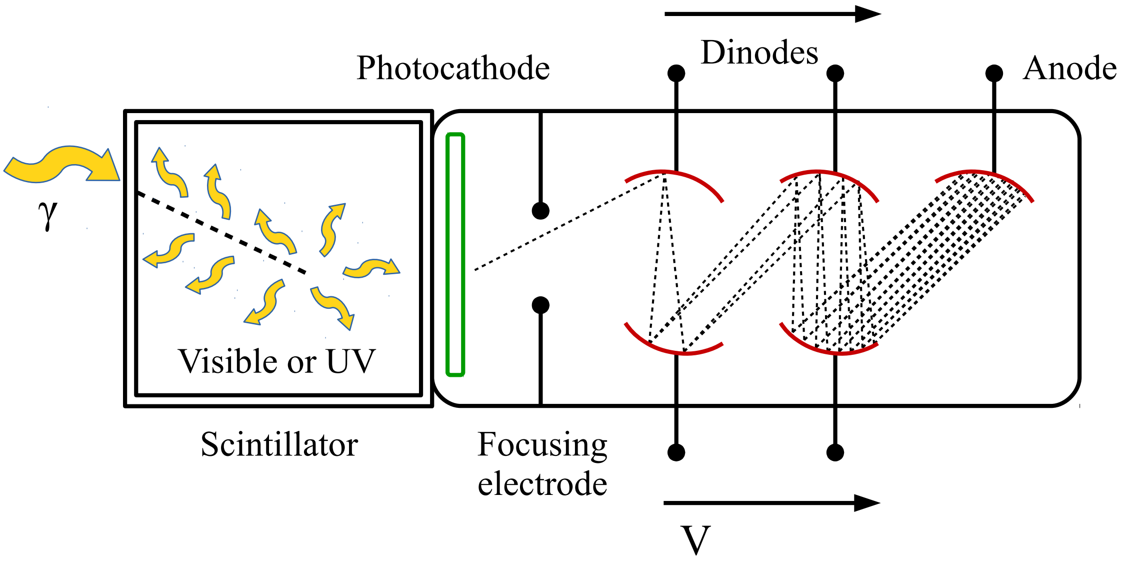

Figure 1.4: Representation of a vacuum photomultiplier tube (PMT).

The scintillator is composed by a material that absorbs the gamma ray and re-emits photons at a lower energy along the ionization path. When one of these visible or near UV photons reaches the photocathode an electron is emitted and it is focused toward the inner part of the tube by an electrode. Other electrodes named dynodes are located between the photocathode and the anode and their task is to generate a second emission. Each dynode is connected to an increasingly intense voltage in order to produce an exponential multiplication due to the avalanche process. In this way from a single photoelectron a million of electrons that are collected at the anode can be produced. The induced current on this last electrode forms the signal observed. The model previously described works well to represent this device. Here the capacitance is composed by that one between the last dinode and the anode and the capacitance generated by the interconnections.

A second example of detectors is represented by the gas-based sensors that are still very interesting for the scientific community due to the their low cost compared to the instrumented area. As explained before, this kind of devices suffers of a weak signal when implemented without an amplification mechanism. Thus also in this case a suitable electric field is applied to exploit an avalanche effect in order to generate more carriers. The intent is to guarantee a proportionality between the primary charge originated by the initial interaction and the detectable signal. Since the length of the avalanche is a function of the creation point of the initial charge, if an equivalent energy is deposited in different regions of the detector the output signals will be characterized by different amplitudes. In other words, it will no longer be possible to maintain a linear measurement. Therefore, the strategy is to bound the avalanche process to a defined region and this result can be obtained with a cylindrical geometry, where the multiplication effect is confined to a volume around a thin wire that acts as the anode. If more than one wire is used, the device is referred as multi-wire proportional chamber (MWPC) where the charges are drifted towards the anodes by the applied electric fields, inducing the signal with a multiplication factor between 10^4 and 10^5. Other solutions have been developed to increase the space resolution as well as the rate such as the Gas Electron Multiplier (GEM). These devices include in the sensitive volume a series of Kapton foils coated with Copper. Holes are opened on these surfaces that are held at a high potential difference. The diameter of the holes is typically in the range of 50 - 70 μm with a pitch between 150 μm and 200 μm. The generated electric field collects the primary charges by drift and when they cross the first foil the multiplication starts. The process is iterated on the second Kapton surface and goes on until the collecting electrode where the signal is induced. The sensor capacitance C_l is around 50 pF, depending on the area of the landing pad.

Semiconductor-based detectors are the last class of detectors considered. Their basic structure is shown in Figure 1.5 where a so called *pn junction* representation is reported [9]. This is formed by the doping process of a semiconductor crystal, namely, when a certain concentration of impurities are introduced into the crystalline structure to improve its conductivity. The concentration range is between 10^{12} - $10^{18} cm^{-3}$. Silicon is a tetravalent atom. If a pentavalent atom is introduced in the lattice, at room temperature the extra electron can be easily promoted to the conduction band due to thermal process and it becomes free to move along the lattice. In this case the dopant, named *donor*, assumes a positive charge and when the semiconductor is doped with this kind of atoms it is referred to as n-type. Conversely, if doping atom is trivalent, a vacancy in the atomic bonds occurs and an additional state is available in the bandgap. Since the energy of this new state is close to the valence band it is particularly easy for the dopant, referred as *acceptor*, to get an electron. Therefore, when the state is filled, a further negative charge is assigned to this atom. At room temperature almost all the dopants states are occupied and the valence band is populated with holes. When the semiconductor is doped with acceptors it is named p-type. Very common semiconductors employed for this kind of detectors are Silicon (Si) and Germanium (Ge), but also compound materials have been explored such as Cadmium Zinc Telluride (CdZnTe). To realize a n-type Phosphorus (P), Arsenic (As) and Antimony (Sb) are commonly used, while Boron (B), Aluminium (Al), Gallium (Ga) and Indium (In) are usually chosen for p-type doping. When these dopants are implanted on the same semiconductor (we consider here silicon) a junction is formed. At the beginning, each side is neutral from an electric point of view. However, the diffusion starts to transport the holes from the p-region towards the n-side, while electrons migration occurs in the opposite direction. As illustrated in the figure, this process leaves a positive charge on the n-side and a negative counterpart on the p-type region, respectively. Therefore, this process builds up a potential and the equilibrium is established when the force exerted by the uncovered dopant atoms compensates the thermal diffusion, limiting further migration of the carriers. Thus, the built-in potential creates a region across the junction named *depletion region* where the net current is zero. This voltage exhibits a dependence on the dopants concentration and the temperature. The depleted volume is almost free of mobile carriers, therefore it acts as a dielectric, and the undepleted sides behave as conductive electrodes since they are populated by free carriers. Because the internal electric field of the depleted volume sweeps the mobile carriers towards the plates, this device is actually an ionization chamber where the width of the depletion region can be increased by applying a reverse bias voltage V_R as shown in Figure 1.5. Hence, when a particle releases energy into the active volume electron-hole pairs are generated and their number is proportional to the deposited energy. When the mobile charged carriers are moving in the sensitive region, a current signal is inducted on the electrodes and the event can be detected. Since this kind of sensors provide a better energy resolution than the previous described classes, they are exploited for X and gamma-ray spectroscopy. In these studies the highly penetrating radiation is investigated, so the sensor must be thick enough to offer a suitable stopping power. For energy below 30 keV the Silicon can be adequately prepared for this task by drifting Lithium ions into the substrate. In this way the impurities trapped in the bulk are

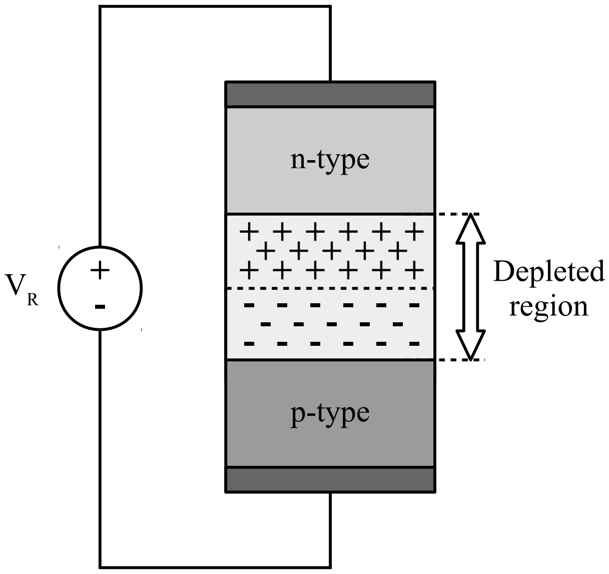

Figure 1.5: Representation of a reverse-biased semiconductor junction.

compensated and the Silicon acts as an extremely pure crtystal leading to a depletion depth of several millimiters. To explore higher energy the Silicon must be substituted with Germanium, but these detectors demand a cooling system. Compound semiconductor solutions are located halfway between an acceptable efficiency in the high energy range and room temperature operation and an examples is represented by Cadmium Zinc Telluride (CdZnTe).

Silicon detectors are also suitable for the detection of low-level light, an example being provided by avalanche photodiode (APD). Similarly to the previous processes described for the other detectors, the electric field drives the electron-hole pairs created by the photons towards an avalanche region where the impact ionization determines the extraction of other charged carriers with a gain in the range of 100 - 1000. This interval is sufficient to spot a small number of photons, however single photon sensitivity is not achieved. To detect even the single interaction the electric field of these devices can be appropriately strengthened. These detectors are called Single Photon Avalanche Diode (SPAD) or Geiger-mode Avalanche Photodiode (GAPD). They incorporate a quenching resistor R_Q in series with the diode to suppress the avalanche process. The voltage at the cathode equals a bias voltage when there is no current. If the latter is increased, the voltage across R_Q brings the cathode voltage below the breakdown point. Unfortunately, this implementation returns a constant output which is independent of the photons in the sensitive volume. To recover an output proportional to the number of incident photons a segmentation of the detector can be adopted. The analog Silicon PhotoMultipliers (SiPMs) presents a structure composed by array of parallel and independent units named microcells whereas the output is obtained as the sum of the signals produced by each diode. To have a small probability of more than one interaction into a single cell, the cell area is around 50 μm x 50 μm. In this topology, the number of firing microcells is equivalent to the number of incident photons and due to their high spacial resolution the SiPMs replaced the photomultiplier tubes in those applications where high granularity and high magnetic field are required. Both the space and time resolution can be exploited to obtain 3D images, deriving the third coordinate with time of arrival measurements. A large number of microcells can be accomodated in parallel, thus the capacitance of these detectors is between 30 pF and 300 pF and the model previously discussed is not valid anymore because the rise time of the signal could be overstimated. However, the literature offers more accurate model description of SiPMs [10]. Finally, CMOS technologies can be used to implement GAPD and they are referred to as digital silicon photomultiplier. Each cell of these device is read-out separately with a very small capacitance which improves the timing performance. In general, the electrodes of a detector can be arranged in different segmentations to obtain a space information about the event. The simplest structure is the partition of the electrodes into strips where the position of a hit is intrinsically one-dimensional. When a tilted particle deposits its charge on more than one strip it is possible to get a better spatial resolution through an interpolation exploiting the ratio between the collected charges on those strips. To increase the spatial resolution up to two-dimensions a further orthogonal segmentation can be implanted. The distance between the centers of two adjacent strips is named pitch. Its value is between 25 and 100 μm in colliding-beam experiments, while the lengths of the strips array starts from centimeters up to 30-40 cm. This sensors organization may suffer of "ghosts" hits at high densities, but this problem can be mitigated adopting a small-angle stereo, where the strips are not perfectly located at 90° as further described in [9]. Another way to achieve two-dimensional information without this kind of compromise is the pixelization of the sensor such as in the case of Charge Coupled Device (CCD) and random access pixel device. Thus, in a possible implementation, the electrodes are segmented into a surface that resembles a chessboard and each partitioned volume is physically connected with solder bumps to a dedicated readout electronics organized in a specular segmentation. The pitch of the pixels is typically between 30 and 100 μm, however the pixel size is constrained by the associated readout unit. Moreover an additional electronics is required to manage the readout control system which means more area is demanded. Furthermore, the split manufacturing increases the cost because the sensors have to be separately produced from the readout chain. In the case of hybrid pixel detectors, this additional budget is due to the bump bonding process which absorbs around half of the cost. From this point of view, a monolithic structure would be preferable because the sensor shares the same pixel area of the electronics, thus their connections are made during the fabrication, requiring a lower material budget. The pixel pitch is reduced down to 5 - 10 μm and the charge collection occurs in a very thin layer, of order μm. When these device are fully depleted (FD), their are named FD-Monolithic Active Pixel Sensors (MAPS) and in the last three decades they became increasingly of interest for particles detection, high-energy photons imaging as well as tracking in space experiments and medical applications. Some examples can be found in [11] [12] [13]. In the next section, some of the architectures developed for the front-end electronics will be discussed, leaving to Section 4.7.1 the discussion about analog amplifiers as a brief prologue to the digital filters.

1.2 Front-end architectures

As well as the sensors, many front-end architectures have been developed due to the variety of possible applications. Indeed, the purpose of the measurement determines the implementation of a specific electronics to execute a required task. For instance, in High Energy Physics (HEP) tracking detectors the aim is to provide a set of space-points to identify a particle track. In this case, a threshold comparator providing a yes/no decision followed by digital memories can be adequate. By contrast, if the energy evaluation is the main target of the experiment, such as in nuclear spectroscopy, the signal amplitude needs to be sampled with adequate resolution, so a high performance ADC is required. Another possible application is represented by Positron Emission Tomography (PET) where both the time information and the amplitude of the event must be derived. Thus, this section will present a not exhaustive list of common front-end solutions.

- The first architecture presented is also the simplest one and it is named *binary* [14] [15]. Basically, the signal is monitored by a comparator and when it is above a configured threshold a digital pulse is generated, neither processing nor storing the amplitude. Therefore, this system is suitable to get space information about the hit channels of a detector. The hits are then stored in digital memories and queued for readout. In most cases, a sparsified approach is adopted, and only the channels that fired within a given time window are readout, suppressing the zeroes.

- As introduced before, some applications require the extraction of other informations rather than the position of an event. In the case of X-ray imaging the quantity of interest is the intensity of the incident radiation [16] [17]. A *counting architecture* can address this need equipping the comparator with a counter to record the number of events occurred in a desired time frame. In this implementation two critical aspects must be considered when designing the counter. The first one is due to the switching activity because if many bits switch at the same time they can generate an higher noise. The other possible issue concerns the storage of the counter value when it is sampled since an error during this step can result into an incorrect extracted value. To get the idea, let's consider a 4-bits counter and the transition from 0111 to 1000. If a bit-flip occurs to the first bit which is the most significant one, the data recorded will be 0000. This means that the error is quantifiable as half of the whole dynamic range. To fix this potentially severe problem, pseudo-random counters can be adopted to limit the number of simultaneous transitions. Using Gray codes is useful as well because the Hamming distance between two consecutive counts is always equal to one.

- A simple approach to derive the amount of charge released by the particle in the sensor is to measure the time spent over a preset threshold by the amplifier signal. Under appropriate conditions, this time interval is proportional to the magnitude of the detected signal. Thus in binary front-ends the width of the comparator pulse can be measured to infer the charge information. In these systems, a common time-stamp is generated and distributed to the channels. The time-stamp is very often provided by a Gray counter driven by a reference clock. The time at which the leading and trailing edges of the comparator signal occur are then captured by dedicated registers in the firing channel. This method is referred to as *Time over Threshold (ToT)* and it widely reported in the literature [19] [18].

- *Time pick-off* systems must be employed when timing information are crucial in the application [20] [21]. This is the case of the particle identification in HEP and PET, where the time of flight of charge particles or photons needs to be measured with a resolution better than 100 ps. In this scenario a counting mechanism as the one described above is unsuitable because the clock frequencies would be in the range 5 - 10 GHz. Hence, while a counter still provides a coarse time base, the time elapsing between the event and a suitable clock transition must be measured by an interpolating circuit. A Time-to-Digital Converter (TDC) is a device able to assign this kind of digital tag to a selected time window with a resolution that today can even exceed 1 ps.

- While the previous architectures only store digital information, there are other approaches in which substantial more analog signal processing is carried out. A typical example is provided by the *sample and hold* systems where a hold capacitor is located between the front-end output and the read-out circuit. Its connections with the input stage and the output one are controlled by switches. As their name suggests, this topology works in two steps: during the acquisition process the input switch of the capacitor is closed, directly connecting it to the front-end, while the output switch is open. If a peak is observed, its value can be stored on the capacitor by opening the first switch. When the detected values have to be read, the output switch is closed in order to digitize it on-chip or to send it out still in analog form. The peak of a signal is the sample-target of a front-end designed with a continuous-time transfer function because this point has the highest SNR. To detect it, both the peaking and the sampling times must be synchronized.

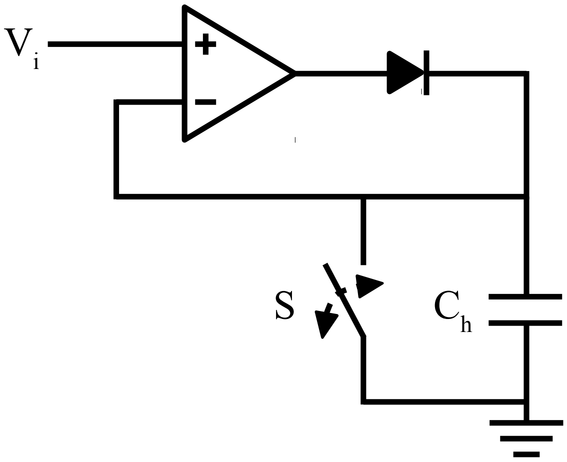

Figure 1.6: Representation of a peak detector

Figure 1.6 illustrates the concept of the *peak detectors* that are used for this purpose. At the non-inverting input (+) of the operational amplifier the front-end output V_i is presented, while the output of the op-amp forms a negative feedback with the inverting terminal (-) where an unidirectional component is added (shown as a simple diode in the figure). The switch S acts as a reset for the hold capacitor C_h, thus when it is closed C_h is discharged to ground. If $V_i \leqslant 0$ no signal is coming from the front-end and the diode is off, namely, the inverting input of the op-amp is zero. Otherwise, the diode is turned on because the difference between V_i and the negative terminal is amplified. Thus, the circuit works as a voltage follower and the voltage on C_h tracks the input one. Once the peak is overstepped, the derivative of the signal becomes negative since V_i is returning back to the baseline. Now the voltage on the hold capacitor will not follow the input signal. In fact, to track again the amplifier input the voltage on C_h should be reduced, but this would imply to sink current from the feedback path. Since the diode is unidirectional, the voltage on the inverting terminal becomes fast larger than V_i. Consequently, the output of the amplifier is pulled down and the diode itself turns quickly off while the peak value is stored in the hold capacitor. In a CMOS implementation, the diode function is implemented with a transistor and on-silicon examples of this are reported in [22] [23]. To reduce the dead time, a discriminator per channel is associated to both sample and hold and peak detectors to point out when a hit occurs. This addition allows for a sparsified readout. To further shrink the dead time, a strategy is to increase the number of storage cells available in each channel.

In this way, when a cell is already busy, another one can be available to store a new peak value.

- The last example of architecture reported is the *analog memories*. Typically, these devices are formed by an array of capacitors, each one equipped with switches for the writing and reading operations. A sample and hold process allows to implement the writing, tracking the output of an amplifier and then storing its value in a single memory unit. When this cell have to be read out, the capacitor is usually connected in the feedback path of an op-amp. A final reset clears the cell, making it available for a new writing. An advantage of analog memories is the possibility to record a full waveform, recognizing specific features such as signal pile-ups. By contrast with an analog to digital conversion process, these devices are characterized by a lower power consumption and for this reason they are suitable to capture fast transients. The acquired values can be converted either on-chip [24] or off-chip [25] at a lower speed when an interesting event is detected.
  Depending on the application, the capacitors can be implemented in CMOS processes exploiting metal-metal or MOS version. The first ones are characterized by an high linearity and a quick response, but as a drawback they have a small density around 1 $fF/\mu m^2$, making them a good choice for high speed applications. The alternative represented by the MOS capacitors is denoted by an higher density even above 5 $fF/\mu m^2$. On the other hand, this kind of capacitors are affected by a poor linearity and a smaller speed compared to the metal-metal counterpart. Thus, they are suitable when many units per channel are required.
  Another aspect to take care of is the sequencer used to control the sampling switches. An approach is to use the clock signal to drive a shift register, using it as pointer in the writing process. Another strategy can be the employment of a digital delay line which is a chain of digital buffers eventually arranged in a closed loop. In this configuration, the buffer propagation delay sets the time difference between two consecutive samples and in CMOS nodes it can achieve values smaller than 100 ps. Hence, this approach becomes interesting for applications where a very fast sampling is demanded.

1.3 Full sampling systems

By "full sampling systems" we mean in this context systems in which an ADC follows immediately the front-end amplifiers and all the relevant signal processing is carried-out in the digital domain. For many years, the power consumption of ADCs represented the bottleneck for full-sampling systems since an on-chip analog to digital conversion with an acceptable resolution required a power budget not affordable by most applications. However, in the last decade the substantial improvement of ADCs power figure has made full-sampling solutions progressively more appealing. In general, ADCs resolution in the range of 6 - 10 bits combined with a sampling rate between 50 - 100 MS/s are adequate for most applications. In this context, a read-out channel can be equipped with a Digital Signal Processor (DSP) to locally select and manipulate the generated data. These tasks can be achieved with configurable thresholds to reject the undesired values and implementing specific filters to extract useful information from the digitized signals such as the energy or the time when the event occurs. This approach leads to a more accurate data processing compared to the analog counterpart because it is not affected by the tolerances of the components. In a digital system the attained precision depends in fact on the algorithm and on the numerical representation. Furthermore, it is possible to program the key parameters of a filter, eventually adjusting them after the fabrication step. Last but not least, state of the art CMOS processes are primarily conceived for digital applications. In the following, the state of the art of full sampling systems is described.

1.3.1 ALTRO

The ALICE experiment is one of the four major detectors equipping the CERN Large Hadron Colliders (LHC), located near Geneva, Switzerland. In particular, ALICE stands for A Large Ion Collider Experiment and it is a detector designed and optimized to study the collisions of ions at ultra-relativistic energies [26]. The three-dimensional reconstruction of the particle tracks is provided by Time Projection Chamber (TPC) which is a cylindrical gas volume held to a uniform electrostatic field. With the process described in Section 1.1, the charged particles ionize the gas and they generate charged carriers later collected by the electrodes. The induced signal has a rise time smaller than 1 ns and a long tail. The latter bounds the event rate since pile-up effects can occur. Because of this the front-end electronics must implement a tail cancellation procedure. This is accomplished by the ALICE TPC Read Out (ALTRO) chip which is a mixed-signal custom integrated circuit designed for gas detectors readout [27] [28]. The chip was fabricated in a 0.25 μm CMOS process and it is formed by 16 channels where independent signals are acquired by an equivalent number of 10-bits ADCs implemented on chip. Then they are digitised and processed to be finally compressed and stored in a multi-event acquisition memory. The chip area is 64 mm^2, while the power consumption is 320 mW at 10 MHz sampling rate. The DSP partitioned the algorithms in pipelined stages where the first one is dedicated to a first baseline correction. At this stage, the digitized signal is manipulated by removing both the low-frequency perturbations and systematic effects to prepare the data for the tail cancellation. Low-frequency spurious signals are up to one kilohertz and they introduce a baseline shift in the order of one ADC count. This kind of perturbations can be generated by the temperature variation of the electronics, while the systematic effects are noise patterns that are superimposed to the original signal. For instance, the trigger of a detector can introduce this signal variations. Due to their nature, they can be removed implementing a pattern memory. Here the input signal is corrected subtracting a set of values previously stored on-chip. Then, the tail cancellation task is accomplished in the following stage where a filter was designed to remove the tail of a pulse within 1 μs after its peak. The signal was approximated by the sum of 4 exponential functions and consequently a transfer function in the Z domain was obtained. This suppression takes advantages of the digital implementation since the filter coefficients are configurable, thus the system can cover a wide range of different tail shapes. Then, a second baseline correction mechanism was carried out exploiting a moving average filter that adjusts the non-systematic perturbations of the baseline [29]. Because at the output of this step the baseline is constant within 1 LSB, a simple zero suppression scheme was adopted. In this way, the data below a programmable threshold are rejected, avoiding to read-out meaningless information. Finally, the chip also manages the data formatting where each packet is assigned its time stamp and size information for the off-chip reconstruction. The 5 kbyte data memory was designed to record up to eight acquisitions and the maximum readout frequency is 60 MHz for a a total bandwidth of 300 Mbyte/s.

1.3.2 Super ALTRO

Super ALTRO (S-ALTRO) was the natural evolution of the ALTRO chip described above. It is a demonstrator that shares the same digital processing such as the baseline restoration and the tail cancellation, but enhancing their performance in terms of power consumption, flexibility, digital noise and stability. A key feature of the S-ALTRO is that it implements on board also the front-end amplifier. The main efforts during this optimization process were put on the digital shaper designed for the undershoots and tail cancellations, since this block accounted for 84% of the digital power budget. A detailed analysis of the IIR filter architectures can be found in [30] where the hardware improvements lead to a gates reduction of 2/3 compared to the ALTRO implementation. Other aspects were also considered for the optimization of the first baseline correction mechanism and the SNR of this stage was incremented by 1.5 dB. However, this results in an increased hardware resources of slighly less than 8%. Since this block is particularly small in comparison to the whole digital processor, this variation is not significant. As well as the first baseline correction, also the second one was improved both on flexibility and digital noise reduction at the cost of 4% more hardware resources. Therefore, the overall hardware resources have been decreased by 20%, while the SNR shown a positive increment of 7.9 dB. Lastly, the power consumption was reduced to 40% changing the power supply from the nominal 1.2 V to 0.8 V at the working frequency of 40 MHz. Under this reduction, the delays increased, but without impairing significantly the overall performance.

After this architecture and hardware review, a prototype has been fabricated in 130 nm CMOS technology which occupies an area of 5.75 mm by 8.56 mm [31]. As its predecessor, S-ALTRO implemented 16 channels where the ADCs have 10-bits resolution and they work up to 40 MHz of sampling frequency. Each channel is equipped with a dedicated memory able to store up to 1000 samples for a maximum time duration of 100 μs at 10 MHz sampling rate or 25 μs at 40 MHz. For a single 10-bits sampled data, the internal data formatting system generates a 40-bit stream which provides a header, a trailer, the time stamp of the event, the configuration information during the acquisition and a set of error flags. The recorded data structure of 80 kbyte can be read out at the maximum frequency of 80 MHz, leaving a dead time of 0.2 ms. The DSP block consumes 65 mW during its normal mode, accounting 4.04 mW/channel.

1.3.3 SAMPA

Periodic LHC shutdowns have been planned to provided the necessary time for accelerator updates in order to achieve higher luminosity and ultimately higher event rate. As a consequence, all the LHC experiments were also upgraded. For what concerns ALICE, an upgrade of its TPC electronics was scheduled in order to enhance the readout capability by two orders of magnitude, thus achieving 50 kHz event rate in Pb-Pb collisions [32]. The efforts to replace the previous front-end led to a new ASIC called SAMPA [33] [34]. Thus, this chip represents the further development of ALTRO and it doubled the number of channel to 32 although the same channel components were maintained: a charge-sensitive pre-amplifier (CSA) is followed by a shaper, then a 10-bit ADC samples the signal at 10 MS/s and a DSP collects and manages the converted data. Two SAMPA prototypes were fabricated in 130 nm CMOS technology. The DSP block inherited the baseline corrections divided into a IIR filter for the low-frequency baseline variations and a moving average filter for the

faster ones and the digital shaper for the tail cancellation. In addition another baseline correction system based on limited slope baseline tracking was implemented. The purpose of this block is to provide a complementary version of the first two restoring methods that may also substitute them. Beside the zero suppression circuit, a data compression scheme was designed based on Huffman coding. The data formatting unit completes the on-chip digital signal processing.
An irradiation campaign with protons was also carried out [35] to qualify the chip for the expected Total Ionizing Dose (TID) of 2.1 kRad and a flux of 3.4 kHz/cm^2. The aim of the test was the investigation when soft errors such as the single event upset and hard error as Single Event Latchup (SEL) occur. For more detail about this topic, the reader can skip to Section 5.2.1. Briefly, the SEL is a permanent effect that can cause the overcurrent of a CMOS component, resulting in a fatal damage for the device. The high current is due to the formation of a low impedance path established between the supply voltage and the ground. The issue can be solved by power cycling the circuit. Therefore, the devices under test have not implemented specific mitigation techniques against radiation, however only the second version of the chip showed SEL events. The adequate safety measures were adopted and implemented in the next prototypes.

1.3.4 CAEN - DT5780

Industrial products provide also a large variety of DSP solutions. For instance, CAEN is a well-know company that designs both high/low Voltage Power Supply systems and Front-End/Data Acquisition modules for nuclear and particle physics experiments. An example of their products is represented by the DT5780 family [36]. These boards integrate 2 independent 16k channels Digital Multi Channel Analyzer (MCA) and they are suitable for Gamma and X-ray spectrometry. They can be used with different detectors such as the High Purity Germanium (HPGe) radiation detectors to obtain high energy resolution as well as a PMT-based sensor where the exponential tail is at least few hundreds of ns long. The on-board ADCs have a resolution of 14 bits and the sampling rate is 100 MS/s. CAEN has also developed a firmware to accomplish a digital signal processing, providing both the pulse height for the energy extraction and timing information. A digital trapezoidal filter was adopted to generate a flat top whose amplitude is proportional to the deposited energy. The systems are even able to show slices of the signal for the fine tuning of the pulse height analysis settings. This MCA are equipped also with a baseline restorer with programmable averaging similar to the ALTRO and SAMPA chips . This means that the baseline is calculated by averaging a configurable number of samples before the trapezoid, then its value is held within the time interval of the trapezoidal signal itself to properly correct the filter output. The high frequency noise can be instead reduced with an adjustable moving average filter. Another useful feature is the integrated pile-up rejection system that appropriately manages the trapezoids overlap when the event rate becomes too high. Furthermore, the dual input allows to realize coincidences and anti-coincidences involving other detectors to improve the measurements.

1.3.5 ADSP-2183 for MINIBALL

This overview terminates with the case of an industrial product employed in a physics experiment. Indeed, as tacitly anticipated in the previous section, custom systems are not always designed and implemented for all the possible applications. Here the case of MINIBALL is reported which is an high-resolution detector array that has been operational at the radioactive ion beam facility named HIE-ISOLDE (High Intensity and Energy) at CERN [37] for over 10 years. This spectrometer was formed by an array of 24 six-fold segmented high-purity germanium crystals that were encapsulated and arranged into a conical shape. This high-resolution γ-ray spectroscopy investigated a wide interval of shell models using many Radioactive Ion Beams (RIB) such as ^{74,76,78}Zn, ^{110,132}Sn, ^{144}Xe [38]. The signal formed by a γ-ray is slighly different from what was introduced before since it involves two steps. The interaction between the incident photon and the sensor material can generate a fast electron and a Compton scattered γ-ray or alternatively a fast electron-positron pair. The second step is the familiar process where the charged particle deposits its energy, then an electron-hole pair is produced and a signal is induced on electrodes. Without discussing further details about other aspects of this generation process, a pulse shape analysis is the way to increase the granularity beyond that one resulting from the detector segmentation. Thus, the events were manipulated by a ADSP-2183 Digital Signal Processor (DSP) from Analog Devices [39]. This unit runs at 40 MHz and it is designed to address 16 bit fixed point representation, integrating 80k bytes for the memory. Three computational blocks can independently work on the processor composed by an Arithmetic Logic Unit (ALU), a multiplier-accumulator unit (MAC) and a shifter. A trapezoidal filter have been implemented on the DSP module to extract the energy of a γ-ray and this algorithms will be specifically analyzed in Section 4.8.1. Therefore, the experiment was able to benefit of the digital processing to correct the ballistic deficit which affects the trapezoidal output. The DSP was also used for the pulse shape analysis since in this applications it is important to determine with an adequate accuracy the time when an event occurs. The Modified

Linear Extrapolated Baseline Crossing (ML-EBC) method was chosen to perform a linear extrapolation exploiting an acquired sample at the start of the signal and the slope defined by its position. The latter was easily derived because it was calculated as difference between two consecutive samples. These pulse shaping analyses were presented and discussed in detail [40].

Chapter 2

Analog-to-digital converters overview

As introduced in the abstract of this thesis, the Analog-to-digital Converter (ADC) is the key device to transfer information from the analog to the digital domain. In a full sampling system, the ADC samples directly the amplifier output and generates a digital stream which is manipulated by the digital signal processing unit. The conversion process is characterized by an intrinsic resolution that reduces the original information and it is affected by several errors due to the non-ideal behavior of the components used in the ADC. When designing the signal processing units, these aspects must be properly accounted for and they will be discussed in the next sections, focusing when appropriate on the successive approximation architecture chosen for this study.

2.1 Quantization error in ADCs

The generic output of an ADC is a digital data represented with a given resolution which is expressed in a number N of bits, leading to a 2^N possible digital codes. Thus, considering a voltage reference V$_{REF}$ as the full scale range of the ADC, the minimum digital step is named Least Significant Bit (LSB) and it is described by:

$$LSB = \frac{V_{REF}}{2^N} \tag{2.1}$$

Because of the limited resolution, a leak of information occurs during the conversion process. This is true also for an ideal converter, because it stems from the need of approximating a continuous quantity with a finite number of possible codes. This error is referenced to as *quantization error* and it is linked with to the SNR as described in the following.

In Figure 2.1 the ideal characteristics of a 4-bits ADC with a V$_{REF}$ of 1 V is depicted. The red line represents a ramp and it is assumed as the input signal, whereas the staircase shape is the response of the ADC. The vertical steps point out the transition levels and the horizontal lines between them are the results of the mapping process from the analog to the digital domain. The error is given by the difference between the ramp and the voltage level associated with the n-th code transition. This means that the error increases from a transition level to the following step, defining a typical saw-tooth shape.

In Figure 2.1 the staircase characteristics is right-shifted by LSB$/2$ from the origin of the system to symmetrize the quantization error. Adopting a ramp as the input signal implies that the output codes will be uniformly distributed on all the bins and the probability density function results to be inversely proportional to the width of the bins. Thus, the RMS quantization error V_q is given by:

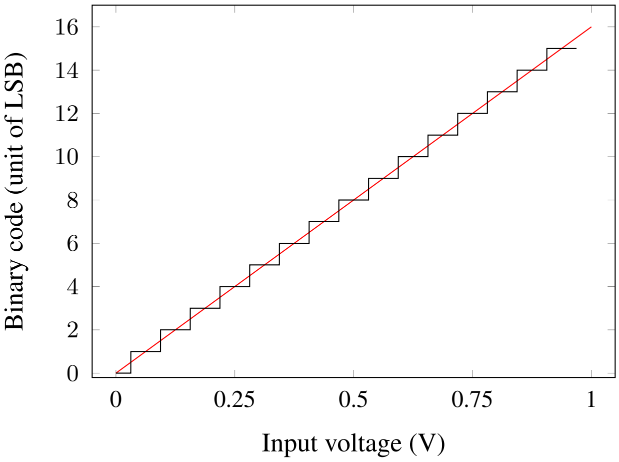

Figure 2.1: Ideal characteristics of a 4 bit ADC

$$V_q = \sqrt{\int_{-\frac{LSB}{2}}^{\frac{LSB}{2}} \frac{x^2}{LSB}\, dx} = \frac{LSB}{\sqrt{12}} \tag{2.2}$$

In general, to define the SNR of a signal we need to know the power of the input signal and the one of the noise . In the case of the SNR of an ideal converter, the quantization error of Eq 2.2 is the denominator of the ratio, while for the numerator the power of a sinusoid is typically used. Sinusoidal waves are a popular choice because they can be generated with very high fidelity. In order to explore all the V$_{REF}$ range of the quantizer, the sinusoid must have an amplitude A = $V_{REF}/2$:

$$V(t) = A \cdot sin(\omega \cdot t) \tag{2.3}$$

The RMS value is written as:

$$V_q = \sqrt{\frac{\omega}{2\pi} \int_0^{\frac{2\pi}{\omega}} \left[\frac{V_{REF}}{2} sin(\omega \cdot t)\right]^2 dt} \tag{2.4}$$

The integral can be solved recalling the remarkable integral:

$$\int_0^{2\pi} sin^2(x)\, dx = \pi \tag{2.5}$$

Hence, the RMS value of the input signal is defined as:

$$V_q = \frac{V_{REF}}{2\sqrt{2}} \tag{2.6}$$

and the SNR is simply the ratio of Eq 2.6 and Eq 2.2:

$$SNR_{dB} = \frac{\frac{V_{REF}}{2\sqrt{2}}}{\frac{V_{REF}}{2^N\sqrt{12}}} \tag{2.7}$$

where the quantization levels have been considered with 2^N. Eq 2.7 can be rewritten as:

$$\begin{aligned} SNR_{dB} &= 20log\sqrt{\frac{3}{2}} + 20Nlog2 \\ &= 6.02N + 1.76dB \end{aligned} \tag{2.8}$$

Eq 2.8 defines the signal-to-noise ratio for an ideal converter. If the SNR is known, the equation can be overturned to find the number of bits:

$$N = \frac{SNR - 1.76}{6.02} \tag{2.9}$$

However, in place of SNR the Signal-to-noise and Distortion (SINAD) is usually employed. SINAD is derived by the spectrum analysis of the ADC output when a sinusoidal wave is provided as the test signal. It is defined as the ratio between the amplitude A$_f$ of the fundamental harmonic located at frequency f and the sum of the other amplitude components where are combined noise and distortion:

$$SINAD = 20log\sqrt{\frac{A_f^2}{\sum\limits_{k=1}^{f-1} A_k^2 + \sum\limits_{k=f+1}^{\frac{N}{2}} A_k^2}} \tag{2.10}$$

The reader should notice that the k index of the leftmost sum at denominator starts from one. This is due to the current dynamic analysis of ADC performance accomplished using a sinusoidal signal, where the DC component located at k = 0 is not conveyed [41] [42]. Substituting now Eq 2.10 in Eq 2.9 the Effective Number of Bits (ENOB) can be obtained:

$$ENOB = \frac{SINAD - 1.76}{6.02} \tag{2.11}$$

which is the actual resolution of the ADC *Methods and draft standards for the DYNamic characterization and testing of Analog to Digital converters* (DYNAD) [43]. In addition to the SINAD and the ENOB, the spurious free dynamic range (SFDR) must also be considered. In a power plot, this quantity is simply the range free of spurious components assuming as reference the principal component frequency centered at 0 dB.

2.2 Thermal noise

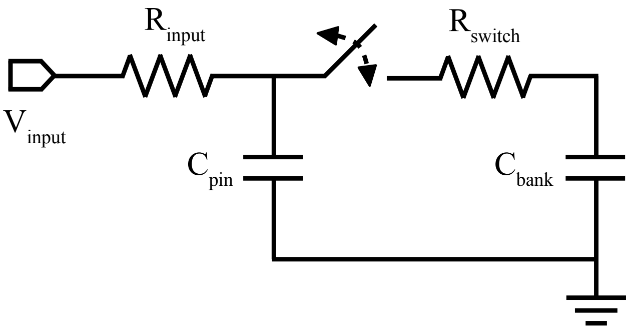

Figure 2.2: RC model of an ADC switch.

Another source of noise is introduced by the sampling process itself and it can be analyzed considering the switching model of an ADC [44] reported in Figure 2.2. Here R$_{input}$ models the external source resistance, R$_{switch}$ represents the internal switch resistance (usually around 1 kΩ), C$_{pin}$ quantifies the pin capacitance and C$_{bank}$ is the sum of the capacitive array of a Successive Approximation Register (SAR) architecture explained in Section 2.4.4. If R$_{input}$ $\ll$ R$_{switch}$ and the errors generated by the capacitance C$_{pin}$ negligible, the model works as an RC filter. Let's introduce the thermal noise V_{th} [45] given by:

$$V_{th} = \sqrt{4k_B T R \Delta f} \tag{2.12}$$

where k$_B$ is the Boltzmann constant ($1.38 \cdot 10^{-23}$ J/K), T indicates the absolute temperature in Kelvin, R is the resistive contribution measured in Ω of the complex impedance Z and Δf defines the bandwidth expressed in Hz. The thermal noise, also called Johnson noise, is due to the thermal agitation of electrons in ohmic paths. Taking into account a frequency band from f$_1$ to f$_2$ and the impedance Z, Eq 2.12 can be rewritten as:

$$V_{th} = \sqrt{4k_B T \int_{f_1}^{f_2} Re(Z)\, df} \tag{2.13}$$

where Re(Z) is the real part of impedance Z:

$$Re(Z) = \frac{R}{1 + (2\pi fRC)^2} \tag{2.14}$$

The total noise voltage of a RC pair can be obtained replacing Eq 2.14 into Eq 2.13 and extending the interval of integration from 0 to $+\infty$:

$$V_{th} = \sqrt{4k_B T \int_0^{+\infty} \frac{R}{1 + (2\pi fRC)^2}\, df} \tag{2.15}$$

Let's focus now on the improper integral. Because of its definition, we can rewrite it as:

$$\int_0^{+\infty} \frac{R}{1 + (2\pi fRC)^2}\, df = \lim_{i \to \infty} \int_0^{i} \frac{R}{1 + (2\pi fRC)^2}\, df \tag{2.16}$$

We define the new variable x and its differential dx as:

$$\begin{gathered} x = 2\pi fRC \\ dx = 2\pi RC df \\ df = \frac{dx}{2\pi RC} \end{gathered} \tag{2.17}$$

where in the last line the expression for df is shown. Substituting it into Eq 2.16, the improper integral becomes:

$$\begin{gathered} \lim_{i \to \infty} \frac{1}{2\pi C} \int_0^{i} \frac{1}{1 + x^2}\, dx = \\ = \frac{1}{2\pi C} \lim_{i \to \infty} \arctan x \bigg|_0^{i} = \frac{1}{4C} \end{gathered} \tag{2.18}$$

The result is achieved using a well-know remarkable integral and the limit value of $\arctan x$ function for $x \to \infty$ which is equal to $\pi/2$. Thus, a last substitution leads to:

$$V_{th} = \sqrt{4k_B T \int_0^{+\infty} \frac{R}{1 + (2\pi fRC)^2}\, df} = \sqrt{\frac{k_B T}{C}} \tag{2.19}$$

That is the expression for the thermal noise. In the case of a SAR ADC, the same result can be reached balancing the energy stored in the capacitor array with the quantity derived by the equipartition energy theorem considering one thermodynamic degree of freedom:

$$\frac{1}{2}C\overline{V}^2 = \frac{1}{2}k_B T \tag{2.20}$$

Eq 2.19 tell us that the thermal noise dominates the ADC noise contribution for high-resolution ADCs [41] [46].

2.3 Jitter

The jitter is the last source of uncertainty taken into account to model the noise at this stage. In order to quantify the effect of jitter, the following input signal is considered:

$$V_{in} = V_0 \sin(2\pi f_{in} t) \tag{2.21}$$

where an ADC with a full scale range of V_{REF} and $V_0 = V_{REF}/2$ is assumed. The point of maximum slope can be easily obtained as:

$$\left.\frac{dV}{dt}\right|_{max} = 2\pi f_{in} V_0 \cos(2\pi f_{in} t)\Big|_{max} = 2\pi f_{in} V_0 \tag{2.22}$$

At this point, the RMS uncertainty V_j on the sampled value is caused by the RMS jitter σ_t:

$$V_j = \pi f_{in} V_{REF} \sigma_t \tag{2.23}$$

To keep this noise below the quantization error, this value can be constrained as following:

$$V_j = \pi f_{in} V_{REF} \sigma_t < \frac{V_{REF}}{2^N \sqrt{12}} \tag{2.24}$$

Thus the value of jitter is bounded to:

$$\sigma_t < \frac{1}{2^N 2\pi f_{in} \sqrt{3}} \tag{2.25}$$

Therefore, the signal generated for this study was characterized by a deviation σ_{ADC} equals to the root sum squared of the quantization error V_q described by Eq 2.2, the V_{th} reported in Eq 2.19 and the jitter contribution V_j of Eq 2.23:

$$\sigma_{ADC} = \sqrt{V_q^2 + V_{th}^2 + V_j^2} \tag{2.26}$$

This sum is allowed because these quantities are uncorrelated [41].

2.4 ADC Architectures

Many ADC topologies have been proposed over the years. In front-ends for radiation detectors the most recurrent ADC architectures are the Flash one [47], the Single ramp [48], the Wilkinson ADC [49] [50] and the Successive Approximation Register [5]. A brief description of each ADC type is provided in the following.

2.4.1 Flash

The Flash architecture partitions the voltage reference V_{REF} using a voltage divider. Each voltage level feeds an array of comparators. The signal to be digitized is connected to the other input terminal of each comparator. Thus, the output of a comparator will be set to one if the input is higher than the associated threshold, otherwise it will be equal to zero. Then the thermometric output due to this comparator bank is converted by an encoder and a binary digital output is available. This process is particularly fast, but it suffers from the complexity due to the large number of comparators required and solutions have been proposed [51] [52] to overcome this issue. Nevertheless, this type of converter is not the best option in a full sampling system. In general, in high density front-ends only low resolution (3 - 4 bits) Flash ADCs are used to obtain a raw information on the input signal.

2.4.2 Single-slope

The Single ramp type is a very affordable solution in terms of design complexity and power budget. Indeed, in this circuit only a comparator is required. One input is connected to the input signal, while the second one is fed by a ramp generator. When the value to be digitized is presented at the input of the comparator, the ramp generation begins. At the same time, a counter value is increased. The synchronization between the ramp and the counter is managed by a dedicated logic. The idea is to freeze the counting when the ramp voltage is equal to the input one. It is straightforward to obtain the analogue input because only the multiplication between the counter output and the slope of the ramp, expressed in volt per clock cycle, is required. The ADC linearity is assured by an accurate design of the ramp generator which is based on a capacitor charged with a constant current.

2.4.3 Wilkinson

When a double ramp is adopted, the converter is named Wilkinson ADC. Basically, the input signal is employed to charge a capacitor, realizing the first ramp. Then, the capacitor is discharged by a constant current which defines the second ramp. Usually, the conversion speed is bound by the discharging ramp because it is slower than the first one. Also in this architecture the counter output is proportional to the amplitude of the signal and its synchronization is still a critical point. The main advantage of the dual-slope approach is the independence of the constant of proportionality by the R or C component used to generate the ramps in contrast to the single-slope converter.

2.4.4 Successive Approximation Register

In recent years the development of deep submicron CMOS technologies made the successive approximation architecture suitable to be used in a full sampling system. This is due to a considerable increase of the conversion speed and to the significant reduction of their power consumption. Over the time, many design solutions have been proposed from the simplest single-ended unipolar input to the differential architecture [53], considering pipeline segmentation approaches [54] as well as non-binary DAC structures [55] and interleaved methods [56]. Each topology exploits some feature to overcome a specific conversion issue. For instance, the differential configuration allows to reject any common mode component that affects the sampled signal, while a multistage pipeline design reduces the DAC area requirements that becomes particularly significant for resolutions larger than 10-bits. A non-binary solution can instead be adopted to recover an error occurred during the bit-decision step. Indeed, any error happening during this process impairs the definition of the following less significant bits, generating a chain of further errors. For example, these issues can be generated by mismatches of the integrated capacitors or a lack of enough settling time to stabilize the conversion result. A possibility to fix this problem is offered by DAC redundancy that permits a bit correction in later stages of the process. Lastly, an interleaved implementation involves an array of ADCs to perform a timing multiplexing in order to achieve a higher sampling rate than the nominal one of the single converter.

For this work, a fully differential SAR ADC architecture with 12-bits resolution has been chosen. In Figure 2.3 a reduced 4-bits version based on charge redistribution method is shown. The yellow boxes highlight the binary weighted capacitor bank which is splitted in two branches. If N is the resolution of the converter, a single array is formed by 2^{N-1} capacitors. Each capacitor of the DAC is realized by physically connecting in parallel a multiple number of a unit capacitor which corresponds to the LSB. The SAR ADC represented adopts the merged capacitor switching method to solve the convergence of the common mode towards zero. This strategy avoids the drawback of using PMOS transistors at the input stage as they are notable slower than their NMOS counterpart. This approach is realized introducing the common mode reference V_{CM} which is set halfway between the voltage reference V_{REF}, set to 1 V in this case, and ground.

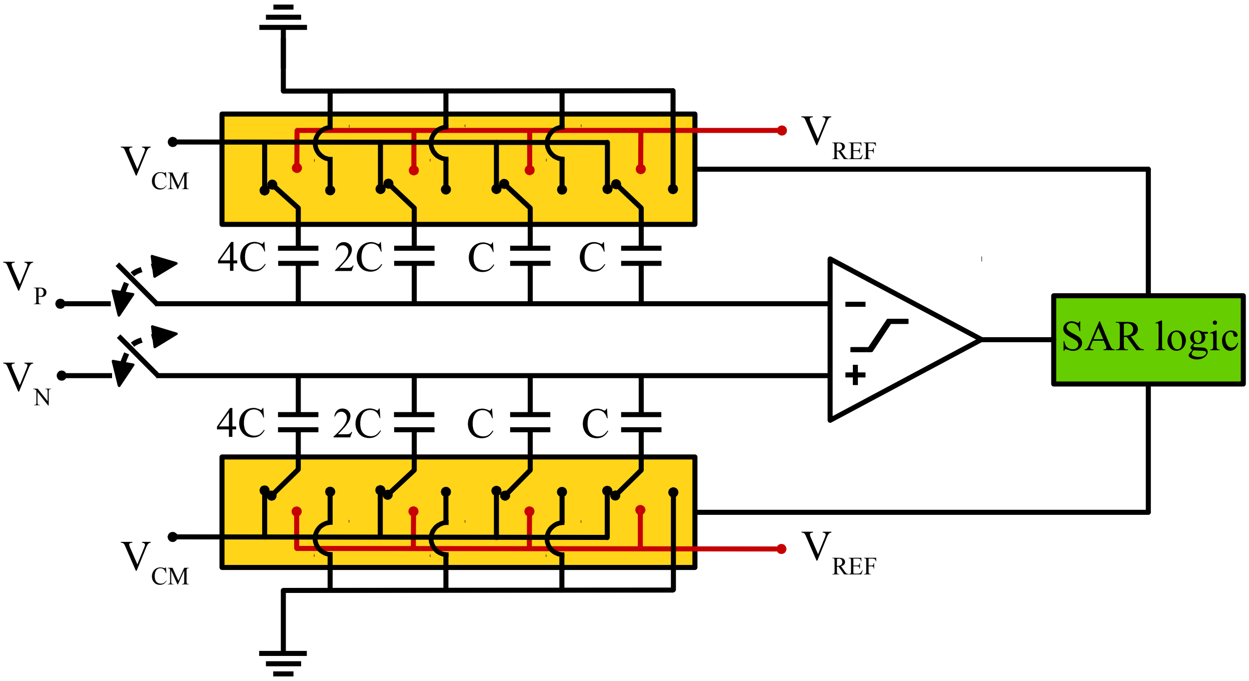

Figure 2.3: The fully differential 4-bits SAR ADC architecture.

Thus, for both the branches, the top plates of the capacitors are in common and they are indicated as V_P and V_N, while the bottom ones can switch between the voltage reference V_{REF}, the common mode voltage V_{CM} and the ground. In this way the common mode rejection is implemented. During the sampling, the input signal is presented at the top plates in differential form, whereas the bottom plates of the capacitor bank is maintained at V_{CM}. Hence, the switches on V_P and V_N lines are opened and the algorithm begins from the Most Significant Bit (MSB) towards the LSB one, evaluating the differential pair $V_P - V_N$ at each bit. If the difference is positive, then the MSB is set to one and the bottom plate of the upper MSB capacitor switches from V_{CM} to ground. By contrast, the bottom plate of the lower MSB capacitor is switched from V_{CM} to V_{REF}. This changes the voltages on V_P subtracting $V_{REF}/4$ and on V_N adding the same quantity. If the difference results negative, the switching scheme is complementary. According to this mechanism, the remaining bits are evaluated in the same way. During this process, the average value of V_P and V_N is constantly equals to V_{CM}. If a common mode component is shown in the signal to sample, it is rejected because only $V_P - V_N$ is processed. Also the logic demanded to the control of the conversion task is not particularly challenging. Furthermore, this circuit is power efficient compared to other SAR solutions.

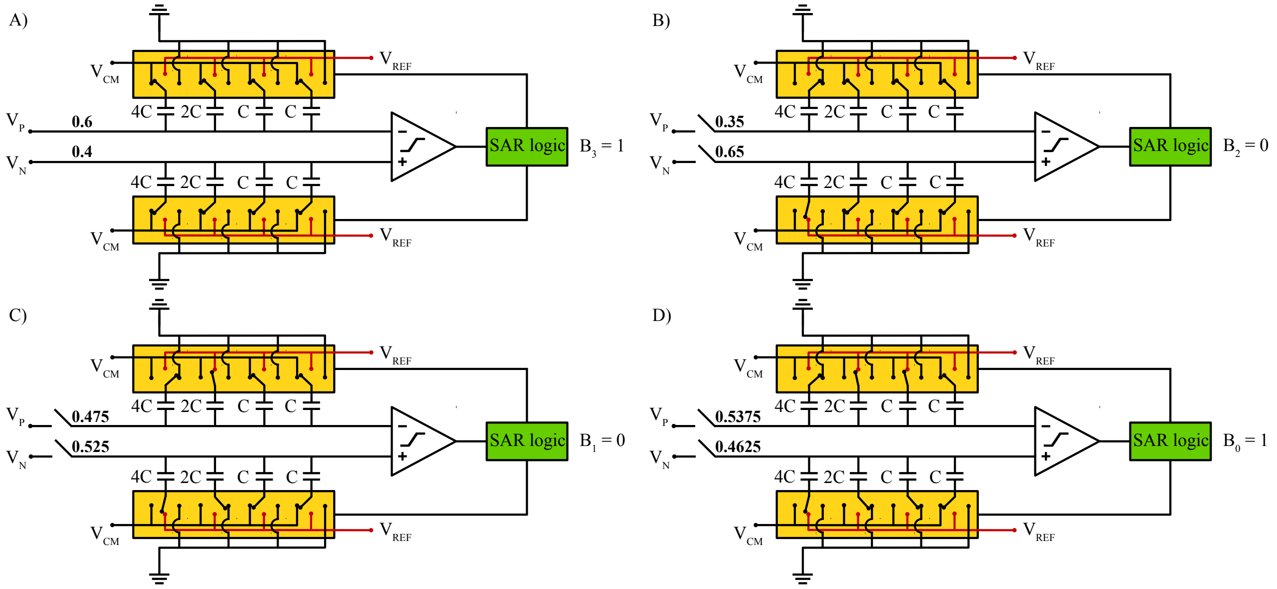

Figure 2.4: Example of a conversion of a fully differential 4-bits SAR ADC architecture.

Figure 2.4 illustrates an example of the method. As introduced before, the algorithm starts the bit evaluation from the MSB to the LSB. Figure 2.4 A) depicts the initial condition: the input switches are closed and on V_P and V_N are present 0.6 V and 0.4 V, respectively, while all the bottom plates of the capacitors are connected to V_{CM}. In this configuration, the condition $V_P > V_N$ is satisfied, thus the MSB results equal to one. This leads to a switching of the bottom plates related to the MSB: on the top array the plate is now connected to ground, while on the bottom one it switches from V_{CM} to V_{REF}. The new connections represented in B) change the value on both the top plates due to the redistribution of the charges. Basically, the double switching subtracts 0.25 V on V_P and adds the same quantity on V_N. Now $V_P < V_N$ and the second MSB named B_2 is set to 0. Accordingly with this outcome, the corresponding bottom plates are switched from V_{CM} to V_{REF} on the bottom plate of the top array and from V_{CM} to ground on its complementary capacitor on the bottom array. The comparator continues the evaluation of the quantity $V_P - V_N$ until the LSB where the same condition of the MSB occurs, hence B_0 is equal to 1. Therefore, the output of the converter is the 4-bit long word 1001. Considering $V_{REF} = 1V$ and the 16 digital levels due to the 4-bits resolution of this example, the value of the LSB is equal to 0.0625 V. If now this quantity is multiplied by 9, it results in 0.5625 V, which is the best possible approximation of V_P for the given resolution of this SAR ADC in an ideal case.

2.5 Integrated capacitors

However, a real capacitor is affected by a parasitic capacitance that involve both its plates. Furthermore, this parasitic contribution can be asymmetric. Since just a short overview on these components is given, the reader can examine [57] for a more detailed explanation. Thus, in ADC implementations, passive capacitors are preferred to their MOS counterpart. MOS capacitors are in fact characterized by the highest capacitance density, but non-linearities arise when a voltage is applied because the values of the capacitors are not constant and may considerably change. Nowadays, for technology nodes below 250 nm the metal-insulator-metal (MIM) capacitors are adopted. This prevents the high parasitic capacitance that dominates the bottom plates of capacitors realized with double polysilicon layers, which are typical of nodes above 350 nm. Hence, in the MIM case, the plates are fabricated with metal layers and the insulator is realized with an oxide. A thickness between 30 nm and 50 nm leads to a capacitance density in the range of 0.7-2 fF/μm^2. In order to make these capacitors the cost of extra masks and further fabrication steps are required. Obviously, this kind of realization increases the manufacturing costs and this is a factor that must be taken into account. However, some applications can exploit the regular thickness between the metal layers to implement the capacitors. This type is named Metal-Oxide-Metal (MOM) capacitor and in case of a two plates structure it can achieve a capacitance of 0.05 fF/μm^2 for a standard oxide thickness between 0.7-0.8 μm. This solution is particularly suitable if small capacitors are demanded or in applications where a reduced number of channels is sufficient. Expanding this approach, it is even possible to take advantage of the vertical direction by adding more metal layers if they are available. So, an increased capacitance density in the range between 0.2 fF/μm^2 and 0.3 fF/μm^2 can be provided. This parallel-plate arrangement creates an asymmetric balance in the parasitic coupling where one plate minimizes this quantity. Thus, the latter has to be connected to the more critical point of the design. The last type of passive capacitor is called Vertical Parallel Plate Capacitors (VPPC) and it takes advantage of the capacitance between close metal lines that belong to the same layer. Adding more layers organized in a fingered pattern allows to further increase the global capacitance of this structure, reaching capacitance density values comparable to the MIM solution or even superior. By contrast to the MIM case, this configuration determines also a symmetric situation, where the top plate and the bottom one are affected by the parasitic capacitance contribution towards the substrate in the same way. However, a VPPC does not require additional masks to be fabricated. The manufacturing process itself improves the matching among the capacitors to minimize the non-linearities impact on the nominal resolution. The factors that determine the value of the unit are the thermal noise and the capacitor mismatch [58]. The thermal noise described by Eq 2.19 is due to the sampling process where the total capacitive bank C of the DAC is composed of the parallel of unit capacitors C_u. The minimum value of C_u is derived by the equality of the thermal noise with quantization error reported in Eq 2.2:

$$\frac{K_B T}{C} = \frac{V_{REF}^2}{2^{2N} 12} \rightarrow C_u = \frac{12 K_B T 2^N}{V_{REF}^2} \tag{2.27}$$

where the last expression is obtained by dividing C for 2^N because of the definition of LSB. The second factor is represented by the capacitor mismatch. Considering the standard deviation of the capacitor mismatch $\sigma(\Delta C/C)$ in a given technology, it is possible to quantify the worst-case deviation σ_W of differential non-linearity (DNL) as:

$$\sigma_W = \sqrt{2^N - 1}\sigma\left(\frac{\Delta C}{C}\right) \tag{2.28}$$

Because the deviation $\sigma(\Delta C/C)$ is inversely proportional to the capacitor area A, we can write:

$$\begin{gathered} \sigma\left(\frac{\Delta C}{C}\right) = \frac{K_\sigma}{\sqrt{A}} \\ C = K_c \cdot A \end{gathered} \tag{2.29}$$

where K_σ is the mismatch parameter measured in $\% \cdot \mu m$ and K_c indicates the capacitor density expressed in F/m^2. To achieve a high yield, it is required to constrain $3\sigma_W$ to be at most equal to $0.5LSB$. Thus the following equation have to be satisfied:

$$\begin{gathered}3\sigma_W = \frac{1}{2}LSB \\ 3\sqrt{2^N - 1}\sigma\left(\frac{\Delta C}{C}\right) = \frac{1}{2}\end{gathered} \tag{2.30}$$

Finally, from $C = K_c \cdot A$ can make explicit the area A to substitute it into the first equation of Eq 2.29. Therefore, taking this result and replacing it into Eq 2.30, we obtain the minimum value of C_u:

$$C_u = 36 \cdot (2^N - 1) \cdot K_\sigma^2 \cdot K_c \tag{2.31}$$

Another important role on the unit capacitance definition is played by the parasitic capacitances affecting the DAC output. However, this aspect is out of the scope of this modelling and further information can be found in [5].

In conclusion, a SAR ADC based on the merged capacitor switching method is a valid candidate for multi-channel applications of radiation detector front-ends. For resolution above 10 bits, digital error correction algorithms become however necessary to counteract the effect of capacitor mismatch.

2.6 SAR ADC state of the art

As introduced in Section 2.4.4, the SAR ADC topology has become popular in recent years due to its power consumption and conversion speed performance. A remarkable example of a 14-bits resolution SAR ADC is reported in [59] where a pipeline architecture has been implemented in a 28 nm CMOS technology. This prototype quantizes the MSB with a fast sub-ADC to obtain a coarse result. Then, the outcome is encoded with a residue transformation technique in order to take control on the generation of the residue in two fine channels. Here two splitted branches are used,where each one is nominally the twin of the other. The final ADC result is the averaged value between the output of the two channels. The difference among these outcomes can be also exploited to derive the error conversion of the two lines and to adjust it. To prevent the same non-linearities in the branches, psuedo-random noise sequences are injected, thus voltage input V_{in} is converted as well as the dithered voltage $V_{in} + \Delta V$. In this way, it is possible to obtain different residue modes. This implementation, assisted with an additional digital calibration, occupies a core area of 0.368 mm^2 with a power consumption of 4.26 mW at 60 MS/s when the clock buffer is included and the calibration block is set to hold-on mode. If the calibration engine is enabled, the power consumption rises to 5.18 mW. The SINAD achieves 66.9 dB, while SFDR records 91.0 dB.

A further interesting result is presented in [60] where another 12-bits pipelined SAR ADC is discussed. In this 28 nm CMOS design the conversion process was carried out with a three-stage architecture to improve its speed by distributing the bit decision sequence among a larger number of steps. A high-linearity open-loop residue amplifier (RA) was adopted to obtain a voltage gain of 8 and an amplification time of $\sim$200 ps. Additionally, a voltage bias circuit compensates the gain variation of the residue amplifier due to the temperature. The whole converter can process a signal at 1 GS/s with a SINAD of 60 dB, consuming 7.6 mW. In a range of temperatures between 0 °C and 80 °C, this SAR prototype achieves an ENOB of 9.

An example of an interleaved approach is shown in [61]. Here a 11-bits SAR ADC was implemented in a 28 nm CMOS process through two channels where identical ADCs are placed. Each converter runs at half of the full clock rate which is 410 MS/s and their input stage is equipped with a bootstrapped front-end sampling switch that removes the time skew between the two branches working at the full rate. A single ADC is composed of 6-bits coarse SAR ADC connected to a RA with a sampling switch as output. The residue of this stage is processed by a further 7-bits fine SAR ADC. Lastly, a multiplexer collects the outputs of the two ADCs and it combines them. A redundancy scheme between the coarse and the fine blocks was exploited to implement a calibration system. The core area of the fabricated chip is 0.11 mm^2, while the SINAD is slightly less than 60 dB at 410 MS/s. At the same rate, the energy per conversion step is less than 12 fJ using a Nyquist input signal.

A last example of interleaved architecture in the same technological node is illustrated in [62] where a pipeline SAR ADC achieves 11 ENOB. The converter is organized in two stages with the first SAR ADC designed to have 7-bits resolution to ensure a noise of the comparator smaller than the redundancy between the two stages, while the second one has 8-bits resolution. The latter is due to the global noise budget where a further 1 bit redundancy is introduced between the stages to manage the noise of the comparator and the offset and RA. Because the converter has an interleaved structure, each ADC branch works at 40 MS/s. The time of the whole operations is organized in a time window assigned to the residue amplification (2.5 ns) and an interval reserved to the conversion time of the SAR (10 ns). This means that a bandwidth of 500 MHz is suitable for the sampling circuitries. Since the RA is a critical component because it gives the appropriate gain for the residue voltage in order to reduce the power consumption of the second stage, its transfer function is an integrator. Indeed, this is the best choice to separate the input signal and the noise because the RA operation is carried out when the input signal seems to be static within a clock cycle. It must also be considered that the larger part of the power budget is taken by the RA, if the thermal noise is not taken into account. Keeping these considerations in mind, an integrator was implemented, based on two integrator sections to take advantage of a low power consumption, a reduced input noise and a low noise bandwidth. Hence, the total power dissipation of the chip is 1.5 mW, with an energy per conversion step of 9.1 fJ. The core area for this standard 28 nm CMOS is 0.13 mm^2.

Chapter 3

Digital Calibration for SAR ADCs

As anticipated in the previous section, for high resolution requirements (above 10 bits) it is indispensable to arrange a correction system to deal with the DAC non-linearities. Both analog and digital domain offer solutions to this issue. Before providing an overview on the possible calibration approaches [63] and then focus in detail on the selected one, it is convenient to discuss the superposition principle.

3.1 Superposition Principle

This principle is the foundation in many scientific fields and its importance will be discussed in more detail in Section 4.2.0.1 dedicated to digital filtering. For the purpose of this section, it is enough to think about a mapping of an analog input signal V_i to a digital output word [65] [66]. Termed this process with the function $Q(x)$, let's suppose to introduce an analog perturbation Δ_P to the input V_i. Once mapped separately, the outputs obtained are $Q(\Delta_P)$ and $Q(V_i)$. Under the assumption of an ideal converter and a linear operation, we can write the digitized output as:

$$Q(V_i \pm \Delta_P) = Q(V_i) \pm Q(\Delta_P) \tag{3.1}$$

Now the above equation is rearranged pointing out $Q(V_i)$:

$$Q(V_i) = Q(V_i \pm \Delta_P) \mp Q(\Delta_P) \tag{3.2}$$

Considering the case of a linear ADC, Eq 3.2 shows that any analog offset can be digitally removed from the output code. Obviously, in a real application the DAC introduces a non-linearity that must be addressed with some method.

3.2 Independent Component Analysis (ICA)

A possible approach to take into account the mismatches is represented by the Independent Component Analysis (ICA) [64], a technique based on bitwise correlation. A single-bit pseudorandom sequence PRS of magnitude Δ_a is injected at the converter input. This quantity is digitized as well as the input signal V_i in the same way described in the previous section, obtaining a code D. This digital word is further processed with a bit-by-bit weighted sum $\Sigma_i W_i \cdot D_i$, let's call it D_W. In an ideal case, it was demonstrated that the superposition principle allows the PRS to be digitally removed and it is independent of the PRS sequence adopted. Because the optimal weights W are unknown at the beginning, this process can not be linear and a residual appears. This discrepancy can be used to adjust the weights and to find their optimal configuration. However, it is important to remember the single-bit nature of PRS that leads to the impossibility to recognize the individual error contribution of each bit. In other words, a single PRS is not informative enough to derive all the DAC weights W_i. A possible solution is provided by multiple PRS injections at the cost of the ADC dynamic range degradation. The ICA overcomes this drawback adopting a digital requantizer. This circuit reproduces the SAR operation decomposing the weighted output D_W back to D and resulting into a new digital code $D_{req} = d_N, ..., d_i$. Then, this last output is correlated with the single pseudorandom sequence in order to manage the learning algorithm of all the weights. The weights are updated employing the following equation:

$$W_i[n+1] = W_i[n] - \mu E[PRS \cdot d_i] \tag{3.3}$$

where μ is a learning parameter, while $E[PRS \cdot d_i]$ is the collective correlation. This correlation is realized with a XOR gate because both $D_{req,i}$ and PRS have the same one-bit dimension. As well as the DAC weights, also the digitized version Δ_d of the magnitude Δ_a must be updated since it is unknown. The corresponding update for Δ_d is given by:

$$\Delta_d[n+1] = \Delta_d[n] - \mu E[PRS \cdot D_W] \tag{3.4}$$

3.3 Redundant Double Conversion (RDC)

Another method reported in literature is referred as Redundant Double Conversion (RDC). In this technique an injection circuit is not required, rather it takes advantage of the internal redundancy of a sub-binary SAR to calibrate the device. The term *sub-*

binary means that the DAC capacitors are slightly larger than their nominal value. As RDC name suggests, a double digitalization of a same sample takes place using two different decision thresholds. Considering a 4-bits SAR ADC and the case of the MSB, the discriminating levels are given by the codes 0111 and 1000. Also in this case a set of weights is used to multiply the two thresholded data which result into the binary words d_+ and d_-. The difference $\varepsilon = d_+ - d_-$ represents the error of the conversion process. If ε is equal to zero, it means that the weights are already optimal for the given sample, otherwise their calibration occurs. The algorithm iterates this steps accounting a sequence of decision levels. The main drawback of this approach is the conversion speed halved during the calibration mode.

3.4 Internal Redundancy Dithering (IRD)

A technique which is not affected by the double conversion limitation such as the RDC one is the Internal Redundancy Dithering (IRD) [67]. Similar to RDC, this calibration proposes to achieve the optimal weights changing the bit-decision thresholds. Here these levels are dithered, namely, called $V_{T,MSB}$ the threshold for the MSB, it is left- and right-shifted by small quantities. The effect of this results in two new thresholds $V_{L,MSB}$ and $V_{U,MSB}$ which defines a dithering window W_D. This window is in turn contained into the redundancy region bound by the transition levels $V_{0,MSB}$ and $V_{1,MSB}$ determined by MSB = 0 and MSB = 1, respectively. Thus, for inputs that fall inside W_D, the outputs of ADC are randomized. If the weight associated with the MSB is ideal, the ADC output is not affected by this randomization, otherwise the digital outcome will show a conversion error which is correlated with the pseudorandom sequence. The zero correlation case is verified when an input falls in the regions included between $V_{0,MSB}$ and $V_{L,MSB}$ or bounded by $V_{U,MSB}$ and $V_{1,MSB}$. This mechanism can be naturally extended to multiple bit weights identification through multiple correlations.

3.5 Code density test

Another possibility to correct the ADC output is given by a statistical method generally named *code density test* [68] or *histogram test* [69]. The dynamic range of the ADC under analysis is spanned by a suitable input signal such as a ramp or a sinusoidal wave. The digital outcomes are collected and organized into an histogram to point out the occurrency of each code. The technique is focused on the discontinuities that may occur in the outputs distribution compared to the ideal one. The histogram of an ADC without mismatches do not exhibit gaps between its bin which are referred to as *missing codes*. Depending on the ADC architecture, this class of algorithms set an optimal group of weights to compensate the DAC non-linearity. Usually, many samples are required to achieve a statistical significance, especially in the case of a sine wave.

3.6 Offset Double Conversion

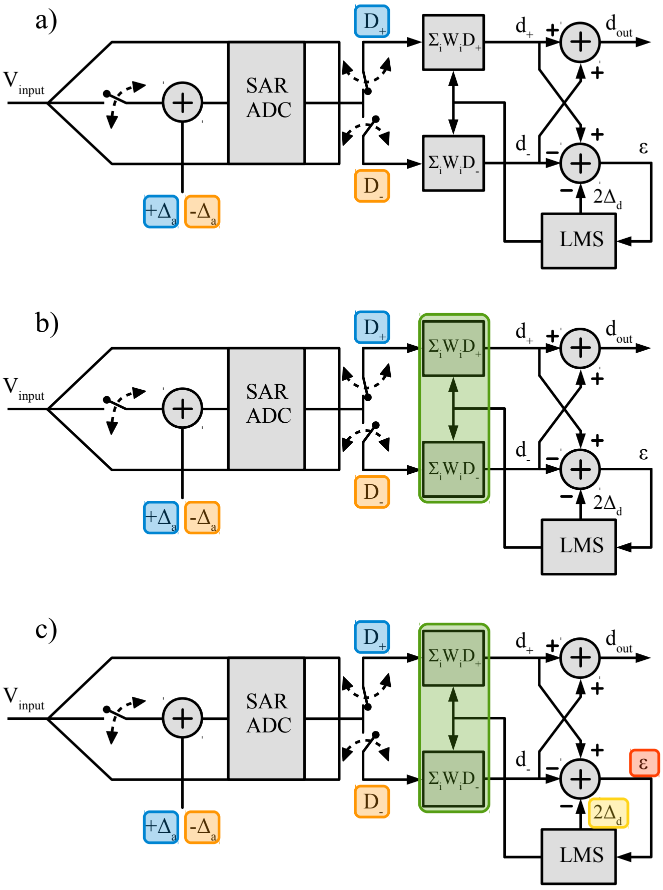

Figure 3.1: Block description of the Offset Double Conversion algorithm.

The last method discussed belongs to the digital background calibration class. It is named Offset Double Conversion (ODC) [63] [70] and it is depicted in Figure 3.1. As the ICA, it is based on the perturbation injection of a small offset Δ_a. Therefore, the conversion process of an analog input V_i is performed twice, one adding to V_i the offset mentioned afore, the other subtracting it. The double sampling results in two raw digital codes indicated in a) with D_+ and D_-, respectively. The core of the algorithm is represented by the weights W_i that are used to perform the weighted sums highlighted with the green box in b):

$$d_{\pm} = \sum_i \left(W_i \cdot D_{i,\pm} \right) \tag{3.5}$$

Having a look in c), the conversion error ε of the ADC can be evaluated as follows:

$$\varepsilon = d_+ - d_- - 2\Delta_d \tag{3.6}$$

where Δ_d is the digital representation of the injected offset Δ_a. If ε results equal to zero or below a desired threshold, the system does not need further calibration as the optimal bit weights are achieved, otherwise the Learning Management System (LMS) adjusts both Δ_d and W_i accordingly to:

$$\Delta_d[n+1] = \Delta_d[n] + \mu_\delta \cdot \varepsilon[n] \tag{3.7}$$

$$W_i[n+1] = W_i[n] - \mu_W \cdot \varepsilon[n] \cdot \big(b_{+,i}[n] - b_{-,i}[n]\big) \tag{3.8}$$

where $b_{\pm,i}$ are the bit-components of the raw codes $D_\pm$, while μ_Δ and μ_W are configurable learning parameters of the Δ_d and W_i update equations, respectively. Due to the double sampling, this method shares the same issue of RDC technique as the conversion speed is halved during the calibration mode. However, an advantage of this algorithm resides in the noise attenuation coming from the quantization error and the comparator because of the averaging. Also the injection circuit does not require particular efforts since is it basically composed by a small capacitor and some control logic to drive it. Compared to other solutions [71] [72] [73], the ODC requires a remarkably shorter time for the error to converge to zero. This benefit combined to a relatively simple hardware implementation were the reasons for the selection and development of this particular algorithm.

3.7 Calibration processor

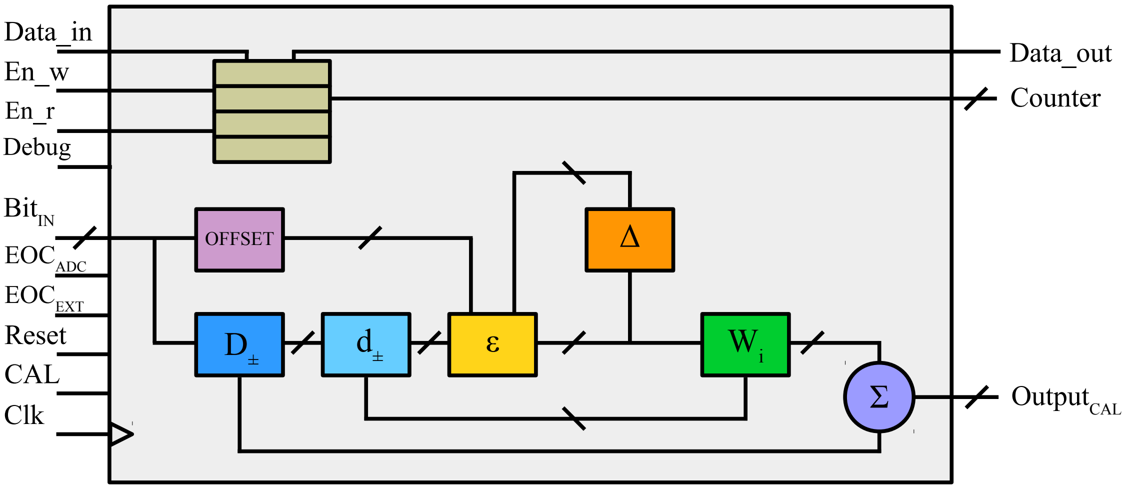

Figure 3.2: Digital calibration processor architecture.

Figure 3.2 illustrates the architecture of the calibration engine at block level. On the leftmost side are gathered the input signals while on the rightmost one are collected the outputs. At first glance, the reader can recognize the computational steps explained in the previous section. The algorithm is organized in seven stages managed by a Finite State Machine (FSM) reported in Figure 3.3 and itemized in Table 3.1. The most obvious first state is an idle one (IDLE), where the processor simply sends out the bits multiplied by the weights, whatever a calibration occurred or not. The device maintains this state if the calibration control signal (CAL) is zero. Otherwise, when CAL switches to one, the engine enters in calibration mode and the following stage is reserved to the acquisition of digitized offset (OFFSET). Here the offset is stored into a memory to be used later in the first evaluation of the error ε. Therefore, both the positive (D_+) and negative (D_-) samples are collected in two dedicated states. In the state diagram reported they are grouped into one stage ($D_\pm$) for convenience. If the offset provided is wide enough (roughly around $20LSB$) the processor is able to distinguish the larger value from the smaller one, thus it is not important which value is provided and converted first. However, this order must be hold for all correction loops. The next stage computes $d_\pm$, namely, the weighted results of the two codes. At this point, the DAC mismatch can be evaluated using Eq 3.6 where Δ_d is the offset previously acquired. This state corresponds to the EPSILON step, while the following one concerns the update of the Δ_d value (DELTA). Finally, the last state updates the array of weights (W). This phase can force the FSM to return to the IDLE stage if CAL is equal to zero or to perform another calibration cycle if CAL continues to be set to one. Thus, the OFFSET state is not longer visited unless the user wants to reset the entire process. Because of this, in Figure 3.2 we can notice

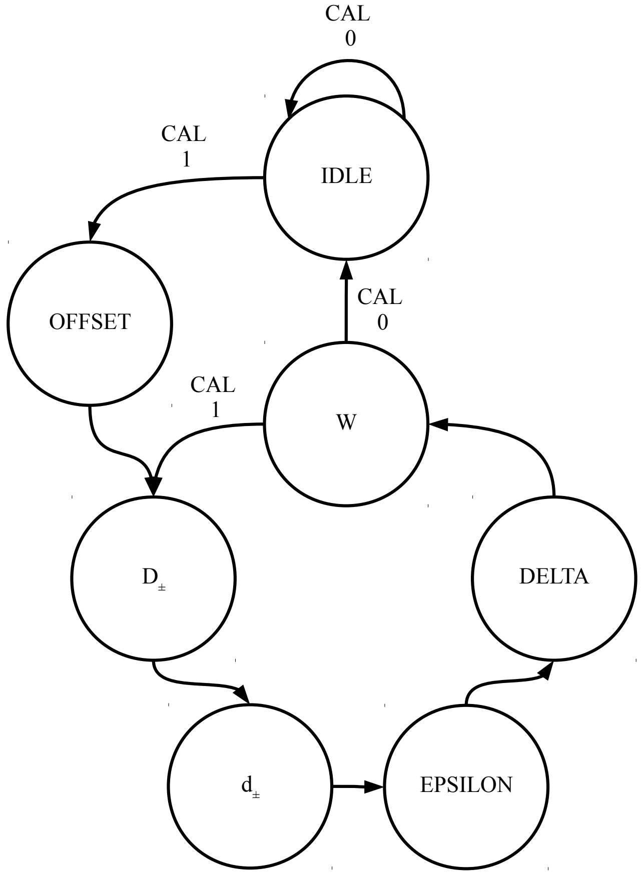

Figure 3.3: Finite state machine of the calibration processor.

the feedback path from the Δ block applied to ε stage. In other words, the following correction loops uses the updated version of Δ to compute the error since the offset is received only once. Lastly, it is important to point out that if a cycle is started, the engine guarantees its completion with the weights adjustment.

State	Value	Description
IDLE	000	If CAL $= 0$, the FSM sends out the weighted sum of the input data, otherwise it enters in calibration mode.
OFFSET	001	The offset is acquired and stored in memory.
$D_{\pm}$	010/011	The system receives the two digital codes.
$d_{\pm}$	100	The algorithm performs the weighted sum of the previous data.
EPSILON	101	For the first calibration cycle, the FSM computes the error associated with the conversion process using both the offset and $d_{\pm}$. For the other correction loops the updated Δ substitutes the offset.
DELTA	110	In this state Δ is updated following the simplified version of Eq 3.7 described in the text.
W	111	The engine adjusts the weights according to a modified version of Eq 3.8. Hence, the FSM can be drived back to the IDLE state set CAL $= 0$ or perform another correction keeping CAL $= 1$.

Table 3.1: FSM description of the digital calibration processor.

From an hardware point of view, to avoid the implementation of a multiplier, Eq 3.7 was simplified introducing a programmable shifter as illustrated in the following code snippet.

```
////////////////////////////////////////////////

always_comb begin
  if (reset) begin
    delta_comb = 32'b0;
  end
  else begin
    delta_comb = 32'b0;
    if (state == 6) begin
      delta_comb = delta_seq + $signed(epsilon >>> delta_shift);
    end
    else if (state == 1) begin
      for(int i = 0; i < `BIT_LENGTH; i++) begin
        delta_comb = delta_comb + W[i]*bit_in[i];
        delta_comb = delta_comb >>> 1;
      end
    end
```

```
18      end
19    end
20
21    /////////////////////////////////////////////////
```

The code describes che combinational logic used to implement the Δ_d update, here named *delta_seq*. An if-else structure discriminates when to apply the reset (line 4) on the dedicated logic, called *delta_comb* and the update process. A sub-condition statement describes how to refresh the Δ_d value: when the FSM reaches the dedicated state at line 9 (state 1 is referred to the offset acquisition, state 6 operates on the updating cycle), two type of operations can occur. If state is equal to 1 (line 12), a MAC was implemented through a for loop to perform the weigthed sum on the 12-bits word sampled by the ADC (*bit_in*) and consequently the computed value is right-shifted by one. The shifting is a signed operation because of the signed declaration of *delta_comb*. The result of this first and unique operation will be stored in a flip-flop register named *delta_seq* at the following posedge of the clock. Otherwise, when state equals to 6 (line 9), the shifted value of the error ε is assigned to the combinatorial output *delta_comb*. We should notice that this is a signed operation too since ε is signed itself. The shifter element is 5-bits long and it is totally configured by the user. As well as the previous process, also in the weights adjustment a simplification of Eq 3.8 was adopted, reducing the multiplier to another configurable 5-bits long shifter. Due to the intensive computational demand at this step, the hardware requirement saving and the timing benefits are particularly evident.

3.8 Fixed-point representation

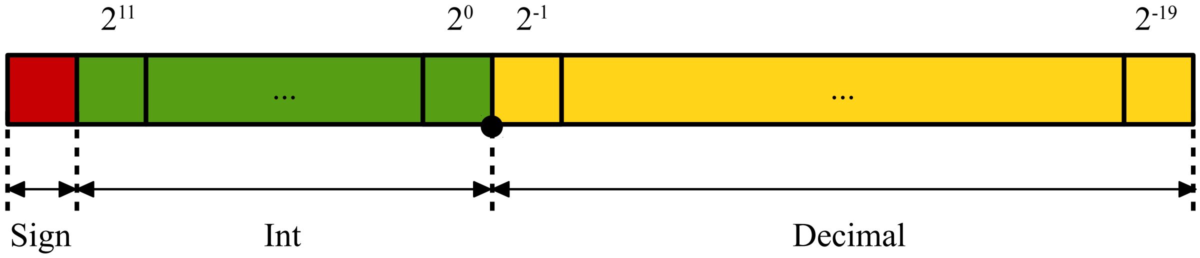

Figure 3.4: 32-bits fixed-point representation adopted. The large black dot highlights the location of the decimal point.

The core of the ODC algorithm is represented by the weights and their updating mechanism. Thus, it was important to ensure an adequate numerical representation for all the steps which lead to the weights adjustment. In general, the word width strongly depends on the application and usually its choice is between 8, 16, 24 or 32-bits for a fixed-point format [74] [75]. Also the floating-point numbers are well explored in modern processors, however for the current work they were not strictly necessary. Despite the floating-point representation assures both a wider dynamic range and precision compared to the fixed-point counterpart, it actually becomes a computational overkill for the following considerations. Let's start from the dynamic range which defines the window of exploitable values without an overflow or an underflow taking place [76]. The dimension of the calibrated output is equal to the input one, since the correction engine must not artificially increase the resolution nor reduce it. Hence, the range of numbers to be represented in the current application is bound by the ADC resolution, which is 12-bits integers. Therefore, the dynamic range of the calibration block is relatively limited without demanding a large numerical representation. The precision instead answers to the question: how many bits should be used to properly represent the numbers? In the floating-point format the precision is given by the mantissa and this implementation takes advantage of an automatic normalization and scaling. This means that the precision is held along all the dynamic range whereas the fixed-point hardware requires the careful intervention of the designer. However, other aspects have to be taken into account to prefer the fixed-point solution rather than floating-point one. Operations in fixed-point format are straightforward [77] once the designer defined the integer and decimal part, whereas the floating-point implementation is more complicated demanding more standard cells and ultimately more area and power consumption [78] [79] [80]. Furthermore, due to the complex operations involved, a floating-point hardware affects also the timing performance of a device. Lastly, the time development was an important factor too. Thus, keeping in mind these considerations, Figure 3.4 depicts the 32-bits fixed-point representation carried out for the weighted sums $d_{\pm}$, ε and Δ_d. In SIF notation [81] [82], these numbers assume the (1/12/19) format where the MSB indicates the sign of the number, the following 12-bits express the integer part and the remaining 19-bits the decimal one. In this convention, the decimal point is between the LSB of the integer word and the MSB of the decimal portion. Since the weights can never be negative, their MSB dedicated to the sign were elided to gain a 1-bit decimal, thus the SIF notation of the array W_i is (0, 12, 20). The updating Eq 3.8 mechanism was implemented hardcoding the addition and subtraction to take into account the sign of the error ε. Finally, the 32-bits word length is also due to the shifting mechanism described in Section 3.7, where it becomes fundamental to have an adequate word representation to guarantee a non-zero shifting. Indeed, it is important to recall that a configurable shifter of 5-bits offers 32 possible shifts and this means that a suitable word length must be guaranteed to prevent a null update.

3.9 Discrete Fourier Transform

In Section 2.1 the ENOB was introduced to quantify the actual resolution of an ADC. This evaluation of ADC performance can be found out analyzing the spectrum of a test signal such as sinewave. The equivalence of information between frequency and time domain is guaranteed by an extremely powerful tool in DSP field called the Discrete Fourier Transform (DFT) [49]. As its name suggests, it is the discrete version of the Fourier Transform that is extended from $-\infty$ to ∞. Thus, the DFT accounts only finite and periodic sequence of samples x[t] which concurs to define the power content X[k] of a signal at a given frequency k through the multiplication with a complex exponential:

$$X[k] = \sum_{t=0}^{N-1} x[t] e^{-\frac{2\pi i t k}{N}} \tag{3.9}$$

where N is the number of samples acquired in the sequence x[t]. The exponential term can be manipulated recalling the Euler's formula $e^{xi} = \cos(x) + i\sin(x)$ and the properties $\cos(-x) = \cos(x)$ and $\sin(-x) = -\sin(x)$ as following:

$$\begin{aligned} e^{-\frac{2\pi i t k}{N}} &= e^{-\left(\frac{2\pi t k}{N}\right)i} \\ &= \cos\left(-\frac{2\pi t k}{N}\right) + i\sin\left(-\frac{2\pi t k}{N}\right) \\ &= \cos\left(\frac{2\pi t k}{N}\right) + i\sin\left(\frac{2\pi t k}{N}\right) \end{aligned} \tag{3.10}$$

Considering now the more general case of a complex signal $x[t] = Re(x[t]) + iIm(x[t])$, the argument of the sum in Eq 3.9 can be written as:

$$x[t]e^{-\frac{2\pi i t k}{N}} = [Re(x[t]) + iIm(x[t])]\left[\cos\left(\frac{2\pi t k}{N}\right) + i\sin\left(\frac{2\pi t k}{N}\right)\right] \tag{3.11}$$

Computing the complex multiplication and then grouping the real part and the imaginary terms, the two components can be defined:

$$
\begin{aligned}
X[k]_{Re} &= \sum_{t=0}^{N-1} Re(x[t] \cdot \cos(\theta) + Im(x[t]) \cdot \sin(\theta) \\
X[k]_{Im} &= \sum_{t=0}^{N-1} -Re(x[t] \cdot \sin(\theta) + Im(x[t]) \cdot \cos(\theta)
\end{aligned}
\tag{3.12}
$$

where $\theta = \frac{2\pi tk}{N}$ is the complex angle. Combining $X[k]_{Re}$ and $X[k]_{Im}$ results in the expression of the magnitude of the signal:

$$
X[k] = \sqrt{(X[k]_{Re}^2 + X[k]_{Im}^2)} \tag{3.13}
$$

The quantity $|X[k]|^2$ is referred as the energy spectral density at frequency k of the signal x[t] and it is evaluated in unit of energy per Hz. If the signal is real, the equations pair in Eq 3.12 becomes:

$$
\begin{aligned}
X[k]_{Re} &= \sum_{t=0}^{N-1} Re(x[t]) \cdot \cos(\theta) \\
X[k]_{Im} &= \sum_{t=0}^{N-1} -Re(x[t]) \cdot \sin(\theta)
\end{aligned}
\tag{3.14}
$$

It is important to put in evidence that the number of points forming the spectrum is not equal to N, but is (N/2 + 1) points long, where the first one represents the DC component of the signal. The DFT is graphically illustrated considering the magnitude on the vertical axis, usually expressed in Decibel (dB), and the frequency f$_k$ on the horizontal axis. The latter can be reported as sample number, namely as k index, or fraction of the sampling rate with f$_k$ = k/N. In general, two issues affect the spectrum of a signal: aliasing and leakage [83]. The first one is due to the undersampling that is the using of a sampling frequency below the Nyquist frequency f$_N$. This last expression stands for the highest frequency component of the signal and it sets an important limit to take into account. Due to the Nyquist-Shannon sampling theorem [76], the sampling frequency must be at least two times the Nyquist one.

However, in real applications, this condition can be never applied and the sampling rate have to be markedly higher than f_N. The second effect is named spectral leakage and it occurs when the sampled signal is not periodic in the acquisition window [84]. This case is referred as non-coherent sampling and it is depicted in Figure 3.5.

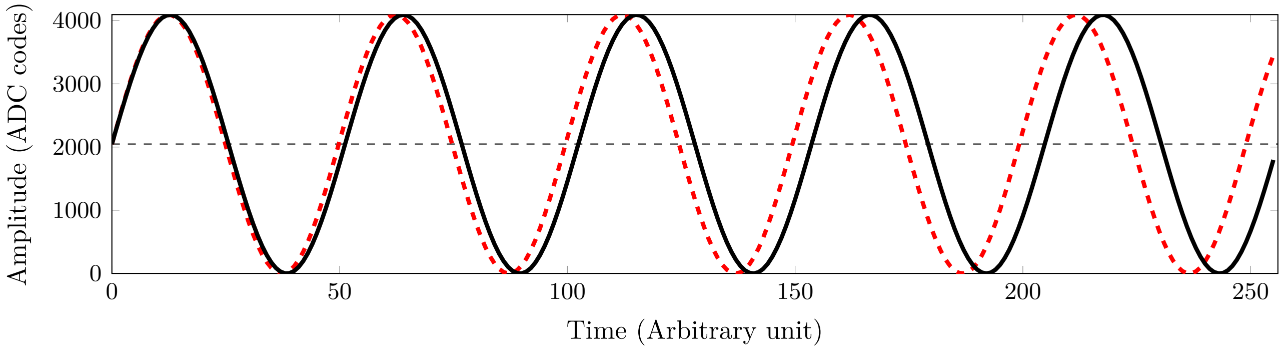

Figure 3.5: Examples of coherent sampled signal (solid black curve) and non-coherent one (dashed red curve).

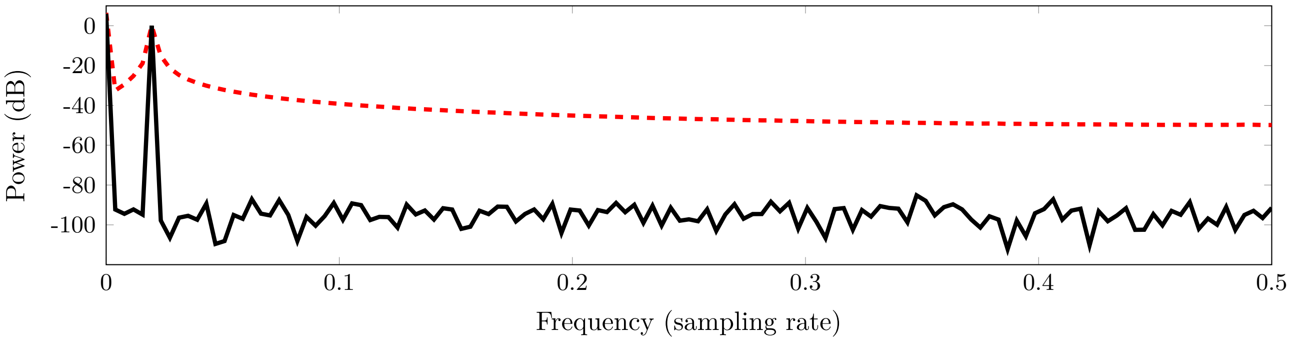

Figure 3.6: Illustration of the spectrum leakage for a non-coherent sampling (dashed red curve) compared to a correct spectrum (solid black one).

In this picture, a 256-points coherent sampled signal (black sinewave) and a non-coherent sampled sequence of the same length (red sinewave) are reported. The dashed signal points out a remarkable discontinuity at the end of the acquisition processing compared to the solid one. On these discrete sequences a DFT was applied, resulting in Figure 3.6. The red spectrum exhibits a noticeable smearing of power located at its fundamental harmonic because of the discontinuity of the signal caused by its non-periodicity. Thus, further harmonics are added to the original spectrum, alterating the energy information of the sequence. There are two ways to fix the spectral leakage: setting up an integer number of cycles or apply a window function. To realize the first solution let's consider the sampling rate f_S and the number of samples N acquired in the time acquisition T. N is given by:

$$N = T \cdot f_S \tag{3.15}$$

Taking into account the input frequency f_{in} now, it required that also the number of cycles M must be an integer in the same timing window T:

$$M = T \cdot f_{in} \tag{3.16}$$

Hence, the ratio between the frequencies is given by:

$$\frac{f_S}{f_{in}} = \frac{N}{M} \tag{3.17}$$

To suppress the spectral leakage it is enough to select M as prime number. If this condition can not be matched, a second approach is available to reduce the undesired leakage effect and it is based on the windowing technique [85]. The input sequence $x[t]$ is weighted for a suitable function $w[t]$ named window function and also referred as tapering function or apodization function. It returns a new signal $h[t] = x[t] \cdot w[t]$ which is characterized by a decreased discontinuity at the edges of the acquisition interval. However, it leads to wider lines, namely, to a reduced resolution. There are different type of functions such as Bartlett window (also called triangular function), Hann window, Hamming window and Blackman-Nuttal window (which belongs to the cosine function class) or Welch window (it is a polynomial function). The selection of the window depends on the signal, but in general the Hanning window is a good choice for most of the cases.

Chapter 4

Digital Signal Processor for radiation sensors

4.1 Introduction

The possibility to implement a fully digital front-end channel has several advantages compared to its analog counterparts. Interesting features such as configurable filtering and data compression can be implemented locally on-chip. In order to realize these tasks, a continuos sampling of the input signal provided by a preamplifier is required. The acquired data can then be stored in a memory as a circular buffer and consequently processed by the system. The digital output data stream is the result of filters manipulation to extract from the signal the physical quantities of interest such as energy, time, position or particle classification, depending on the requirements. The programmability of the filters guarantees an important flexibility. In the next sections an overview on the DSP advantages and of key aspects of digital filters is given. Then a detailed description of the employed filters and the adopted solutions for this work will be discussed.

4.1.1 Difference between analog and digital implementation

Digital signal processing offers a number of important advantages:

- The convenient system reconfiguration is perhaps one of the most interesting aspect of the DSP field. For instance, some specific coefficients of a filter can be programmed without changing any other configuration of the chip. This fact ensures an important flexibility during an experiment or a general purpose application.

- A digital device can extract energy, time and pulse shape information from a signal. An appropriate design guarantees low cost and also reliability.

- Some undesired effects such as the pile-up and baseline drift can be easily corrected on digital side with high accuracy.
- Automatic calibration circuits can be employed preventing any manual intervention.
- Many compression algorithms have been developed to reduce the required bandwidth.
- A digital approach is suitable for the management of a massive number of parallel channels.

They also present a few issues:

- Usually the algorithms are complex and require more development time.
- In some applications, the real-time computation represents the bottleneck of the system speed.
- Power consumption at high processing speed can be significant.

4.2 Introduction to digital filtering

4.2.0.1 Linear time-invariant systems (LTI)

This section discusses the linear time-invariant (LTI) systems due to their importance in DSP field [87]. Firstly, in contrast to the continuous systems that handle continuous signals x(t), a discrete device is based on a sequence of samples x[n] eventually stored into a memory. The linearity is a property of those systems whose output is the linear combination of at least two distinct inputs. Let's consider the case of an input x_1[n] applied to a system and its output y_1[n] as shown in Eq 4.1:

$$x_1[n] \rightarrow y_1[n] \tag{4.1}$$

Similarly, an independent input x_2[n] results in y_2[n] as reported in Eq 4.2:

$$x_2[n] \rightarrow y_2[n] \tag{4.2}$$

If the system is linear, the sum of the individual input signals gives the superposition of the distinct outputs as shown in 4.3:

$$x_1[n] + x_2[n] \rightarrow y_1[n] + y_2[n] \tag{4.3}$$

Another property of a linear system is the homogeneity. Given two constant factors f_1 and f_2, if the inputs are scaled by these constant, then also the output changes accordingly:

$$f_1x_1[n] + f_2x_2[n] \rightarrow f_1y_1[n] + f_2y_2[n] \tag{4.4}$$

Finally, the time-invariant property can be expressed using once again Eq 4.1. If a delay D is applied to the input, for a time-invariant system it is expected that also the output will be shifted by the same quantity. This concept results in Eq 4.5:

$$x_D[n] = x_1[n-D] \rightarrow y_D[n] = y_1[n-D] \tag{4.5}$$

These properties are the most fundamental building ones of the DSP.

4.2.0.2 Impulse response

The impulse response h[n] to the discrete delta function δ[n] is the way to characterize a LTI system. The function δ[n] is a normalized impulse, which means that when n = 0 its value is 1, otherwise it is zero:

$$\delta[n] = \begin{cases} 1, & \text{if n = 0} \\ 0, & \text{if n} \neq 0 \end{cases} \tag{4.6}$$

Hence, using the linear property of the system:

$$\delta[n] \rightarrow h[n] \tag{4.7}$$

which is the impulse response of a LTI system when the input is δ[n].

4.2.0.3 Convolution

Let's consider now an input signal x[n] formed by the sequence x[0], x[1], ..., x[n]. By definition of the discrete delta function in Eq 4.6, x[n] can be expressed as the decomposition of a set of impulses, where each one is a scaled and shifted version of the delta function:

$$x[n] = x[0]\delta[n] + x[1]\delta[n-1] + ... + x[n]\delta[n-m] \tag{4.8}$$

For the LTI system properties and Eq 4.7, the output of the system stimulated by the sequence x[n] can be written as:

$$y[n] = x[0]h[n] + x[1]h[n-1] + ... + x[n]h[n-m] \tag{4.9}$$

which is the sum of individual impulse responses as expected. This formula can be also compacted into the following expression by changing the summation index:

$$y[n] = \sum_{m} h[m]x[n-m] \tag{4.10}$$

Eq 4.10 is named *convolution* and it is the cornerstone of the DSP. It is widely adopted in several algorithms and implemented in many applications. When a filter is considered, the impulse response is termed *filter kernel* or *convolution kernel*. In the next section, the two classes of filters used in digital processing will be presented.

4.2.1 FIR and IIR filters

Filtering is probably the most important task that a DSP device has to carry out. It concerns the data manipulation of an input signal in order to obtain a desired result. Typically a filter changes the shape of a signal in the time domain to improve the SNR or it modifies its spectra to cut off high frequencies components or it lets a specific band pass unaltered. All these processings are realized with two classes of filters in LTI systems.

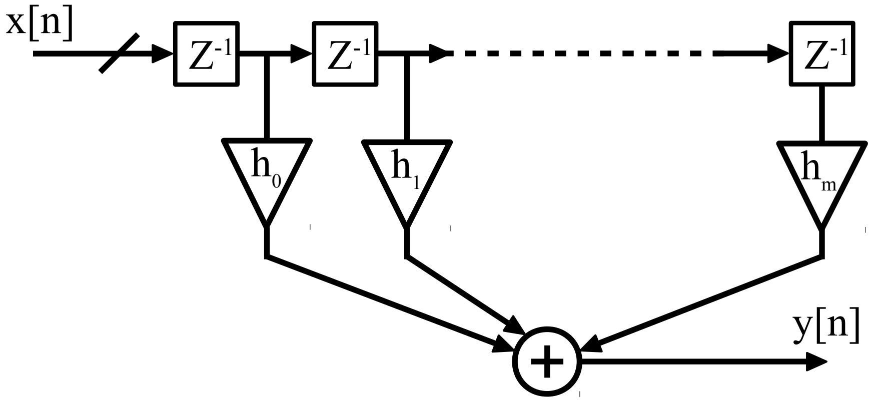

Figure 4.1: Block diagram realization of a generic FIR filter.

The first category of digital filters is referred to as finite impulse response (FIR) and a generic block diagram description is reported in Figure 4.1. Its name is due to the bound interval [0, M] where the impulse response h[n] has nonzero value. M is the filter order, while the length L is equal to M + 1. Depending on the field, the impulse response coefficients h[n] are called coefficients, weights or taps of the filter. Because of the limited interval, Eq 4.10 assumes the form:

$$y[n] = \sum_{i=0}^{M} h[m]x[n-m] \tag{4.11}$$

where the output y[n] is computed only on the current sample x[n] and the previous M records. An example of this kind of filter is the moving average one used in this work to realize the baseline subtraction. It will be described later in Section 4.5. The second class of filters differs from the FIR implementations becasue of their interval which is extended to infinite [0, ∞). For this reason, they are termed infinite impulse response (IIR). Consequently, due to the infinite range Eq 4.10 for a IIR filter is:

$$y[n] = \sum_{i=0}^{\infty} h[m]x[n-m] \tag{4.12}$$

Expressed in this way, a IIR filter output is simply non-computable. A workaround is to define a sub-set of the filter coefficients coupling them to each other through constant-coefficient linear difference equations. Thus, the IIR outcome computation can be realized through recursion, namely, it depends in general on the current sample x[n], the previous L samples and the previous M outputs as formulated in Eq 4.13 and depicted in Figure 4.2:

$$\begin{split} y[n] = b[0]x[n] + b[1]x[n-1] + ... + b[L]x[n-L] + \\ +a[1]y[n-1] + a[2]y[n-2] + ... + a[M]y[n-M] \end{split} \tag{4.13}$$

which is the extension of:

$$y[n] = \sum_{i=0}^{L} b[i]x[n-i] + \sum_{i=1}^{M} a[i]y[n-i] \tag{4.14}$$

This peculiar feedback characteristic makes the IIR filters potentially unstable. Indeed, in contrast to a FIR output which has a zero value when the input signal becomes zero, a recursive implementation can lead to an infinite nonzero outcome. If it is not carefully designed, the feedback path introduces perturbations that cause infinite oscillations. However, the high performance and the small hardware requirements largely compensate all the efforts in their design. An example of recursive filter will be shown in detail in Section 4.7 where the CR-RC^4 block will be discussed.

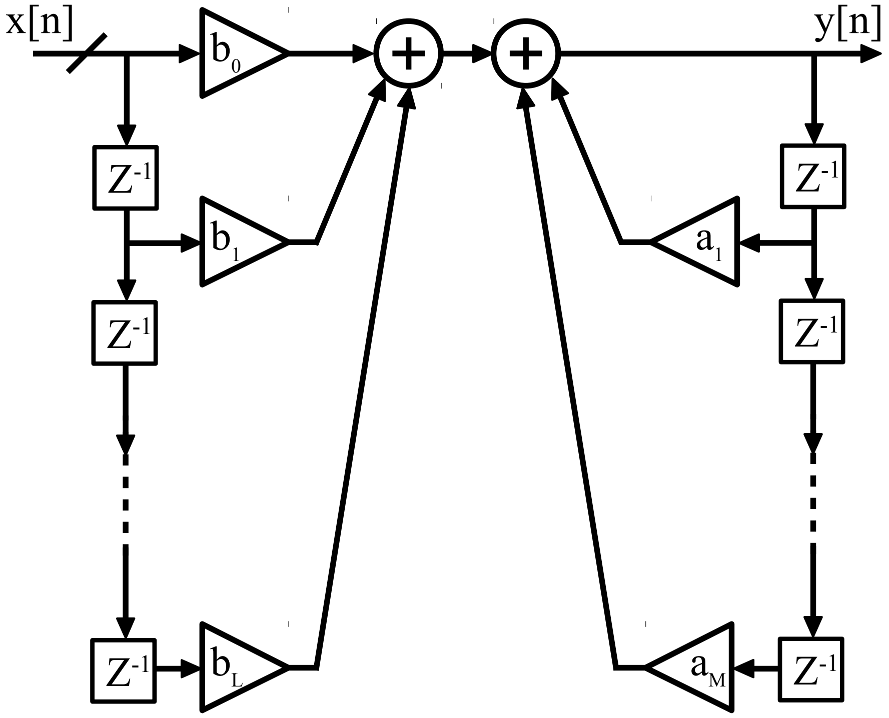

Figure 4.2: Block diagram realization of a generic IIR filter.

These two classes can be equivalently described in the z-domain using the z-transform. If the Laplace transform is useful to study continuous signals, the z-transform is a similar tool for discrete ones. Considering the impulse response h[n], its z-transform is defined in Eq 4.15:

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} \tag{4.15}$$

where z is the complex variable. H(z) is named the *transfer function* of the filter. The z-transforms give a powerful analysis tool for LTI basing on three properties: linearity, delay and convolution property. The linearity and delay properties are the same of Section 4.2, while the convolution affirms that given two discrete signals h[n] and x[n], the z-transform of their convolution is simply the product of their individual z-transforms [88]:

$$y[n] = h[n] * x[n] \Leftrightarrow Y(z) = H(z)X(z) \tag{4.16}$$

From Eq 4.16, H(z) can be expressed as the ratio of the two z-transforms X(z) and Y(z). Indeed, using the linearity, let's consider the z-transform of the Eq 4.14:

$$Y(z) = \sum_{i=0}^{L} b[i]z^{-i}X(z) + \sum_{i=1}^{M} a[i]z^{-i}Y(z) \tag{4.17}$$

and collecting the Y(z) terms the expression can be rearranged as:

$$Y(z)\left[1 - \sum_{i=1}^{M} a[i]z^{-i}\right] = \sum_{i=0}^{L} b[i]z^{-i}X(z) \tag{4.18}$$

and finally the general transfer function of a IIR filter is given by:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{L} b[i]z^{-i}}{1 - \sum_{i=1}^{M} a[i]z^{-i}} \tag{4.19}$$

If the coefficients a_i in Eq 4.19 are equal to zero, the expression returns the FIR filter representation.

4.3 Digital Signal Processor overview

In this section an overview of the designed digital signal processor is presented. Figure 4.3 reports the general architecture at block level. On the left and top sides the input signals are collected, while on the rightmost side the outputs are grouped.

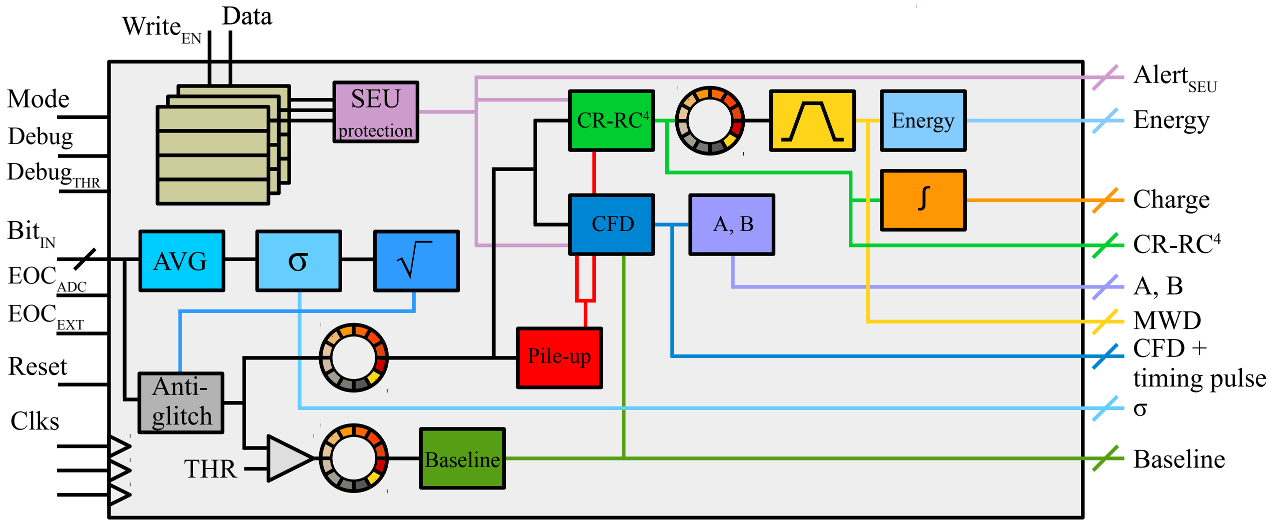

Figure 4.3: Digital signal processor architecture.

The whole chip is managed by a Mealy Finite State Machine (FSM) equipped with 7 states whose state diagram is shown in Figure 4.4. Table 4.1 itemizes a brief description of each step.

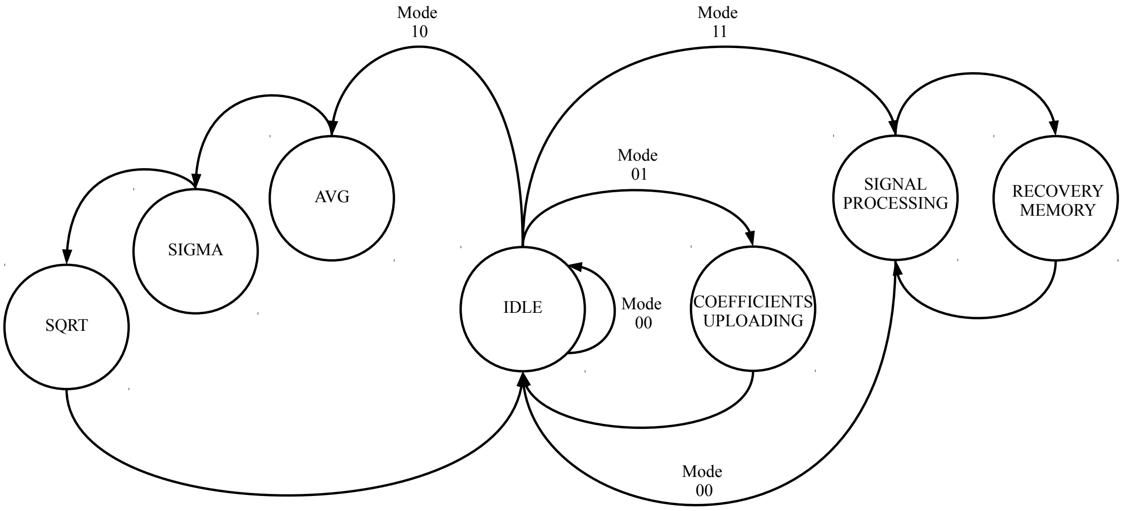

Figure 4.4: State diagram of the finite state machine for the digital signal processor.

State	Value	Description
IDLE	000	This is the default state of the FSM.
COEFFICIENTS UPLOADING	001	In this configuration the user can upload the memory content of the system.
AVG	010	If selected, the processor computes the average noise.
SIGMA	011	The chip derives the variance of the noise.
SQRT	100	This task performs the computation of the standard deviation.
SIGNAL PROCESSING	101	The core of the processor. Briefly, this state allows to obtain both energy and timing information adopting an adequate filtering described in detail in later sections.
SEU_RECOVERY_MEMORY	110	In this stage the system recovers the memory errors caused by bit inversions.

Table 4.1: FSM description of the digital signal processor.

The default state is represented by the IDLE. Here the system waits for further instructions provided by the *Mode* input signal. The COEFFICIENTS UPLOADING is a stage used to upload the coefficients used by the filters to update the configurable thresholds, to modify the delay for the timing filter and to change other parameters. The uploading channel (Data) is enabled with a dedicated signal (Write$_{EN}$). AVG is the state designed to compute the average noise collecting 4096 samples into an accumulator and performing a bit-shift for the required division. The state SIGMA shares the same accumulator of the AVG one, deriving the variance of the noise. This state, as well as the following, is automatically reached by the system without any user intervention. Afterwards the variance is passed to SQRT which is the task reserved for the computation of the standard deviation. The processor uses the bisection algorithm to complete this assignment. The SIGNAL PROCESSING state is the core of the processor. After an anti-glitch cleaning, the signal is splitted into two branches: one is filtered by a digital comparator and it is used to calculate the baseline, the other is stored into a circular buffer and further distributed among three submodules. The most important blocks are the CR-RC^4 shaper adopted as first stage for the energy extraction and the Constant Fraction Discriminator (CFD) used to obtain timing information. For what concerns the energy derivation, a chain formed by the CR-RC^4 and a trapezoidal filter was implemented. The trapezoidal shape was obtained carrying out a Mobile Window Deconvolution (MWD) filter. To provide a complementary information, the CR-RC^4 outcome is integrated in time when it is above a configurable threshold. Considering the timing, a linear approximation was assumed around the zero crossing region for the CFD result, thus a linear interpolation was employed to get the angular coefficient and the zero crossing time. If the shaper wasn't enough a pile-up rejection system guarantees that no pulse overlap will occur during the data processing. Lastly, a SEU mechanism protects the main memory against errors generated by an undesired particle interaction. This precaution was implemented through a Triple Modular Redundancy (TMR) of the memory contents. The triplication of the registers justifies also the three clocks depicted in Figure 4.3. To recover the bit inversion errors, a state have been dedicated: SEU_RECOVERY_MEMORY takes care about the bit flips of the main memory where the coefficients and the other parameters are stored. The state is operative when the FSM is in the SIGNAL PROCESSING stage only. Indeed, it is not possible to correct any coefficient without uploading it before.

Command	Value	Description
IDLE	00	The FSM is maintained in its default state, waiting for instructions.
COEFFICIENTS UPLOADING	01	The coefficients upload is enabled.
AVG + SIGMA	10	As mentioned before, the processor computes the average value of the noise and it derives its sigma using a specific module for the square root. Because the tasks involved form an automatic processing chain, this command can return to the IDLE after an initial trigger.
SIGNAL PROCESSING	11	This task is dedicated to the processing of the signal provided by the Bit$_{IN}$ input.

Table 4.2: Possible configuration modes.

The *Mode* is a 2-bits long input signal and it defines the current task that the processor have to execute. The mode is configurable by the user and the possible commands are summarized in Talbe 4.2. The chip is also equipped with a debug mode. To consider valid the data coming from Bit_{IN} an EOC_{ADC} signal is checked. Ideally, this confirmation is given by the ADC, but in a real application a pitfall can occur. Therefore, a *Debug* signal enables the acceptance of an external check signal (EOC_{EXT}). As described in more details in the following sections, the system is able to evaluate a suitable threshold based on the noise level. However, similarly to the previous case, $\textit{Debug}_{THR}$ permits to bypass this default threshold generation, setting the selected one by the user. Finally, a *Reset* signal cleans all the internal registers as well as the memory content.

About the hardware evaluation, another consideration played a role in the final decision. This DSP chip has been implemented following as much as possible a shared-memory architecture [86]. For instance, the registers of the main circular buffer broadcast their contents to the CR-RC^4 block described in Section 4.7 as well as to the CFD timing filter discussed in Section 4.9.4.

4.4 Anti-glitch system

The typical signal of an amplifier can be very noisy, making difficult the discrimination of the signal from the noise. An impulsive contribution can be also introduced, generating a false pulse recognition flag. Thus, the first processing step led to the implementation of an anti-glitch filter [30] as shown in Figure 4.5.

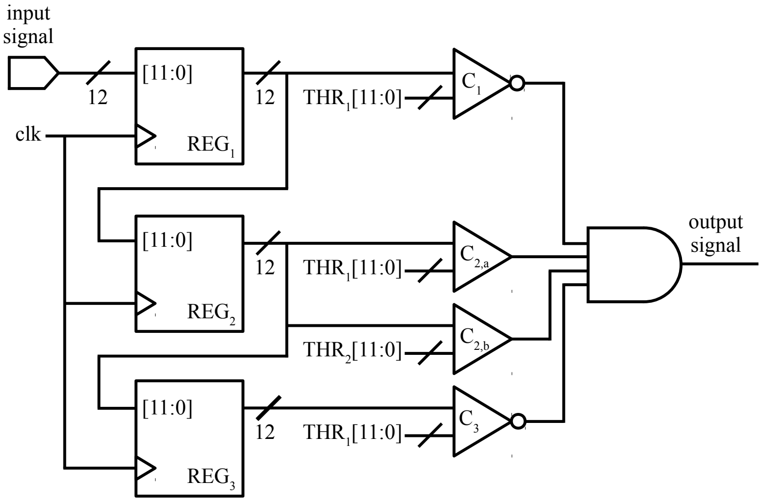

Figure 4.5: Anti-glitch circuit formed by three 12-bits long registers and four comparators.

This system is based on three registers, each one 12-bits long, that form a chain. Their outcomes are continuously monitored by four comparators. Three thresholds, termed C$_1$, C$_{2,a}$ and C$_3$ in the picture, are set to the automatic value or the selected one by the user. The last threshold C$_{2,b}$ is set to a higher value of the previous ones and it is connected to the central register. If the three samples satisfy the four threshold conditions at the same time, then a logic output is generated and the content of REG$_2$ is sent to the circular buffers. This double threshold architecture was chosen for the reason illustrated in Figure 4.6. Given a threshold on the leading edge, it is straightforward to distinguish the signal above the threshold from the noisy baseline. By contrast, it becomes tricky on the falling edge of the pulse. In particular, the problematic area was magnified in the picture. Without the double threshold scheme, a part of the signal would be cut off and it would not be stored into the circular buffers. It is important to stress that those buffers directly feed the next filter stages. As consequence, the CR-RC^4 block, the moving window deconvolution filter and the constant fraction discriminator would share an abrupt output at that time and part of the information would get lost and distort. Furthermore, in this case also the memory reserved for the baseline computation stores an undesired bump, altering the baseline level. This issue could affect the baseline restoration of an incoming close signal. Hence, THR$_2$ avoids this drawback implementing a NAND logic $C_{out} = \overline{C_1 \cdot C_{2,a} \cdot C_{2,b} \cdot C_3}$.

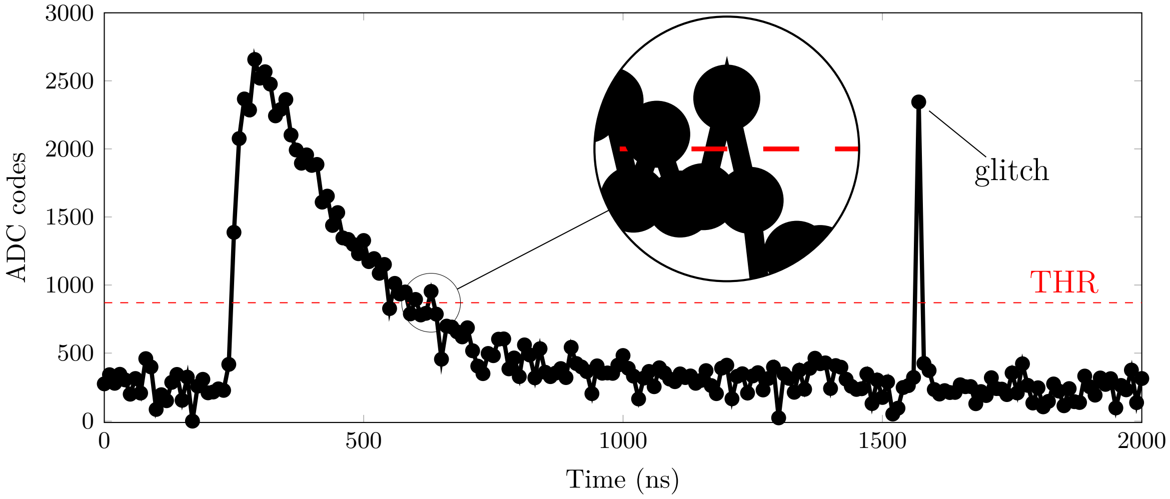

Figure 4.6: Magnification of a typical noisy amplifier output with a single threshold level.

In this work both THR$_1$ and THR$_2$ are externally programmable. It is also possible to let the system find a suitable threshold above the noise. This feature is based on the computation of the average noise level and its deviation σ_{rms}, extensively explained in Section 4.4.1. A successive comparator decides if to include the validated data into the buffer dedicated to the baseline subtraction. This mechanism is exposed in Section 4.5.

4.4.1 Self-adjustable thresholds

After the stage dedicated to uploading the coefficients, a state of the FSM is reserved to the computation of the average value of the noise and the associated σ_{rms}. This operation is performed on a relatively large number of fixed samples M and it is useful for a self-adjustable thresholds scheme implemented on the DSP chip. Indeed, assuming a white noise and jitter explained in Chapter 3 and Section 4.9.1, the idea is to find the average value of the noise perturbation in order to define the thresholds THR$_1$ and THR$_2$ introduced in Section 4.4 as follow:

$$THR_1 = AVG + \sigma_{rms} F_{THR} \tag{4.20}$$

$$THR_2 = AVG + \sigma_{rms} F_{THR2} \tag{4.21}$$

where both F$_{THR}$ and F$_{THR2}$ are 8-bits fixed-point configurable factors. The details about the fixed-point representation are provided in Section 4.7.5, however F$_{THR}$ and F$_{THR2}$ are expressed as Q(4.4). This means that the 4 MSBs are dedicated to the integer part, while the others are used for the decimal representation. Therefore the maximum value for the quantity σ_{rms}F$_{THR}$ can be $15\sigma_{rms}$ and the minimum available value is $0.0625\sigma_{rms}$. To complete the task, a first stage is used to the computation of the average defined in Eq 4.22:

$$AVG = \frac{1}{M}\sum_{i=0}^{M-1} x[i] \tag{4.22}$$

The sum is iterated by acquiring the first M = 4096 available samples and the result is stored into an adequate accumulator, then a 12-bits right shift carries out the division. The following state needs to evaluate the σ_{rms}:

$$\sigma_{rms} = \sqrt{\frac{1}{M-1}\sum_{i=0}^{M-1}(x[i] - AVG)^2} \tag{4.23}$$

where the approximation M - 1 $\approx$ M was made considering the large value of M. This slight change is required to simplify the calculation by using once again the shift operation. The registers involved in this task are dimensioned to tolerate a SNR around 10, which is a typical value for this applications. Once computed, THR_1 and THR_2 are defined accordingly with Eq 4.20 and Eq 4.21 by default. However, the user can bypass this configuration with a debug command. Although a *on the fly* computation of the sigma was possible adopting the circular buffer as introduced in Section4.5, the current pipelined state architecture was preferred. Indeed this choice prevent the many operations involved with the circular buffer approach and ultimately it saves power. This current version reduces also the number of the required registers. The way to compute the square root is the topic of the next section.

4.4.2 Square root computation: the bisection algorithm

Many algorithms for the square root computation have been proposed such as Newton-Raphson method [89], SRT-Redundant method [90] and Taylor-series expansion algorithm [91]. Most of them required seed initialization, look-up tables or complex circuitry to converge toward the root [92].

In this work we adopted the bisection method [93] [94]. This approach is particularly simple and the hardware implementation is not challenging. In the following code snippet a Python code that carries out the algorithm is reported, based on an initialization and a while cycle.

```
##################################################

if num < 1:
    lower = num
    upper = 1
else:
    lower = 1
    upper = num

while (upper - lower > epsilon):
    guess = (upper + lower)/2.0
    if pow(guess, 2) > num:
        upper = guess
    else:
        lower = guess

##################################################
```

The aim of this script is to find a square root approximation of $\sqrt{num}$ given the number *num* > 1 and using an error *epsilon* as stop condition. The two numbers *upper* and *lower* define the interval [upper, lower]. The first if/else statement initializes these numbers as reported. During the iterations of the while cycle, the process maintains the inequality upper $< \sqrt{num} <$ lower and halves the interval by computing the quantity *guess*. Indeed, if the square of this value is larger than the number num, the upper value will be set to guess, otherwise lower is equal to guess. The repetition of these two steps until the while condition is satisfied leads to the solution. The hardware implementation was marginally modified and it is shown in the pseudocode below based on the SystemVerilog language:

```
////////////////////////////////////////////////

always_comb begin
    if (reset)
        initialize guess, guess2;
    else begin
        if (counter > 5'b0 && counter <= accuracy) begin
            guess = (upper + lower) >> 1;
            guess2 = guess*guess;
        end
    end
end


always_ff @(posedge clk) begin
    if (reset)
        initialize counter;
    else begin
        if (counter == 5'b0) begin /*To initialize the values*/
            lower <= 1;
            upper <= num;
        end
        else begin /*loop*/
            if (guess2 > num)
                upper <= guess;
            else
                lower <= guess;
        end
```

```

            if (counter == accuracy) /*output*/
                sigma <= guess;
            counter <= counter + 1;
        end
    end

    /////////////////////////////////////////////////
```

The hardware description is composed by one fully combinatorial block and a sequential one. The first circuit is dedicated to the computation of the *guess* and its power of two (referred to as *guess2* in the pseudo-code). In contrast to the while loop, here it was introduced a *counter* to guarantee a break condition and to prevent a non-convergence state. Some preparatory tests have demonstrated that in most cases the convergence to a solution occurs within 15-20 clock cycles. Therefore a 5-bits programmable register is reserved to set the *accuracy* (which is homonym in the pseudo-code), i.e. the maximum number of approximation steps. The sequential block is sub-divided in three parts. After a reset, the first statement initializes the upper and lower variables as indicated in the Python code. Next, a loop is performed until the equality $counter = accuracy$ is achieved. When this condition is satisfied the content of guess is stored in the output register *sigma* and it is available for the THR$_1$ and THR$_2$ formulation. At the same time, the counter is updated at each clock cycle. The RTL outcome of σ_{rms} is in accordance with the result returned by the software implementation.

4.5 Baseline subtraction

A fundamental data manipulation concerns the baseline restoration. This operation is required in many processing units because of the fluctuations that affect the reference level against which the signal is measured [29] [34]. This amplitude shift can be classified in two categories: the first class belongs to the systematic effects that can derive by the combination of the trigger generation, the baseline drift induced by temperature variations, noise injected by the supply voltage or AC/DC coupling. The second type of perturbations are the complementary ones and can be termed non-systematic. For the first type of signal distortion a look-up table can be adopted. Indeed, due to their systematic nature, the knowledge of the these effects can correct the digital output by storing into a memory a compensation pattern. However, due to the lack of this time-dependent information and the general purpose approach of

the current DSP, only non-systematic effects were considered in the current work. In particular, white noise and jitter effects have been taken into account as described in Chapter 3 and Section 4.9.1.
To face the problem, a Moving Average Filter (MAF) was chosen [96]. This filter belongs to the low-pass FIR class as described in Section 4.2.1 and its coefficients are equal to $1/N$, where N corresponds to the number of delays:

$$MAF_{out}[n] = \frac{1}{N}\sum_{i=0}^{N-1} x[n-i] \tag{4.24}$$

The equivalent transfer function in Z domain is:

$$H_{MAF}(z) = \frac{1}{N}\Big(1 + z^{-1} + z^{-2} + ... + z^{-(N-1)}\Big) \tag{4.25}$$

We can observe that using a little trick [96], Eq 4.24 can be rewritten in its recursive form:

$$MAF_{out}[n] = MAF_{out}[n-1] + \frac{x[n]}{N} - \frac{x[n-N]}{N} \tag{4.26}$$

The Eq 4.26 takes advantage of many points:

- Only two operations, one addition and one subtraction, are required to compute the output, whatever is the length of the filter. Furthermore, these operations are not time-consuming as the multiplication is.
- The computation can be realized using a fixed-point representation. Additional details will be provided in Section 4.7.5, but in this context the benefit is basically the speed of the operations compared to the use of the floating-point representation. Moreover, a integer representation gives a further convenience about the round-off error. Indeed, this algorithm is not affected by this error since no drift quantity is introduced by integers during the calculations and no caution is demanded [97].
- The pointers are trivial to implement in hardware.

All these points contribute to make the recursive form of Eq 4.26 a faster algorithm compared to other solutions and suitable to be implemented on silicon. Considering the last benefit, this type of filter is particularly straightforward to realize using a circular buffer [98]. Figure 4.7 illustrates the equivalence between this type of buffer and a FIFO. Essentially, a circular buffer is a FIFO rolled up with a pointer that overwrites the older data in the register. After the writing step, the pointer is incremented by one and the operation is repeated. A similar mechanism takes place when the buffer must be read. In contrast to the FIFO, the main advantage of this implementation is the storing of data inside the buffer without any internal shift. Moreover, a wise choice of N as power of 2 leads to a simplified circuit. Thus, this architecture is convenient for fast processing data tasks.

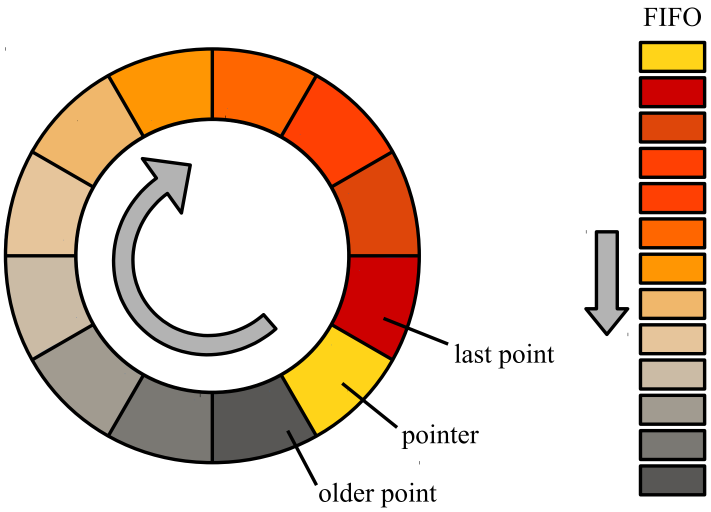

Figure 4.7: The circular buffer (on the left) and its equivalent FIFO (on the right).

The last important topic concerns the length N of the filter. This decision was a trade-off between performance, usefulness and hardware demanding. In Figure 4.8 the frequency response of the MAF for different values of N analyzed is reported. Apart from the impossibility to have a direct control on the cutoff frequency once N is fixed, it is quite evident that this circuit behaves as a bad low-pass filter because it is characterized by a slow roll-off and a poor stop-band attenuation at high frequencies. These weak results are counterbalanced with a good performance in the time domain and therefore this work looked also at the rms noise σ_{rms} values, reported in Table 4.3.

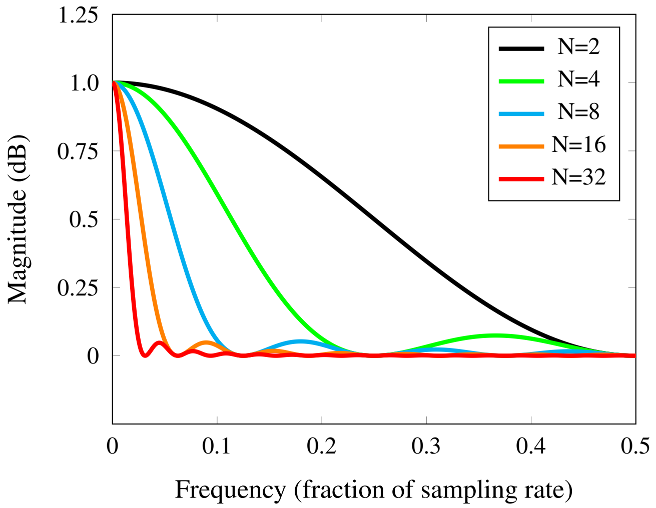

Figure 4.8: Frequency response of the MAF to a impulse considering different N.

N	Average	σ_{rms}	%
2	270.3	145.3	1
4	270.3	102.7	0.71
8	270.3	72.5	0.50
16	270.3	51.3	0.35
32	270.2	36.4	0.25

Table 4.3: σ_{rms} for different N values.

The last column itemizes the ratio between the current σ_{rms} and the σ_{rms} of N = 2 kept as minimum value reference. The increasing of N progressively reduces the σ_{rms} values and the best result is achieved for N = 32 in the listed group. However, the difference between N = 16 and N = 32 represents a small improvement compared to the σ_{rms} variation among N = 2 and N = 16 that is already a quite good outcome. Furthermore, the maximum value of N would implies also a larger demand of memory confronted to N = 16, keeping in mind that each stored word is 12-bits long. Ultimately, based on these considerations, an order N = 16 was chosen to restore the baseline of the raw signal in order to feed the timing filter discussed later in Section 4.9.4. Thus, this calibration circuit uses the accumulator depicted in Figure 4.9 to store the sum of Eq 4.24, then the result is shifted by 4 positions on the right to perform the division. The accumulator is physically the same of the average and sigma blocks to share resources and to further reduce the hardware area requirement.

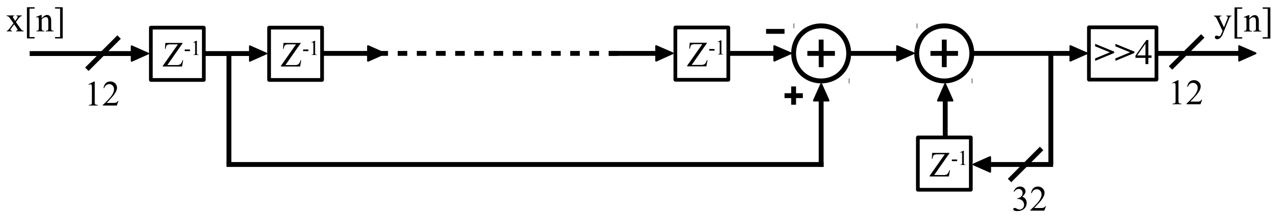

Figure 4.9: Block diagram realization of the MAF.

This computation is iterated until the double threshold scheme introduced in Section 4.4 detects a signal. When it occurs, the last computed baseline is no more updated. It is kept constant within the acquisition window of the pulse and its current value is subtracted to the raw signal. If the double threshold condition of the anti-glitch system is not satisfied anymore, the baseline is updated again.

A reserved baseline restorer is dedicated to the digital pulse shaper described in Section 4.7.2. In this specific case the filter strongly manipulates the raw signal, obtaining a significantly increased SNR at the output. For this reason, the circular buffer implemented for this module has just N = 4. The other components of this circuit are similar to the others introduced before.

4.6 Pile-up

When at least two consecutive pulses are too wide or when they occur close in time, an overlap of their signals takes place. This event is named pile-up effect and it affects the amplitudes of two or more peaks [9]. The obvious consequence of this superposition is a corrupted output, whose energy measurement may be useless. A way to recover such distorted signal is to adopt a digital shaper as the one that will be described in Section 4.7 or deconvolve the data stream as will be explained in Section 4.8. These methods can reduce the occurrence of this effect and its impact on the data quality as shown in Figure 4.10. However, these algorithms could not be able to totally remove the pile-up [99]. Thus, a specialized circuitry is required to reject these kind of events and the next Section is dedicated to this topic.

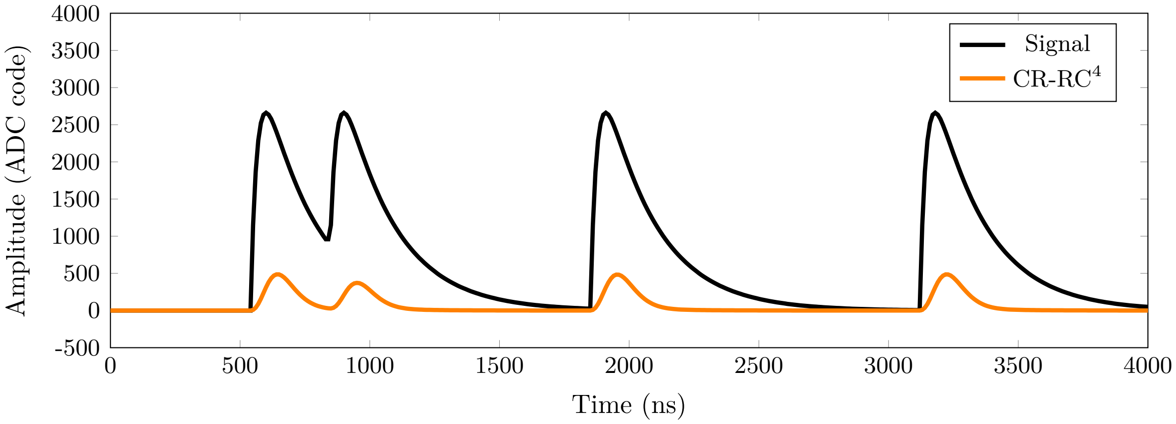

Figure 4.10: Pile-up effect.

4.6.1 Pile-up rejection circuit

Basically, the idea was to implement a circuit to adequately separate two or more input shapes too close to each other. The use of a counter to monitor the time-distance between two events appeared as the more reasonable choice. Thus, the first step was to find a robust enough method to check the signal and a differentiator was a suitable option. Based on a backward difference [100], the differenziator D of length N can be carried out as FIR filter where the coefficients *c[i]* are the unit organized in a positive set and a negative one:

$$D[n] = -\left(\sum_{i=0}^{\frac{N}{2}-1} c_-[i]x[n-i] - \sum_{i=\frac{N}{2}}^{N-1} c_+[i]x[n-i] \right) \qquad (4.27)$$

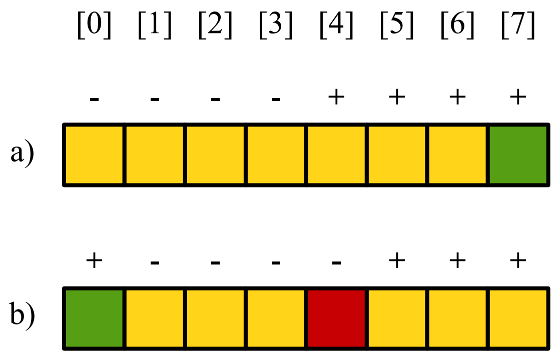

Figure 4.11: Differentiator mechanism with circular buffer.

Figure 4.11 clarifies the concept behind the diffentiator realized in Eq 4.27. In a) N = 8, hence the eight squares represent a circular buffer where the green one points out the current pointer position to write the incoming data. In this configuration, the differentiation can be accomplished performing the subtraction of the first half registers and the sum of the last half memories of the buffer. In b) the pointer is advanced by one position by writing the new data in the first register and it dragged also the sign. The previous computations should be performed once again to obtain the difference D. This operation is quite expensive in terms of computation power. Thus, in this thesis a strategy was adopted to achieve an efficient calculation performance. Compared to the initial situation a), in b) the minor changes were highlighted. More in detail, the first register content *[0]* was deleted and replaced with a new value, while the data stored in *[4]* is the same of a), but with the sign flipped. The other yellow boxes do not suffer any change in content or sign. This means that Eq 4.27 is reduced to the sum of the previous value in *[0]*, the addition of the new data that will be stored in *[0]* and the subtraction of twice the value saved in *[4]*, namely, the differentiator filter is given by the recursive form:

$$D[n] = D[n-1] + buffer[0] + new\_data - 2 \cdot buffer[4] \tag{4.28}$$

Thus, the basic algorithm formed by 8 operations of sum was shrunk to 3 sum and one multiplication/shift, leading to the same output shown in the picture. In the physical implementation a length of N = 8 was selected as compromise between the tracking performance and the employment of a too large accumulator. The pointer values were adjusted to run on the main circular buffer of length equal to 16. This lengths discrepancy was due to the maximization of the resource sharing among the modules. Therefore, the pile-up rejection circuitry is composed by a digital comparator that confronts the accumulator outcome with a configurable threshold THR$_p$. When the value is above THR$_p$, a 12-bits counter is incremented by one by starting a loop. The iteration continues until a programmed count W is achieved. When it happens, the counter is set back to zero and another cycle is available. This loop defines a programmable window W as reported in Figure 4.12.

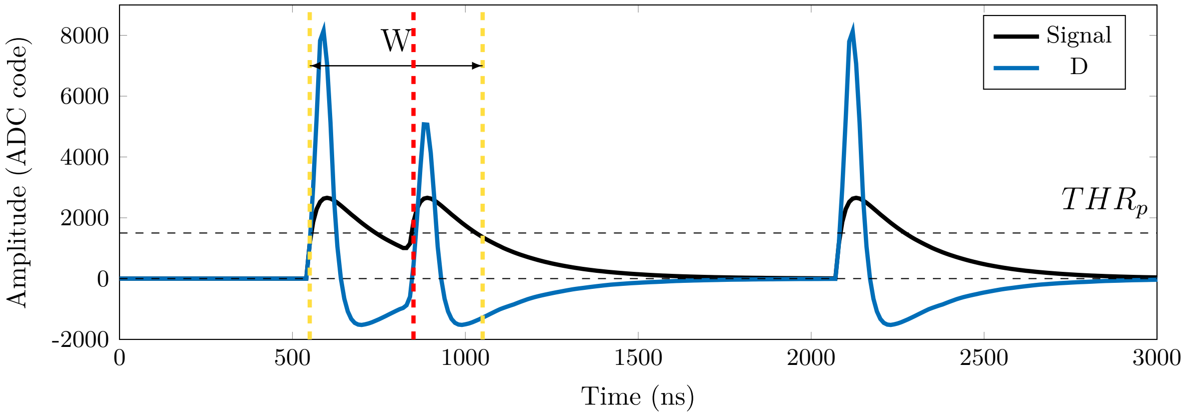

Figure 4.12: Differentiator trend for N = 8.

If the digital comparator fires again inside this range because the differentiator value is larger than THR$_p$, a signal of pile-up rejection is generated (red dashed line) and the counter is set to zero before its nominal time given by W. The signal freezes the activities of both CR-RC^4 digital shaper and the Constant Fraction Discriminator. Then, the system waits an amount of time equals to the selected window. At the end of this cycle, all the registers return to their initial state.

4.7 Digital CR-RC^4 pulse shaper

The charge sensitive amplifier output is the input stage of the DSP circuitry. This signal can directly feed the constant fraction discriminator block to preserve the slope-to-noise-ratio as will be shown in Section 4.9.4. However, it must be manipulated to extract energy information from the signal to maximize the SNR. Thus, this section is dedicated to describe how the preamplifier output was modelled and to provide a description of the digital CR-RC^4 shaper implemented.

4.7.1 Charge sensitive amplifier

In Figure 4.13 the first stage of a typical front-end amplifier named Charge Sensitive Amplifier (CSA) [5] is depicted. I$_s$ is the current from the detector, while C$_l$ is the sum of different contributions due to the detector capacitance, the input transistor capacitance and parasitic capacitance caused by the connection between the detector and the electronics stage. C$_f$ and R$_f$ are the feedback and resistance capacitance, respectively. This circuit is the preamplifier of a readout chain where the following stages, implemented with a $CR - RC^n$ network, determine the shape of the signal. Thus, due to this function, they are called *pulse shaper*.

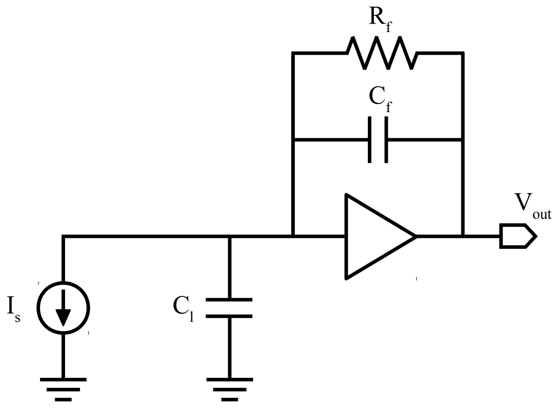

Figure 4.13: Charge sensitive amplifier schematic

The input of the preamplifier is directly connected to the sensor where the detector signal is approximated by the analog version δ(t) of the discrete Dirac-delta function introduced in Section 4.2.0.2. Hence, if the Q$_{in}$ is the total charge stored in the pulse, the input results in I$_s$(t) = Q$_{in}$ δ(t). As discussed in Chapter 1, Q$_{in}$ is the quantity of interest because it is proportional to the energy released into the sensor by the particle. In an ideal case infinite gain and bandwidth for the core amplifiers and an infinite value for the feedback resistor R$_f$ are assumed. The resistor defines the amplifier DC operating point for the biasing. In this way, the CSA operates as an ideal integrator and the output is described accordingly to Eq 4.29:

$$V_{out}(t) = \frac{1}{C_f}\int i(t)\,dt \tag{4.29}$$

Therefore, keeping the δ(t) approximation for the detector signal:

$$V_{out}(t) = \frac{Q_{in}}{C_f}u(t) \tag{4.30}$$

where u(t) is the integral of the delta function referred to as *unit step function* and defined as:

$$u(t) = \begin{cases} 1, & \text{if t} \geq 0 \\ 0, & \text{if t} < 0 \end{cases} \tag{4.31}$$

Hence, the R$_f$ = ∞ leads the CSA output to be a voltage step with zero rise time for t $\geq$ 0. However, this case is unrealistic in practical applications because of the approximation assumed:

- The typical value for the R$_f$ can not be infinite, but it is typically of order of a few MΩ.
- The core amplifiers show a frequency-dependence voltage gain.

Thus, from these aspects the CSA output due to a δ(t) stimulus is modeled as [101] [102]:

$$V_{out}(t) = \frac{Q_{in}}{C_f}\frac{\tau_f}{\tau_r - \tau_f}\left(e^{-\frac{t}{\tau_r}} - e^{-\frac{t}{\tau_f}}\right) \tag{4.32}$$

where τ_r and τ_f are the constant rising and falling times, respectively. Typically the condition $\tau_r \ll \tau_f$ is assumed. This equation was adopted to simulate the preamplifier output in this work by adding the noise level discussed in Chapter 3 and the jitter analyzed in Section 4.9.1. For test purpose, the values collected in Table 4.4 were adopted.

Parameter	Value
Q$_{in}$	$3.5 \cdot 10^{-15}$
C$_f$	$4 \cdot 10^{-15}$
τ_r	$50 \cdot 10^{-9}$
τ_f	$0.2 \cdot 10^{-6}$

Table 4.4: Parameters selected for the simulation.

4.7.2 Digital shaper description

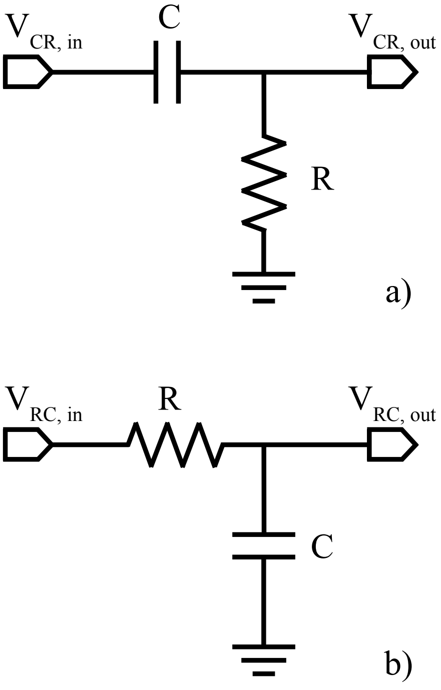

Figure 4.14: a) CR b) RC schematics.

In order to extract energy information from the input signal, a CR-RC^n shaping filter is a popular choice in measurements systems. This is due to the fact that it is not strictly necessary to preserve the original sensor signal shape to fulfill this task. Thus, many implementations have been proposed [104] [103] [105]. From the wide available literature a pole zero correction version was selected and carried out in the current work [106].

A CR-RC^n circuit is a chain of stages where the first element is the CR differentiator block followed by a number n of RC integrators modules. The number of the integrators defines the order number of the filter. To derive the digital network is useful to analyze its components starting from the analog counterpart reported in Figure 4.14. In a) the V$_{CR,in}$ and V$_{CR,out}$ satisfy the equation:

$$RC \cdot \frac{dV_{CR,out}(t)}{dt} + V_{CR,out}(t) = RC \cdot \frac{dV_{CR,in}(t)}{dt} \tag{4.33}$$

In the discrete domain, V$_{CR,in}(t)$ and V$_{CR,out}(t)$ become V$_{CR,in}[n]$ and V$_{CR,out}[n]$, respectively. Thus Eq 4.33 is turned into:

$$\frac{V_{CR,out}[n] - V_{CR,out}[n-1]}{T} + \frac{V_{CR,out}[n]}{RC} = \frac{V_{CR,in}[n] - V_{CR,in}[n-1]}{T} \tag{4.34}$$

where T is the sampling period. Let's substitute now $d = RC/(RC + T)$ and rearrange Eq 4.34 as:

$$V_{CR,out}[n] = d \cdot (V_{CR,in}[n] - V_{CR,in}[n-1]) + d \cdot V_{CR,out}[n-1] \tag{4.35}$$

In Eq 4.35 we recognize a recursive form belonging to the IIR filter class previously described. By using the z-transform, the equivalent transfer function description H$_{CR}$(z) can be conveniently derived as:

$$V_{CR,out}(z) = d \cdot V_{CR,in}(z)(1 - z^{-1}) + d \cdot V_{CR,out}(z)z^{-1} \tag{4.36}$$

Then, remembering Eq 4.19, H$_{CR}$(z) is given by:

$$H_{CR}(z) = \frac{V_{CR,out}(z)}{V_{CR,in}(z)} = \frac{d \cdot (1 - z^{-1})}{1 - d \cdot z^{-1}} \tag{4.37}$$

In Figure 4.14 b) the RC schematic is reported and by repeating a similar derivation, the V$_{CR}$, in and V$_{CR}$, out are related as:

$$RC \cdot \frac{dV_{RC,out}(t)}{dt} + V_{RC,out}(t) = dV_{RC,in}(t) \tag{4.38}$$

The discrete description of Eq 4.38 is given by:

$$V_{RC,out}[n] = (1 - d) \cdot V_{RC,in}[n] + d \cdot V_{RC,out}[n-1] \tag{4.39}$$

where *d* is the same substitution of the CR stage. Also in this case the transfer function H$_{CR}$(z) is derived by applying the z-transform to both sides of Eq 4.39:

$$V_{RC,out}(z) = (1 - d) \cdot V_{RC,in}(z) + d \cdot V_{RC,out}(z)z^{-1} \tag{4.40}$$

Thus H$_{CR}$(z) is obtained:

$$H_{RC}(z) = \frac{1 - d}{1 - d \cdot z^{-1}} \tag{4.41}$$

By combining Eq 4.37 and Eq 4.41 the generic transfer function for a CR-RC shaper of order *n* is obtained:

$$H(z) = \frac{d \cdot (1 - d)^n \cdot (1 - z^{-1})}{(1 - d \cdot z^{-1})^{n+1}} \tag{4.42}$$

Unfortunately, this CR-RC^n pulse shaper version is not already undershoot-free as pointed out in Figure 4.15. This is due to the non-compensated pole. A possible solution is described in the next Section.

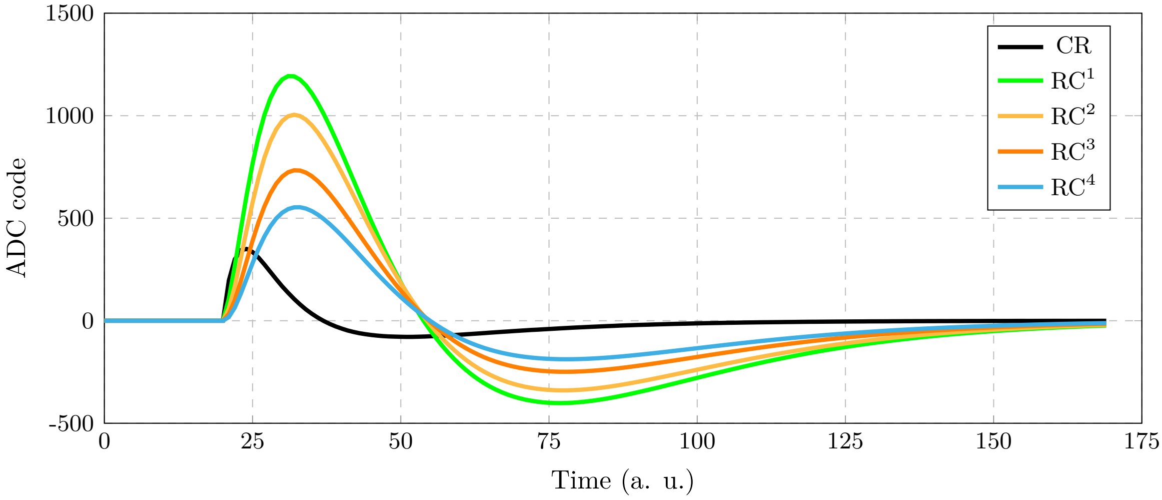

Figure 4.15: Undershoot of CR-RC^4 shaping filter.

4.7.3 Pole zero cancellation

Figure 4.16 shows the frequency response of the filters RC^n to an impulse in the range [1, 8]. A way to compensate the undershoot depicted in Figure 4.15 was proposed in [107] and it is based on the introduction in Figure 4.14 a) of a resistance kR in parallel to capacitance C. Now, the equation that V$_{CR}, in$ and V$_{CR}, out$ have to satisfy is given by:

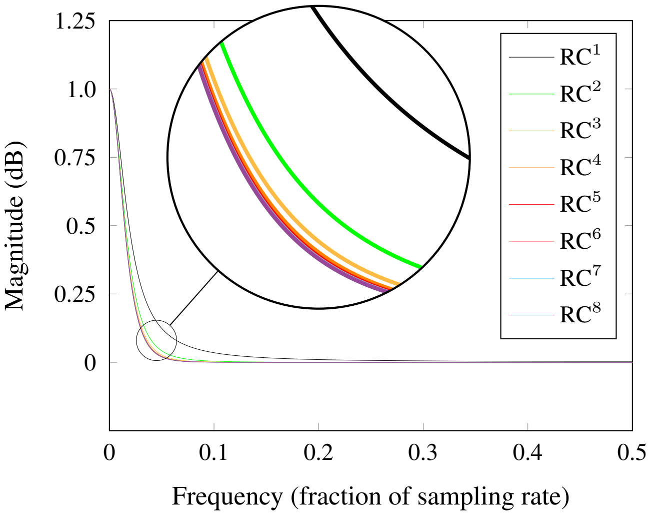

Figure 4.16: Frequency response of RC^n to an impulse considering n = [1, 8]

$$\frac{V_{CR,out}(t)}{R} = \frac{V_{CR,in}(t)}{k \cdot R} - \frac{V_{CR,out}(t)}{k \cdot R} + C \cdot \frac{dV_{CR,in}(t)}{dt} - C \cdot \frac{dV_{CR,out}(t)}{dt} \tag{4.43}$$

Let’s repeat the discretization process to obtain the following equation:

$$\begin{aligned}\frac{V_{CR,out}[n]}{V_{CR,in}[n]} = \frac{V_{CR,in}[n]}{k \cdot R} + \frac{V_{CR,out}[n]}{k \cdot R} &+ C \cdot \frac{V_{CR,in}[n] - V_{CR,in}[n-1]}{T} - \\ &+ C \cdot \frac{V_{CR,out}[n] - V_{CR,out}[n-1]}{T}\end{aligned} \tag{4.44}$$

In order to simplify the long Eq 4.44, the two quantities C and F are defined as:

$$\begin{gathered} C = \frac{kRC + T}{kRC + kT + T} \\ F = \frac{kRC}{kRC + kT + T} \end{gathered} \tag{4.45}$$

If the fraction RC/T is substituted with $d/(1-d)$, Eq 4.45 becomes:

$$\begin{gathered} C = \frac{kd + 1 - d}{k + 1 - d} \\ F = \frac{kd}{k + 1 - d} \end{gathered} \tag{4.46}$$

Hence, Eq 4.44 can be compacted into:

$$V_{CR,out}[n] = C \cdot V_{CR,in}[n] - F \cdot \big(V_{CR,in}[n-1] - V_{CR,out}[n-1]\big) \tag{4.47}$$

The z-transform returns the following equation:

$$V_{CR,out}(z) = V_{CR,in}(z) \cdot (C - F \cdot z^{-1}) + F \cdot V_{CR,in}(z) \cdot z^{-1} \tag{4.48}$$

Finally, the transfer function H$_{CR,PZC}$ of the CR stage equipped with pole zero compensation is given by:

$$H_{CR,PZC}(z) = \frac{C - F \cdot z^{-1}}{1 - F \cdot z^{-1}} \tag{4.49}$$

To simplify the calculations let's assume the approximation $\tau_r \ll \tau_f$ of Section 4.7.1. Thus, Eq 4.32 is reduced to:

$$V_{out}(t) = \frac{Q_{in}}{C_f} \cdot e^{-\frac{t}{\tau_f}} \tag{4.50}$$

As expected, before to operate with the z-transform is necessary to discretize V$_{out}$(t) into V$_{out}$[n], then:

$$V_{out}(z) = \frac{\frac{Q_{in}}{C_f}}{1 - d_{pzc} \cdot z^{-1}} \tag{4.51}$$

where d$_{pzc}$ = e$^{\frac{-T}{\tau_f}}$ and k = τ_f/RC. The pole zero compensation occurs when k = τ_f/RC, namely, the undershoot is filled up. Therefore, the transfer function of a generic CR-RC^n filter with the additional pole zero correction is given by:

$$H(z) = \frac{C - F \cdot z^{-1}}{1 - F \cdot z^{-1}} \cdot \left(\frac{1 - d}{1 - d \cdot z^{-1}} \right)^n \qquad (4.52)$$

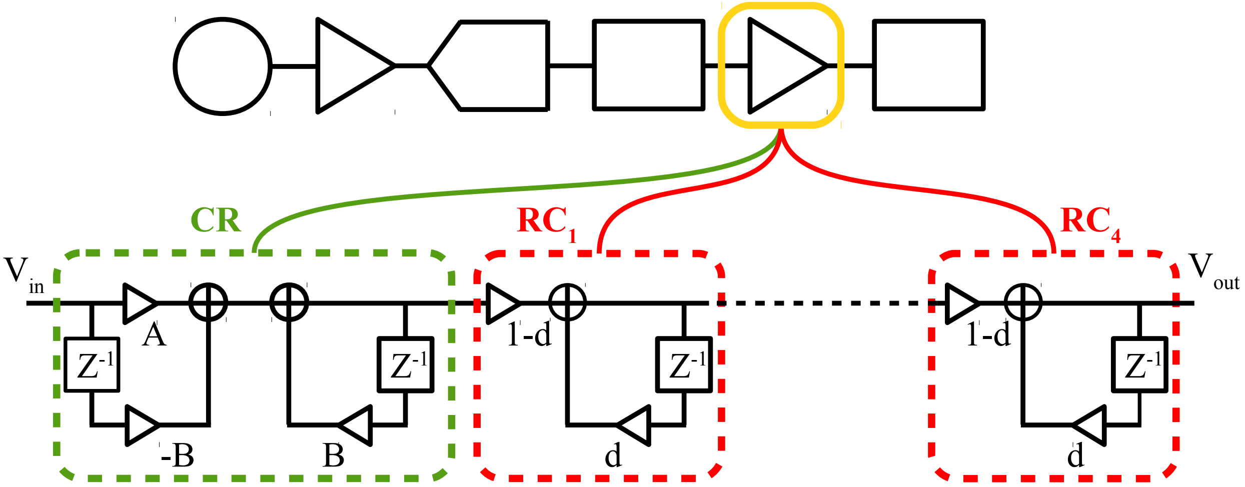

Figure 4.17: CR-RC^4 block diagram.

Figure 4.17 reports the block diagram representation of the filter described by Eq 4.52. It is evident how much this pole zero correction implementation is straightforward compared to other solutions as [105] where a division is required. Indeed, to realize a filter of any desired order, the three parameters *C*, *F* and *d* are enough. In terms of hardware requirements it means that the memory to allocate these values is smaller compared to a FIR filter and, ultimately, it reduces the power consumption due to the calculations.

4.7.4 SNR

Pulse shaping affects both the peak signal amplitude and the total noise of a signal at the output of the shaper. In other words, the shaping tries to accomodate two conflicting objectives at the same time. Because one target of the shaping is to process consecutive pulses often at high rates, it is common to face the pile-up effect described in Section 4.6. To avoid this inconvenience, the shaper reduces the signal width by an adequate length. On the other hand, to improve the SNR the shaper reduces the bandwidth which implies a broader pulse at the output. During the filter design these two opposite targets must be conciliated with an appropriate compromise, depending on the the application.

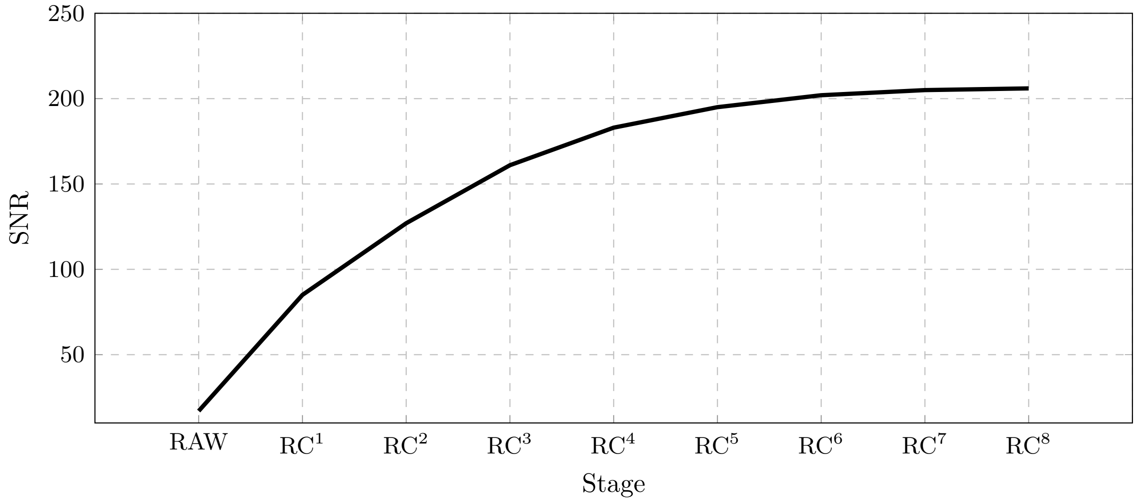

Figure 4.18: SNR as function of the RC^n stage.

In the present work a CR-RC^4 shaping filter was carried out. The motivation of this order, as anticipated, was due to a tradeoff between performance and hardware requirements. A preliminary implementation was realized in Python to investigate the filter performance as function of the number of stages. These results are reported in Figure 4.18 which was assumed as touchstone for the hardware implementation. The first term called *RAW* indicates the preamplifier signal without any processing and it was obtained collecting 1000 samples of peak pulses with the parameters itemized in Table 4.4 as well as the other points on the horizontal axis. The coefficient values adopted for the simulation are C = 0.52, F = 0.47 and d = 0.5 since they compensate the undershoot issue. The SNR was computed acquiring the peaks of the pulses at the same time reference and calculating their RMS value. Similarly, an equivalent number of points was collected on the baseline of the signal to evaluate its RMS noise. The ratio of these two quantities returned the SNR:

$$SNR = \frac{Signal}{Noise} = \frac{\sqrt{\frac{\sum_{i=0}^{N}(V_{i,peak})^2}{N}}}{\sqrt{\frac{\sum_{i=0}^{N}(V_{i,baseline})^2}{N}}} \tag{4.53}$$

Then, the SNR values were represented in Decibel usign the well know formula:

$$SNR_{dB} = 20 \cdot log_{10}(SNR) \tag{4.54}$$

In Table 4.5 are listed the values of the trend shown in Figure 4.18, where F_{SNR} represents the multiplicative factor of the SNR assuming F_{SNR} = 1 for the raw data. It is noticeable that the major improvements are located at the first RC stages, whereas a longer chain tends to progressively reduce the noise decreasing. Indeed, the F_{SNR} of RC^8 is not significantly different from the factor of RC^4 despite it demands a larger circuitry. This result is comparable with that one presents in the literature as reported in [104]. These considerations led to choose a compromise between the noise reduction level and the delay and complexity of the hardware. Thus, the final circuit was developed accounting 4 RC integrators. Due to the nature of the algorithm, a pipelined architecture was selected as more suitable design to implement the shaper. Therefore each stage is computationally autonomous during a clock cycle, requiring only the outcome of the previous stage and providing the input to the follow one. To further improve the energy resolution, the output of the RC^4 stage feeds the next Mobile Window Deconvolution filter (MWD).

Stage	SNR	SNR_{dB}	F_{SNR}
RAW	17.24	24.73	1
RC^1	85.95	38.68	4.99
RC^2	127.83	42.13	7.42
RC^3	161.80	44.18	9.39
RC^4	183.44	45.27	10.64
RC^5	195.92	45.84	11.37
RC^6	202.69	46.14	11.76
RC^7	205.91	46.27	11.95
RC^8	206.92	46.32	12.00

Table 4.5: SNR value for 8 RC integrator stages.

4.7.5 Fixed-point representation

As mentioned in Section 4.7.3, the recursive algorithm for the CR-RC^4 filter requires a set of programmable coefficients. This collection was carried out as a 32-bits fixed-point representation where the MSB is dedicated to the sign of the number, because of the signed multiplication. The second MSB represents the integer value, while the remaining bits form the decimal part as depicted in Figure 4.19. Therefore, in SIF notation [81] [82], these numbers are represented as $(1/1/30)$ format. As well as described in Section 3.8, the decision of adopting this format was due to some general considerations. Even in this case the dynamic range was the first element taken into account. The range of numbers that must be represented in the current application is limited once again by the 12-bits resolution of the ADC. This is a constraint also for the digital shaper output since it can not exceed this numerical domain, excepting the signed bit to avoid undesired overflows. Thus, the fixed-point representation was the most natural choice.

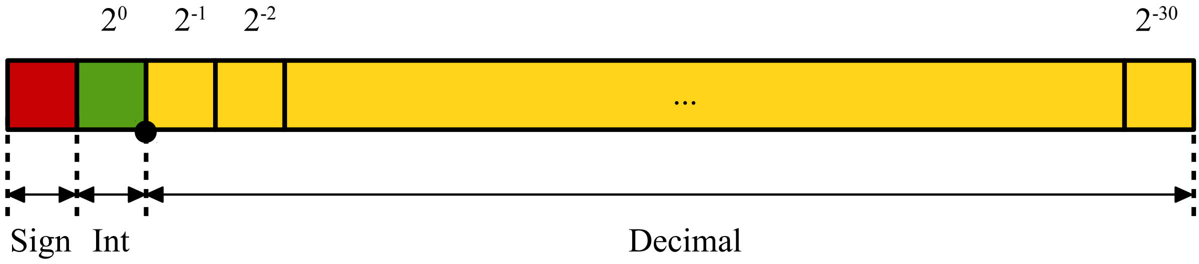

Figure 4.19: 32-bits fixed-point representation of a coefficient. The large black dot represents the location of the decimal point.

Moreover, the complexity of the circuit played a central decision role, keeping in mind the possible timing slack issues due to the floating-point implementation. As for the calibration processor, the time development was not an underestimated factor. Indeed, to speed up the design and the test step of all the algorithms presented in this work, convenient Python scripts were firstly implemented. Then, the core snippets of the algorithms were ported into RTL domain, simulated and verified. So, the typical DSP development flow [108] was partitioned between the double float precision evaluation on the Python environment and the direct fixed-point porting on the RTL side.

4.8 Deconvolution

In Section 4.2.0.3 the convolution process of two input signals was introduced. In some applications it may happen that the intent is to return back to unconvoluted inputs from a given signal. This method is named *deconvolution* and it can be exploited to obtain an accurate value of the pulse amplitude. To further increase the final SNR, a trapezoidal filter was realized with this approach and it was pipelined with the $CR - RC^4$ digital shaper discussed before. In the following the deconvolution process itself and the adopted strategy for the energy extraction are described.

4.8.1 Mobile Window Deconvolution

To acquire the energy information from the signal, a Mobile Window Deconvolution (MWD) was adopted. This algorithm have been widely explored as well as documented in many applications [109] [110] [111] [112] [113], implemented in recursive forms [114] [115] or bipolar versions [116]. The aim of this filter is to manipulate the preamplifier output to obtain a trapezoidal shape with a flat top. It is based on the deconvolution technique that is the reverse operation of the convolution as mentioned before. This processing is due to the undesired convolution between the distribution function of the charge characterizing the detector and the impulse response of the CSA, resulting in the preamplifier output. As presented in Eq 4.32, the charge collection causes a fast rising step of the signal with constant time τ_r followed by a longer exponential decay depending on the constant time τ_f. The decay is due to the feedback resistor R$_f$ depicted in Figure 4.13 and the continuous discharge reduces the peak height, namely, it affects the energy information. This phenomenon is named *ballistic deficit* [117]. The MWD removes this decrease restoring the full ideal amplitude of the signal [118]. An exaustive work on the MWD algorithm can be found in [40]. Therefore, let's assume the condition of a single exponential decay with $\tau_r \ll \tau_f$ and rewrite Eq 4.32 as follows:

$$f(t) = \begin{cases} \frac{Q_{in}}{C_f} e^{-\frac{t}{\tau_f}}, & \text{if t} \geq 0 \\ 0, & \text{if t} < 0 \end{cases} \tag{4.55}$$

where $A_0 = Q_{in}/C_f$ is the initial amplitude. If the value of f(t$_n$) is known at initial time t$_0$, it is possible to obtain A_0 as:

$$\begin{aligned} A_0 &= f(t_n) + A_0 - f(t_n) \\ &= f(t_n) + A_0\Big(1 - e^{-\frac{t}{\tau_f}}\Big) \\ &= f(t_n) + \frac{1}{\tau_f}\int_{t_0}^{t_n} f(t)\,dt \\ &= f(t_n) + \frac{1}{\tau_f}\int_{-\infty}^{t_n} f(t)\,dt \end{aligned} \tag{4.56}$$

where the last integral expansion to $-\infty$ is justified by the equation system 4.55 because the amplitude A$_0$ has value equal to zero for t $<$ 0. In the digital domain the last step of Eq 4.56 is given by:

$$A[n] = x[n] + \frac{1}{\tau_f}\sum_{i=-\infty}^{n-1} x[i] \tag{4.57}$$

Eq 4.57 is the discrete deconvolution that we were looking for. The shape of A[n] is an infinite staircase as result of the deconvolution operation. To bound this progression to a finite value and to get the desired step height, a differentiation is required:

$$MWD_N[n] = A[n] - A[n-N] \tag{4.58}$$

and substituting Eq 4.57 in Eq 4.58, it leads to the final form of the MWD:

$$MWD_N[n] = x[n] - x[n-N] + \frac{1}{\tau_f} \sum_{i=n-N}^{n-1} x[i] \tag{4.59}$$

with N equals to the window length of the filter. In Figure 4.20 are depicted MWD responses to the preamplifier output for different values of τ_f. In the case a) (green) the outcome of the deconvolution is adequately compensated and the top is flat, providing a correct extraction of the energy information. The other two cases b) and c) (red) report respectively an undercompensation and an overcompensation due to an erroneous setting of the constant τ_f.

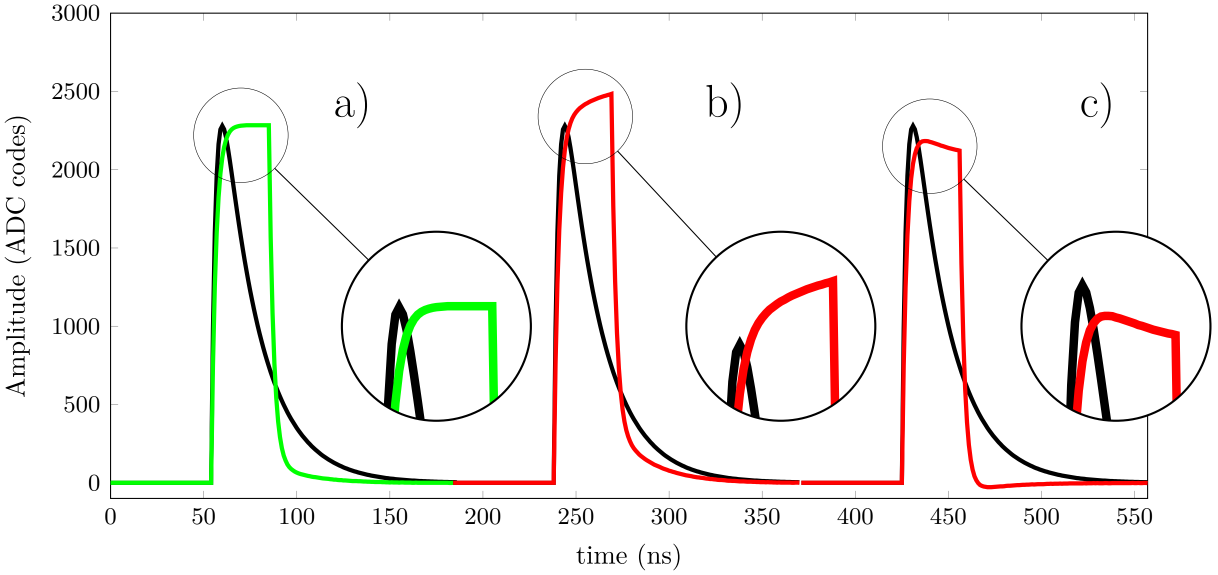

Figure 4.20: MWD filter responses to the preamplifier output for different values of τ_f: a) compensated, b) undercompensated, c) overcompensated.

The satisfactory response of the MWD to the ballistic effect is shown in Figure 4.21. Here three ideal input signals are reported characterized by different value of τ_r, while τ_f is maintained constant. The variation of the rising time has no effect on the flat top of the deconvolution filter which is fundamental to extract the energy information. To achieve an adequate flatness, it is important to select a suitable window N. The reader has to keep in mind that the deconvolution of the input signal is performed only within a fixed range defined by the window. Should this number be too small, the filter can not reach the plateau and a loss of information occurs. For this work, a τ_r range of 40 - 80 ns was considered and it was figured out that a deconvolution window N equals to 32 is suitable to obtain a flat top. This window was implemented adopting an accumulator to carry out the sum of Eq 4.59 and a circular buffer with dedicated pointers to accomplish the difference x[n] - x[n - N]. The trapezoidal outcome does not require any baseline restoration since it was already corrected in the CR-RC^4 chain. It is important to point out that the SNR is not improved after the deconvolution process and this fact demands a further elaboration of the output signal through an averaging. However, the digital shaping increases the SNR as demonstrated in Section 4.7.4. Thus, in this work the outcome of CR-RC^4 stage was conveyed to the input of the MWD instead of the raw signal. This approach has also been confirmed in the literature [119]. It seems to be very robust because it takes advantage of the ballistic correction provided by both the digital shaper and the MWD filter.

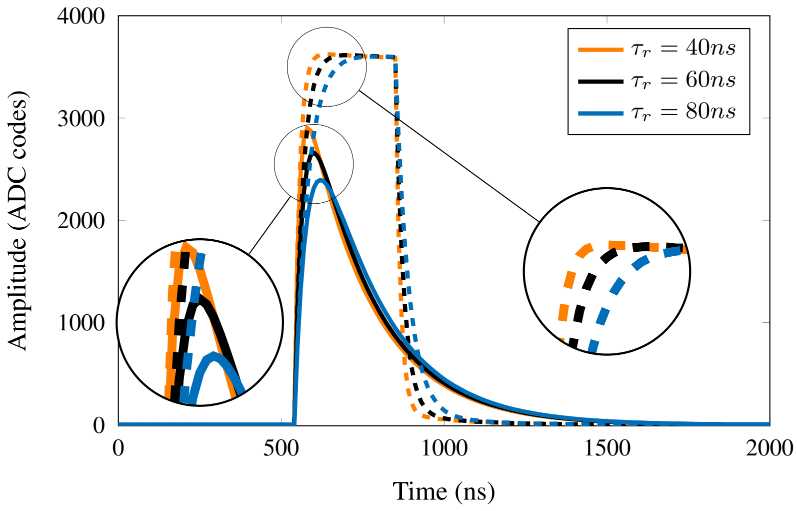

Figure 4.21: Ideal response of MWD filter to the ballistic deficit for different τ_r values.

4.8.1.1 Energy extraction

The last step of this processing is the effective energy extraction. At this point a natural question should arise: when is it more appropriate to acquire and store the data from the MWD output? This is not a trivial point since the magnified area on the right in Figure 4.21 highlighted how the flat top is not immediately reached. Thus, if an acquisition system collects the data too early, the final energy estimation will be affected by a systematic error and the derived value will be underestimated. In the current work a simple solution similar to the one described in Section 4.6.1 was proposed to prevent this undesired circumstance and to further increase the SNR. In particular a digital comparator was coupled with the first-order finite difference filter depicted in Figure 4.22 and this configuration allows to pick up the data outside the transition region. The following pseudo-code snippet reports the SystemVerilog implementation of this module, clarifying each step.

```
////////////////////////////////////////////////

module energy_filter (

  /* Here were declared the input/output interfaces
  of the module */

);

/* Instantiation of the internal registers */

always_comb begin
  if (reset)
    diff_thr = 13'b0;
  else
    /*Here is set the configurable threshold diff_thr.
    Its range can be positive or negative, with a maximum
    variation of [-128, +127] */
end

always_ff @(posedge clk) begin
  if (reset) begin
    /* All the registers are initializated to 0 */
  end
  else begin
    if (state == 5) begin

      /* Here were omitted some if-statements to ckeck
      the validity of the samples */

      MWD_previous <= MWD_current;
      difference <= MWD_previous - MWD_current;

      if (MWD_current > MWD_thr && ~energy_en) begin
        if (difference < diff_thr) begin
          energy_counter <= 2'b0;
          energy_acc <= 18'b0;
```

```
38           energy_module <= 13'b0;
39         end else if (difference >= diff_thr) begin
40           energy_counter <= energy_counter + 1'b1;
41           energy_acc <= MWD_current;
42           e_flag <= 1'b1;
43           energy_en <= 1'b1;
44         end
45       end else if (MWD_current <= MWD_thr && ~e_flag) begin
46         energy_counter <= 2'b0;
47         energy_acc <= 18'b0;
48         energy_output <= 13'b0;
49         energy_en <= 1'b0;
50       end
51 
52       if (e_flag) begin
53         energy_counter <= energy_counter + 1'b1;
54           if (energy_counter == 2'b11) begin
55             energy_output <= (energy_acc + MWD_current) >>> 2;
56             e_flag <= ~e_flag;
57           end else begin
58             energy_output <= 13'b0;
59             energy_acc <= energy_acc + MWD_current;
60           end
61     end
62   end
63 end
64 
65 endmodule: energy_filter
66 
67 ////////////////////////////////////////////////////
```

The lines between 2 and 11 are dedicated to the declaration of the input and output port interfaces of this module as well as to instantiate the internal registers of the block. The combinatorial description of lines 12-19 initializes a configurable threshold named $diff\_thr$. Its utility will be explained later, however it is bounded in the range [-128, +127] since only 7 bits are programmable. The sequential code that describes the core of the extraction method starts from line 21. After the reset, when the FSM is entered into the suitable state and some conditions are checked with if-else statements, the system records the previous output of the MWD filter into a dedicated register (line 31). At the same posedge of the clock, the difference between the previous value and the current one is computed (line 32), namely, another backward difference [100] with changed sign was realized. This quantity is the gradient of the trapezoidal filter. It is well know that a flat trend of the gradient indicates when the curve has the first derivative equals to zero, which is the region we are interested in. Therefore, the comparator at line 34 matches the trapezoidal outcome with a configurable threshold called MWD_thr to eventually reject some specific data range, holding only the selected one. Although it was not explicitly reported in the code, this is a signed operation. If it is above the desired level, the result of the differentiator is constantly checked. Another if-else statement compared the $difference$ value with the configurable threshold $diff\_thr$. If the condition $difference \geqslant diff\_thr$ occurs (line 39), a counter is incremented by 1 (line 40), the accumulator $energy\_acc$ is re-initialized to the current value of the MWD output (line 41), a flag is set to 1 (line 42) and the enable variable $energy\_en$ changes to 1 (line 43) in order to disable all the block at line 34. Now, only the sub-block at line 52 is active. Here $energy\_counter$ is further incremented at each new validated data, while the accumulator iteratively collects them (line 59). When the counter is equal to four (line 54), the output is calculated as the signed shift of the accumulator content (line 55) and e_flag is set back to 0. This step is highlighted in Figure 4.22 with the vertical dashed lines that define a sub-window on the plateau where these samples are included. Thus, in this way one condition of the else statement at line 45 is enabled again and at some point, depending on the value of $MWD\_current$, the full conditions will be satisfied. As a consequence, the counter, the accumulator and the enable will be reset again. The latter enables the second condition at line 34, thus another computation cycle is possible. Although the backward difference with just two points is the rough discrete

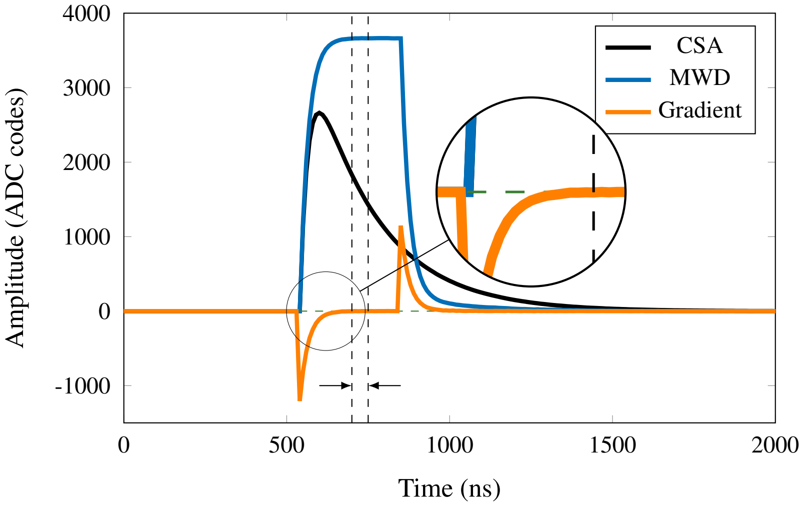

Figure 4.22: Approximation of the first-order derivative of the MWD output.

approximation of the first-order derivative, it was tested that it is enough accurate to track narrowly the trend of the trapezoidal shape. Moreover, the main advantage of this strategy is the processing of the energy information from a signal of variable amplitude without a specific threshold on the trapezoidal output. Lastly, the whole energy extraction system composed by the MWD and the described differentiator circuitry allows to improve the SNR of the outcome signal and it will be demonstrated in Section 5.2.3.

4.8.2 Charge extraction

A complementary evaluation about the charge deposited in the detector is provided by a dedicated module beside the MWD filter [120]. The circuitry dedicated to this task is particularly elementary. It is based on a digital comparator that confronts the CR-RC^4 output stage with the same threshold configured for the trapezoidal filter. If the digital shaper values are higher than the selected level, the CR-RC^4 outcomes are accumulated into a memory. This register is properly dimensioned to prevent the overflow during the sum operation. When the shaper output returns below the threshold, the sum process is terminated and the result is sent out while the accumulator is cleaned. In this case it was not necessary to implement a double threshold scheme to face the problem depicted in Figure 4.6 and described in Section 4.4 because the CR-RC^4 shape is smoothed enough and no signal bounce occurs.

4.9 Time pick-off

In many applications an important role is played by the definition of the time occurrence of a pulse with respect to a time reference [121] [122] [123]. When a signal is detected a time-tag is associated with it. The logic pulse generated by this time pick-off circuit should be in principle independent on the shape and the amplitude of the signal. In a real application these conditions are difficult to match since each acquisition is affected by different source of errors. These kind of inaccuracy defines two classes [128] [9]. When the amplitude of the input pulse is constant it is named *time jitter*. More precisely, it is classified in two categories: if the frequency variation is less than 10 Hz it is termed *wander*, if the fluctuation is above 10 Hz is properly defined as *jitter*. The other is referred to as *amplitude walk* in the case of a variable amplitude. In the following sections common approaches for time measurements and their limitations will be discussed, pointing out the strategies to overcome some of them.

4.9.1 Jitter

In a digital circuit, the jitter is the time deviation from the nominal value that affects both the clock and the signals in many ways [124] [143]. Thus, it represents a benchmark of their quality. Sources of this uncertainty could be thermal noise, cross talk, PLL oscillators and power supply rippling and it has different impact on the performance of the devices. The error contribution due to the jitter becomes more relevant with the increasing of the clock frequency.

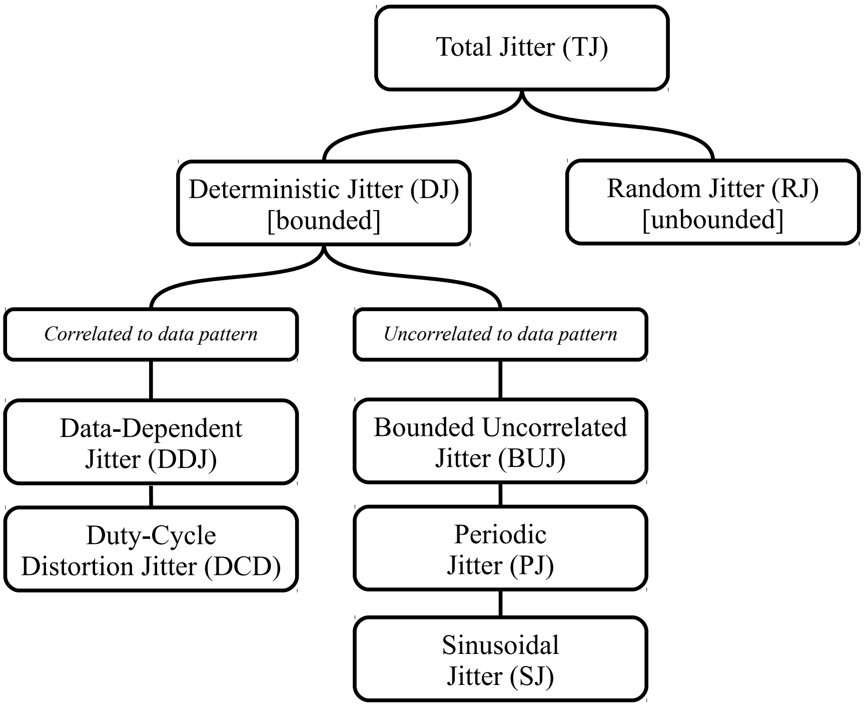
Total Jitter (TJ)
Deterministic Jitter (DJ) [bounded]
Random Jitter (RJ) [unbounded]
Correlated to data pattern
Uncorrelated to data pattern
Data-Dependent Jitter (DDJ)
Duty-Cycle Distortion Jitter (DCD)
Bounded Uncorrelated Jitter (BUJ)
Periodic Jitter (PJ)
Sinusoidal Jitter (SJ)

Figure 4.23: Jitter classification based on its characteristics.

Figure 4.23 shows an overview about the jitter classification. When the clock period is considered, the jitter is called *Period Jitter* and it is characterized by a Gaussian distribution due to the scattering of electrons that generate a thermal noise at temperatures larger than zero degrees Kelvin. This quantity provides the time window available for the setup and hold time accomodation within a clock period, hence it is critical for digital circuits timing [127]. The *Absolute Jitter* shares the same random nature of the Period Jitter except that is referred to the rising edge of the clock rather than its period. This uncertainty arises taking into account the non-zero time required for a clock transition and its not constant value for subsequent rising edges. These slightly different effects of the clock jitter must be added to another term called the *Cycle-to-Cycle Jitter*. It defines the time deviation between two suc-

cessive periods. Due to the comparison of two consecutive periods, this component expresses the highest frequency contribution of the clock jitter.
By examining the jitter which affects the data, Figure 4.23 illustrates the main division that takes part in the definition of the *Total Jitter* (TJ) of a signal. The discriminator resides in the bounding of its components. The *Random Jitter* (RJ) is quite self-explaining, however its nature is not correlated to the data pattern or other type of signals and it follows an unbounded Gaussian distribution. The rms value, equal to the standard deviation of the distribution, defines this source of jitter. By contrast, the *Deterministic Jitter* (DJ) has a bounded histogram representation. Its nature is systematic and it hosts two sub-classes categorized by their correlation with the data pattern [125]. The *Data-Dependent Jitter* (DDJ) occurs when the edge of the signal is affected by early or late deviation with respect to the nominal values [129]. The *Duty-Cycle Distorsion Jitter* (DCD) instead appears when there is a dependency of the edge timing on only a single bit. This distorsion arises in case of nonlinearities or asymmetric rise and fall times.
The uncorrelated to data pattern class collects all the terms coming from the other possible jitter source. The main container of this category is the *Bounded Uncorrelated Jitter* (BUJ) caused, for example, by the crosstalk among near signal lines or the noise introduced from the power and ground lines. The *Periodic Jitter* (PJ) belongs to a sub-division of the BUJ and may be generated by the switching power supply of the circuit. In turn, the *Sinusoidal Jitter* (SJ) is a subcategory of PJ and it is intentionally generated for compliance testing purpose.

In order to quantitatively evaluate the jitter deviation an amplifier output must be considered. In Figure 4.24 a collection of these kind of signals is depicted. The different spreading of the jitter among the rising and the trailing edge in Figure 4.24 is mathematically explained recalling the output of an amplifier around its rising edge point [5] [130]:

$$V_{out}(t) = V_{out}(t_0) + \left.\frac{dV}{dt}\right|_{t=t_0} (t - t_0) \tag{4.60}$$

Eq 4.60 represents the first order Taylor expansion of the amplifier output, where t_0 indicates the nominal crossing point of the signal when it reaches the threshold (THR). In this description, $\left(\frac{dV}{dt}\right)\Big|_{t=t_0}$ is the slope of the signal. If any voltage variation (ΔV) occurs when the signal crosses the threshold, a corresponding time variation (Δt) affects the signal achieving that point. Thus, the voltage uncertainty can be expressed as:

$$\Delta V_{out} = \left.\frac{dV}{dt}\right|_{t=t_0} \Delta_t \tag{4.61}$$

In terms of noise standard deviation σ_t, it is possible to write:

$$\sigma_V = \left.\frac{dV}{dt}\right|_{t=t_0} \sigma_t \tag{4.62}$$

and finally the jitter can be defined:

$$\sigma_t = \frac{\sigma_V}{\left.\frac{dV}{dt}\right|_{t=t_0}} \tag{4.63}$$

Eq 4.63 justifies in a mathematical way what we graphically appreciated in Figure 4.24: where the slope of the signal is higher, the value of the jitter is smaller. This last point suggest also the complementary approach in time measurements compared to the extraction of energy information from a signal.

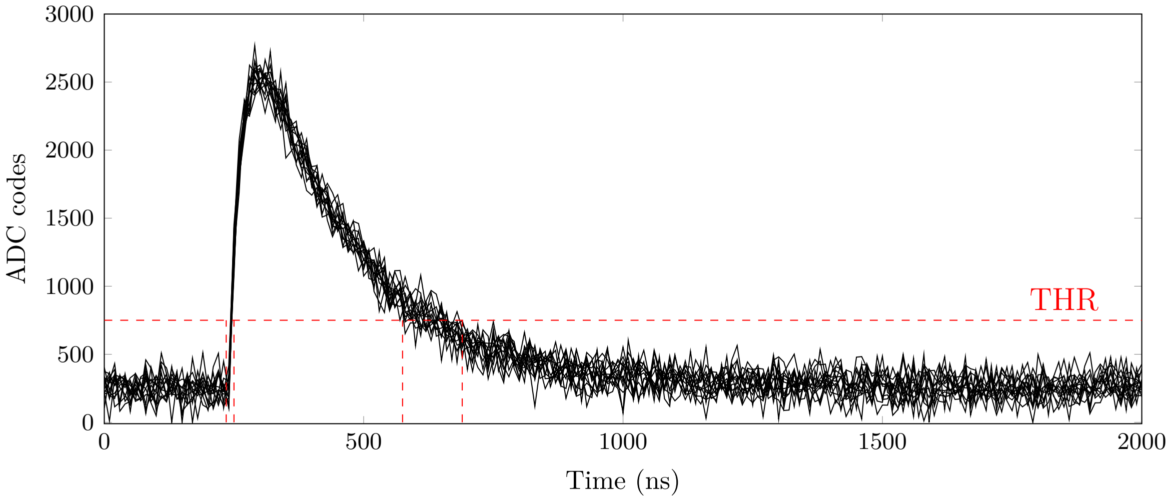

Figure 4.24: Jitter spreading. The different jitter distributions between the rising and trailing edges are noticeable.

Indeed in the first case the slope-to-noise-ratio is the quantity to minimize to ensure a good timing performance [9]. On the other hand, it is desirable to maximize the signal-to-noise-ratio, namely, minimize the amplitude jitter σ_V, in the energy measurements of a signal. Also other sources of uncertainties contribute to the time resolution such as the geometry of the sensor and differences in energy deposits that cause shape variations of the signal. However, these evaluations would involve other type of simulations that are out of scope of this thesis.

Therefore a typical time resolution is the combination of time uncertainties due to both the detector and the electronics [131] and it strongly depends on the application [132], spanning a relatively narrow range. As a reference, we can recall the ALICE introduced in Section 1.3.1. One of the detectors of ALICE is named Time of Flight (TOF) and it is based on a Multigap Resistive Plate Chambers (MRPC) technology with an overall time resolution around 60 ps. Another example is represented by the Silicon PhotoMultipliers that are characterized by a time resolution in the range 80 - 90 ps. However, the future colliders will implement a 4-D tracker where precise time information will be demanded. Indeed, considering the beamspot time spread of the LHC which is in the range of 150 - 180 ps, a time resolution of few tens of ps per layer will be a strict requirement. Nowadays, the current clock jitter for high energy physics experiment is between 10 and 20 ps RMS for a detector [126] and the generic time resolution for both radiation detector and electronics is in the window $20-120$ ps [133]. Considering this general range, in this work the time deviation was modelled taking into account two contributions for the CSA output model adopted in Section 4.7.1. As mentioned earlier, the jitter defines the time deviation of a signal from its nominal reference and it is composed by two principal terms: a random component RJ and a deterministic part DJ. The evaluation of both terms was carried out in a Python3 script, generating a jitter signal with a realistic 12-bits SAR ADC including capacitor mismatches as described in Section 2.4.4. The RJ was simulated using the Gaussian random routine of Python3 [140]. Because the RJ is unbounded, its correspondent peak-to-peak value is dependent on the number of samples used [135]. Keeping in mind this, a noise was added to the original pulse shape, thus by setting the threshold as low as possible, the average value of the slope and noise σ_V were computed. For a SNR in the range 10-20, the deviation in term of ADC codes for this isolate gaussian contribution was $\sim 1225 \pm 52$. Then, using Eq 4.63 a peak-to-peak jitter value of 47.8 ps was obtained, leading to a $\sim$ 7 ps rms jitter [134]. This value is equal to the standard deviation of a normal distribution centered on the time zero point. The number of samples collected was 1000 to get a not unreasonable optimistic evaluation. Indeed, during a simulation campaign, a slight variation based on the number of simulated samples was appreciated. As for the deterministic part, an uncorrelated to data pattern component was separately considered to simulate a power supply noise [141] [144]. In particular, the sum of sinusoidal waves PJ_{tot} was adopted to stimulate the signal input [139] [142]:

$$PJ_{tot}(t) = \sum_{j=0}^{N} A_j sin(\omega_j t + \phi_j) \tag{4.64}$$

where N expresses the number of sinusoidal components, A_j is the amplitude of each tone, ω_j corresponds to the angular frequency, t is the time and ϕ_j represents the jitter phase expressed in radians [136]. For the run simulation, the frequency domain was bounded up to 1 kHz and the amplitude was limited up to 5 LSB. For each sinewave the frequency, amplitude and phase were picked up randomly within these ranges. The total jitter J_{tot} is simply the sum of RJ and PJ parts [137]:

$$J_{tot} = J_{RJ} + J_{PJ} \tag{4.65}$$

Jitter	Δ_{ADC}	$\Delta_{Jitter,p-p}$ (ps)
Gaussian	1225 ±52	±47.8
Sinusoidal	1226 ±50	±46.4
Total	1319 ±104	±97.3

Table 4.6: Jitter results for 1000 samples.

As well as RJ, the results were collected in Table 4.6 where the overall resolution is also reported. This modeling is qualitatively shown in Figure 4.25. The two Gaussian distributions are deliberately enlarged with arbitrary units due to the scales. The final resolution for the simulated CSA output is quite comparable to the magnitude order of real implementations [131] [138]. However, the reader must remember the approximate nature of this simulation, without any study about the detector performance or variations in energy deposits.

4.9.2 Leading edge

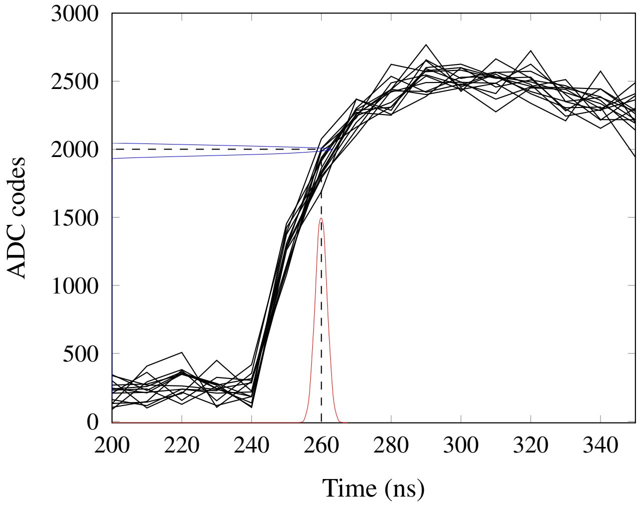

Figure 4.25: Example of jitter distribution and ADC codes spreading.

As mentioned in Section 4.9, a timing discriminator circuit must be able to recognize at least the arrival time of an event with an adequate resolution. This basic function have to be independent from the final application whetever it is time spectroscopy or time coincidence of several detectors [145]. Furthermore it has been shown that jitter is one of the primary uncertainties faced by electronic circuits. During the years, many solutions have been proposed to address this task [146]. The simplest discriminator is based on the leading-edge timing approach. The circuit monitors continuously if the input signal crosses a threshold and it generates a pulse when this event occurs. Despite its simplicity, the method is sensible to the amplitude of the signal. This effect is illustrated in Figure 4.26. In this case, two pulses characterized with the same rising timing and shape cross the threshold at different times, in particular when their amplitudes are different. The time difference among the generated pulse in the plot below, highlighted with the face-to-face arrows, is named *time walk*. This undesired drawback can break down the time resolution of the discriminator when pulses with not constant amplitudes are processed. The same effect is also shown in the case of input signals with constant amplitudes, but different rising times [147] as depicted in Figure 4.27.

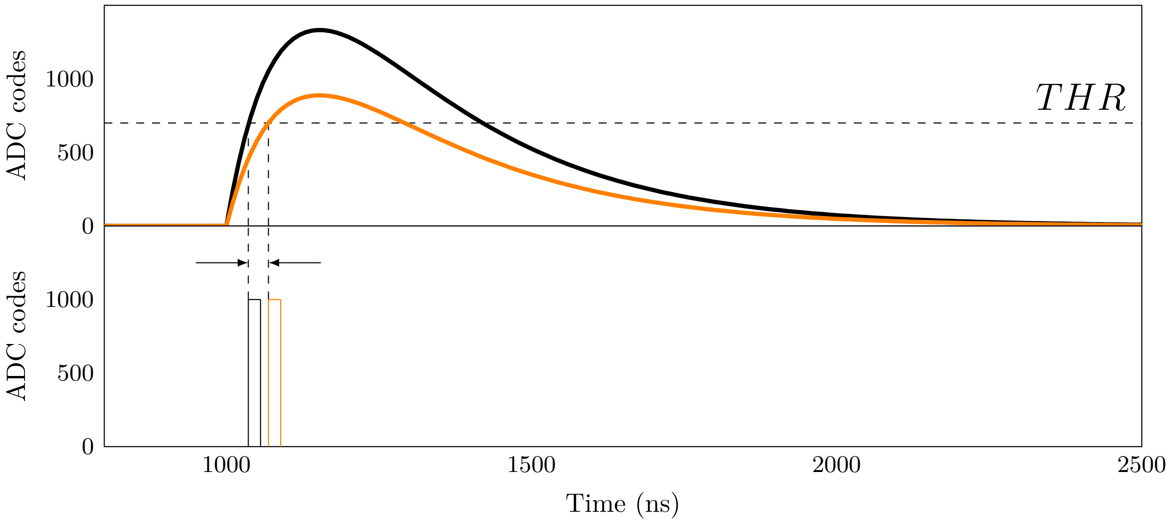

Figure 4.26: Time walk: different amplitude, constant rising time

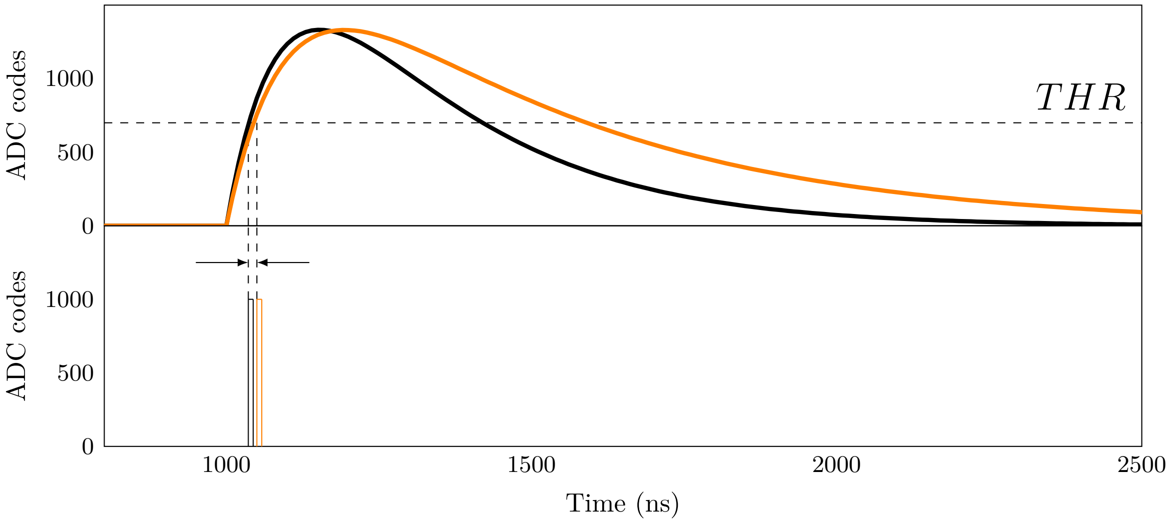

Figure 4.27: Time walk: constant amplitude, different rising time

Moreover, if the discriminator is not charge-balanced, a further degradation of the time walk is introduced by the *charge sensitivity* [148]. This effect adds a delay in the output pulse generation and it is due to the additional charge required by the switching circuits to change their state. Furthermore, the charge sensitivity also changes the nominal threshold by introducing an error much greater for the closer input signals [149].
To overcome this side effect many solutions had been proposed [150] [151] [152]. However, the golden reference for timing measurements has been attributed to a totally different approach and it will be explained in a later section.

4.9.3 Zero crossing discriminator

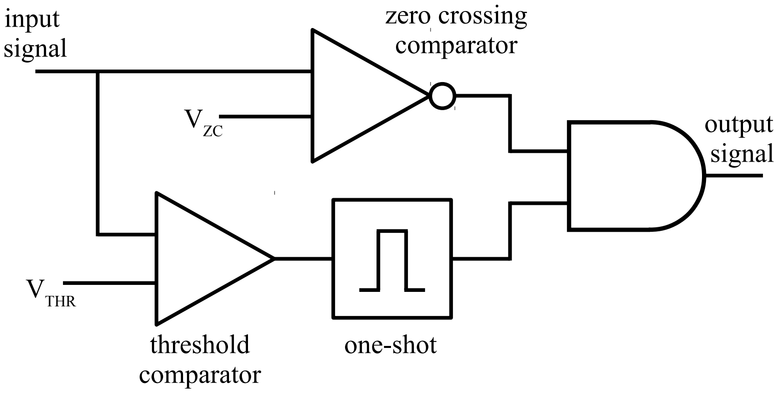

Figure 4.28: Zero crossing discriminator scheme

The time walk effect can be mitigated adopting a zero crossing discriminator as reported in Figure 4.28. This circuit introduces an additional comparator confronted to the single version of the leading edge method [9]. The function of this element is to continuously trigger on noise, therefore it monitors when the signal crosses the zero level. In order to execute this task, its voltage reference have to be set to zero. The threshold comparator instead generates an output when the leading edge of the input signal crosses a selected threshold. In contrast to the zero crossing comparator, its threshold is adequately set high to avoid firing on noise. The following one-shot is used to extend the comparator outcome, ensuring enough overlap with the zero crossing comparator. At the end of the chain, the AND gate returns the timing output signal. The time resolution of the system is limited by the zero crossing comparator. Although this solution reduces the time walk, if a bipolar signal is used a degradation of the slope occurs because the signal has a larger time jitter compared to the unipolar counterpart.

4.9.4 Constant fraction discriminator

A popular and performing choice for timing pick-off purposes is the Constant Fraction Discriminator (CFD) which finds wide applications [153] [154]. This method compensates both the amplitude variation and the rising time of the input signal. The original analog version of this circuit [155] was carried out into the digital domain with some changes and it is properly referred to as dCFD. The concept underlying this filter is shown in Figure 4.29.

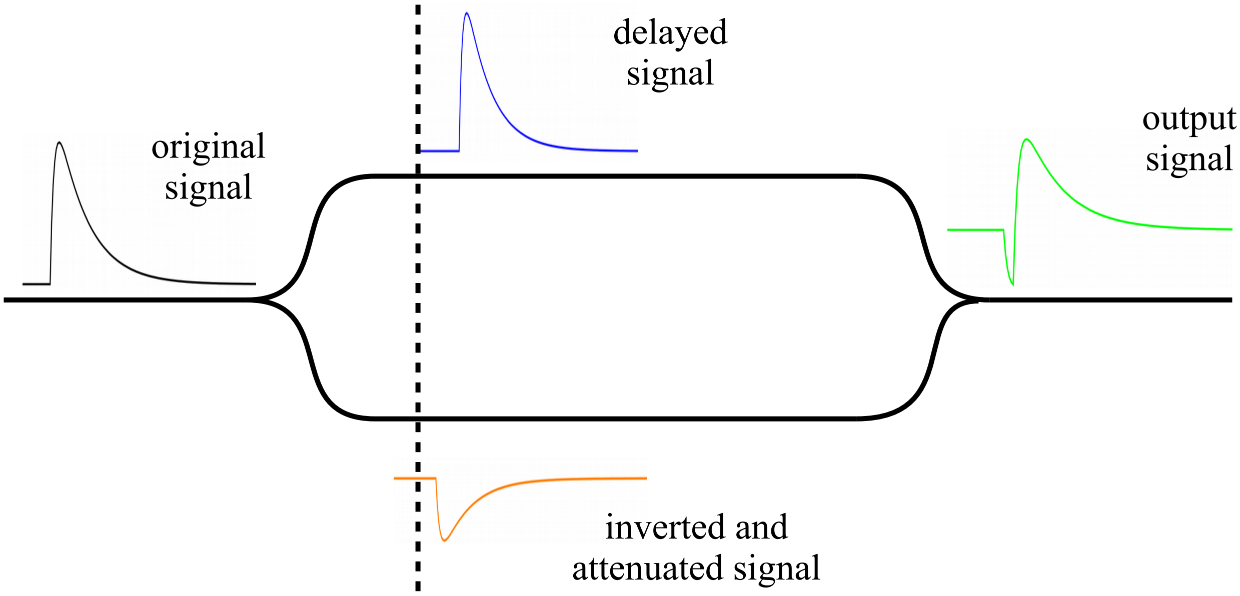

Figure 4.29: CFD algorithm

This specific version follows a comparative approach based on the splitting of the original signal into two branches [156].

$$CFD_{out}[t] = S[t - D] - S[t] \cdot F \tag{4.66}$$

One branch is simply retarded by a programmable delay D, while the other is inverted and attenuated by a factor F. The sum of these two terms leads to a bipolar output signal illustrated in detail in Figure 4.30 and reported in Eq 4.66. Due to its nature, the generated signal crosses the zero at a certain time (dashed line in the picture) and that point coincides with the timing information that it was looking for. The reader can observe that the zero cross spot occurs when the maximum amplitude of the inverted signal (Δ_B) is equal to the the same fraction of the input signal on the delayed shape (Δ_A) [121]. Depending on the application and scope, this fraction is in the range 10% to 70% of the pulse amplitude [157] [158] [159] [160]. Accordingly, this technique allows to achieve a result independent of the pulse amplitude.

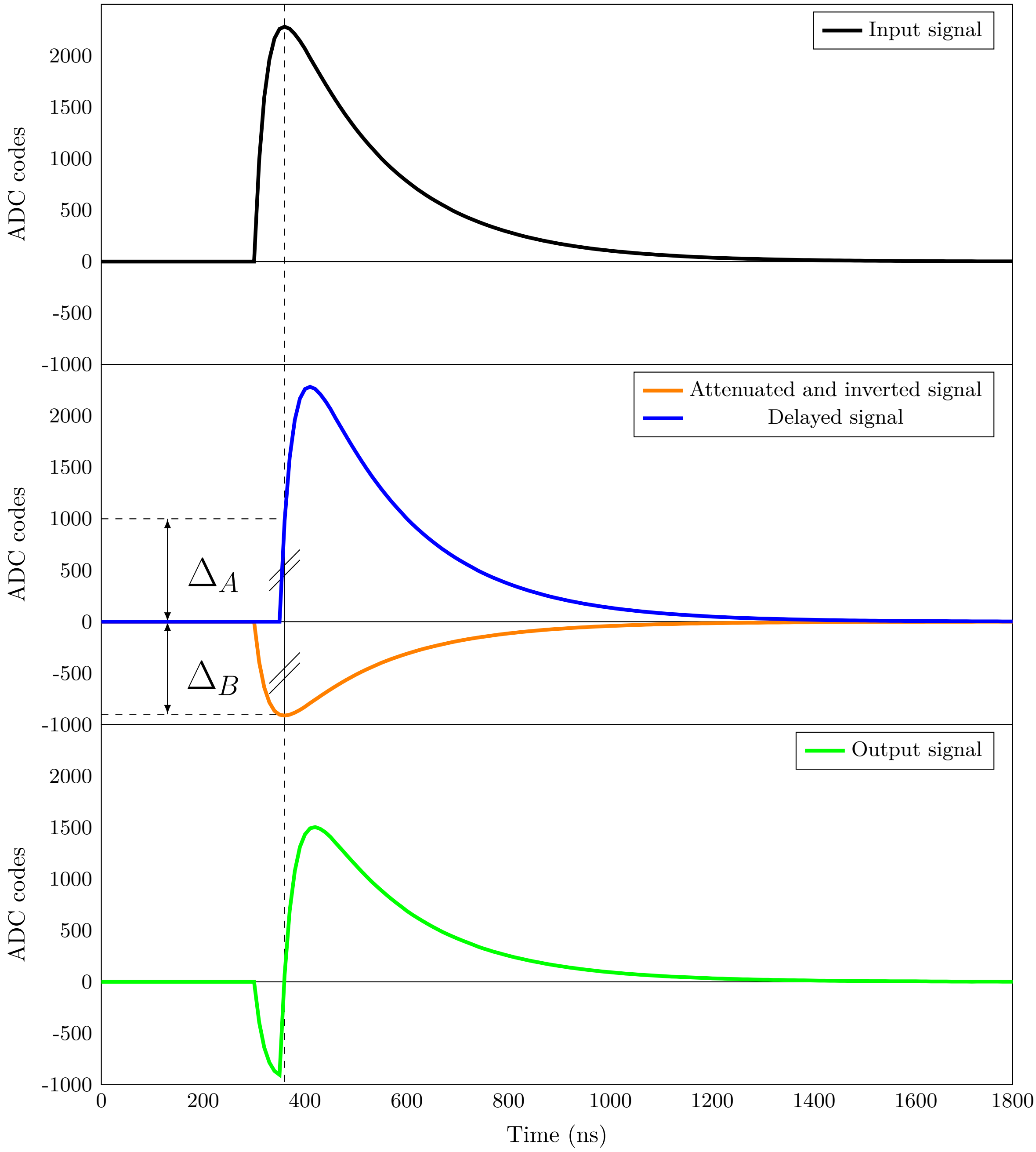

Figure 4.30: The constant fraction method (ideal case).

To compute the zero crossing point it is necessary to store at least two points, the less negative and closer to zero point and its positive complementary above the zero level. Once collected, an interpolation leads to the result. Although the cubic interpolation returns the best result in some circumnstances [161], a linear interpolation could represent a good solution, especially if an intensive-computation resource is not available for a on-chip processing. This approach assumes that the region around the transition point can be linearly approximated. However, it is very impor-

tant to take care of the rising edge of the signal. Indeed, the dependence of the time resolution from the rising time [120] has been demonstrated. In particular, the observations coming from simulations and tests discriminated two regions. If the rise time is larger than 5 times the value of the sampling period, then the error associated with the computation of the zero crossing is mostly due to the quantization error of the converter. On the other hand, when this value is smaller, the linear approximation of the curve around the zero crossing point could not be considered valid anymore.
The quality of the result in terms of time resolution is affected also by the choice of the parameters D and F as indicated in Figure 4.31. This plot was obtained by another high level modeling of the CFD carried out in Python3. The picture reports the trend of the time resolution, named *Sigma*, as function of both the fraction F, expressed as percentual of the input signal amplitude, and delay D. This preliminary analysis takes into account only a first set of all the possible delays because the changing in the slope between $F = 0.4$ and $F = 0.5$ points out a common increasing of the sigma value, hence no more delays have to be studied. The magnified area denotes that for the simulated signal, characterized with the noise level and jitter distribution examined respectively in Section Chapter 3 and Section 4.9.1, the best choice is $D = 3$ and $F = 0.45$. Thus, these values were set for the simulation post P&R.

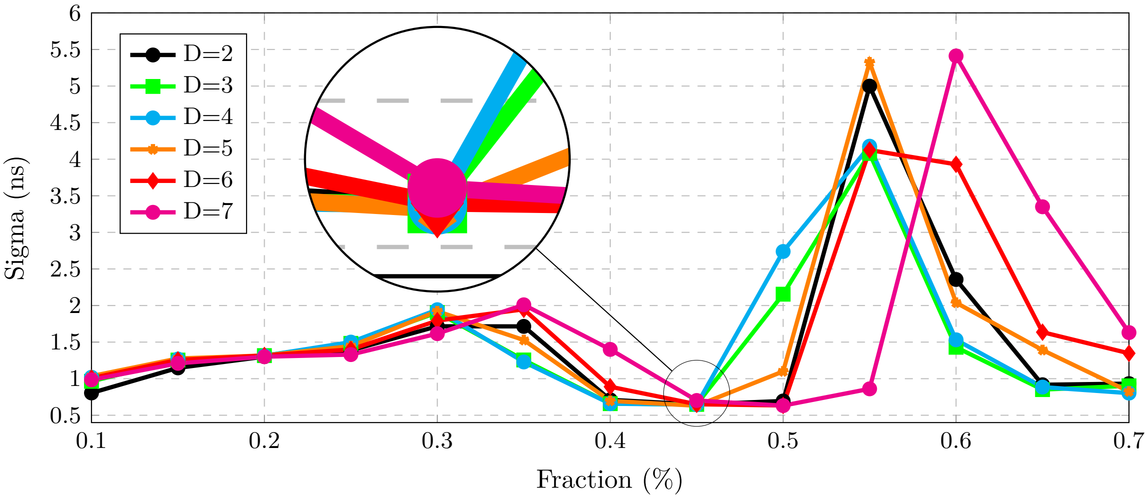

Figure 4.31: Sigma distribution as function of delay D and fraction F.

The physical implementation of the CFD algorithm was particularly straightforward. Indeed, this block shares the same circular buffer used to store the data processed by the CR-RC^4 module. In addition just a further pointer was required to pick-up the delayed value selected through the parameter D. Since the content of the circular buffer was the raw data, a baseline restoration was demanded. For the fractional term of Eq 4.66 it was fundamental to include the baseline subtraction within the multiplication operation. This action prevented a unwanted vertical shift of the CFD output and a consequent uncorrected computation of the zero crossing point. In the next section it will be explained how to compute this point.

4.9.5 Least square minimization and interpolation

Once obtained an analyzable signal, whatever it is the raw data or the CFD outcome, many techniques exist to compute the zero point. Possible solutions are optimal filtering [162], deconvolution [163], LUT [164], least square minimization [165] and interpolation [166]. In this section the last two methods will be shortly explored.

In case of multiple points acquisition, the least square minimization is a suitable way to calculate the zero crossing transition. A non-recursive solution was initially explored in this work for the computation of the zero point on the leading edge. As mentioned in Section 4.9.4, it is assumed that the points around the zero level transition can be approximated with a linear function as depicted in Eq 4.67:

$$y = A + Bx \tag{4.67}$$

where A is the intercept and B is the slope of the line. From the theory [167] it is known that A and B can be derived from a i-th set of points and expressed as:

$$A = \frac{\sum_i^N x^2 \sum_i^N y - \sum_i^N x \sum_i^N xy}{\Delta} \tag{4.68}$$

$$B = \frac{N \sum_i^N xy - \sum_i^N x \sum_i^N y}{\Delta} \tag{4.69}$$

$$\Delta = N \sum_i^N x^2 - \left(\sum_i^N x \right)^2 \tag{4.70}$$

where N is the number of samples, x is the time tag and y is the data. At first glance, all these quantities change in time because of the continuous acquisition process. However, choosing an appropriate reference system the equations 4.68, 4.69, 4.70 can be remarkably simplified. For instance, let's consider to collect the first four data on the leading edge at a certain clock frequency. A pulse signal can be generated when the input signal is above the desired threshold, in order to get the

timestamp value of a timing counter to point out when the transition occurs and the collection starts. At the same time, that point can be *locally* considered the origin of the acquisition, thus with a time x = 0. If the sampling period is equal to T$_{clk}$, then the second point will be located at x = 1 T$_{clk}$, the third one at x = 2 T$_{clk}$ and so on. This consideration simplifies the Δ expression since only fixed terms are involved now in Eq 4.70. These values can be calculated once during the reset state, saving timing and power consumption. The benefit also regards the division operation and this is due to the possibility to compute and freeze the inverse value $(1/\Delta)$ just one time, in order to multiply it with the numerators later. Naturally the changing quantity y as well as the xy mixed term are immune to this simplification.
This approach was initially tested in 110 nm CMOS technology. The simulations post P&R were in accordance with the Python results, proving the good response of the method. However, the limitations explained in Section 4.9.2 led to discard this technique to adopt a CFD algorithm and a lighter computational method. In particular, a linear interpolation replaced the calculation on leading edge [168]. The new system considers only two points: the negative point closer to the zero level and a following positive data selectable with a threshold. Also in this case, a linear approximation was kept valid for this study. The relation between these two points is expressed by the Eq 4.71:

$$y = \frac{y_p - y_n}{x_p - x_n}(x - x_n) + y_n \tag{4.71}$$

where the couples (x$_n$, y$_n$) and (x$_p$, y$_p$) are the points of interest and the pedix p (n) indicates the positive (negative) data. Even in Eq 4.71 a simplification can be operated. Indeed, the denominator is the time distance between the negative and the positive value. If we set the time origin of our local system to the negative term another once again, the denominator is equal to the sampling period T$_{clk}$ or its multiplicative factor and this quantity will never change anymore. Now, to compute the slope m of the line it is sufficient to perform the subtraction at the numerator among the y variables, then divide for the T$_{clk}$. The formula can be rewritten as:

$$y = m(x - x_n) + y_n \tag{4.72}$$

We are interested in value of x when the y is equal to zero. Thus, setting both y and x_n to zero (recalling that the negative number is our local time origin) and rearranging the equation:

$$x = -\frac{y_n}{m} \tag{4.73}$$

Therefore the slope allows to extract the time occurence of the point at the zero level threshold. The computational cost of the entire process is just one subtraction and two divisions. Under the given approximation, the results coming from the interpolation are fully compatible with the least square minimization returns, but they involve a drastically smaller number of operations. The details about the division implementation are discussed in the next section.

4.9.6 Goldschmit's algorithm

In the DSP field there are many digital algorithms to carry out a division on a chip [169]. They can be classified evaluating different characteristics such as the mathematical approach, the convergence speed and the hardware requirements. One popular class is the iterative one. It contains two types of division algorithms, based on their iterative operators and ultimately on their convergence speed. The first group is defined by the subtractor operator and it includes well know methods such as the nonrestoring technique. These kind of algorithms are not particularly fast because the time to find the solution is proportional to the divisor length [170]. The multiplication is the iterative operator of the second ensemble. In this case the convergence rate is quadratic with the time required to execute the operations proportional to $\log_2$ of the divisor length.

The implementation of a division algorithm is sensitive both to the hardware resources required and the latency related to a cycle. It was demonstrated that a large latency causes a performance degradation as well as a too low divider latency due to the exponential increase of the area demand or the timing execution [171]. In order to efficiently implement the required division for the zero point computation, the Goldschmit's algorithm [172] was selected. This method belongs to the division-by-convergence algorithms class and it iteratively computes the quotient Q by multiplying both the numerator N and the denominator D by the same term F$_i$ as illustrated in Eq 4.74.

$$Q = \frac{N \prod_{i=0}^{n} F_i}{D \prod_{i=0}^{n} F_i} \tag{4.74}$$

The factor F_i is initially computed as the inverse of the divisor, resulting in an approximation of the reciprocal D:

$$F_0 = \frac{1}{D} \tag{4.75}$$

Then, the first iteration results in two parallel multiplications:

$$\begin{aligned} N_0 &= N \times F_0 \\ D_0 &= D \times F_0 \end{aligned} \tag{4.76}$$

The following step updates all the three quantities:

$$\begin{aligned} F_1 &= 2 - D_0 \\ N_1 &= N_0 \times F_1 \\ D_1 &= D_0 \times F_1 \end{aligned} \tag{4.77}$$

Through this loop, the denominator converges to 1 as well as the numerator converges to the quotient in parallel [173] [174].

In the current implementation, N, D and F are defined as 33-bits fixed-point words. In SIF notation they are expressed in (1/12/20) format with the MSB used for the sign, the following 12 bits reserved for the integer part and the other 20 bits for the fractional term. Figure 4.32 explicitly illustrates this partition.

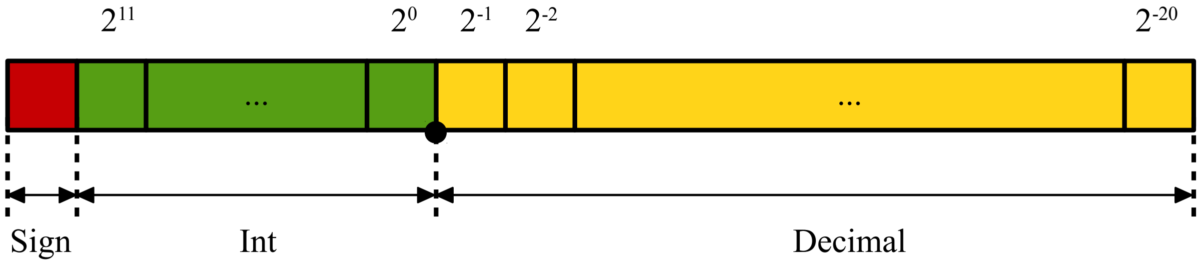

Figure 4.32: 33-bits fixed-point representation of the numerator, denominator and factor F.

The whole conversion is executed in 6 main steps presented in Figure 4.33. The finite state machine starts with the reset stage where all the registers involved in the computation are set to their initial zero values. Then the FSM enters in the acquisition mode. This step continuously receives the CFD outcome and when a negative/positive pair is detected it passes to the following state, otherwise it remains in this mode. If a pile-up rejection signal occurs, the CFD can momentarily disable the acquisition of the two points. A step is then dedicated to the setting of the numerator, computed as difference of the previous negative and positive values, the denominator and the factor F. Also an upper and lower thresholds are set at this point. The range defined by these values is a conservative choice to guarantee the convergence of the system. Because in this version only two points are collected, it is possible to directly define the denominator as the time distance of the two consecutive acquisitions. This quantity is equal to the time difference between the two points considering the first negative one as the origin of the local system. Hence, programming this term in the coefficient array, the initialization of Eq 4.75 is realized just by shifting the denominator of 14 bits to right and storing the value in register F. It was noticed that the nominal shift to right (13 bits) can not be enough to guarantee the convergence of the solution. Thus, the shift parameter was set to a smaller value (14) than the nominal one to ensure the achievement of the correct result in a reasonable computation time. This further conservative approach has a totally negligible impact on the other expected results.

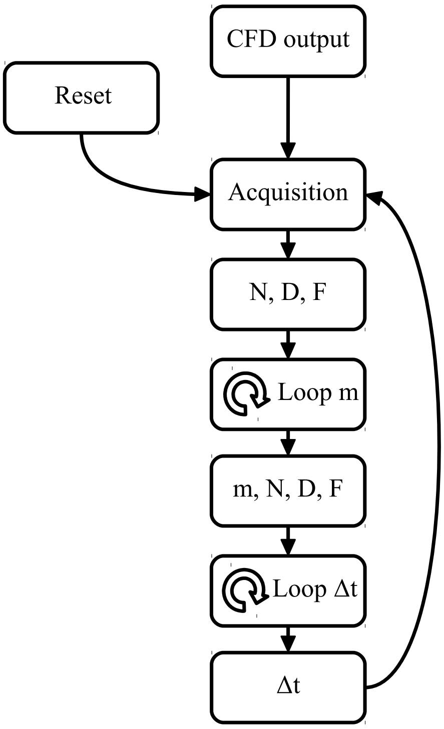

Figure 4.33: FSM of the Goldschmit's algorithm.

Therefore, the FSM iteratively computes the coefficient m in the next step. Typically it requires around 5 clock periods. When the denominator converges within the configurable range an output with the value of m is generated. As well as the bisection algorithm explained in Section4.4.2, an internal counter takes into account the number of iterations to avoid undesired endless loops. Also in this case, the maximum number of steps can be programmed by the user. After this computation, a new instantiation for N, D and F happens. A last loop calculates the Δt that corresponds to the zero crossing point in a reference system where the negative point is the time origin. It is very easy to reconstruct when this point occurs since a pulse signal is generated between the acquisition step and the next state. Finally, the FSM returns to the acquisition mode, ready for another cycle.

Chapter 5

Physical implementation

Both the calibration engine described in Chapter 3 and the digital signal processing discussed in Chapter 4 were carried out in SystemVerilog (SV) IEEE standard language [175]. It is an hardware description language (HDL) used to modeling and simulating an electronic device at many stages, from the behavioral description to register transfer level (RTL) and gate-level. Therefore, dedicated Electronic Design Automation/Computer-Aided Design (EDA/CAD) tools generate the technology dependent netlists driven by associated Synopsys Design Constraints (SDC) files. The technology targets identified for this thesis were 6-metals 110 nm and 9-metals 65 nm CMOS processes. The Place & Route step was accomplished for the placement of the standard cells, their routing, the generation of the clock trees and the power planning. Diodes and antenna cells were added to drive and protect the input ports. The final results were a Verilog netlist coupled with a Standard Delay Format (SDF) file where all the timing annotations of the designs were reported. The simulations were performed in the typical corner and they proved the functionality of the calibration unit as well as the digital processor.

A fault tolerance system against undesired particle interactions was mandatory for radiation applications. One of the next sections will be dedicated to this implementation.

5.1 Calibration processor implementation

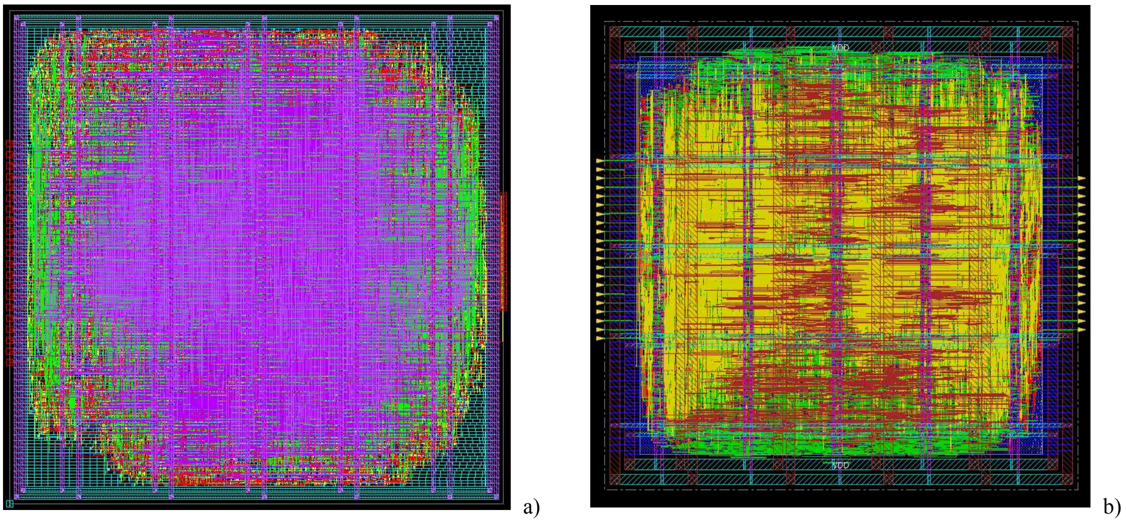

Figure 5.1: Hardware implementation of the digital calibration processor: a) 6-metals 110 nm CMOS node, b) 9-metals 65 nm CMOS technology.

Figure 5.1 shows the layouts of the calibration processor implemented in 6-metals 110 nm and 9-metals 65 nm CMOS technologies. The whole chip areas are 452 x 452 μm^2 and 265 x 265 μm^2, respectively. For this work a non-segmented SAR ADC architecture was chosen as target. A preliminary study in C++ was carried out to address the problem without taking care of the physical implementation. Thus, both a SAR ADC model and the calibration engine were simulated and the analysis of the algorithm led to some important considerations which drove the next hardware implementation. The first one concerned the value of the learning parameters μ_Δ and μ_W. It was figured out that for a reasonable time conversion, these parameters must be set to small values, otherwise the algorithm requires too much time to calibrate the weights or the error risks to oscillate without reaching any convergence point. Another important inference was about the updating mechanism of the weights. Indeed, the learning parameters must be scaled considering the significance of the bit to be effective. In other words, in Eq 3.8 the result from the multiplication of the learning parameter with the error have to be incrementally right-shifted by 1 bit from the LSB to the MSB. In this way, the contribution to the updating of the array is properly weigthed. Therefore, to realize both these conditions and to avoid an increased hardware complexity due to the multiplication stages, a simplified version of the ODC algorithm was carried out as discussed in Section 3.7. Once the error ε is derived, the updating equations Eq 3.7 and Eq 3.8 were computed with a signed-shift to right of

ε by configurable 5-bits long values. Since the shifter mechanism has been already explained, the following code shows the updating scheme adopted for the weights.

```
 1  ////////////////////////////////////////////////
 2
 3  always_comb begin
 4    if (reset) begin
 5      epsilon_shifted = 32'b0;
 6      for (int i = `BIT_LENGTH - 1; i >= 0; i--)
 7        W_wire[i] = 32'b0;
 8    end
 9    else begin
10      for (int i = `BIT_LENGTH - 1; i >= 0; i--)
11        W_wire[i] = W[i];
12      epsilon_shifted = 32'b0;
13
14      if (state == 7) begin
15        for (int i = 0; i < `BIT_LENGTH; i++) begin
16          epsilon_shifted = $signed(epsilon >>> (w_shift - i));
17          case({D_plus[i], D_minus[i]})
18            2'b00: continue;
19            2'b01: W_wire[i] = (W[i] + epsilon_shifted);
20            2'b10: W_wire[i] = (W[i] - epsilon_shifted);
21            2'b11: continue;
22          endcase;
23        end
24      end
25    end
26  end
27
28  ////////////////////////////////////////////////
```

As well as the Δ_d update system, even in this case a main if-else statement divides the initial reset (line 4) from the core of the combinatorial logic (line 9). When the updating state reserved to the weights W is achieved (line 14) the shifted version of ε is computed. To accomplish this passage, the iterative i quantity is subtracted from the configurable w_shift parameter. Thus, depending on the significance of the bit processed, the $epsilon\_shifted$ value is used to update the weights through a case statement where the individual components of the double samples D_+ and D_- are considered (line 17).

Mode	En_w	En_r
Reset	0	0
Reading	0	1
Selecting	1	0
Writing	1	1

Table 5.1: Bit configuration for writing and reading operation.

For the combinantion 00 and 11 the algorithm skips the update task, otherwise it performs the update coherently with Eq 3.8. Since W are always positive and unsigned-defined, the sum and subtraction at lines 19 - 20 are hardcoded in order to maintain the sign information of ε (this is not shown in the code above because of just illustrative purpose).

5.1.1 Serializer

The calibration processor is equipped also with a system to write and read the internal registers, in order to monitor their trend without a dedicated output and to provide a debug system. The data to be stored are fed through a dedicated input ($Data\_in$), while a 1-bit output ($Data\_out$) is used to serialize the data to readout. To select the operation, the signals En_w and En_r have to be configured as reported in Table 5.1. Once the selecting mode 1, 0 is enabled, the serialized bits provided by $Data\_in$ are stored into a memory at each clock period. The stream is 37-bits long and it is partitioned in 5-bit header and 32-bits data. The header is used to select the register following the schema shown in Table 5.2 where each memory allocation has a unique identification code. To actually write the content into the selected register, it is mandatory to confirm the operation with the last configuration 1, 1 reported in table. Otherwise, if the user wants to accomplish the reading task, it is enough to enable the appropriate mode. The stream out is composed by the whole

Header code	Register	Width
00000	delta_seq	32
00001	D_plus	12
00010	D_minus	12
00011	d_plus	32
00100	d_minus	32
00101	epsilon	32
00110	delta_shift	5
00111	w_shift	5
01000	W[11]	32
01001	W[10]	32
01010	W[9]	32
01011	W[8]	32
01100	W[7]	32
01101	W[6]	32
01110	W[5]	32
01111	W[4]	32
10000	W[3]	32
10001	W[2]	32
10010	W[1]	32
10011	W[0]	32
10100	counter_out	5
10101	counter_rw	5
default	no writing	-

Table 5.2: Header configuration for register selection to write.

37 bits both to read the data chosen and to verify the correct register with the header.

5.1.2 Simulation results

Parameter	Value
Voltage Supply V_{DD}	1.2 V
Voltage reference V_{REF}	1 V
Common mode voltage V_{cm}	0.5 V
Resolution	12 bits
Clock ADC frequency f_{ADC}	750 MHz
Sampling frequency f_S	50 MS/s
Injected offset	20-30 LSB
Training samples	30000
DFT samples	4096
Random capacitor mismatch σ_c	15%

Table 5.3: Simulation parameters for the signal generation used for the calibration testing.

In order to calibrate the circuit and to perform the DFT, sinewaves were adopted. A Python script was written to generate a new sinusoidal pattern for a given sampling frequency at each simulation. It considered suitable signal frequencies to satisfy the condition of Eq 3.17, thus the spectral leakage is avoided. The injected offset is also picked up randomly to test the adaptability of the calibration engine to several values. It was figured out that the algorithm was robust enough to tolerate these variations. In Table 5.3 the parameters adopted for the generation of the sinusoidal signals are summarized. For a preliminary investigation, a randomized non-segmented SAR ADC model has been simulated. To mimic the DAC mismatches due to the fabrication process, each capacitor was composed by the sum of unitary elements individually affected by a random capacitance mismatch σ_c. To take into account a remarkable DAC mismatch, a $\sigma_c = 15\%$ was assumed as worst case. As discussed in Section 3.7 the algorithm is managed by a 7-states Mealey FSM. Dedicated modules described in SV language gather all the combinatorial logic required to compute the updating equations and the results are stored into memories at the posedge of the clock. The state progress is scanned by an end-of-conversion (EOC) signal coming from the SAR ADC to report when a new digital code is available. The post P&R timing diagram of the FSM and main registers is depicted in Figure 5.2. First of all, the behaviour of the algorithm before the calibration is reported in the picture. When the control signal CAL is low, the input data (bit_in) is not processed by the correction engine and the calibrated output (not shown in the image) follows the input as expected. If CAL switches to one (vertical red dashed line), the system enters in calibration mode and the offset is the initial digital word to store. The two following input codes represent the double acquisition when the offset is injected. They are kept in memory into D_+ and D_- registers to be used to compute d_+ and d_- at 160 ns.

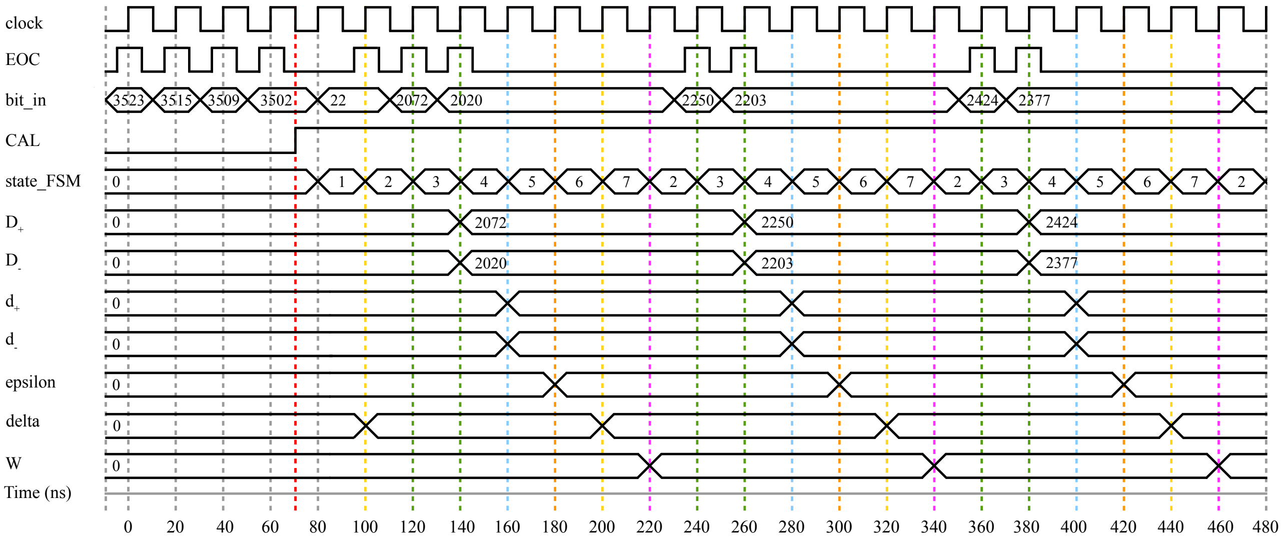

Figure 5.2: Post P&R simulation of the Offset Double Conversion algorithm.

The further step involves the derivation of ε to evaluate the error of the SAR ADC conversion. Finally, both Δ_d and the set of weights W are updated with another two clock cycles. The reader had noticed that because of the time required to compute the whole updating chain $D_{\pm}$, $d_{\pm}$,ε, Δ_d, W, the refreshing of bit_in can not be constant. An entire adjustment cycle takes 120 ns from the double acquisition to the last W update.

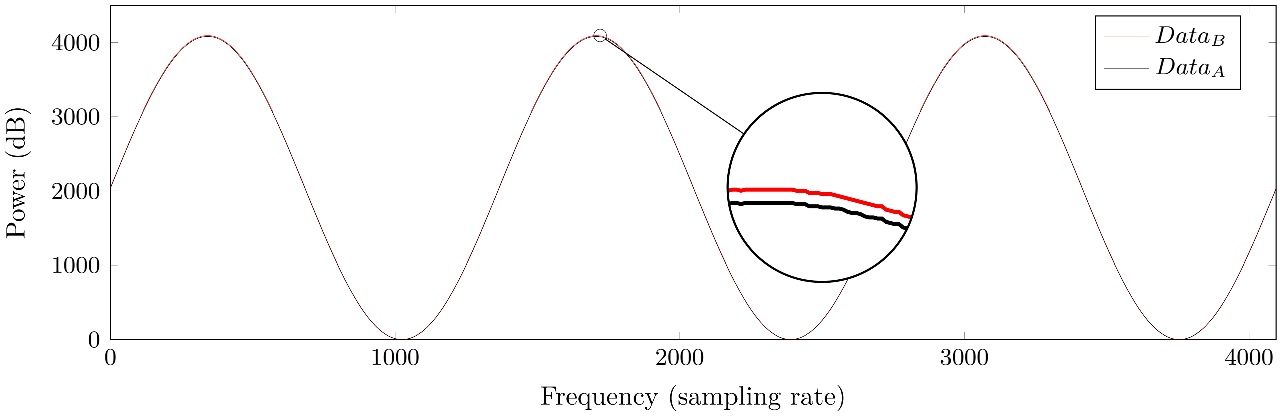

Figure 5.3: Example of two sinusoidal signals before the calibration (red) and after the correction (black).

An example of two sinewave test signals used to estimate the performance of the algorithm is illustrated in Figure 5.3. The calibration unit and its response were simulated post P&R in 65 nm process. $Data_B$ represents the output of the raw data, namely, the digital codes without the correction, while $Data_A$ is the sinusoidal outcome once the calibration is performed. The magnified area highlights the differences that are not perceptible at this scale. Figure 5.4 reports the same information, but in the form of histograms. The plots are obtained collecting 4096 samples and they point out an appreciable discrepancy in the bin content.

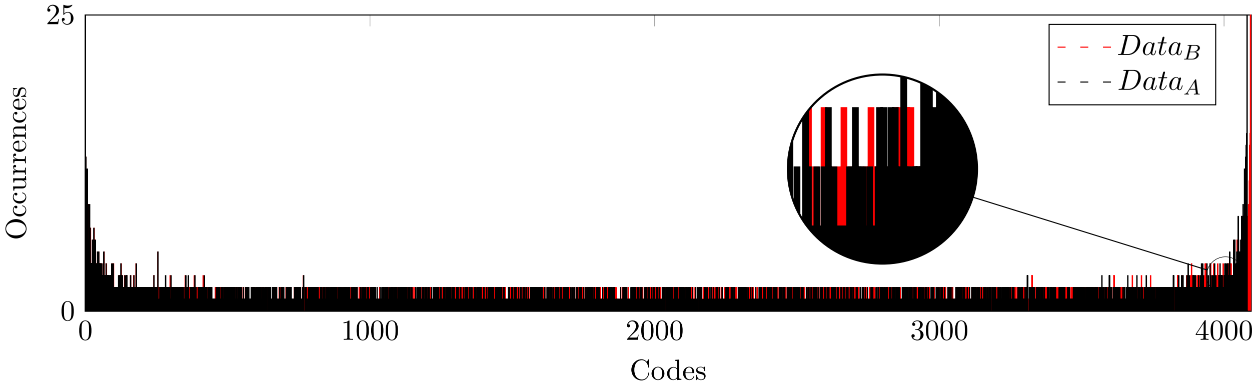

Figure 5.4: Histogram of the signals reported in Figure 5.3

As explained in Section 3.9, a signal can be expressed also in the frequency domain and this representation allows to remark some characteristics as shown in Figure 5.5. Figures of merit extracted from the two spectra are collected in Table 5.4. In the reported power plot the principal component of the signals is close to the DC contribution and it does not appear to be totally separated. However, the two spectra are remarkably different. The power trend before correction shows a considerable incidence of spurious components at low frequencies. They are due to the nonlinearity which affects the DAC bank. After calibration most of them are cut off and it is very well demonstrated by the SFDR that pass from 58.2 dB to 80.3 dB, proofing the remarkable attenuation of spurious components. Also the other figures of merit benefit from this recovery. The signal-to-noise and distortion is increased slightly more than 13 dB. The ENOB was 8.49 before the calibration and it achieves the significant value of 10.72 after the correction, with an increment of +2.23 ENOB. These achievements are comparable to those of one examined paper [65]. According to it, the number of samples used to calibrate the ADC with an offset of 25-LSB is 22000.

Figure of merit	Before	After	Variation
SINAD	52.9 dB	66.3 dB	+13.4 dB
ENOB	8.49	10.72	+2.23
SFDR	58.2 dB	80.3 dB	+22.1 dB

Table 5.4: Figures of merit of the sinusoidal test signals.

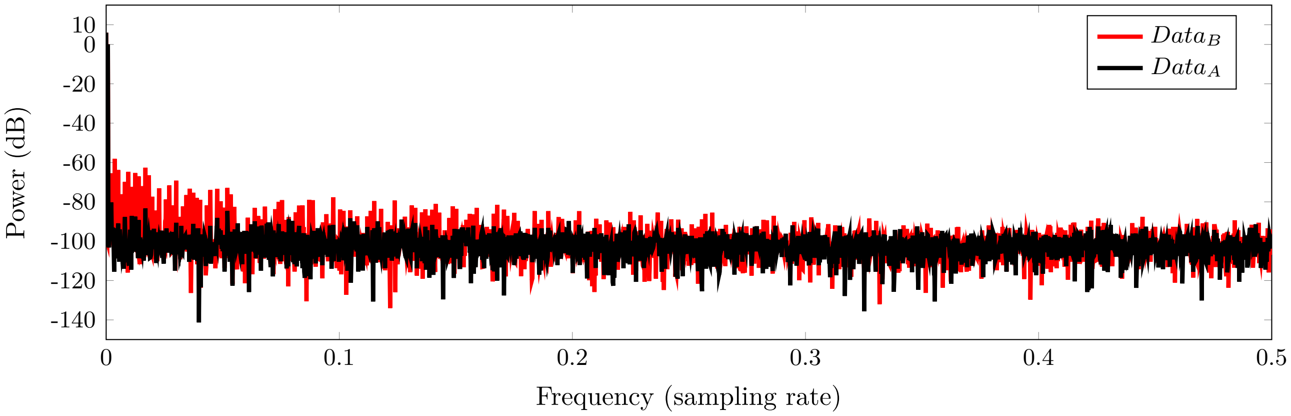

Figure 5.5: Power spectra of two sinusoidal test signals before the calibration (red) and after the correction (black).

The other figure of merits are also close to the results of this work. For instance, the two implementations share the same order of SINAD improvement which is around 10 dB and the SFDR is also roughly the same. These outcomes are satisfactory taking into account the simplified on-chip calibration. Considering a rate of 50 MS/s an 30000 training samples, the time required to correct the DAC mismatch is 0.6 ms. Table 5.5 reports the comparison with other researches used to evaluate the performance of the algorithm.

	[73]	[176]	[177]	[178]	[179]	[65]	This work
Technology (nm)	MATLAB	28	28	135	MATLAB	130	65
Area $10^3(\mu m^2)$	-	16	9.3	260	-	30	70
Sampling rate (Ms/s)	-	4	16	-	-	45	50
Supply voltage (V)	-	1.0	1.0	-	-	1.2	1.2
Power (mW)	-	0.115	0.710	-	-	0.23	0.82
Calibration algorithm	ICA	Redundant search tree	Digital redundancy	Radix < 2	ODC	ODC	ODC
Training samples	$60 \cdot 10^6$	-	2^{17}	-	$0.5 \cdot 10^6$	$22 \cdot 10^3$	$30 \cdot 10^3$
Time calibration (ms)	600	-	-	-	-	< 1	0.6
ADC architecture	Pipelined SAR	SAR	Pipelined	Pipelined	Pipelined	SAR	SAR
Resolution (bits)	12	12	15	14	15	12	12
SINAD (dB)	69	-	91.8	78	52	70.7	66.3
ENOB	-	10.1	14.9		12.7	-	10.7
SFDR (dB)	100	-	117	97	108	94.6	80.3

Table 5.5: Comparative table of calibration techniques for ADCs.

The comparison is particularly arduous due to the large spread of technologies and architectures involved as well as the parameters considered to estimate the calibration ability of each techniques. In addition, the investigations are classifiable in four categories:

- High level simulations: these works are realized with softwares as MATLAB that can not guarantee a realistic hardware implementation since a slack analysis can not be performed.

- EDA/CAD simulations: only these results are reliable for the performance evaluations and the manufactoring process.
- Silicon results: the hardware implementation is the best test bench to widely analyze the designed circuits. The power consumption estimations coming from this stage are definitive.
- Hybrid results: this class collects all the hardware integration with a software counterpart. In this scenario the evaluation of performance could be very challenging.

This further categorization makes harder an adequate comparative study among the methods. A significant case is [65] where the digital calibration is carried out at software level, letting the evaluation about the area to a synthesis tool. In contrast, the present work considers the area reserved to the communication system and the power rings as well as the one exclusively dedicated to the correction engine. The same problem is addressed for power estimations, since the works often do not specify how the power consumption is evaluated. Thus, keeping in mind these limitations, the implementation presented in this thesis is competitive with other solutions when a digital ADC calibration is required. A further study was carried out to test the minimum number of necessary corrected bits in order to hold comparable performance with the 12-bits full case [180]. If the conversion error is exclusively dominated by the DAC mismatch without a noise component, it was figured out that the ENOB degradation is limited even calibrating only the first 6-MSB. This result depends on the choice of the learning parameters and the offset, thus it is more conservative to correct the first 8-MSB to ensure the ENOB preservation. The power budget could be further reduced as a consequence of the shorter calibration.

5.1.3 Test chip results

A prototype ot the calibration engine has been fabricated in 110 nm CMOS process. The digital block receives the data from a 12-bits SAR ADC and the chip hosts two main serializers and a LVDS bank. The fully differential architecture of the converter was carried out as segmented structure. The MOM unit capacitor were chosen particularly small with a 2 fF value. The present work also included the design of the serializers to send out the samples converted by the SAR ADC and those ones calibrated by the correction processor. Both of them were realized to process the samples at 100 MHz which is the same frequency of the converter. The data stream generated is composed by a 4-bits header, required for the alignment, and the 12-bits converted word. The calibration circuit operates at the lower frequency of 20 MHz and it is equipped with a further serializer for writing and reading operations. The chips were bonded on boards to provided the required power supply and to interface them with a FPGA in order to read out the data. The digital block was not directly testable due to a hardware level problem. Indeed, during the integration of the circuit two signals with the same name on the layout were connected to each other. In this way a single input signal was shared between the SAR ADC and the calibration module. Unfortunately, this issue was critical because the two signals were designed to behave differently when the whole circuitry is on. In particular, this undesired sharing involved the calibration control signal named CAL which was previously described in Section 3.7. The digital processor requires a steady state signal to pass from the default idle to the calibration state, while the SAR ADC demands a dedicated signal switching to operate properly. Since this connection was made on silicon, a cheap workaround was not feasible. However, it was possible to extract the samples through the serializer dedicated to the converter, named *Raw serializer* in Figure 5.6 a), in order to feed the simulated engine in the technology target. The test campaign collected the data composed by two different collections. One acquisition set was constituted by the simple raw output by using a sinewave as input signal. The second bunch was formed by enabling the injection circuit designed to provide the double offset on-chip. The outcome of this unit is a sequence of triplets made by three consecutive data: the not injected value, the positive offset and the negative one as shown in Figure 5.7.

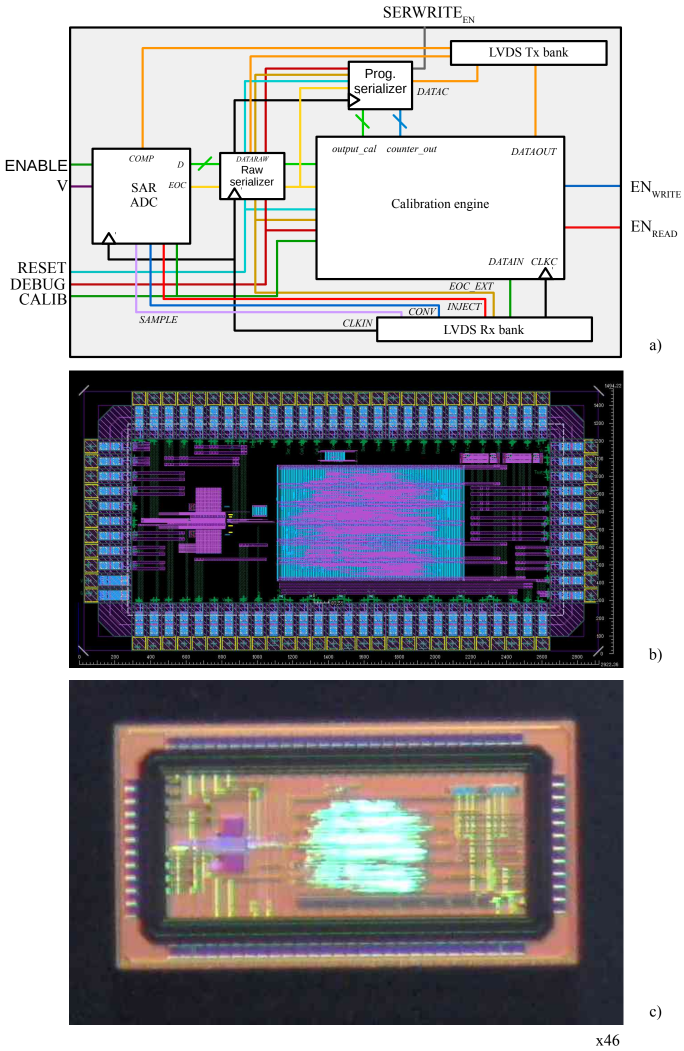

Figure 5.6: In the upper section a) is illustrated the block level architecture of the chip, while the middle picture b) shows the layout of the chip and c) reports its microphotograph.

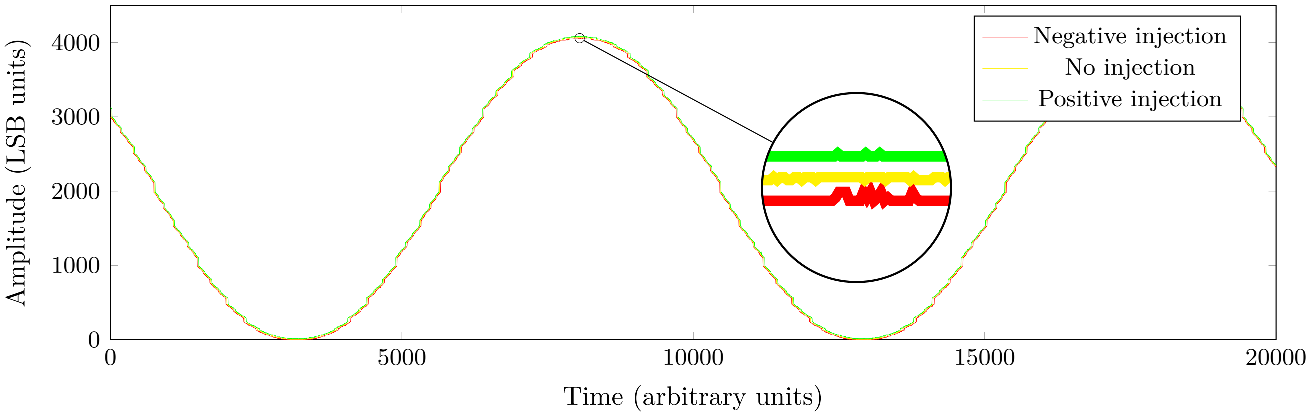

Figure 5.7: The injection circuit triplets formed by a no injected data (yellow) a positive injection value (green) and a negative one (red)

Table 5.6 summarizes the parameters set for the module test. Given the nominal clock frequency of 100 MHz for the ADC and the number of samples selected for the DFT, the input frequency f_{in} was carefully chosen to limit as much as possible the spectral leakage, as well as the previous simulations presented in Section 5.1.2. Moreover, f_{in} was intentionally set in the range of few kHz or above to avoid the introduction of a further offset during the injection process. Thus, the offset peak-to-peak value was bounded around 30 LSB although it can approximately reach a value of 100 LSB at some points of the converter range. These gaps occur at the bit decision boundaries [184] and they involve many ADC codes. In addition, the input sinewave exhibits a staircase shape superimposed to the ideal signal as appreciable in the magnified area of Figure 5.8. Here the *Raw data* represents the samples before the calibration and its step trend is due to the DAC mismatch. The algorithm reshapes the collected samples resulting in the output *Calibrated data* where the sinewave clearly appears to be smoother than the original one. The correction involves all the available digital codes and the calibrated signal amplitude is determined by the learning parameters

Parameter	Value
Voltage Supply V_{DD}	1.2 V
Voltage reference V_{REF}	1 V
Common mode voltage V_{cm}	0.5 V
Nominal resolution	12 bits
Clock ADC frequency f_{ADC}	100 MHz
Input frequency f_{in}	0.762 kHz
Injected offset$_{peak-to-peak}$	$\sim$ 20-100 LSB
Training samples	300k
Power spectra samples	131073
Δ_d	10001
Δ_w	01100

Table 5.6: Parameters used for the chip test.

μ_Δ and μ_W. These values basically remap the original range into the desidered one. Thus, the method compacts the original spreaded codes into a smaller range, trying to take advantage of the nominal 12-bits resolution redundancy to reduce the non-linearity. The algorithm operates both at the bit decision boundaries and at the other step-like gaps.

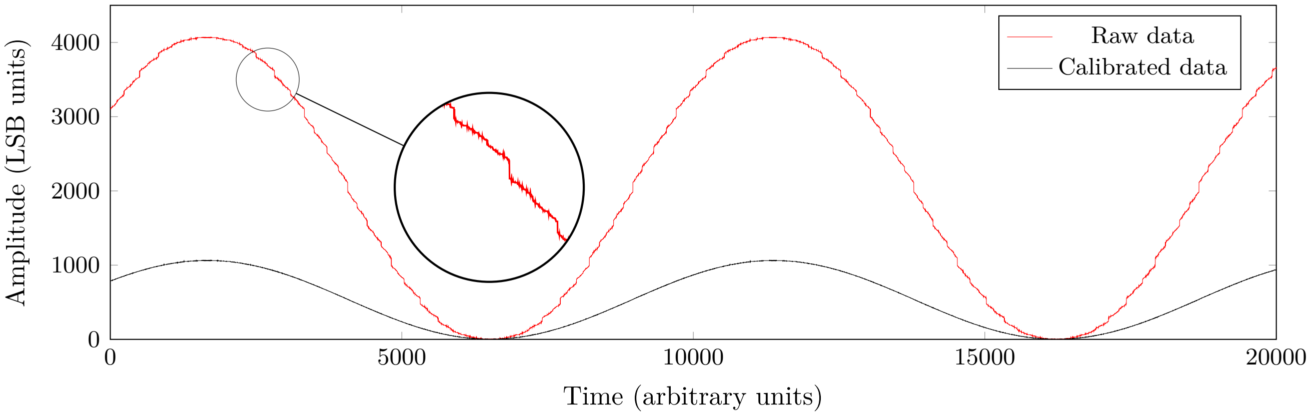

Figure 5.8: The two sinusoidal signals before the calibration (red) and after the correction (black) obtained from the sperimental setup.

The correction is also glaring in Figure 5.9 where the red histogram reports the collected codes. In this case, the nominal resolution is particularly degraded and only 866 codes are available, distributed in a kind of bunches along all the 12-bits resolution range. This could be ascribed to the lack of information about the Montecarlo simulations for the MOM capacitors provided by the foundry.

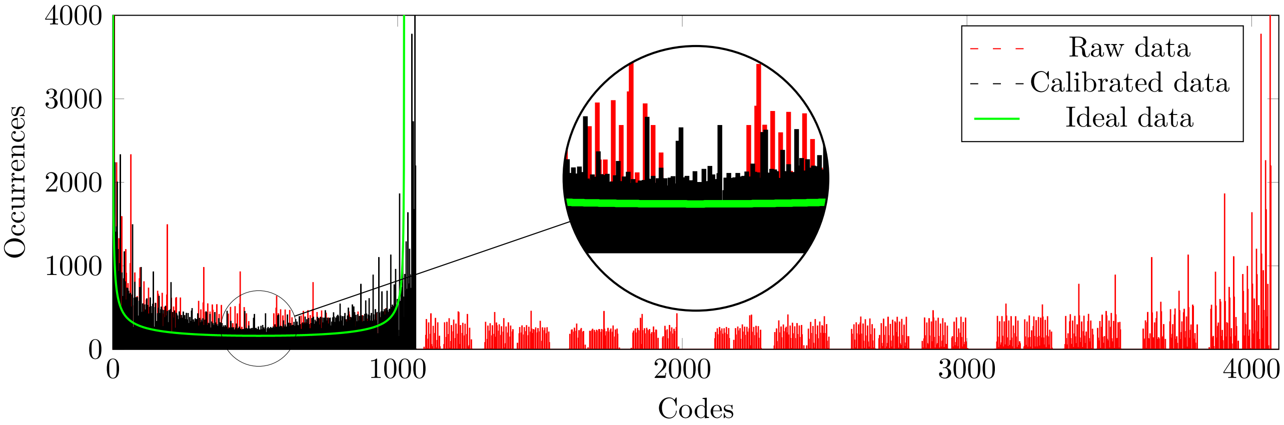

Figure 5.9: Histogram of the two sinewaves shown in Figure 5.8

The magnified area highlights the MSB decision boundary of the new target range which was set to 10-bits of nominal resolution now. Before the calibration no codes are gathered in many intervals of this range, while after the correction the same code-window was populated. This means that the method manipulated the initial 866 codes to increase the ENOB, trying to obtain the ideal distribution highlighted with the green shape in the picture. Naturally, the number of the original available codes is not enough to achieve the full 10-bits resolution. Additional post P&R simulations proved that the range can be further shrinked to 9-bits reducing the number of missing codes to about ten on 512 nominal codes. However, this solution pays the price of a final ENOB smaller than those one reported in Table 5.7. This should not surprise the reader since the ENOB is not just a mere measure of the number of missing codes, but it accounts also how the codes are well-distributed along the range. In other words, even if the outputs of the converter apparently cover the conversion range, it could be possible that significant distortions degrade the quality of the conversion itself. Hence, because of the initial extreme distortion of the output signal, for this elaboration it was chosen to privilege a higher linearity gain rather than a reduced number of missing codes. This choice was done only to prove both the configurability and effectiveness of the algorithm in the data manipulation. The samples remapping is even evident in the frequency domain as shown in Figure 5.10. The raw signal is affected by a large number of spurious frequencies generated by the non-linearity introduced during the conversion phase. The method is particularly effective to decimate these undesired frequency contributions along all the sampling rate range. The depicted power spectra was obtained by collecting 2^{17} samples after a training where 300,000 data were used. The initial SINAD of 31 dB was increased to slightly less than 51 dB, while the ENOB achieved a remarkable increment of +3.26. Also the SFDR met a positive variation of the roughly same magnitude of the SINAD with an increment of +22.45 dB.

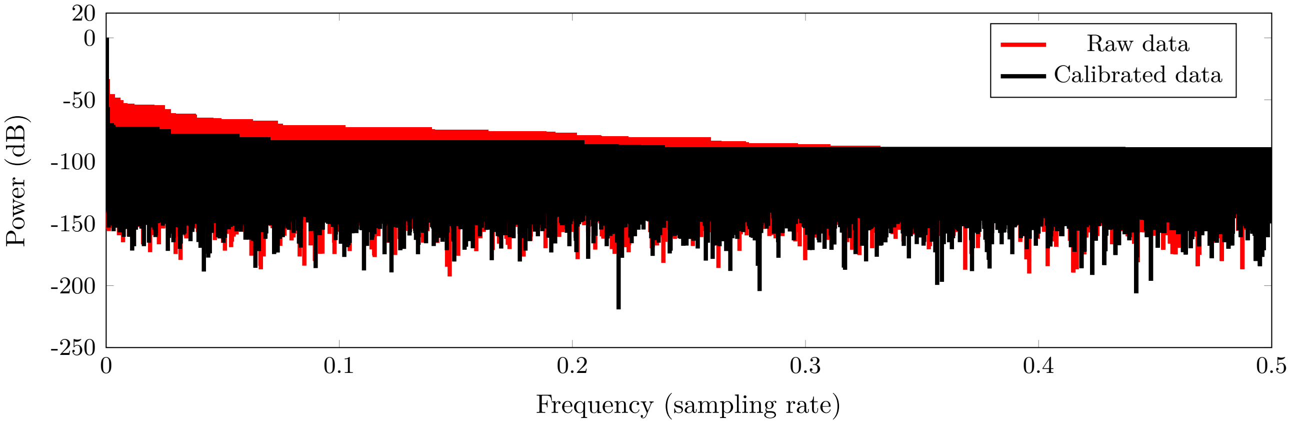

Figure 5.10: Power spectra of two sinusoidal signals before the calibration (red) and after the correction (black).

Let's consider now the first data column of Table 5.7 that lists the figures of merit before the calibration and the second one which considers the outcome after a correction run with a sinewave. Unlike the results summarized in Table 5.4 obtained with a DFT code implemented from scratch for this purpose, these are derived by using a Fast Fourier Transform (FFT) routine available on Python 3 to speed up the analysis. The recorded improvements are considerable by taking into account a quite large number of double training samples, itemized in the rightmost column.

Figure of merit		sinewave		ramp		Training samples (10^3)
	Before	After	Variation	After	Variation	
SINAD (dB)	31.29	44.63	+13.34	50.21	+18.92	100
ENOB	4.90	7.12	+2.22	8.05	+3.14	
SFDR (dB)	33.55	46.93	+13.38	60.39	+26.84	
SINAD (dB)	31.29	50.25	+18.97	52.35	+21.06	200
ENOB	4.90	8.06	+3.15	8.40	+3.50	
SFDR (dB)	33.55	54.57	+21.02	60.13	+26.58	
SINAD (dB)	31.29	50.93	+19.64	49.88	+18.59	300
ENOB	4.90	8.17	+3.26	7.99	+3.09	
SFDR (dB)	33.55	56.0	+22.45	54.25	+20.70	
SINAD (dB)	31.29	51.02	+19.73	51.02	+19.73	400
ENOB	4.90	8.18	+3.28	8.18	+3.28	
SFDR (dB)	33.55	56.76	+23.21	56.21	+22.66	
SINAD (dB)	31.29	50.73	+19.44	50.97	+19.68	500
ENOB	4.90	8.13	+3.23	8.17	+3.27	
SFDR (dB)	33.55	55.86	+22.31	56.76	+23.21	
SINAD (dB)	31.29	50.81	+19.53	51.33	+20.05	600
ENOB	4.90	8.15	+3.24	8.24	+3.33	
SFDR (dB)	33.55	56.60	+23.05	58.45	+24.90	
SINAD (dB)	31.29	51.77	+20.48	51.33	+20.05	700
ENOB	4.90	8.31	+3.40	8.24	+3.33	
SFDR (dB)	33.55	59.43	+25.88	58.45	+24.90	
SINAD (dB)	31.29	51.44	+20.15	51.65	+20.36	800
ENOB	4.90	8.25	+3.35	8.29	+3.38	
SFDR (dB)	33.55	60.11	+26.56	58.58	+25.03	

Table 5.7: Figures of merit of the tested chips.

These results are graphically represented in Figure 5.11 where the parameters are plotted as a function of the number of training samples. Both the SINAD and the ENOB show a clear plateau that starts from a training samples of 200,000 acquisitions, while in the SFDR plot this saturation is not so evident. This observation is important to contain the calibration to a reasonable time window.

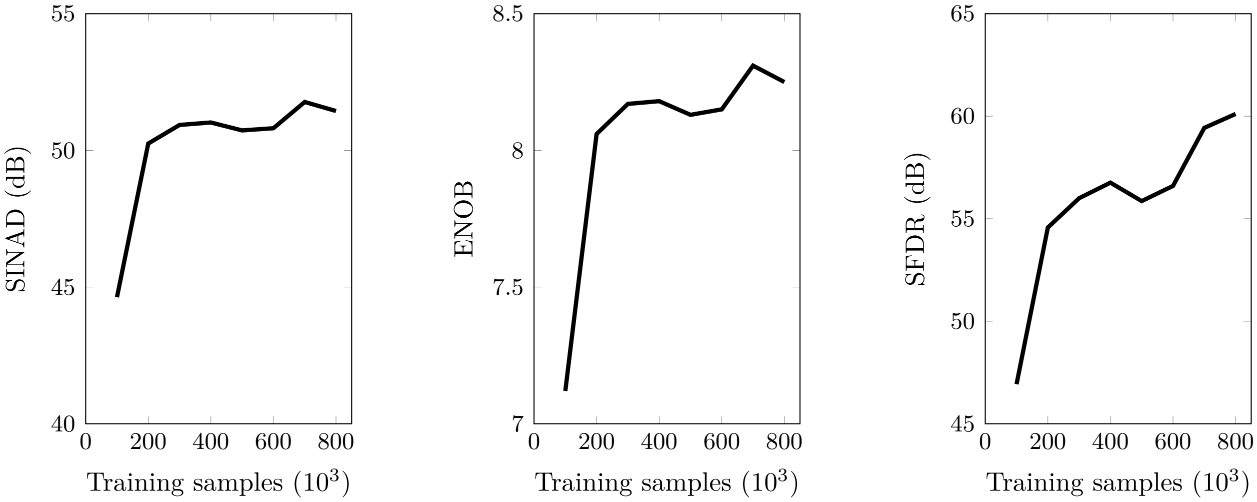

Figure 5.11: Figures of merit outcomes for different numbers of training samples using a sinusoidal signal as input.

Indeed, a quick comparison with the results collected in Table 5.4 shows that even in the smaller training set two out of three figures of merits are similar to the pure simulations case. The most important differences are about the training sets. To achieve these comparable outcomes a number of samples which is three times the preliminary estimations is required. At this point it is necessary to highlight the different initial conditions since the original evaluations did not considere an extreme DAC mismatch as that one observed in the laboratory test. However, the algorithm responds adequately even in this not favorable condition. A general improvement occurs for all the training sets used, achieving the maximum positive ENOB variation of +3.40 in the best case for the given learning parameters and by using a number of training samples equal to 700,000. As mentioned before, the input used to calibrate the weights is a sinusoidal signal which amplitude was properly chosen to avoid the saturation during the injections. However, although the availability of a signal generator is not problematic during a laboratory test, it becomes trickier to generate a sinewave on-chip for standalone applications. A more suitable signal for a local generation is a ramp which is enough for calibration purposes. Thus, the algorithm performance was also tested by providing a ramp signal to the correction engine. The outcomes were collected in the third column of Table 5.7 and they are fully competitive with the ones achieved with a sinewave. It is worth to mention that for most training sets the results are even more better than the sinusoidal signal case. For instance, the highest performance was achieved for a number of training samples equal to 200,000, where the SINAD is incremented by +21 dB, ENOB passes from the initial 4.90 bits to 8.40, recording a positive variation of +3.50, and the SFDR reaches slightly more than 60 dB. The increased outcomes can be explained by considering that the equal flat code distribution could be more suitable than the sinusoidal input signal. Given the selected learning parameters μ_Δ and μ_W, the non-linear points are equally well explored in this case and the final performance benefits by this feature. Although the three pictures of Figure 5.12 point out a shared bump located at 300,000 training samples, the general trend resembles the plateau achievement of Figure 5.11 with global better results. This aspect is particularly interesting in terms of an embedded on-chip integration for multichannel systems and it proves its feasibility.

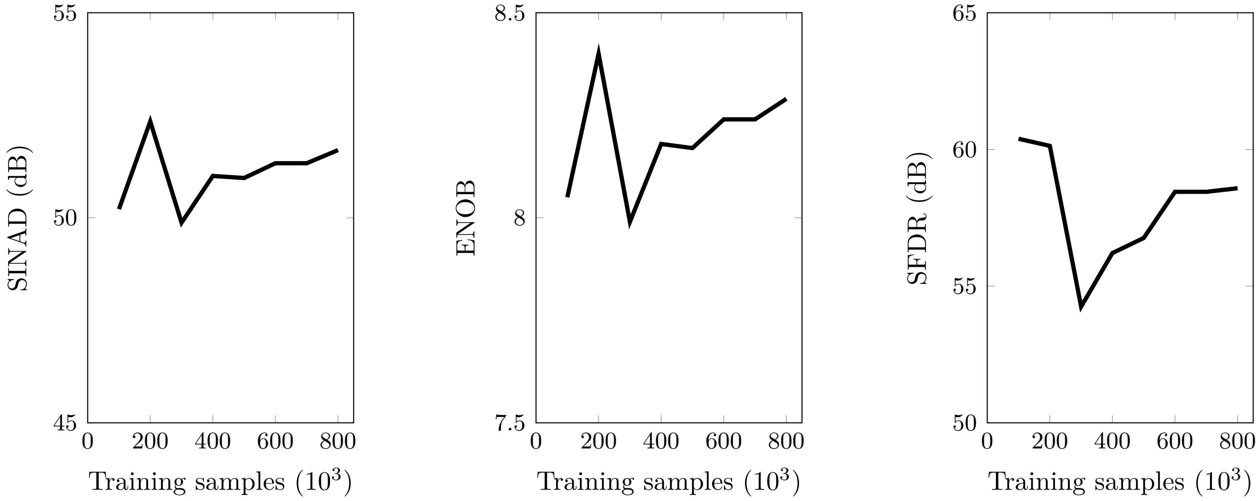

Figure 5.12: Figures of merit outcomes for different numbers of training samples using a ramp signal as input.

5.1.4 Power analysis

To complete the study of this unit, the power consumption was also investigated because the gate delays strongly depend on the applied voltage. The digital power dissipation is formed by two components: the static power and the dynamic one [181]. The first contribution is relevant to the case when the standard cells are freezed into some logic states and no charge and discharge event occurs. To get the idea about this configuration, let's recall a simple CMOS inverter powered by a VDD voltage source. When the input is 0, the p-MOS transistor is closed, while the n-MOS is open and this results into an output equals to VDD. If the input is 1, the complementary situation happens and the output is equal to 0. Ideally, no current can flow from the terminal VDD to ground because the path is always opened due to one transistor switched off. Then, the static power dissipation should be equal to zero. However, the path between the diffused regions of drain and source terminals and the p-substrate gives a static power consumption because of the reverse-bias leakage.

The total power related to the leakage dissipation is then calculated as the product between the sum of all possible leakage current sources I_L and the supply voltage VDD:

$$P_{leakage} = \sum_{i=0}^{n} I_L \cdot VDD \tag{5.1}$$

A detailed discussion about the leakage can be found in [182]. Thus, to define the power associated with the static fraction, the latter quantity is summed to the internal power due to the charged and discharged states of the internal capacitances of each cell. Therefore, the average switching power is also added and computed using the formula:

$$P_{switching} = \frac{1}{2}\alpha \cdot VDD^2 \cdot C \cdot f \tag{5.2}$$

where α quantifies the activity factor, C represents the power-dissipation capacitance and f is frequency of the input signal. Eq 5.2 is required by the tool to compute the average switching power [183] as part of the global static power. The activity factor is set to a default value, but this quantity is overwritten if the user provides a post-layout simulation to obtain a more precise result as the case of this thesis. The IR drop along the power grid, named static rail analysis, is also extracted from this estimation. Conversely, when steady-states are not kept anymore by the logic gates, the dynamic power consumption is evaluated. In this second contribution the current flows during the switching activity of each transistor that changes its current logic state. The switching power component is detailed computed based on a vectorless timing database and the effective switching derived from the simulation waveforms. The dynamic internal power takes into account also the short-circuit powers due to a momentary short circuit current that occurs between the P and N transistors of each cell during the switching. These values are collected into the libraries provided by the foundry. As well as the static version, even in this case the previous results can be exploited to estimate the IR drop, called dynamic rail analysis. For the current design two modes of the processor were examinated, namely, when the calibration is enabled and when the correction does not take place, which is the default operation mode. Hence Table 5.8 summarizes the results for the 9-metals 65 nm CMOS technology implementation using the data collected from the test chip discussed in Section 5.1.3.

Power	without calibration		with calibration		with calibration	
	Static (mW)	(%)	Static (mW)	(%)	Dynamic (mW)	(%)
Internal	0.3220	86.11	0.4079	85.47	0.7301	86.19
Switching	0.0500	13.38	0.0673	14.11	0.1150	13.58
Leakage	0.0019	0.52	0.0020	0.42	0.0020	0.24
Total	0.3740		0.4772		0.8471	

Table 5.8: Static and dynamic power budget without calibration and during correction.

Table 5.8 itemizes the power estimations derived with a dedicated software tool. The static power is divided into two columns, in particular when the calibration block runs without the correction (first column) and when the weights adjustment is enabled (second column). The last column sums up dynamic budget. Although the percentages remain quite the same for the three cases dominated by the internal power, the total values of the static parts are remarkably different compared to the dynamic evaluation. The main differences concern both the internal and switching powers, while the leakage is barely larger in the second case. These values are explained by considering the calibration engine activation that requires more power due to the weights computation. The dynamic contribution on the third column is calculated by taking into account the most power consuming interval along 1600 ns post P&R simulation in the typical corner. The larger power demand compared to the static configuration is evident because the dynamic estimation is slightly less than twice the static value in calibration mode. Tables 5.9 and 5.10 show the different power distribution among the power groups. Indeed, when the calibration is not enabled the combinational contribution to the total 0.374 mW is limited to 5.985% and the power budget is dominated by the sequential portion with almost the 80%. This partition changes by passing to the calibration mode as indicated in Table 5.10 where the combinational part is increased to $\sim$0.11 mW, thus roughly less than 24%. The variation is justified by the computational power required to figure out the sequence $d_{\pm}$, ε, Δ_d and finally the weights.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	0.2974	0.0004	0.0002	0.2979	79.66
IO	0	0	$1.4 \cdot 10^{-9}$	$1.4 \cdot 10^{-9}$	$3.8 \cdot 10^{-7}$
Combinational	0.0078	0.0130	0.0016	0.0224	5.985
Clock (Combinational)	0.0169	0.0367	$8.6 \cdot 10^{-6}$	0.0536	14.33
Total	0.3221	0.05	0.0019	0.374	100

Table 5.9: Static power contributions without calibration. The total value as well as the internal, switching and leakage ones are expressed in mW.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	0.3085	0.0016	0.0002	0.3103	65.02
IO	0	0	$1.4 \cdot 10^{-9}$	$1.4 \cdot 10^{-9}$	$3.0 \cdot 10^{-7}$
Combinational	0.0825	0.0291	0.0017	0.1132	23.73
Clock (Combinational)	0.0169	0.0367	$8.6 \cdot 10^{-6}$	0.0536	11.23
Total	0.4079	0.0673	0.0020	0.4772	100

Table 5.10: Static power contributions enabling the calibration engine. The power is expressed in mW.

The trend is confirmed in Table 5.11 where the sequential power budget plummets to 38% while the combinational one consistently rises to 55%, thus more than half of the total power in calibration mode.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	0.3221	0.0020	0.0002	0.3243	38.28
IO	0	0	$1.4 \cdot 10^{-9}$	$1.4 \cdot 10^{-9}$	$1.7 \cdot 10^{-7}$
Combinational	0.3912	0.0769	0.0018	0.4699	55.47
Clock (Combinational)	0.0169	0.0361	$8.6 \cdot 10^{-6}$	0.0529	6.248
Total	0.7301	0.115	0.002	0.8471	100

Table 5.11: Dynamic power contributions during the calibration. The power is evaluated in mW.

To better visualize this quantitative dynamic analysis on the chip area, some heatmaps are reported in the following.

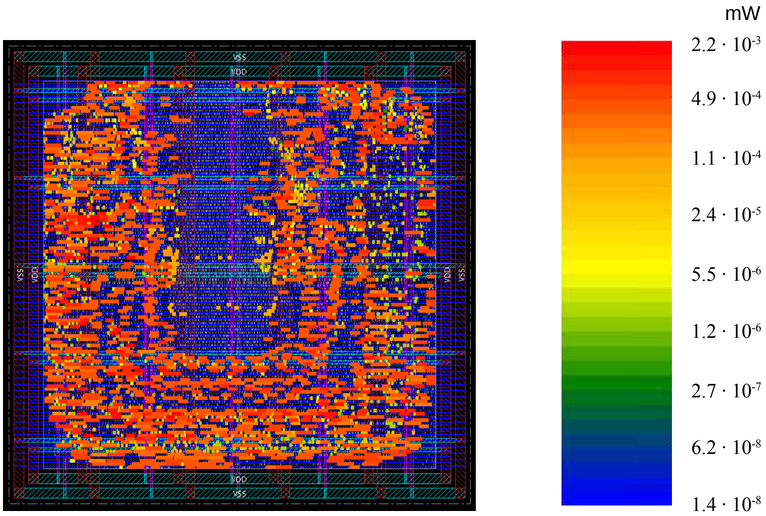

Figure 5.13: Heatmap of the internal power over the chip area of the calibration block.

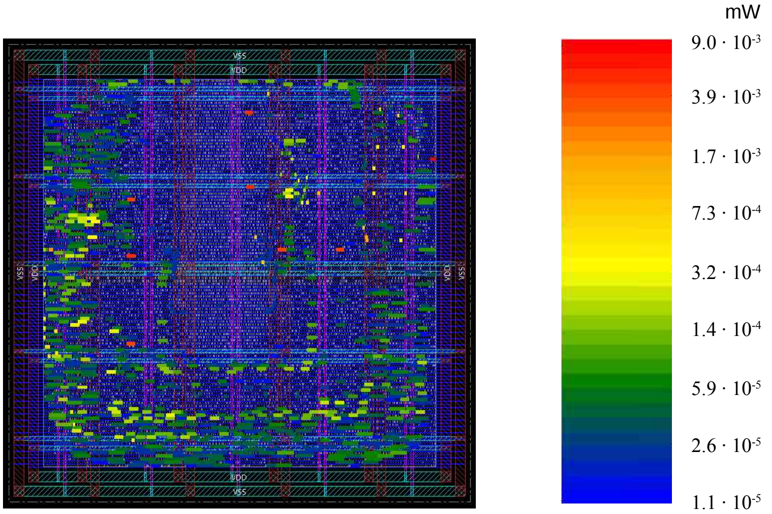

Figure 5.14: Heatmap of the switching power of the instances.

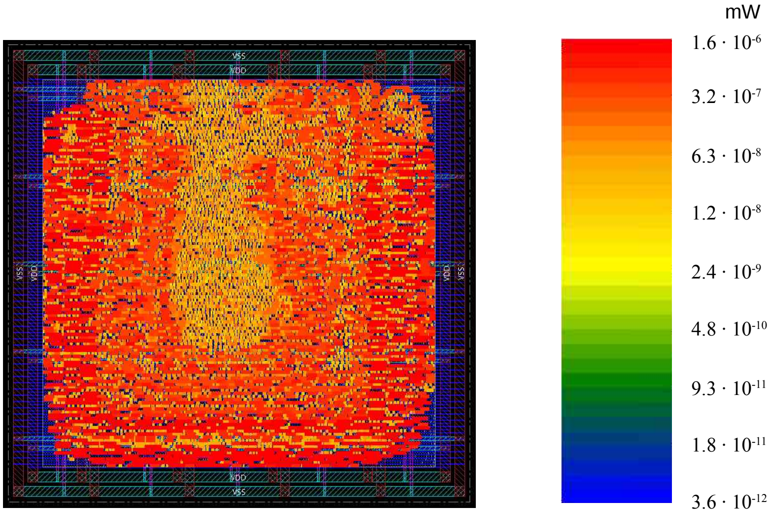

Figure 5.15: Heatmap of the leakage power of the standard cells.

These maps are characterized by an apparently singular U-shape that is due to the optimization process of the P&R tool used. The combinatorial logic of the module dedicated to the computation of the weights was mainly allocated in the middle area of this shape. This hardware is surrounded by the registers to store the outcomes and even outermost the other blocks take place. This implementation appears a quite natural choice to minimize the slack among the sub-modules since the circuitry used to update the weights is strictly interfaced with the others. Indeed, the reader have to recall again the algorithm discussed in Section 3.6 where the calculations clearly show how the weights are involved in the derivation of $d_{\pm}$, ε and Δ_d. Therefore, based on these considerations the recognizable shape is fully justified. The previous maps merge into Figure 5.16 where the total power is represented. Figure 5.17 visually describes the total power density.

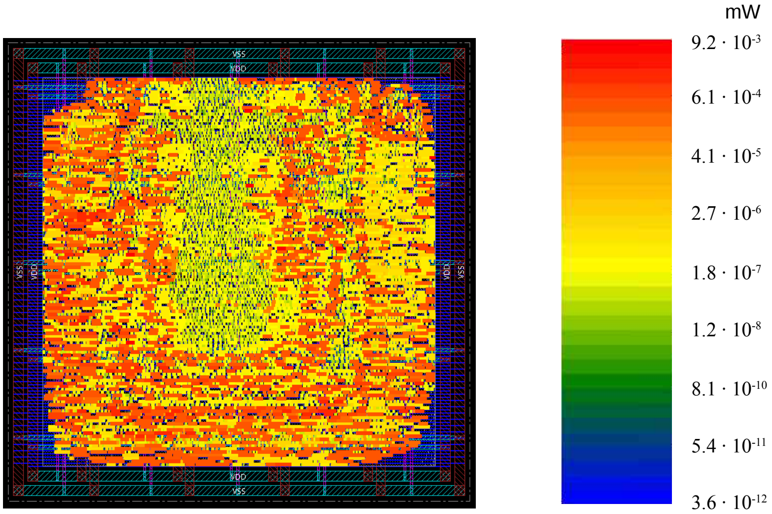

Figure 5.16: Heatmap of the total power that characterize the implemented ODC algorithm.

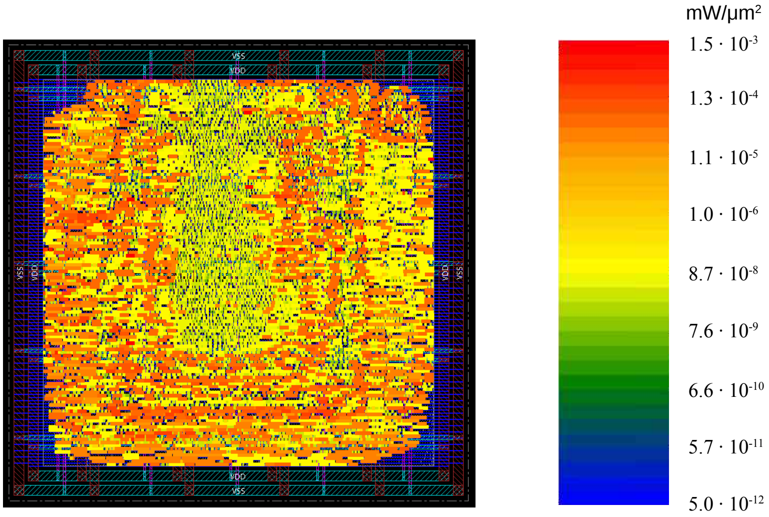

Figure 5.17: Heatmap of the total power density.

The last heatmap of Figure 5.18 illustrates the IR drop on the chip area when the voltage source VDD is equal to 1.2 V. Some spots of the map exhibit the larger value reported in the thermometer bar. These extreme points are probably due to the width of the vertical power metal lines and could be mitigated by increasing the area dedicated to these stripes or adding further lines. However, the general IR drop is uniform enough on the chip area and it is acceptable.

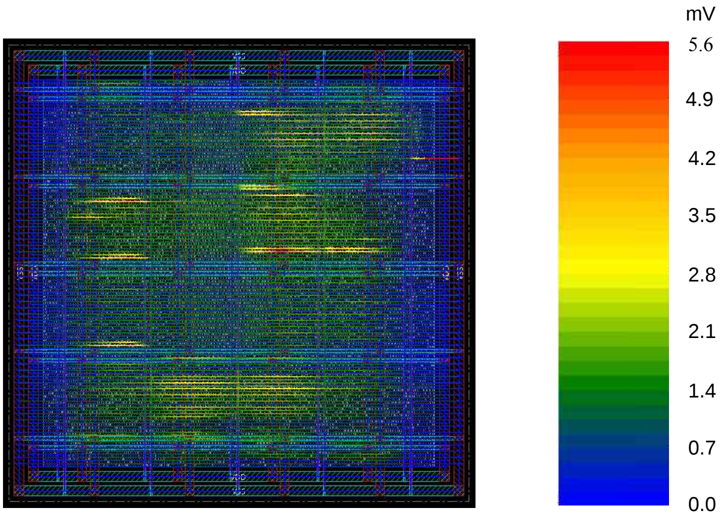

Figure 5.18: Heatmap of the IR drop with the voltage source VDD equals to 1.2 V.

The pie chart representing the dynamic power partition among the sub-modules of the block is reported in Figure 5.19. It was obtained by considering five double acquisitions, including the initial offset injection. This latter explains why the delta module contribution is prominent in the chart since the circuit dedicated to the updating of Δ_d is enabled one more time. The slices reserved to the computation of $d_\pm$, the weights and the final output, respectively named *d_pm_module*, *W_module* and *out_module* in the chart, are clearly comparable in terms of power consumption. It was expected because the implementation is similar in the RTL code and these passages involve digital words with the same length. Thus this means that their switching activity can be also guessed roughly comparable. Lastly, the ε derivation presents the lower power impact since it implements only subtractions and no multiplications are required.

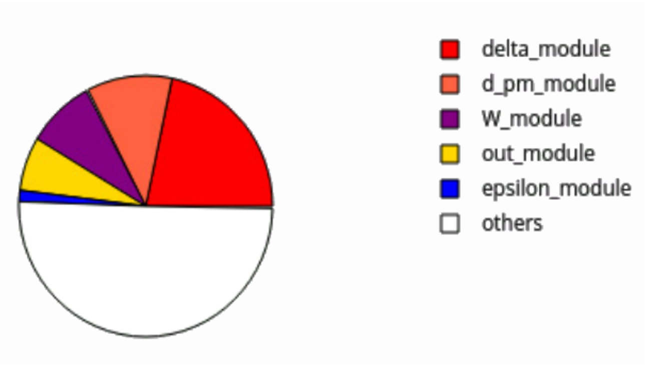

Figure 5.19: Pie chart of the power budget for the calibration engine.

5.2 Digital signal processor implementation

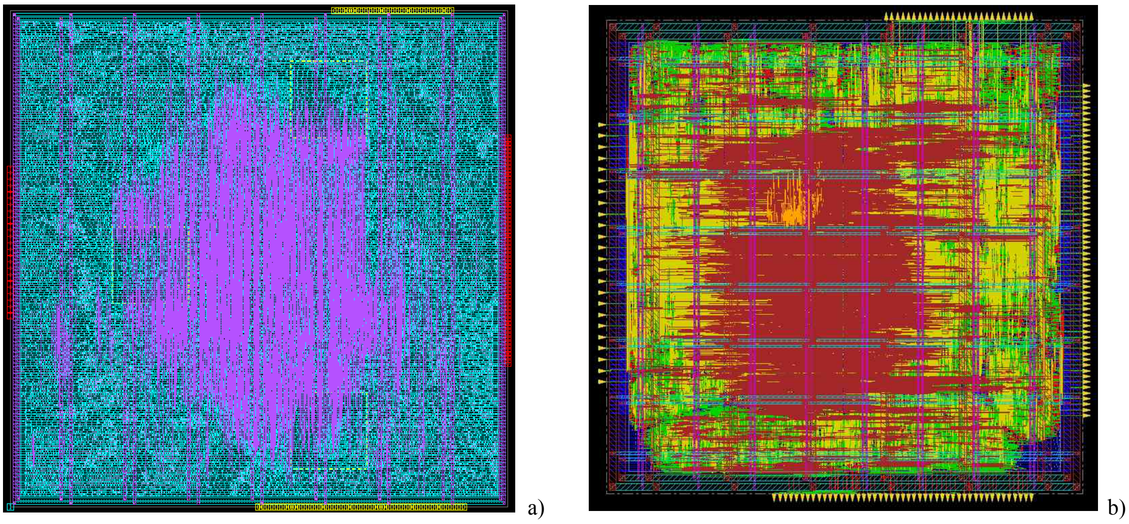

Figure 5.20: Layout of the digital signal processor: a) 6-metals 110 nm CMOS process, b) 9-metals 65 nm CMOS technology.

Also the digital signal processor was implemented both in 6-metals 110 nm and 9-metals 65 nm CMOS technologies. The final layouts occupy an area of 781 x 781 μm^2 and 424 x 424 μm^2, respectively. In addition, due to the number of crucial registers for the operativity of the circuit, this unit was equipped with a protection against radiation. In next sections this topic and the strategy assumed will be discussed.

5.2.1 Radiation hardness

Let's consider a space mission where many electronics systems are employed both for pure automatic tasks such as satellites or rover missions and crew transportation toward the International Space Station. This environment is deprived of the atmospheric protection, becoming hostile for human beings, but also for integrated circuit as well. The section of a generic silicon device is reported in Figure 5.21, showing what happens when a charged particle passes through it [185]. The red arrow represents the dense ionization track of a particle, such as proton or a heavy ion, into the material. The crossing from the passivation layer to the substrate creates free electron and hole pairs as discussed in Section 1.1. If the particle interacts with the nucleus of an atom its scatter contributes to another ionization process. Many electron and hole pairs recombine back along the path, but when the interaction involves the Si substrate depletion region, a collecting effect of electrons both by drift (darker green arrow) and diffusion (lighter green one) happens. This is due to the higher voltage of the NMOS drain diffusion and it can produce a bit flip in a memory cell. This phenomenon is named Single Event Upset (SEU) and its effect on the storage of a SRAM cell [186] is examinated in Figure 5.22.

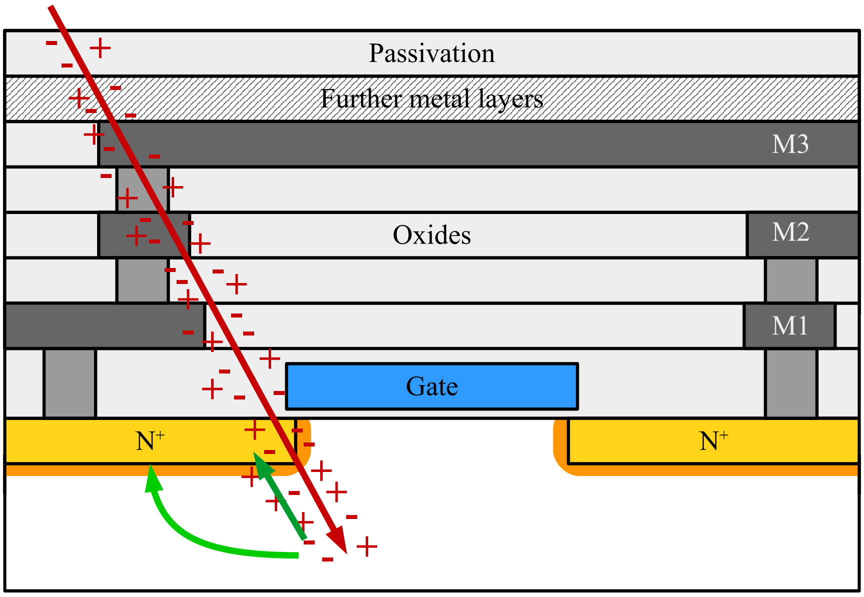

Figure 5.21: Example of single event upsets (SEU) in a silicon device.

Let's suppose to have the initial voltage value VDD on the net named IN and the complementary value GND on the net referred as OUT. Then, a radiation source deposits a charge at the drain of M1. A transient current occurs and the state of IN is temporarily changed from VDD to GND. Thus, the second inverter composed by the pair M3 and M4 evaluates this switch and changes accordingly the output value from GND to VDD. The new state of OUT in turn enforces a wrong state also on the net IN and the bit inversion remains stored in the memory cell. This type of system failure belongs to soft error class and its effect can be classified as *static* to distinguish it from the *transient* one caused by a Single Event

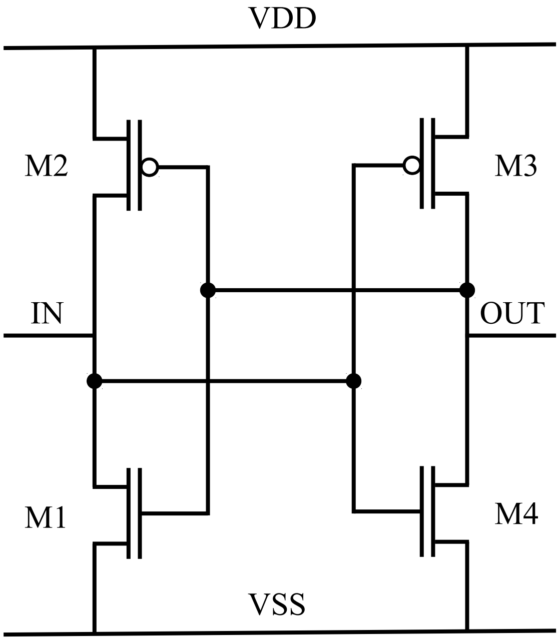

Figure 5.22: SRAM cell.

Transient (SET). A SET occurs when a charged particle passes through a sensitive node of a combinational logic block. This event has a distinct signature in polarity, amplitude of the waveform and its duration [187] depending both on the radiation source and on technological parameters as the doping. SEU is classified into three order effects by the number of upsets that take place. If a single bit inversion occurs, it is defined as first order effect. When Multiple Bit Upsets (MBU) happen in the circuit, they are categorized as second or third order effects. A MBU can be generated by a very tilted charged particle that ionizes two sensitive nodes at the same time. MBU in turn is classified into three categories. The first type describes a second order effect and it is represented by a single particle that strikes two near junctions, located in distinct memory cells. The second MBU case appears when a particle hits two adjacent nodes placed in the same memory unit. This event belongs to the third order class of effects. Lastly, when multiple particles deposit energy in different sensitive junctions and this results into multiple bit flip of memory elements, a third order class of effects occurs. This case is assimilable to a group of SEU that simultaneously strikes memory cells. The charge deposition can be approximated as a double exponential current pulse [188]:

$$I(t) = I_0\left(e^{-\frac{t}{\tau_\alpha}} - e^{-\frac{t}{\tau_\beta}}\right) \tag{5.3}$$

where I_0 represents the maximum current, while τ_α is the collection time constant of the junctions and τ_β is another time constant for initially set the ion track. This model was derived from device simulations where different types of junctions and charge collection mechanisms were considered [189]. The curve shows the shape of the charge collection considering the depletion and funneling regions, including the diffusion process. The funneling region is generated when the electron-hole pairs cross the depletion region around the junction which is characterized by a high field. The carriers cause a distortion of this field that is extended now along the track, occupying a field-free region. This new configuration of the field leads to a drift collection of the carriers in the track. The extension of this mechanism depends on the concentration of impurities into the substrate. A detailed analysis can be found in [190]. To evaluate the incidence of radiation into the material the energy and the flux of particles must be considered. The energy is measured in *rad*, where 1 rad is equal to 10^{-s} (J/s). The flux is expressed as the number of particles that pass an area of 1 cm^2 during 1 second. If the ionization event is particularly energetic, the electronics can show permanent damages as mentioned in Section 1.3.3. However, these types of errors are out of the scope of this thesis and they are not discussed here.

These sources of fault were predicted in the first years of 1960s and confirmed in the late 1970s for what concerns both space and terrestrial microelectronics. The first evidence of SEU due to cosmic rays interaction with electronics is dated 1975 and it involved satellite operations. On terrestrial applications, the alpha particles were recognized as the primary generator of errors at the ground level because they were emitted from ^{238}U impurities in the packaging of DRAM [191]. These soft errors were confirmed in later years with cumulative evidences and a well documented history can be found in [192].

5.2.2 SEU mitigation techniques

As the knowledge about the SEUs and their effects has increased during the years, many solutions have been proposed to detect and mitigate the problem [193]. Nowadays, the initial shielding protection that reduces the particle flux through the devices is not enough. Indeed, modern circuitries took benefit from the scaling process, power consumption reduction and increased clock speeds. However, the resulting reduced noise margin makes the recent very deep submicron technologies more sensitive to radiation influence. Thus, the development of more sofisticated fault-tolerant techniques becomes of interest for the scientific community and industry. Many approaches have been explored during years and they are strongly dependent on the target device. Three main classes can be defined: fabrication process-based techniques, design-based methods and recovery systems. The first category adopts solutions based for instance on epitaxial CMOS processes or silicon-on-insulator (SOI). The design-based systems are divided into two branches, where the first one collects all the detection techniques such as the hardware redundancy, the use of an error detection coding or self-checker approach. The second branch is based on mitigation techniques. Members of this sub-class are the Triple Modular Redundancy (TMR) with voters or hardened memory cell level. Lastly, the recovery systems are only applied to programmable logic and they account reconfiguration, partial configuration and rerouting design. Each approach can be more suitable than others depending on the application and the available resources in terms of area, power budget and performance. Some variants were also derived from these techniques as reported in [194] where a logic partition have been applied.

5.2.2.1 Triple Modular Redundancy implementation

For this work, a generic protection against radiation was adopted and the triple modular redundancy with triplicated voters was selected [195]. Because the core of the current digital signal processor is represented by its memory where the coefficients of the filters and the other configurable parameters are stored, the triplication involved these registers. Therefore, the first step concerned the development of a detection scheme. This circuit was simply accomplished comparing each triplicated register TMR_n with the other two and requiring that these three equalities were simultaneously fulfilled. Thus, the scheme followed for the detection of a changed register is shown in the following code snippet:

```
////////////////////////////////////////////////

    foreach(TMR_1[i]) begin
        if (TMR_1[0] == TMR_2[0] && \
            TMR_2[0] == TMR_3[0] && TMR_1[0] == TMR_3[0])
            matching_TMR_1[i] = 1'b0;
        else
            matching_TMR_1[i] = 1'b1;
    end

////////////////////////////////////////////////
```

where the foreach loop is performed on the registers named $TMR\_1[i]$ and the signals $matching\_TMR\_1[i]$ point out when an inequality occurs. This detection circuit is replicated also for TMR_2 and TMR_3. Once an inequality condition is recorded, the FSM reaches the dedicated $SEU\_RECOVERY\_MEMORY$ state for the correction. In Figure 5.23 the majority voter implemented for this purpose is reported.

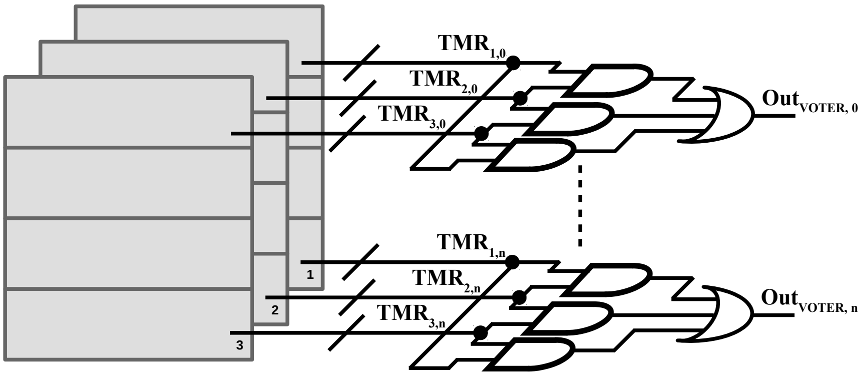

Figure 5.23: Majority voters for the memory recovery.

The recovery is realized with a combinatorial logic able to perform the bitwise

AND operation between the registers content combined with a final bitwise OR operator. The following code summarizes this description:

```
////////////////////////////////////////////////

//STATE SEU_RECOVERY_MEMORY

foreach (out_voter[i])
   out_voter[i] = (TMR_1[i] & TMR_2[i]) | \
                  (TMR_2[i] & TMR_3[i]) | (TMR_1[i] & TMR_3[i]);
end

////////////////////////////////////////////////
```

As mentioned before, also the majority voter is triplicated and the output provided by $out\_voter[i]$ is then used to rewrite the data stored in triplicated memories TMR_n. The physical implementation of this correction circuitry required some caution. Firstly, all the synthesis tools try to simplify the hardware modeled at RTL stage. This intrinsic operation makes sense because the designer have to focus the most part of efforts on the functionality and algorithmic efficiency of the design, leaving the CAD to manage often complex circuits. However, sometimes a fine control on this optimization process is required as the case of TMR. Hence, a *dont_touch* command was applied into the Tool Command Language (TCL) scripts to avoid the deleting of these nets. Unfortunately, this is not enough when the layout have to be generated. Even in this step, the dedicated tools try to optimize the timing performance by minimizing the space between connected cells. The corrector circuit introduced before is strictly connected through the triplicated voters and this means that the tool will try to place the cells near each other. If this placement improves the timing of the processor, at the same time it nullifies the protection against radiation. Indeed, a particle might generate multiple bit flips inside the core area of the design. If these inversions affect two locally close registers of the same memory slice, a correction is not possible anymore. A good strategy to overcome this placement-default behaviour is to define a bound, namely, an area in the floorplan of the design where to harvest all the cells of a certain type or with a specific name. The idea is depicted in Figure 5.24. The grey darker region represents the core of the design, while the three lighter squares are bounds. They are physically distantiated and each one contains registers beloging to just one triplicated memory, symbolized by the numerical subscript. Generally, the tools give detailed options to control these areas [196] [197] and in this work *move bounds* were created.

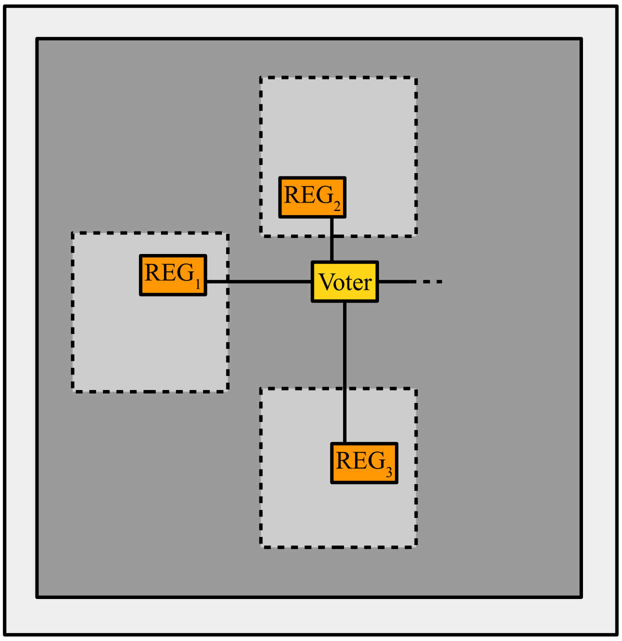

Figure 5.24: Floorplan partition into bounds for TMR implementation.

Thus, the standard cells grouped by their post synthesis netlist name have been constrained to be placed inside regions with specific coordinates to prevent any timing optimization. On the other hand, the possibility of placing also other cells into the bound areas was allowed. This choice was done to not overcomplicate the design, leaving the CAD free to optimize the timing inside and around the bounds.

5.2.2.2 SEU and MBU simulations

Finally, to test the obtained layout, some simulations were run. To simulate the ionizating particle strike the *force* command was used [198]. In particular, the *-deposit* option was adopted to temporary force the change and to allow the recovery of the stimulated cells. The inversion bits were introduced both at the input and at the output nodes of the triplicated flip flops. The system is successfully able to detect the changed memory content and to correct it by restoring back the original data. More in detail, the refreshing circuit can identify and adjust SEU, MBU of second order effect and MBU of third type when they indifferently interest one of the triplicated memory unit TMR_n.

5.2.3 Simulation results

For both the technology targets adequate testbenches were prepared to stimulate and gather the outcomes from the simulated chips in post P&R typical corner. The simulation tool allowed to test all the tasks presented in the overview Section 4.3: by starting from the ideal state, the FSM takes care of the coefficients uploading required for the successive processing. Then, the average computation of the noise takes place, followed by its square root derivation through the bisection algorithm. At this point the FSM is ready to manage the main signal processing by feeding the CR-RC^4 chain to apply later the MWD in order to extract the energy information from the incoming data. At the same time, the charge is computed as well as the timing information is obtained with the CFD. Finally, the pile-up rejection system is enabled to prevent undesidered effects on the output data. In addition, as explained in the previous sections, a TMR module restores the memory content in case of particle interactions. Most of these processes are not particularly interesting since they involve just circular buffers and accumulators to realize addition and multiplication operations. However, it could be more stimulating to inspect the implementation of two specific algorithms. Thus, the RTL simulation of the bisection algorithm discussed in Section 4.4.2 is depicted in Figure 5.25. The passages described in the following were also verified with post P&R simulations of the two technologies with a clock period of 20 ns. The signal *signal_to_root* presents to the combinatorial logic the value to be rooted, 6561 in the reported example. The register *counter_accuracy* keeps a record of the number of iterations. When its value is equal to zero and the FSM enables the square root computation, the initialization of the memories named *lower* and *upper* occurs (red dashed line).

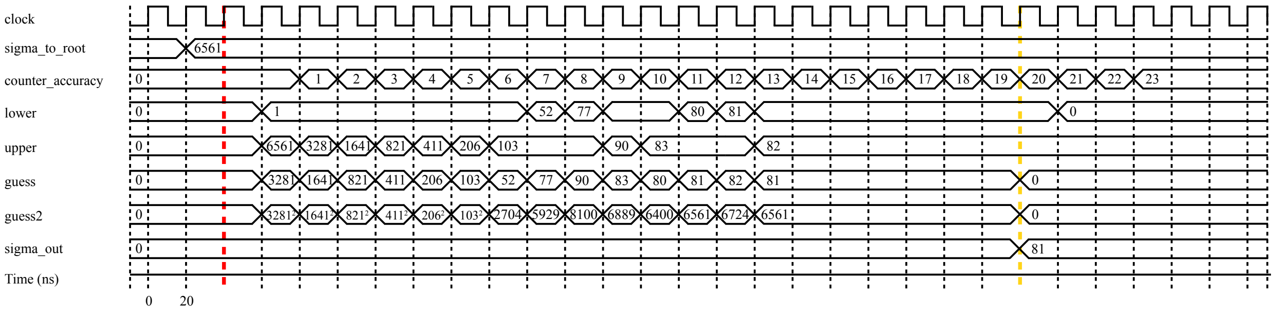

Figure 5.25: Timing diagram of a behavioural simulation of the bisection algorithm for the square root computation.

In particular, the smaller value, which is equal to 1, is assigned to *lower* while the data contained into *signal_to_root* is stored into *upper*. Consequently, the iteration starts and at each clock cycle the value of *guess*, its squared value *guess*2 and both *lower* and *upper* themself are updated. The process continues until a pre-uploaded value for the counter is achieved, highlighted with the yellow dashed line in the picture. During this process, the reader can appreciate the gradual convergence towards the final quantity. At this stage the value of *guess* is truncated and a 12-bits data word, called *sigma_out* in the image, is sent out from the module. In the illustrated example the accuracy of the algorithm can be proved by computing the square root of 6561 which is properly 81. The bit-width of each register was dimensioned to avoid miscalculation due to the approximation. The other fascinating on-chip application is represented by the implementation of the Goldschmit's algorithm introduced in Section 4.9.6. Even the example depicted in Figure 5.26 illustrates a RTL simulation and as for the bisection algorithm its regularity was tested with a post P&R check.

The latter returns the same results of the behavioral simulation.

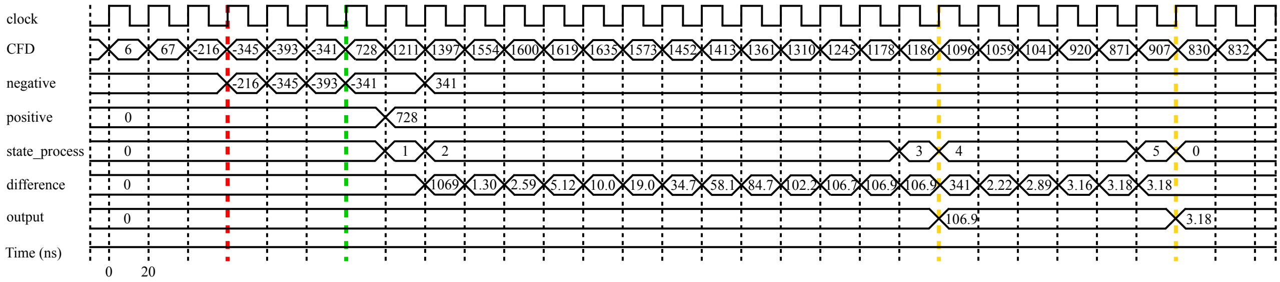

Figure 5.26: Timing diagram of a behavioural simulation of the Goldschmit's algorithm for the implementation of the division.

In Figure 5.26 the clock is set to 50 MHz and at its posedge the CFD output is refreshed. Similarly to the bisection case, a register named $state\_process$ manages the computing stages of the method. When it is 0, the $negative$ register is continuously updated if the CFD output is negative. This step corresponds to the interval between the red dashed line and the green one. If a positive CFD outcome is presented at the input and its value is above a configurable threshold, the refreshing of $negative$ memory is stopped, holding the last stored value. The content of $positive$ is also updated with the detected data and a pulse signal is generated. In the reported example, the threshold was set to zero, thus the positive word recorded was the first one available. When the $state\_process$ is equal to 1, the circuitry is initializated, while at the state 2 it begins to compute the angular coefficient referred as m. If the convergence is achieved into a configurable number of steps, the unit sends out the computed value, going on with the process, otherwise the state returns to zero to avoid dangerous loops. After this stage the block takes advantage of m derivation to calculate the zero crossing time. Hence, the state 4 is reserved to this task and the value is obtained with the same procedure. The previous counter is exploited to prevent another potential unwanted loop.

These algorithms and the other implemented features discussed in Chapter 4 have been tested on two different configurations with post P&R typical corner simulations. In the first one, 1000 noisy pulses with the same nominal height were provided to the DSP module. These values are biased with a time-variable offset and they were extracted by a Python script implemented from scratch. This program simulated the outcome of a 12-bits SAR ADC running at 50 MHz. To model a realistic DAC, a 15% random capacitor mismatch has been considered in the data generation. Furthermore, the RMS quantization error V_q described in Section 2.1, thermal noise V_{th} introduced in Section 2.2 and lastly the jitter contribution defined in Section 2.3 have been taken into account as well in the script. The data coming from the CR-RC^4, the trapezoidal filter and the energy output were collected and analyzed to compute the final SNR. Three pulses of this set were reported in Figure 5.27. The reader can appreciate the baseline correction on the CR-RC^4 output by observing the flatness of the processed data. In the magnified area the plateau formed by the MWD manipulation was enlarged. The blue step represents the energy extracted after the averaging of 4 trapezoidal outputs once the plateau was reached. The energy output is held until the MWD value returns below a configurable threshold.

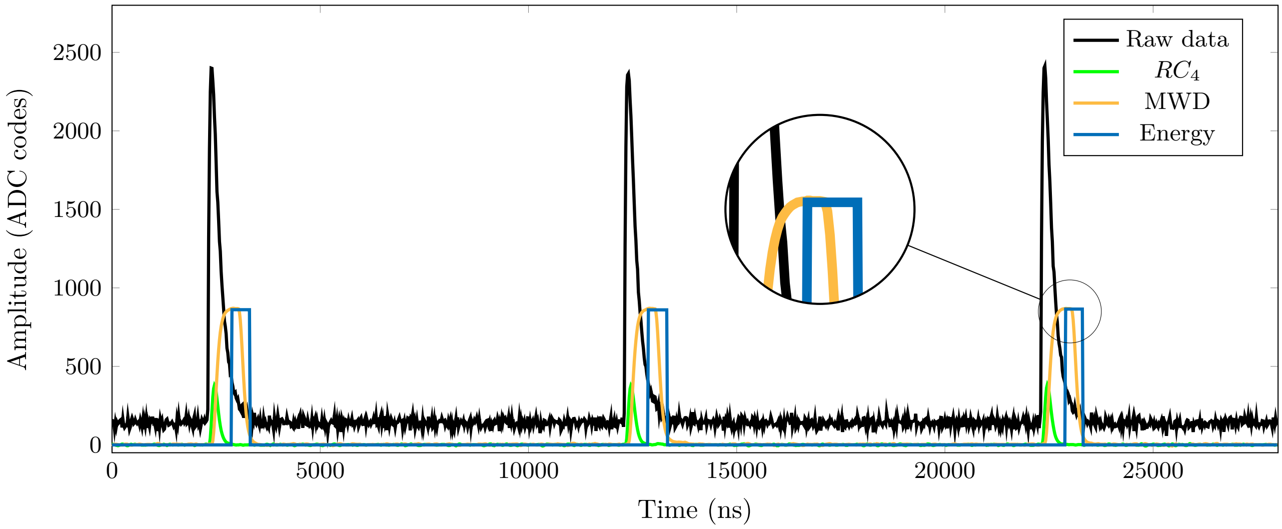

Figure 5.27: Example of DSP elaboration of pulse signals after the CR-RC^4 filtering (green) and the MWD processing (orange) pipelined with a dedicated energy extraction circuit (blue).

The stored outcomes of the CR-RC^4 chain were elaborated in the same way presented in Section 4.7.4. For the trapezoidal signal instead a single top value of each energy step was acquired and the RMS of the signal was calculated on this dataset. The RMS noise was derived by considering an equal number of points collected from the MWD baseline. The results of 1000 analyzed samples are reported in Table 5.12 where the first row recalls the raw data evaluation shown in Table 4.5. The SNR of the pulse shaper is slightly smaller than the value 183.44 of the previous Python analysis. This discrepancy could be attributable to two factors: the first one concerns the mere approximation used in the physical implementation. In fact, the CR-RC^4 output is formed by only 12-bits and an additional one is reserved to the sign. By contrast, the Python simulation used to initially evaluate the SNR of the same chain exploits 64-bits. A second reason is due to an inherent difference in the data processing between the post P&R simulations and that one carried out with Python. The latter used a simplified version where the baseline is not continuously updated, but its value is freezed at the beginning of the script. Conversely, the final chip design takes care of the baseline fluctuations and the circuit corrects them sample by sample. Although the SNR of the pulse shaper is a little bit smaller than expected, the final result is fully satisfactory by looking at the ratio of the MWD. Actually, the acronym MWD in Table 5.12 indicates the pipelined circuitry composed by the MWD filter coupled with the energy extraction module. Thus, at the end of the signal processing, the chip is able to increase by 16 times the initial SNR, ensuring a SNR of 48.93 dB.

Stage	SNR	SNR_{dB}	F_{SNR}
RAW	17.24	24.73	1
RC^4	170.18	44.62	9.87
MWD	279.52	48.93	16.22

Table 5.12: Post P&R SNR values for RC^4 and MWD coupled with the energy extraction circuitry.

The second simulation proved the ability both of the CFD method and the Goldschmit's algorithm to detect the zero crossing time with a precision below 1 ns, if properly configured. For this purpose, another set of 1000 pulses of different amplitudes fed the DSP chip. The minimum generated amplitude was around 1700 ADC codes, while the maximum one was of 3045 ADC codes. The level of noise injected was the same of the previous simulation. The post P&R simulation for the CFD block is reported in Figure 5.28 where the orange signal represents the output of the CFD circuit.

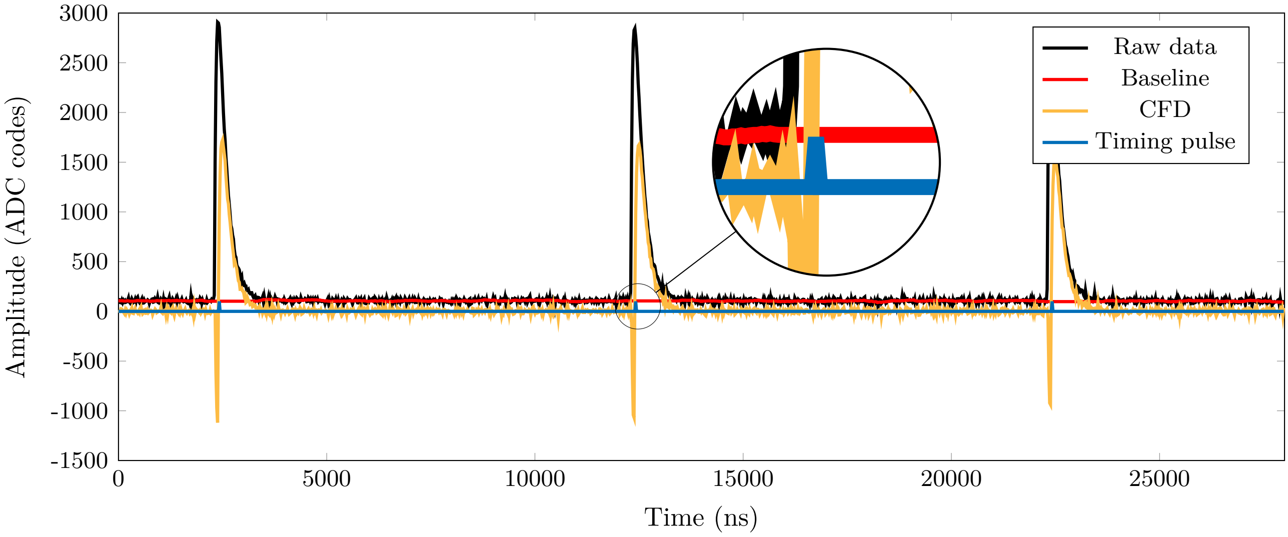

Figure 5.28: Example of CFD filtering with baseline correction and the pulse generated for signals of different amplitudes.

The sign changing of the function when a pulse is detected on the raw data is easily recognizable. A step is contextually genereted in order to reconstruct the event even offline (it is intentionally enlarged in the magnified area for visualization purposes). The continuously updated baseline is also depicted in the image. This is maintained constant inside the acquisition window of the signal which is detected with two programmable thresholds as explained in Section 4.4. This value is then used to correct the CFD output and its baseline restoring is effective to cancel both the injected offset and the noisy modulations. From the simulation two set of data composed by angular coefficients and time zero points were also collected. Each one is expressed as 33-bits long word, where the first bit is reserved to the sign, followed by 12-bits slice assigned to the integer part and the last 20-bits part used for the decimal component. On this data structure the average value of the 1000 time zero points was computed, then their standard deviation was calculated. To investigate the robustness of the method, the nominal rising time $\tau_r = 50 \cdot 10^{-9}$ was additionally randomized pulse-by-pulse besides the injected noisy background of the previous simulation. Table 5.13 summarizes the results where the first column indicates the window of randomization for each pulse and the second one lists the average values of the cross points measured in ns. For these values the zero time reference were defined by the negative value whose occurrence is externally flagged by the step signal. Lastly, the last column shows the standard deviations σ_{zp}.

Δ_{τ_r} ($\pm$%)	$\overline{Zeropoint}$ (ns)	σ_{zp} (ns)
0	10.32	±0.64
5	10.29	±0.69
10	10.25	±0.70
15	10.23	±0.88
20	10.23	±0.84
25	10.16	±1.08
30	10.02	±1.64

Table 5.13: Timing resolution obtained with the CFD method for different rising times and amplitudes.

The best result is (10.32 ± 0.64) ns, hence when no further perturbation on the rising edges are introduced. However, the CFD method is strong enough to tolerate a 20% of randomization without overcoming the 1 ns of resolution. The non-total monotonicity of σ_{zp} is due to the intrinsic differences between the generated datasets. For instance, each simulation is characterized by its own baseline fluctuations and noise contributions. Thus, under this conditions, the circuit is able to resolve both the time walk effect that arises when signal amplitude is not constant and the rising time variations within a time resolution below 1 ns.

5.2.4 Power analysis

Finally, the DSP circuit performance in terms of power consumption was investigated. Different tasks performed by the digital processor were considered: the initial average computation, the sigma derivation, the baseline definition and the pulse processing were inspected both on their static and dynamic components. The latter examination takes into account the same two most energy-intensive intervals, one for the average and sigma, the other one for the baseline and pulse elaborations. These two couplings are due to the order of execution of the tasks. Indeed, the average and sigma calculations share the same toggle rate because they are activated earlier and neither the baseline elaboration, nor the pulse signal processing are still enabled at those points. On the other hand, the static dissipation records these variations and it rises progressively from the average to the pulse elaboration of the signal. It was expected because more and more circuitries are involved from the initial idle state to the full final computation.

Power	Average Static (mW)	(%)	Sigma Static (mW)	(%)	Baseline Static (mW)	(%)	Pulse Static (mW)	(%)	Pulse Dynamic (mW)	(%)
Internal	1.9883	82.42	2.1316	80.77	2.7301	77.03	2.9016	77.46	4.1207	76.94
Switching	0.4191	17.37	0.5026	19.04	0.8090	22.83	0.8390	22.40	1.2296	22.96
Leakage	0.0050	0.21	0.0050	0.19	0.0051	0.14	0.0051	0.14	0.0051	0.10
Total	2.4124		2.6392		3.5442		3.7457		5.3554	

Table 5.14: Static and dynamic power budget of the DSP chip during different tasks.

For this reason, the static power of all the listed tasks are reported in Table 5.14 while the dynamic slice takes into account slightly more than 17,000 ns during the pulse processing operation. In this time window the chip processed 10 pulses of variable amplitude and different time distribution to mimic a realistic condition.

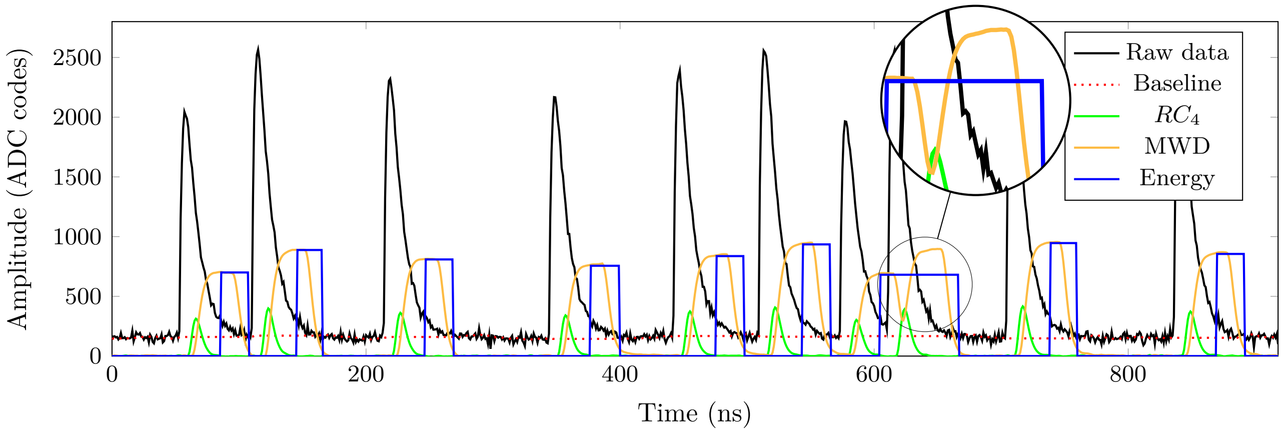

Figure 5.29: Example of DSP elaboration of pulse signals.

These signals are also biased with an offset and they are depicted in Figure 5.29 where the magnified area shows the activation of the pile-up rejection system. In fact, the pulse located at 600 ns is correctly processed while the next one is rejected. This fact is visible in the energy level (blue) which is not updated. Although the chip was still able to process this pile-up event, the feature was enabled to include it in the power evaluation. These values are referred to the 9-metals 65 nm CMOS technology running at 50 MHz clock and by considering the typical corner simulation. The average, sigma and baseline power consumptions were computed considering a time window of 200 ns. In Tables 5.15, 5.16, 5.17 and 5.18 the static power distribution changes from the initial state towards the full operativity in terms of computation. In particular, it is significant to take into account the combinational percentages. During the average value derivation no other module is directly activated, thus the value is around 30%.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	1.432	0.0050	0.0010	1.438	59.59
IO	0	0	$4.4 \cdot 10^{-8}$	$4.4 \cdot 10^{-8}$	$1.8 \cdot 10^{-6}$
Combinational	0.4705	0.2551	0.0038	0.7295	30.24
Clock (Combinational)	0.0863	0.1589	$4.4 \cdot 10^{-5}$	0.2452	10.17
Total	1.988	0.4191	0.0050	2.412	100

Table 5.15: Static power distribution expressed in mW computing the average value.

Once the average value is obtained, the FSM passes to the sigma elaboration and the same power group rises up to 35%. It was expected since in this task two subtractions and one multiplication are involved, while by contrast the average operation is carried out with a single addition into the accumulator register.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	1.451	0.0064	0.0010	1.458	55.26
IO	0	0	$4.4 \cdot 10^{-8}$	$4.4 \cdot 10^{-8}$	$1.6 \cdot 10^{-6}$
Combinational	0.5943	0.3372	0.0038	0.9354	35.44
Clock (Combinational)	0.0863	0.1589	$4.4 \cdot 10^{-5}$	0.2452	9.291
Total	2.132	0.5026	0.0050	2.639	100

Table 5.16: Static power derived for the sigma computation. Unit expressed in mW.

As mentioned before, the baseline task marks the first step into the full processing state. Basically, the combinational power around 49% is calculated when all the processes dedicated to the data filtering are activated. This means that the CR-RC^4 chain, the further trapezoidal deconvolution coupled with the energy extraction, the charge accumulation and the data manipulation by adopting the CFD approach, including the pile-up rejection system, take part to this component.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	1.533	0.0121	0.0010	1.547	43.64
IO	0	0	$4.4 \cdot 10^{-8}$	$4.4 \cdot 10^{-8}$	$1.2 \cdot 10^{-6}$
Combinational	1.11	0.638	0.0038	1.752	49.44
Clock (Combinational)	0.0863	0.1589	$4.4 \cdot 10^{-5}$	0.2452	6.919
Total	2.73	0.809	0.0050	3.544	100

Table 5.17: Static power annotated for the baseline task and measured in mW.

The percentages variation between Table 5.17 and Table 5.18 where the combinational power is slightly more than 54% is to ascribe to the average toggle rate during the pulse processing.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	1.529	0.0119	0.0010	1.542	41.18
IO	0	0	$4.4 \cdot 10^{-8}$	$4.4 \cdot 10^{-8}$	$1.2 \cdot 10^{-6}$
Combinational	1.286	0.6682	0.0039	1.958	52.27
Clock (Combinational)	0.0863	0.1589	$4.4 \cdot 10^{-5}$	0.2452	6.547
Total	2.902	0.839	0.0051	3.746	100

Table 5.18: Static power contributions expressed in mW during the pulses processing task.

Power group	Internal	Switching	Leakage	Total	Percentage
Sequential	1.559	0.0157	0.0010	1.576	29.42
IO	0	0	$4.4 \cdot 10^{-8}$	$4.4 \cdot 10^{-8}$	$8.3 \cdot 10^{-7}$
Combinational	2.475	1.058	0.0041	3.538	66.06
Clock (Combinational)	0.0863	0.1558	$4.4 \cdot 10^{-5}$	0.2422	4.522
Total	4.121	1.23	0.0051	5.355	100

Table 5.19: Dynamic power contributions during the elaboration of a pulse signal. The power is evaluated in unit of mW.

Finally, Table 5.19 reports the values for the dynamic power where the total power budget is bound to slightly more than 5.3 mW. The larger power consumption is again required by the combinational group, followed by the sequential slice. These values could be further reduced by enabling a zero rejection circuitry to send out only significant data. This feature was not implemented in the current version of this DSP chip because the intention was to explore the power performance of the presented filters in their worst case.

Even for this digital block some heatmaps were generated. In Figure 5.30 a sort of tilted T-shape is recognizable and it is clearly defined by sharp borders characterized by the lower internal power. This area is mainly populated by filler cells and logic used to refresh memories. This apparent void is required to physically separate the tripled registers as discussed in Section 5.2.2.1 and illustrated in Figure 5.2.2.2. The almost blue pattern is due to the absence of stimuli of simulated particles.

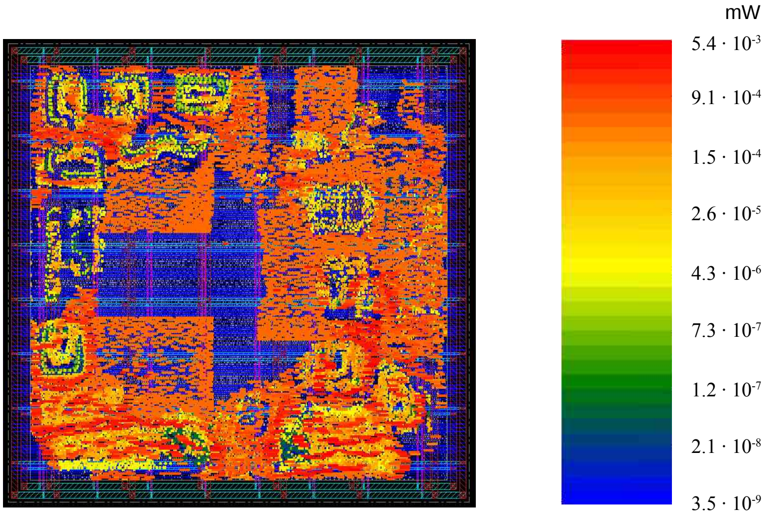

Figure 5.30: Heatmap of the internal power for the DSP chip.

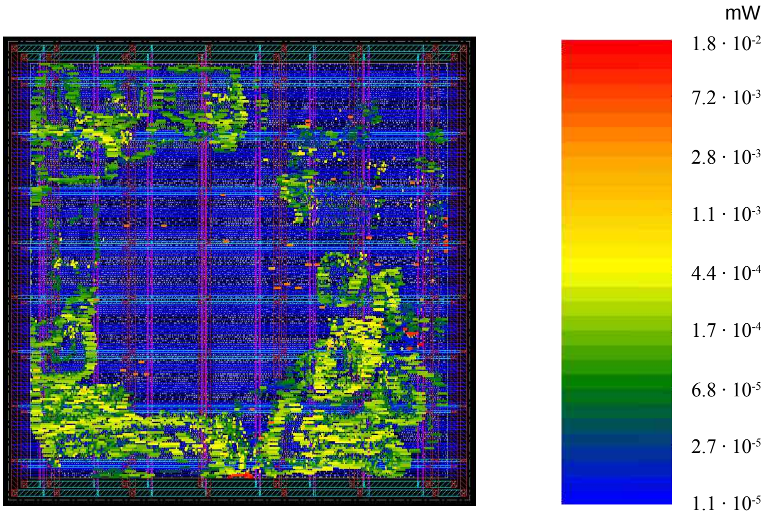

Figure 5.31: Heatmap of the switching power derived by the dynamic analysis.

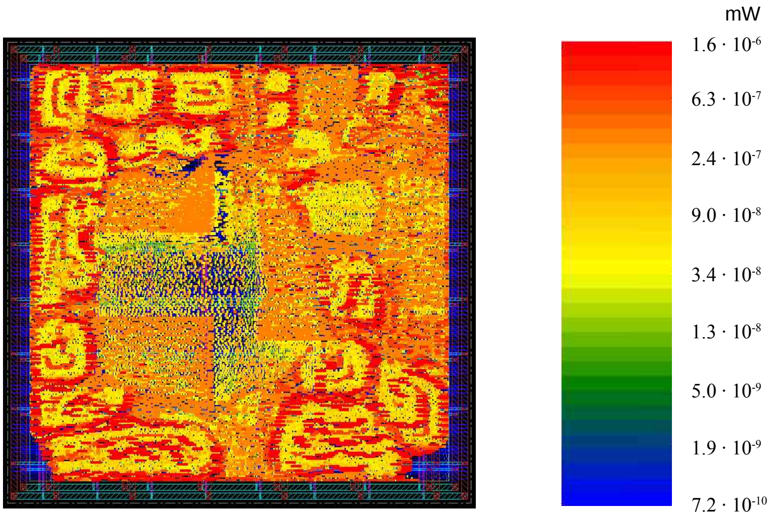

Figure 5.32: Heatmap of the leakage power accounted for the standard cells.

The total power of the standard cells summarized in Table 5.19 is illustrated in Figure 5.33. Figure 5.34 depicts the total power density. The heatmap represented in Figure 5.35 shows the IR drop across the chip area by considering a voltage source VDD set to 1.2 V. In general, the distribution appears flat, but some spots are characterized by a larger value compared to the average trend. In particular, the most critical point is located at the bottom of the design. A more detailed inspection figured out that the involved cells belong to the output of the CFD filter. However, these variations do not invalidate the functionality of the timing block.

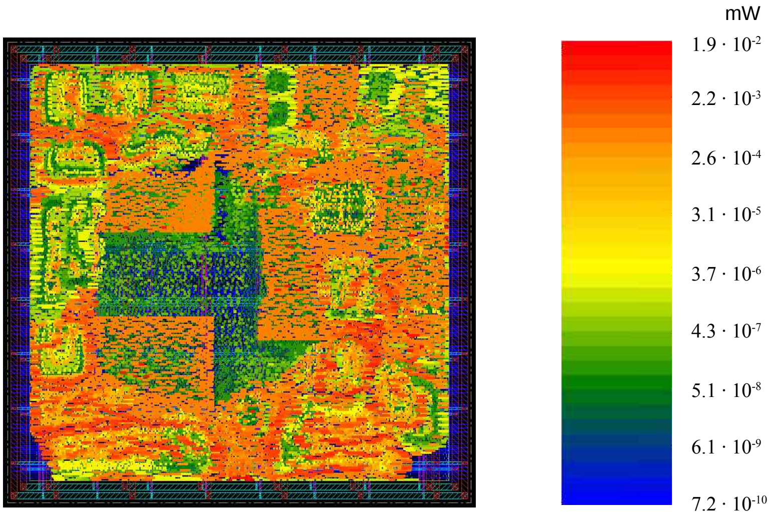

Figure 5.33: Heatmap of the total power of the DSP block.

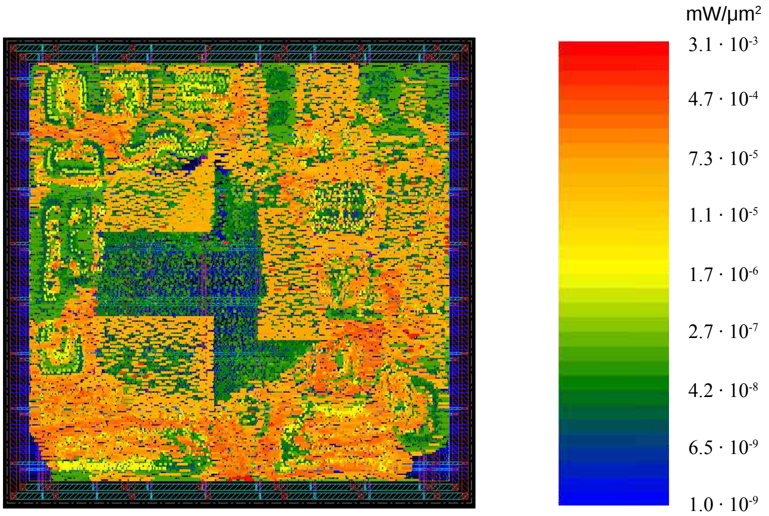

Figure 5.34: Heatmap of the total power density for the whole DSP chip.

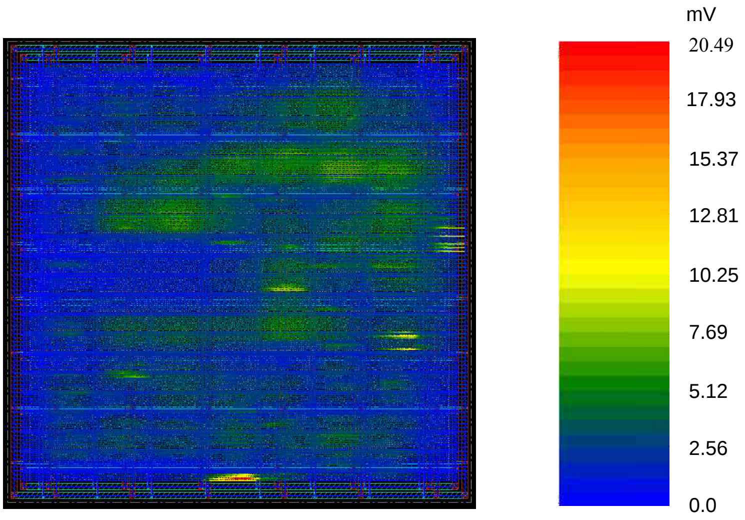

Figure 5.35: Heatmap of the IR drop. The voltage source VDD is equal to 1.2 V.

Similar considerations can be carried out for the rightmost regions where some differences are recorded. These areas are managed by the trapezoidal filter and the standard cells interested are some flip-flops of the dedicated circular buffer, some logic and register of the reserved accumulator and a couple of flip-flops to obtain the final outcome. Even in this case, the higher IR drop does not negatively affect the performance of the filter. Therefore, the power distribution among the several blocks reported in the pie chart of Figure 5.36 was quite expected. The most power consuming module is the CFD block which is justified by the complex calculations. In fact, they involve the discriminator and the division implemented with the Goldschmit's algorithm as explained in Section 4.9.6. The CR-RC^4 module requires less power, nevertheless the number of operations is not trivial. Hence, the pipelined architecture seems to be a reasonable design choice to contain its power consumption. The trapezoidal module should be coupled with the energy one because they formed another pipelined structure. The budget for the MWD manipulation is far higher than the one accounted for the energy extraction. Even this variation is expected since onerous cumulative additions are carried out in the trapezoidal filter.

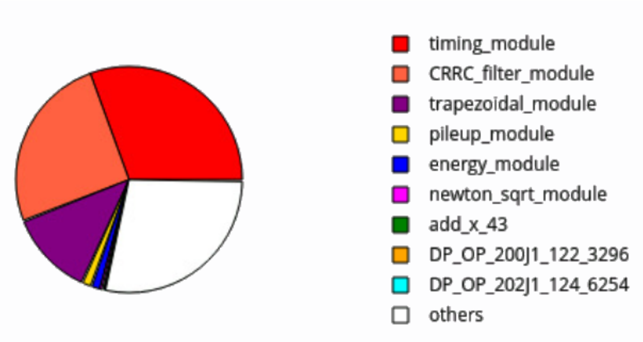

Figure 5.36: Pie chart of the power budget distribution among the sub-modules of the DSP chip.

Chapter 6

Data compression

Data compression represents an important branch of the digital signal processing. This term collects all those techniques that reduce the original data size during memory allocation and the algorithms developed to efficiently exploit the bandwidth during the data transmission. At the same time, this category includes also the fundamental methods to reconstruct the compressed data. The reduction of information can be also realized without a proper compression scheme. For instance, the occupancy expected in ALICE was low, thus the development of ALTRO and its heirs adopted a zero-suppression system to achieve an initial data reduction. This approach can be classified as the early stage of data formatting rather than a proper data compression.

The compression algorithms are divided into two groups referred as *lossy* and *lossless*. As their name suggest, the difference is about the loss of information. In the first set of techniques a lack of the exact reconstruction of the original data is expected. This situation is not critical in some applications such as transmitting speech because a precise reproduction of that sound is not required and a distortion is assumed acceptable. On the other hand, this compromise is widely compensated by general higher compression ratios than the lossless counterpart. By contrast, the second class of algorithms does not allow any loss of information. For example, let's consider a medical application as the PET where an accurate reconstruction of the radioactive tracker distribution inside the patient is fundamental. In this case a deficiency of information that may prevent a correct diagnosis is totally unacceptable. From this point, only the lossless group will be taken into account since in nuclear and particle physics experiments the loss of information cannot be tolerated. Thus, how do you quantify the performance of a compression scheme? At first glance one could be focused only on the amount of compression, but many factors contribute to a global evaluation. In fact, the data similarity after the reconstruction process must be also considered as well as the general complexity of the employed algorithm and the required memory to store it. Additionally, the processing speed plays an important role too. All these aspects become particularly significant for an on-chip implementation. A quantitative measurement of the performance achieved with a given method is the *compression ratio* (CR) which is defined as:

$$CR = \frac{S_A}{S_B} \tag{6.1}$$

where S_A is the size of a bitstream after the compression and S_B is the original data before the compression occurred. The reverse of this quantity is named *compression factor*, while to estimate the saving percentage (SP):

$$SP = \frac{S_B - S_A}{S_B} \tag{6.2}$$

6.1 Compression techniques

Many compression algorithms have been developed over the years and they are optimized for specific tasks. For an exaustive list of these techniques a self-explained title is reported in bibliography [199]. Even the particle physics field benefits of compression schemes, especially when the data reduction represents the only feasible way to send out or to store the data during the experiments. An example is shown in [200] where once again the TPC of the ALICE is involved. The expected rate of events for the experiment was estimated around 300 Hz with an event size of 66 Mbyte by enabling the zero-suppression scheme. Therefore, the maximum data rate was simply the multiplication of these two quantities which is close to $\sim$ 20 Gbyte/s. This would have been an impressive amount of data and the issue to store it would have not been trivial. For this reason the zero-suppression circuit was developed. Although the data reduction was estimated around a remarkable 80%, this was not enough and a compression algorithm was implemented and tested on FPGA. The method took advantage of three techniques named vector quantization, delta calculation and Huffman coding. The maximum reduction obtained by combining these three steps was 38% at 40 MHz.

In the following sections a selection of three popular and effective algorithms useful to reduce the size of a data stream will be presented. The last section will be reserved to discuss an application of these methods on a dataset generated for the CMS experiment at CERN. This activity have to be considered as complementary work of the thesis due to its preliminary nature.

6.1.1 Run-Length Encoding

The first method presented is named *Run-Length Encoding* (RLE) and it is based on the idea of replacement. Let's suppose that a given data d is repeated n times in a stream. The consecutive replicated word is usually referred as *run*. In this case it is convenient to replace the recurring values with the shorter nd packet. For instance, in the string TMEEEEERAAAAS two runs can be identified, the first one of length 5 on symbol E and the second one of length 4 on the symbol A. Then, this stream can be encoded as TM§5ER§4AS where § defines a special character to flag when a recurrence happens. In this trivial example the CR is 0.69 and the saving percentage is 30%. The lossless nature of this scheme is quite evident once the reduced string is decompressed. The stream is read character by character until a marker § is found. Hence, the following number is recorded and the consecutive letter is repeated a number of times equal to that number.

6.1.2 Huffman tree

The second compression technique introduced is called *Huffman coding* [201] [202] [203] and it is based on the knowledge of the frequency distribution for a given set of values. To get the idea, let's consider the number of occurrences in a given language. For instance, in English the letters E, T, A, O, I are remarkably most frequent than K, X, Q, J, Z [204]. If a text compression is required, this asymmetrical distribution can be exploited by assigning shorter codes to the letters more frequently used and longer codes to those letters less utilized. Thus, Huffman coding is a statistical method, belonging to the fixed-to-variable class. This means that each fixed length symbol is associated with a prefix code of variable size. The following example will be focused on the binary coding for simplicity. Therefore, we take into account the ensemble reported in Table 6.1.

Symbol	T	M	E	R	A	S
Frequency	11	1	5	7	1	13

Table 6.1: Ensemble of six symbols and their occurrences.

Here six simbols with their occurrences are collected. The first step is to sort the group, obtaining the Table 6.2. Now it is possible to build the so termed *Huffman tree* as follow. The algorithm starts by picking up the two elements of the group which are less frequent (A and M in our case).

Symbol	S	T	R	E	A	M
Frequency	13	11	7	5	1	1

Table 6.2: Ensemble of six symbols and their occurrences.

Then, they are combined into a new node whose frequency is the sum of A and M occurrencies. This new element, often named *node* or *leaf*, is included into the original ensemble and it substitutes both A and M. The rightmost branch that connects the generated node with the child is assigned 1, while the other branch is marked with 0. This assignment will be also replicated for the successive steps. The explained passage is realized in the panel P_0 of Figure 6.1 where the new node is highlighted with a red ring around it. Afterward, the same operation is repeated among the remaining elements, thus E is selected and another leaf of the tree takes place with the frequency equals to 7. The process is iterated also in P_2 with the creation of an additional node with a doubled occurrence.

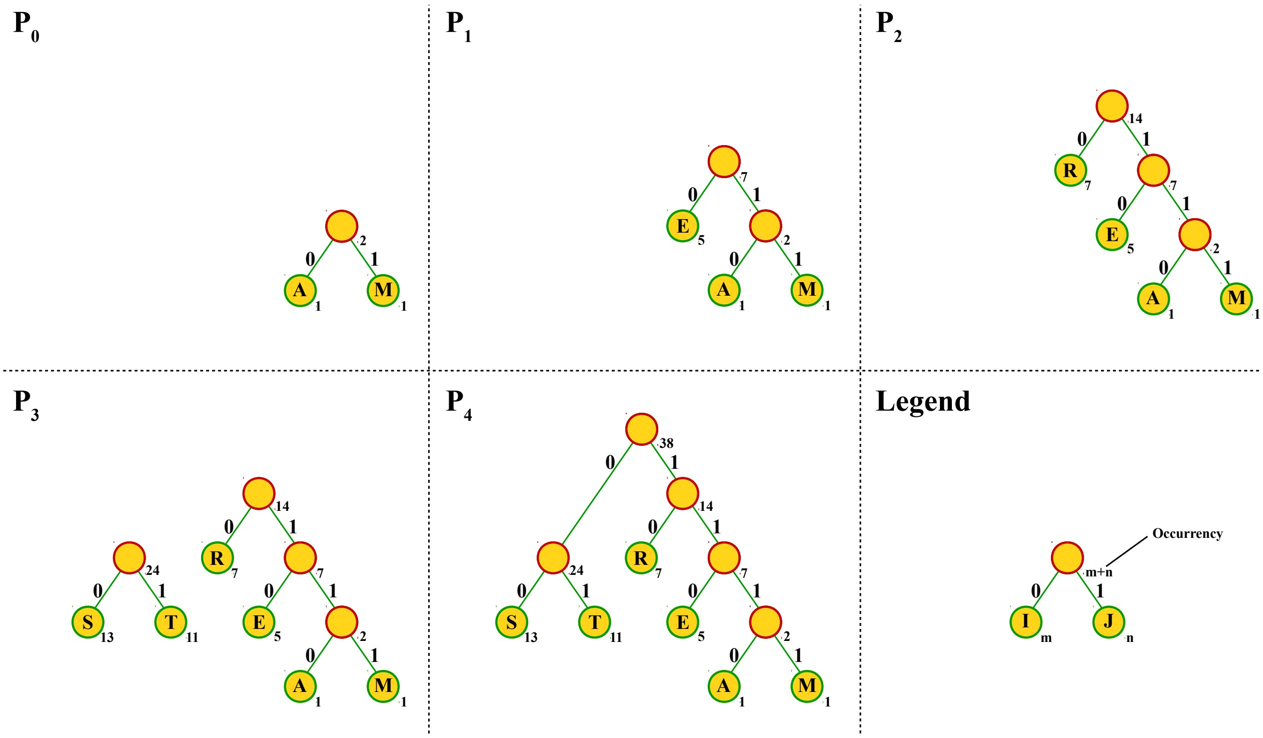

Figure 6.1: Huffman tree for the ensemble {S,T,R,E,A,M} with the occurencies {13,11,7,5,1,1}.

At this stage the ensemble is composed by S, T and the last created node which represents all the branches built up to now. Let's have a look at their frequencies that are 13, 11 and 14, respectively. In this case the frequency of the node is larger than those ones associated with S and T. So, the leaf can not be combined neither with S nor with T and in the box P_3 a new branch is generated. Finally, in P_4 the full Huffman tree is realized by combining the two last nodes into the higher one denoted with the frequency equals to 38. The last panel contains only a legend on the general core of the described process. In order to assign at each symbol a variable length code it is sufficient to start from the higher node and to point to the desidered symbol. The 0s and 1s met along the path are the sequence assigned to that symbol. Table 6.3 summarizes the assignment obtained in this way with the current ensemble.

Symbol	Frequency	Huffman code
S	13	00
T	11	01
R	7	10
E	5	110
A	1	1110
M	1	1111

Table 6.3: Ensemble of six symbols and their occurrences.

These codes are unique for each symbol and no ambiguity can occur. To prove it, the sequence MSSRT is considered. Given those Huffman codes, the string is converted by an encoder resulting into the bit stream 111100001001. The decoder accomplishes the reconstruction process bit-by-bit by examining the internal dictionary. If the input sequence corresponds to a Huffman code the symbol is recognized. The first bit of the string is equal to 1 and it is stored into a variable length array, but there is not a single bit associated with any symbol into the memory. Thus, the decoder appends the second bit which is another 1. Even in this case the interal research does not return any symbol. The process is iterated until the fourth 1 is added. The word 1111 is assigned to M and the first symbol is correctly reconstructed. The array is flushed and another cycle is enabled. The following bit is a 0 and once again there is not a valid symbol to associate with. After the appending of the next 0, the letter S is sent out without uncertainty. The same instructions are repeated for the other symbols and the original string is obtained. For what concerns the compression performance of this algorithm, it is easy to find that at least 3 bits are required to represent those 6 elements in the raw case. The total frequency is equal to 38, so the uncompressed original stream is 114 bits long. To figure out the benefit of the method each frequency must be multiplied by the bit-length of the corresponding Huffman code. The sum of this products results in 85 bits. Therefore, the compression ratio in this case is 0.75, while the saving percentage equals 25%. As expected, the Huffman coding is not particularly effective since only six symbols are considered, however also in the current example an appreciable data reduction can be observed.

This last paragraph is dedicated to some preliminary considerations about the possibility to apply the Huffman coding to the DSP output. In particular, the MWD outcome was considered as field investigation. Thus, the first step is to obtain a histogram of the collected codes to analyze their distribution along the range. This study adopts the same data set used for the CFD evaluation described in Section 5.2.3. This choice is due to the wide variation in amplitude of the raw signal which is a quite realistic condition. The histogram illustrated in Figure 6.2 provides a first useful information about the benefit of a zero-suppression which is not currently implemented. In fact, the lower values that were cut off in the plot form basically the baseline of the MWD filter and they can be rejected. The interesting data are included between the threshold (the red dashed line) and the code 1200. This interval defines almost 700 codes that should be represented with an equivalent number of Huffman codes. Such a large dictionary is certainly not feasible for each channel, but it could be integrated at a higher hierarchy level by sharing it among the channels. However, a further significant reduction in the size of the dictionary is possible with the approach described in the next section.

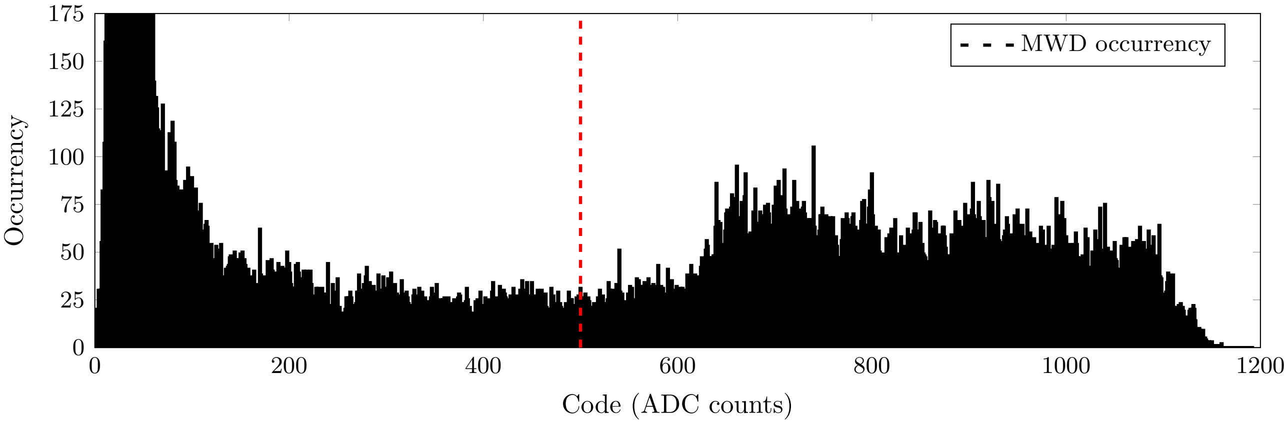

Figure 6.2: MWD codes distribution.

6.1.3 Delta encoding

If the samples are quite similar to each other or they difference is small a *delta encoding* can be employed. It is also called *relative encoding* or *differencing* [205] and as its name suggested it is based on the difference between data. Therefore, let's suppose to compress the stream {113, 117, 107, 120, 115, 111}. The encoding machine will produce the output {113, 4, -6, 7, 2, -2} where the first element is unchanged since it is used as reference for the compressed sequence. In this example, the original number of bits required to represent each component of the string was

7. After the compression this number is scaled down to only 4 bits since 3 bits are spent to the magnitude and the MSB is reserved to the sign. The first word is naturally excluded from this reduction. Thus, if the initial size of the stream was 42 bits, after the compression the sequence is reduced to 27 bits. In this case the CR is 0.64 with a saving percentage around 35%.
This technique becomes interesting also in the case of the trapezoidal input. The consideration comes from a close look at the output of this filter. Indeed, after the rising edge, the signal reaches a plateau, then it returns to the baseline. If we extract only the codes above the threshold defined in Figure 6.2, the reduced interval of amplitudes reported in Figure 6.3 is obtained.

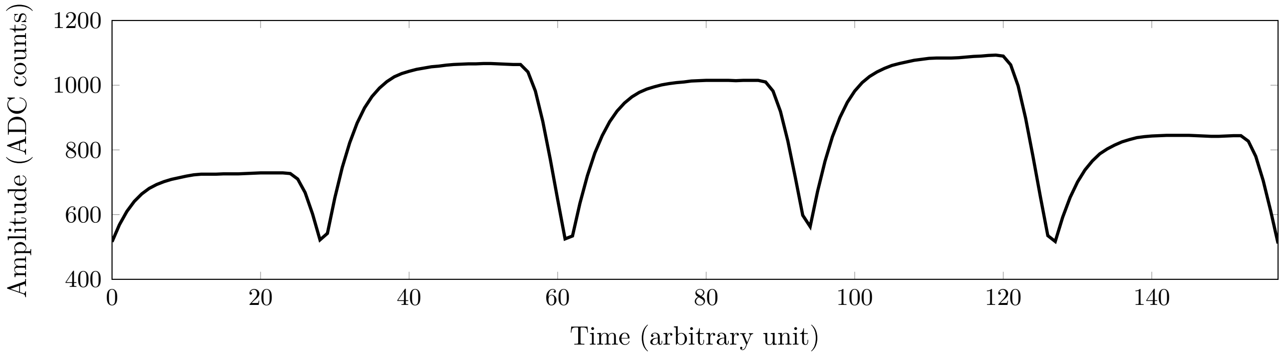

Figure 6.3: MWD amplitudes above the threshold set into Figure 6.2.

Since the delta encoding is effective when the difference between adjacent values of a sequence is small, the plateau clearly appears to be the ideal window for the application of the method. The information contained in Figure 6.3 can be arranged in a more significant way by plotting the difference Δ = MWD_output[t + 1] - MWD_output[t] as show in the following picture.

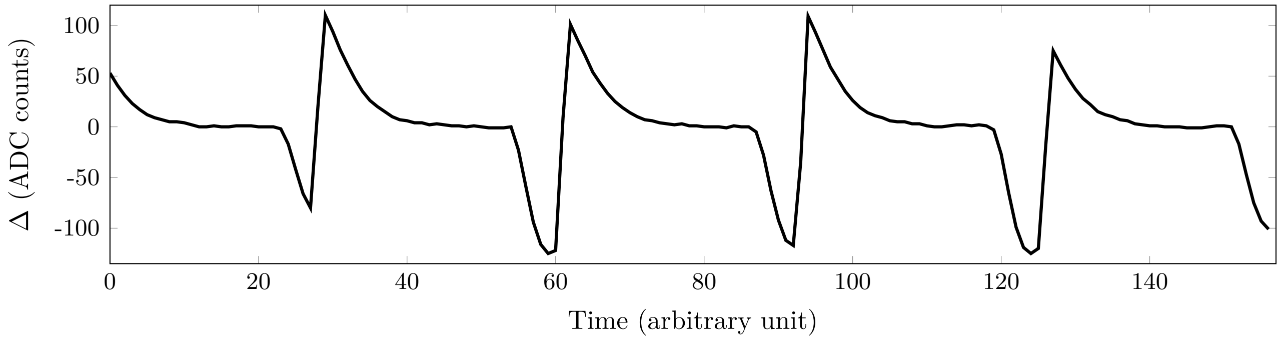

Figure 6.4: Difference between consecutive MWD amplitudes of Figure 6.3

Figure 6.4 is a more useful representation of the MWD output because it highlights the falling edges of the signal as negative values. These ranges of samples are totally unnecessary to the definition of the energy which is represented by the baseline of this signal. Therefore, if these values are filtered with a simple threshold set to zero, a plot of the occurrences can be derived.

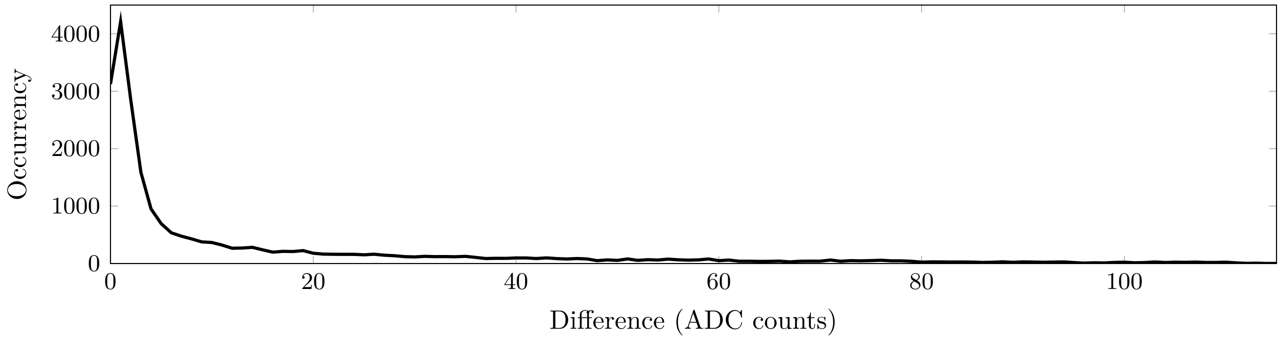

Figure 6.5: Occurrences of the positive differences MWD[t+1] - MWD[t]

Figure 6.5 shows the positive occurrences of the collected differences by taking a dataset of 1000 pulses. Most part of them are distinctly grouped in the lower part of the graph, reflecting the small variations between the points of the plateau. Although the trend is extended along a wide range, the frequencies drop down quicly and the important values are confined near the peak that occurs for $\Delta = 1$. Now, if a new threshold is set on these positive differences, a range of acceptability for the MWD outputs can be defined in the same way of that one proposed in Section 4.8.1.1. Thus, a sequence of MWD outputs precisely located inside the flat top window is formed. This approach is totally independent by the amplitude, because only the differences are considered and no further thresholds are required. The resulting stream is more convenient for a serialization. Indeed, let's assume to send out ten points of the plateau. If the acceptable Δ between them is set to eight for instance, only three bits will be taken to represent the component of the string after the first full-length element. Since the difference is strictly positive, no additional bit for the sign have to be accounted. So, in the nominal case of 12-bits resolution, the stream would be composed by 120 bits, while the delta encoding leads to 39 bits. The compression ratio in this case is equal to 0.32 with a remarkable saving percentage of 67%. However, this value can be still slightly increased if a Huffman coding is applied on the computed Δ. Since only 3 bits are used, the dictionary will be extremely small. By taking into account the eight most frequent differences the resulting Huffman tree is depicted in Figure 6.6

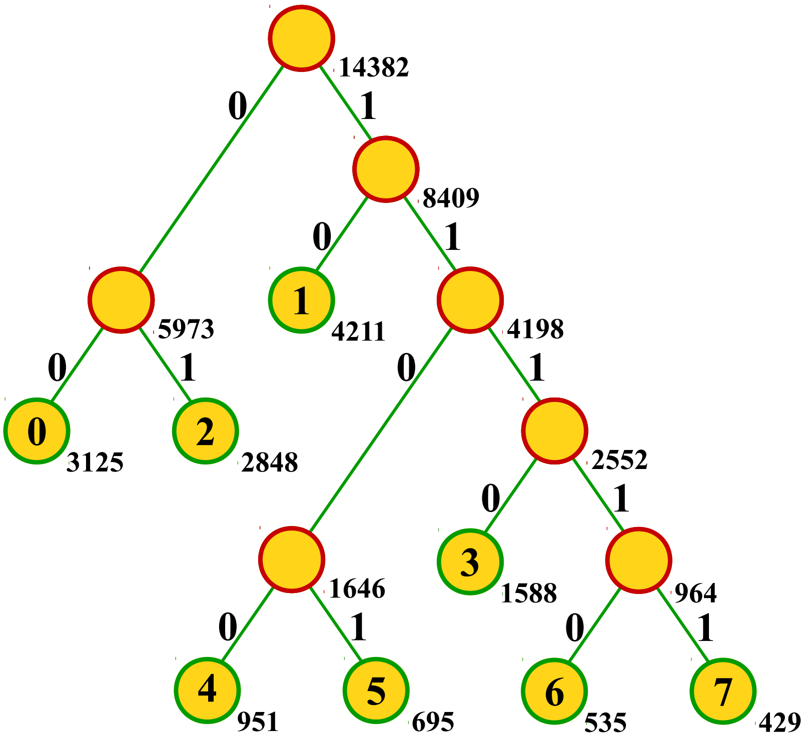

Figure 6.6: Huffman tree built on the eight most frequent differences.

Then, the Huffman codes are derived from the developed tree as summarized in Table 6.4. The dictionary length benefits enormously from the use of a delta encoding compared to the raw indexing of 700 codes required by the single Huffman coding.

Δ	Frequency	Huffman code
0	3125	00
1	4211	10
2	2848	01
3	1588	1110
4	951	1100
5	695	1101
6	535	11110
7	429	11111

Table 6.4: Huffman codes of the eight most frequent differences.

If 1000 pulses are considered and 10 samples of their flat tops are recorded for each one, the raw stream will be formed by 120,000 bits since:

$$1000 pulses \cdot 10 samples \cdot 12 bits = 120,000 bits \tag{6.3}$$

Then, the use of a delta encoding results into a drastic bit sequence reduction. Since the string referred to a single flat top is formed by a full 12-bits reference followed by 9 elements of 3-bits length each, only 39,000 bits are accounted for the same dataset:

$$1000 pulses \cdot (12 bits + 9 \cdot 3 bits) = 39,000 bits \tag{6.4}$$

Finally, if the delta encoding is coupled with an Huffman coding, the non uniform distribution of the codes can be exploited to further reduce the number of bits as demonstrated by:

$$1000 pulses \cdot 12 bits + 21970 bits = 33,979 bits \tag{6.5}$$

where 21970 bits are obtained by the use of the dictionary built in Table 6.4 on the 9 elements of the sequence. After this additional stage, the compression ratio is 0.28 while the saving percentage is increased up to almost 72%.
This analysis should be read as a preliminary investigation about the benefits of a data compression on the MWD output. Similar considerations can be done for the other relevant outcomes in order to evaluate the feasibility of these schemes for a on-chip implementation. Additional details about the signal characteristics can drive to better compression strategies by adequately weighing the trade-off between the required area and power consumption and device performance.
In the last section another application of these techniques on a simulated data sample for a CERN experiment will be presented.

6.2 RD53A

CMS is another of the four detectors employed by the LHC and its name stands for Compact Muon Solenoid. It shares the same scientific targets of the ATLAS experiment, nevertheless it is equipped with a different magnet-system design and it adopts other technical solutions. CMS was designed as a general-purpose detector to explore the high energy physics domain and to investigate the heavy ion collisions. The solenoid magnet is the core of CMS. It is formed by a cylindrical coil of superconducting cable which is able to generate an electromagnetic field of 4 Tesla. The collision products are detected by a stacked layers structure composed by a inner tracker, an electromagnetic calorimeter, a hadronic calorimeter, the solenoid magnet and a muon detector [206]. In 2013 the discover of the Higgs boson was confirmed by both CMS and ATLAS experiments.
As well as ALICE, CMS was also subjected to upgrades [207] [208] [209]. In

this context, a chip demonstrator developed by an international collaboration called RD53 was designed to prove the suitability of 65 nm technology for both ATLAS and CMS experiments, including radiation tolerance [210]. It was not intended to be a production IC, thus different design variations were implemented. The chip size is 20 x 11.6 mm^2 organized in 400 by 192 pixels of 50 μm pitch. The top hosts a row of test pads placed for debugging purposes, while peripheral circuitry is located at the bottom and it is dedicated to the bias, the configuration, the monitoring and the chip readout.

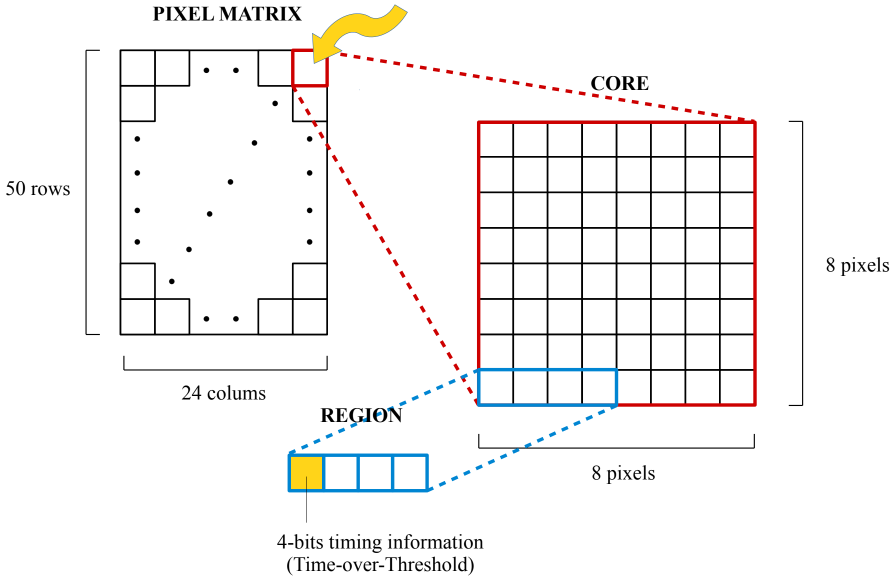

Figure 6.7: Hierarchy level of the RD53A pixel matrix.

The pixel matrix is segmented into 8 by 8 *pixel cores* distributed on 50 rows by 24 columns as shown in Figure 6.7. A core is partitioned into 16 *pixel regions*, each one equipped with 4 front-ends. A single sub-region provides 4-bits timing information. These values are the ToTs discussed in Section 1.2 and they are counted independently at 40 MHz clock. The readout was implemented with two different architectures named *Distributed Buffer Architecture* (DBA) and *Central Buffer Architecture* (CBA). In the first flavor the ToT information is stored by pixel, while its counterpart records the same information into a common region memory. Thus, the CBA architecture suppresses ToTs whose values are zero, but this advantage requires the recording of a hit map. By contrast, the DBA does not need an additional hit map at the expense of memory usage to store zero ToT values. Due to these differences, the DBA results efficient in case of high region occupancy, so when multiple pixels are hit. On the other hand, if the region occupancy is low the CBA flavor is efficient.

A very high hit rate of 3GHz/cm^2 was estimated for the upgrades of the tracker. This implied the exploration of some compression scheme to sustain the generated data. A first study about this feature is reported in [211] where an arithmetic encoder [212] at behavioural description level was designed. This work was based on 50 Monte Carlo simulations in which proton-proton collision events were considered. The obtained saving percentage was in the range between 41% and 54%, according to different readout configurations.

As a complementary work of this thesis, another preliminary investigation about data compression methods was done. The available dataset was not physically validated with the last expected results, thus the performance can not be considered well grounded. A mix of Monte Carlo simulations and internally generated data was adopted for the current study. The latter was initially carried out by considering a single column data stream and the DBA architecture was selected. The raw data string taken into account was composed by 6 bits for the column addressing, other 6 bits reserved to the core, 4 bits accounted for the pixel region address and 16 bits for the 4 ToT values. The data was collected and analyzed to estimate an appropriate approach. Figure 6.8 reports the distribution of the 4-bits ToTs along all their range.

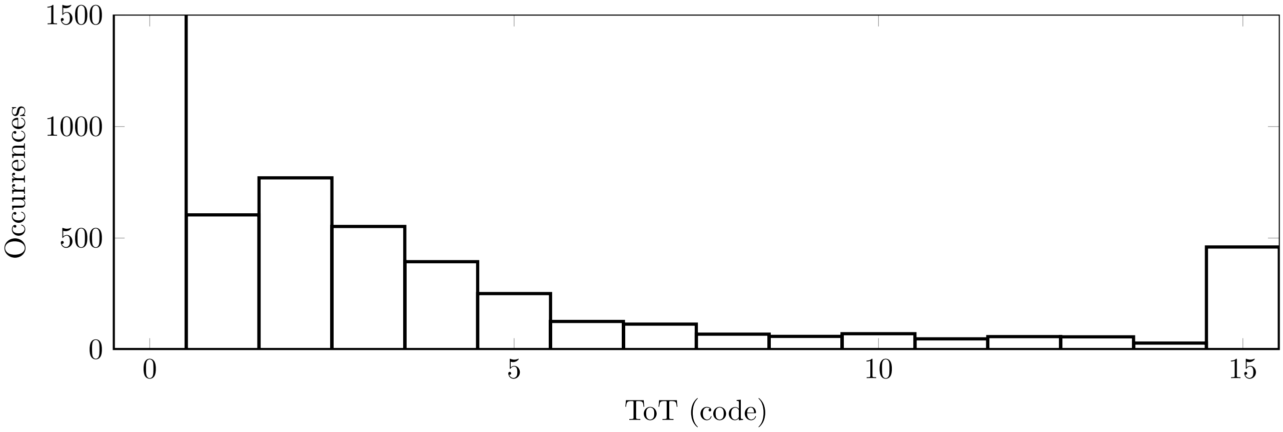

Figure 6.8: Histogram of ToTs distribution

In the DBA flavor the no hit condition of a pixel was indicated with the word 0000. This protocol justifies the huge number of occurrences which populate the zero bin whose height is around 5000 for the current dataset, not totally shown in the histogram. Since these values are not suppressed and due to the non equal distribution of the ToTs along the remaining range, a Huffman coding was adopted. The frequencies of each bin fed a C++ code implemented from scratch in order to generate the associated Huffman codes listed in Table 6.5.

ToT code	Frequency	Huffman code
0	604	1000
1	770	111
2	552	1001
3	394	1100
4	251	10101
5	126	101001
6	114	110101
7	69	1010000
8	59	1010001
9	71	110111
10	48	1101100
11	58	1101000
12	57	1101001
13	29	1101101
14	460	1011
15	4939	0

Table 6.5: ToTs distribution.

A behavioural model of this dictionary was implemented in SystemVerilog to verify this effectiveness. Thus, compared to the raw information, the saving percentage for the timing information was around 40%.
Afterwards, the core rows were considered. Although they are equal to 50, their addresses must necessarily be expressed with 6 bits, leaving some codes never explored. This fact can be exploited since it generates an asymmetrical condition in their distribution. Thus, it was proposed to split the 6 bits of the core into two subgroups, each one used as pointer of a grid arranged as depicted in Figure 6.9. Then, the idea was to superimpose the two grids as illustrated in the second stage. To distinguish the most significant pointer (MSP) from the least significant one (LSP) a bit D is reserved. Based on this bit fashion, a delta encoding of 3-bits length was applied by setting as initial reference the origin 000. In this way, due to the unequal distribution of the pointers, the system was forced to generate smaller differences with a higher probability and the final string was formed by $\{D,\Delta_1,\Delta_2\}$. An additional Huffman coding was coupled with the described scheme, leading to a saving percentage of 9%. A similar approach was adopted also for the pixel regions with a SP equals to 30% compared to the full addressing. As well as for the ToTs case, other two dictionaries were modeled in SV language to test the algorithms. By combining the performance of these three compression schemes, the final bit stream reduction was $\sim$37%. For this evaluation the column address was not considered because the compression engine was designed at single column level.

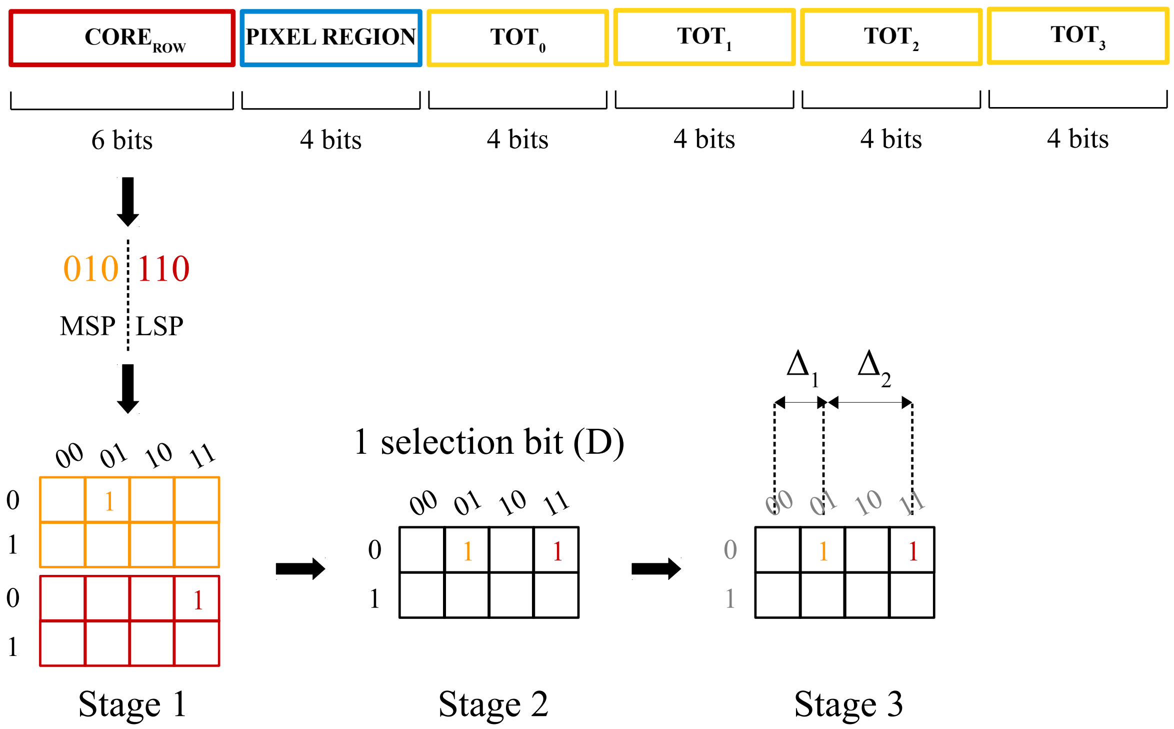

Figure 6.9: The considered raw data stream is composed by the core row address followed by the pixel region and the values of the four ToTs. Below is reported the proposed data reduction scheme for the core address.

A comparison can be done with another work on the same dataset [213]. This study implemented a RLE and a Variable Length Coding (VLC) to reduce the original data. It is important to keep in mind that the mentioned analysis considered only the active regions which included at least one hit pixel. Since intra-column and inter-column modes were even developed, the size reduction ranged between ∼36% and ∼38%.

Chapter 7

Conclusion and outlooks

The purpose of this thesis was to investigate digital signal processing techniques suitable to be implemented in multi-channel front-end ASICs for radiation sensors with the aim of pushing forward the state-of-the-art in the field in term of integration density and performance. The proposed digital processing chain was partitioned into two sub-systems: a digital calibration block for analog-to-digital converters and a processor reserved to the data manipulation. Both circuits were implemented in 110 nm and 65 nm CMOS technologies: the calibration unit occupies an area of 452 x 452 μm^2 and 265 x 265 μm^2, while the digital processor has an area of 781 x 781 μm^2 and 424 x 424 μm^2, respectively.

The ADC correction engine is based on an algorithm named Offset Double Conversion which adjusts a set of digital weights to calibrate the ADC outputs. The method is reported in literature and the efforts were focused on the on-chip realization, simplifying the technique without compromising the performance. A prototype have been fabricated in 110 nm CMOS process and this chip was equipped with a SAR ADC with a nominal resolution of 12 bits, two serializers and a LVDS bank. Due to an hardware level issue, it was not possible to directly test the calibration engine. However, the raw data of the SAR ADC was collected and processed by a post P&R simulation running in the typical corner. Therefore, the performance of the ADC was compared with the one after the bits adjustment, extracting the figures of merit through FFTs. The results are summarized in Table 7.1.

Figure of merit		sinewave		ramp		Training samples (10^3)
	Before	After	Variation	After	Variation	
SINAD (dB)	31.29	50.93	+19.64	49.88	+18.59	300
ENOB	4.90	8.17	+3.26	7.99	+3.09	
SFDR (dB)	33.55	56.0	+22.45	54.25	+20.70	

Table 7.1: Summary of the figures of merit obtained in the test campaign.

For this training set the learning parameters μ_Δ and μ_W were chosen to remap the original data into a 10-bits resolution range. Due to the high number of missing codes, it was not possible to recover them in this interval with the given parameters. However, the calibrated output benefits of a general improvement in terms of linearity as demonstrated by the SINAD which rises from the initial 31.29 dB to slightly less than 50.93 dB after calibration. The ENOB follows the trend with a positive variation of +3.26, achieving an effective resolution of 8.17 bits. This results is satisfactory, considering the initial ENOB is equal to 4.90. The SFDR records the same improvement with a final value of 56 dB that highlights the effectiveness of the correction algorithm. Since 300,000 samples were used during the training step and the circuit was running at 20 ns clock, the total time required to complete the calibration was only 6 ms which is still an acceptable interval. Finally, it is important to point out that also when a ramp signal is used to calibrate the system, the response is still satisfactory with the given learning parameters. Indeed, the SINAD is almost equal to 50 dB, while the ENOB is very close to a resolution of 8 bits and the SFDR is up to 60 dB. In a multi-channel ASIC implementation, a ramp signal is much easier to deploy than a sinusoidal one. Such ASICs are in fact intended for large systems where the accessibility to the detector is an issue and connections to the outside world must be minimized. It is important to keep in mind that these values are a function of the number of training samples employed as well as the learning parameters that define the final resolution range and the shape of the signal. If we recall Table 5.7, when the training set is equal to 200,000 points the ramp signal performance is even higher that the sinewave counterpart. Ultimately, to achieve the best performance it is important to find a balance between the parameters. A possible procedure to accomplish this task is to start with small values of μ_Δ and μ_W and a fixed signal shape. Then, the number of training samples can be progressively increased, recording the figures of merit in which one is interested. Afterwards, the signal shape can be changed to differently explore the sources of non-linearities.
The power consumption of the calibration engine implemented in 65 nm CMOS is below 1 mW. In particular, the total static power is 0.37 mW when the correction circuit is not enabled. This value is increased up to 0.47 mW once the weights are updated. The dynamic contribution is 0.84 mW for the same calibrating mode and the larger part of the power budget in these three cases is due the internal one with a 86.19%, followed by the switching (13.58%) and the leakage (0.24%).

After this initial stage, the core of the design is the digital signal processor. This unit is equipped with an anti-glitch system and a baseline restorer to correct the pulse input. Both these circuits exploit the amplitude intervals defined by two pro-

grammable thresholds. However, these levels are self-adjusted by default. They are based on the signal itself, computing its average noise value and standard deviation. The latter is derived using the bisection algorithm. The two thresholds are defined as the sum of the average value with configurable multiple values of the deviation.
Hence, the raw data is processed by a pipelined stage composed by a digital CR-RC^4 pulse shaper and a trapezoidal filter carried out through a Mobile Window Deconvolution. This block is coupled with a dedicated circuit to extract the amplitude which is proportional to the energy. The final SNR results further improved compared to the single CR-RC^4 as summarized in Table 5.12, reported below for convenience.

Stage	SNR	SNR$_{dB}$	F$_{SNR}$
RAW	17.24	24.73	1
RC^4	170.18	44.62	9.87
MWD	279.52	48.93	16.22

Table 7.2: Post P&R SNR values for RC^4 and MWD coupled with the energy extraction circuitry.

The raw data was generated with a Python script accounting for a non-ideal 12-bits SAR ADC with a DAC affected by a 15% of random capacitor mismatch. Furthermore in this generator other possible sources of noise were also considered such as the quantization error, the thermal noise and the jitter contribution. The values show in Table 7.2 were elaborated by the DSP that processed 1000 pulses with the same nominal amplitude. The initial SNR of the raw data is around 17 which is a typical value for a front-end. The SNR after the CR-RC^4 filtering is increased by 10 times. This value is slightly smaller than the simulated one in Python, but this deviation is justifiable considering two factors. The first one is related to the numerical representation, since the script was implemented on a 64-bits machine. By contrast, the CR-RC^4 outcome is only 12-bits long, excluding an additional bit reserved to the sign. The second reason can be attributed to the differences between the RTL and Python codes. Due to its nature of preliminary study, this latter does not realize a continuous update of the baseline. Conversely, the final circuit has dedicated block that monitors the baseline of the digital shaper at each clock cycle, correcting the CR-RC^4 output. Therefore, the MWD filter coupled with the module to extract the amplitude of the signal further increases the final SNR up to 16 times. The average amplitude, expressed in ADC codes, is 860.5 ± 11.7, hence the relative error is 1.3%.
A second set of raw data was used to feed the Constant Fraction Discriminator. Besides the noise already introduced with the previous simulation, the generated 1000 pulses were characterized by variable amplitudes between 1700 and around 3000

ADC codes to simulate time walk effects. Additionally, the nominal rising time τ_r was randomized pulse-by-pulse with a maximum variation Δ_{τ_r} of $\pm$ 30%. The hardware implementation of the Goldschmit's algorithm ensures an accurate division to carry out the interpolation process. The results are collected in Table 5.13 where the variation is associated with the corresponding zero point and the standard deviation. The time events can be easily reconstructed using as reference the pulse generated when the CFD signal crosses the zero level. The best result was achieved for Δ_{τ_r} = 0% where the standard deviation is 0.64 ns. However, the circuit can tolerate Δ_{τ_r} = 20% remaining below 1 ns of uncertainty. Thus, the timing block can handle both the time walk effect and the rising time variation with an acceptable performance in terms of resolution.

The power consumption of this digital unit was analyzed under different working conditions as shown in Table 5.14. The maximum dynamic dissipation is 5.3 mW when the pulse is processed. This is expected because during the signal elaboration all the filters are enabled. The power is divided between internal (76.94%), switching (22.96%) and leakage (0.10%). The entire processor handles pile-up events with a dedicated filter and it is protected against radiation. The strategy adopted to avoid undesired bit inversions in the memory content is the well-explored Triple Modular Redundancy. It was demonstrated that the protection is effective against Multiple Bit Upsets in post P&R simulations.

7.0.1 Future perspective

A comparison with other works reported in Section 1.3 is quite difficult since they are designed and optimized to specific tasks for the most part. By contrast, this thesis was focused on a more general DSP for multipurpose uses where several filters have been implemented, as introduced in the abstract. The same reason motivated the implementation of a calibration block that can be used for a large topology of converters and it is not limited to the SAR ones.

However, some comparison such as area and the power consumption allows few interesting considerations on future perspective. The comparative Table 7.3 reports some features of the digital processors implemented for HEP experiments. In the following, the technology target for this work will be the 65 nm CMOS. The total circuit area is 0.25 mm^2 considering the digital occupancy per channel, accounting both the calibration unit and the DSP. Since ALTRO and S-ALTRO have been fabricated with 16 channels, the same number of digital units for the current design would be equal to 4 mm^2 on silicon. This is a reasonable value even in the case of a comparison with SAMPA, where the channels were doubled.

Parameter	ALTRO	S-ALTRO	SAMPA	This work
Process (nm)	250	130	120	110 - 65
Area (mm^2)	64	∼49	∼86	0.81 (110) 0.25 (65)
Sensor topology	gas-based	gas-based, GEMs	TPC, Muon Chamber	general purpose
Channels	16	16	32	1
Power supply (V)	2.5	1.5	1.25	1.2
ADC resolution	10	10	10	12
Frequency (MHz)	10	10	10	50
Total Power (mW)	320	757	1.5 (ADC)	-
DSP Power (mW)	67.5-250	65-84	-	-
DSP Power/ch (mW/ch)	4.2	4.04	-	4.1-6.2
Embedded memory	800Kb	1280Kb	-	No
ADC calibration	No	No	No	Yes
Data formatting	Yes	Yes	Yes	No
Zero-suppression	Yes	Yes	Yes	No
Pile-up rejection	No	No	-	Yes
Radiation hardness	No	No	No	Yes

Table 7.3: Comparative summary with other works.

A power consideration is more tricky due to the different operation conditions of the chips. For what concerns ALTRO, the power consumption ranges between 67.5 and 250 mW while the variation of S-ALTRO is smaller with the minimum value around 65 mW and the maximum one equal to 84 mW. From these power windows a value for the single channel can be derived considering the lower value, hence 4.2 and 4.04 mW, respectively. These values are quite comparable with the range of the present design which is 4.1 (static) - 6.2 (dynamic) mW obtained as sum of the power dissipations of the two processors. However, this comparison is not totally fair because of the functionalities carried out. For instance, ALTRO takes advantage of lookup tables to perform a baseline correction. This feature was designed taking into account the gas-based detector and it involves particularly energy-intensive operations that lead to 250 mW of power dissipation when the lookup tables are enabled. On the other hand, this chip realized a tail cancellation filter with two types of baseline restoring and an additional zero-suppression scheme without any further digital elaboration on-chip. S-ALTRO inherited basically the same features, while the digital power dissipation of SAMPA can not be evaluated due to a lack of information at the time of writing. The power budget evaluated for this work includes the baseline correction, the CR-RC^4 pulse shaper coupled with a MWD filter, the charge accumulator, the CFD module, the pile-up rejection system and the TMR protection at channel level, additionally equipped with the calibration engine. Furthermore, the power estimation of this design have to be considered as a worst case. Indeed, at higher level of ASIC architecture it might be planned to share the core of the calibration engine between more channels, leaving only the updated weights stored in memories at the single front-end level. It makes perfectly sense because the more expensive computation costs can be shared between channels, reducing the overall impact on the power budget. In the following a definitely non-complete list of other items is reported that can further improve the current implementation, optimizing the overall digital design:

- The first step should be the realization of a zero-suppression scheme which is a low-cost change. In order to accomplish this upgrade, a more precise model of the expected signal should be obtained.
- Another important investigation may be about the outputs of the filters as a function of the numerical resolution. In particular the SNR degradations of the CR-RC^4 shaper, the MWD filter and CFD could be studied progressively reducing the current 32-bits long words. In the same way, an analysis concerning the performance of the calibration engine should be explored with a reduced bit-length of the weights.
- A hardware implementation of a compression scheme such as the one proposed in Section 6.1.3 could be a first step for a smarter management of the data to sent out. The delta encoding is a valid candidate at least for the MWD output. However, an extension to the CR-RC^4 and CFD output appears reasonable at the expense of a memory bank dedicated to their dictionaries.
- In order to reduce the area, both the calibration unit and the digital processor must be integrated at ASIC level. As reported before, this choice would benefit of a reduced power consumption as well as a more compact design since now the blocks were developed as standalone modules, equipped with dedicated cumbersome power rings.
- Despite the independent implementation of the two digital blocks presented in this thesis, some considerations have been made in view of an integration at chip level. In Figure 7.1 a) and b) are shown two possible layouts in the case of strip sensors. The latters are represented on the left side of the figures considering a generic pitch of 50 μm (the sensors are not proportionally scaled on the horizontal axes). Taking into account the 65 nm version of this work, the calibration

block occupies an area of 265 x 265 μm^2. Since each strip is equipped with a dedicated digital converter, a sharing of the calibration engine among the sensor could be more convenient. For this reason the computational core of the design can be placed in a dedicated area (the green box). In order to reduce the area on silicon reserved to the memories, the use of RAMs is a reasonable choice. In this way, the calibration of each ADC is obtained by multiplexing the digitized output of a single strip. Additionally, these outputs can now be directly connected to an accumulator (blue boxes) to independently obtain the adjusted codes.

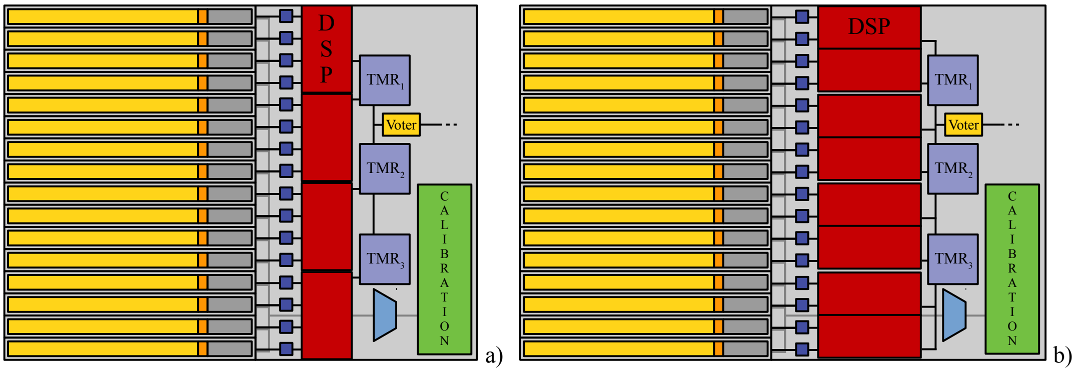

Figure 7.1: Block level representation of possible layouts for the integration of a chip.

The current implementation of the DSP module requires 424 x 424 μm^2 which is not suitable for a pitch of 50 μm. However, the resources sharing strategy can be applied once again. The three trenches defined to implement the TMR as explained in Section 5.24 occupy a total area around 51k μm^2. Since these memory cells store the values of the coefficients and parameters that will be used by all the processors, a different configuration of this bank makes sense at chip level. Therefore, the TMR can be carried out by distributing the tripled registers along a DSP array of replicated modules. Each one is fed with the nearest memory and at the same time multiple outputs from the strips are accomodated by using a multiplexer scheme. In this way, also the spatial distribution of the TMRs is properly mantained while the routing congestion is avoided. The total area of a single DSP block is now reduced around 126k μm^2 from the initial 176k μm^2. If the timing becomes a critical parameter for the application, the layout depicted in Figure 7.1 b) could be adopted. Here each DSP block is connected to a reduced number of outputs in order to decrease the work queue. The power impact on this elongated design is expected to be trivial because of the pipelined architecture of the processor itself. Indeed the block level representation of Figure 4.3 shows the partition of each filter into a dedicated module and the data stream is developed as a pipeline structure. Basically, this block organization can benefit from a larger placement without suffering a drop of the power performance. Anyway, the worst case can ward off by adding a suitable number of vertical power stripes with the same width of the current implementation to guarantee a uniform power distribution across the area.

- Finally, a down-scaling to a smaller technological node such as 28 nm would be very interesting to study the area and power consumption of both the calibration engine and the digital signal processor. Moreover, the circuitry can also take advantage of the reduced gate dimensions for what concern the radiation hardness. Indeed the damage caused by particle interactions is proportional to gate oxide volume. By contrast, ultra-scaled technologies involve cautions on the analog counterpart which is reasonable to face in a real mixed-signal system. For instance, if the lower supply voltage reduces the power consumption on the digital side, the same trend can not be derived for the analog part. This is due to the decreased SNR for a given noise power compared to larger technologies. As consequence of the reduced supply voltage, the threshold voltage have also to decrease because of an adequate current capability. However, the ratio between the two voltages is smaller in ultra-scaled nodes if compared to other older CMOS technologies and this fact leads to a reduced dynamic range. These second order effects and other issues such as PVT and mismatch variations which are minor in larger technological nodes must be properly addressed during the down-scaling and porting of the design.

List of Figures

List of Tables

Bibliography

[1] Claus Grupen, Irène Buvat, *Handbook of Particle Detection and Imaging*, Springer-Verlag Berlin Heidelberg, 2012.

[2] Stefaan Tavernier, *Experimental Techniques in Nuclear and Particle Physics*, Springer-Verlag Berlin Heidelberg, 2010.

[3] Gerhard Lutz, *Semiconductor Radiation Detectors*, Springer, Berlin, Heidelberg, 2007.

[4] Claus Grupen, Boris Shwartz, *Particle Detectors*, Cambridge University Press, 2011.

[5] Angelo Rivetti, *CMOS Front-End Electronics for Radiation Sensors*, CRC Press, 2017.

[6] Valerio Re, *Front-end electronics for silicon trackers*, IV Scuola Nazionale Rivelatori ed Elettronica per Fisica delle Alte Energie, Astrofisica, Applicationi Spaziali e Fisica Medica", Laboratori Nazionali di Legnaro (Padova), 2011, pp 23-29.

[7] J. Härkönen, E. Tuovinen, P. Luukka, T. Mäenpää, E. Tuovinen, E. Tuominen, Y. Gotra, L. Spiegel, *ACcoupled pitch adapters for silicon strip detectors*, 17th RD50 Workshop, CERN, 2010.

[8] Gianluigi Casse, Marko Milovanovic, Paul Dervan, Ilya Tsurin, *Comparison of the AC and DC coupled pixels sensors read out with FE-I4 electronics*, 25th RD50 Workshop, CERN, 2014.

[9] Helmuth Spieler, *Semiconductor Detector Systems*, Oxford University Press, 2005, chapters 2, 4.

[10] F. Corsia, A. Dragone, C. Marzocca, A. Del Guerra, P. Delizia, N. Dinu, C. Piemonte, M. Boscardin, G.F. Dalla Betta, *Modelling a silicon photomultiplier (SiPM) as a signal source for optimum front-end design*, Nuclear Instruments

and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 572, Issue 1, 1 March 2007, Pages 416-418.

[11] S. Holland, H. Spieler, *A Monolithically Integrated Detector-Preamplifier on High-Resistivity Silicon*, IEEE Transactions on Nuclear Science (Volume: 37, Issue: 2, April 1990).

[12] Raffaele Aaron Giampaolo, Andrea Di Salvo, Lucio Pancheri, Tommaso Croci, Jonhatan Olave, Angelo Rivetti, Manuel Da Rocha Rolo, Serena Mattiazzo, Piero Giubilato, *Depleted MAPS on a 110 nm CMOS CIS Technology*, 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS).

[13] Lucio Pancheri, Raffaele A. Giampaolo, Andrea Di Salvo, Serena Mattiazzo, Thomas Corradino, Piero Giubilato, Romualdo Santoro, Massimo Caccia, Giovanni Margutti, Jonhatan E. Olave, Manuel Rolo, Angelo Rivetti, *Fully Depleted MAPS in 110-nm CMOS Process With 100-300 μm Active Substrate*, IEEE Transactions on Electron Devices (Volume: 67, Issue: 6, June 2020).

[14] J. Kaplon, W. Dabrowski, *Fast CMOS Binary Front End for SiliconStrip Detectors at LHC Experiments*, IEEE Transactions on Nuclear Science (Volume: 52, Issue: 6, Dec. 2005).

[15] Michela Chiosso, Ozgur Cobanoglu, Giovanni Mazza, Daniele Panieri, Angelo Rivetti, *A fast binary front-end ASIC for the RICH detector of the COMPASS experiment at CERN*, IEEE Nuclear Science Symposium Conference Record, 2008.

[16] P. Grybos, A. E. Cabal Rodriguez, W. Dabrowski, M. Idzik, J. Lopez Gaitan, F. Prino, L. Ramello, K. Swientek, P. Wiacek, *RX64DTH - A Fully Integrated 64-channel ASIC for Digital X-ray Imaging System with Energy Window Selection*, IEEE Symposium Conference Record Nuclear Science, 2004.

[17] Roger Steadman, Christoph Herrmann, Oliver Mülhens, Dale G. Maeding, *ChromAIX: Fast photon-counting ASIC for Spectral Computed Tomography*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 648, Supplement 1, 21 August 2011, Pages S211-S215.

[18] Ivan Peri, Laurent Blanquart, Giacomo Comes, Peter Denes, Kevin Einsweiler, Peter Fischer, Emanuele Mandelli, Gerrit Meddeler, *The FEI3 readout chip*

for the ATLAS pixel detector, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 565, Issue 1, 1 September 2006, Pages 178-187.

[19] I. Kipnis, T. Collins, J. DeWitt, S. Dow, A. Frey, A. Grillo, R. Johnson, W. Kroeger, A. Leona, L. Luo, E. Mandelli, P.F. Manfredi, M. Melani et al., *A time-over-threshold machine: the readout integrated circuit for the BABAR Silicon Vertex Tracker*, IEEE Transactions on Nuclear Science (Volume: 44, Issue: 3, June 1997).

[20] S. Martoiu, A. Rivetti, A. Ceccucci, A. Cotta Ramusino, S. Chiozzi, G. Dellacasa, M. Fiorini, S. Garbolino, P. Jarron, J. Kaplon, A. Kluge, F. Marchetto, E. Martin Albarran, G. Mazza, M. Noy, P. Riedler, S. Tiurianemi, *A Pixel Front-End ASIC in 0.13 m CMOS for the NA62 Experiment with on Pixel 100 ps Time-to-Digital Converter*, IEEE Nuclear Science Symposium Conference Record (NSS/MIC), 2009.

[21] Peter Fischer, Ivan Peric, Michael Ritzert, Martin Koniczek, *Fast self triggered multi channel readout ASIC for time and energy measurement*, IEEE Transactions on Nuclear Science (Volume: 56, Issue: 3, June 2009).

[22] Gianluigi De Geronimo, Anand Kandasamy, Paul OConnor, *Analog peak detector and derandomizer for high rate spectroscopy*, IEEE Nuclear Science Symposium Conference Record (Cat. No.01CH37310), 2001.

[23] Gianluigi De Geronimo, Jack Fried, Paul OConnor, Veljko Radeka, Graham C. Smith, Craig Thorn, Bo Yu, *Front-end ASIC for a GEM based time projection chamber*, IEEE Transactions on Nuclear Science (Volume: 51, Issue: 4, Aug. 2004).

[24] Eric Oberla, Jean-Francois Genat, Hervé Grabas, Henry Frisch, Kurtis Nishimura, Gary Varner, *A 15 GSa/s, 1.5 GHz bandwidth waveform digitizing ASIC*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 735, 21 January 2014, Pages 452-461.

[25] M. J. French, L. L. Jones, Q. Morrissey, A. Neviani, R. Turchetta, J. Fulcher, G. Hall, E. Noah, M. Raymond, G. Cervelli, P. Moreira, G. Marseguerra, *Design and results from the APV25, a deep sub-micron CMOS front-end chip for the CMS tracker*, Nuclear Instruments and Methods in Physics Research Section

A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 466, Issue 2, 1 July 2001, Pages 359-365.

[26] CERN/LHCC/9571 LHCC/P3, *ALICE - Technical Proposal for A Large Ion Collider Experiment at the CERN LHC*, https://cds.cern.ch/record/293391/files/cer-000214817.pdf, CERN, 1995.

[27] CERN EP/ED, *ALICE TPC Readout Chip - User Manual*, https://ep-ed-alice-tpc.web.cern.ch/, CERN, 2002.

[28] R. Esteve Bosch, A. Jiménez de Parga, B. Mota, and L. Musa, *The ALTRO chip: a 16-channel A/D converter and digital processor for gas detectors*, IEEE Transactions on Nuclear Science, (Volume: 50, Issue: 6, Dec. 2003).

[29] Bernardo Miguel Lopes Leitao Mota, *Time-domain signal processing algorithms and their implementation in the ALTRO chip for the ALICE TPC*, PhD thesis, Ecole Polytechnique, Lausanne, 2003, pp 41-54.

[30] Eduardo José García García, *Novel Front-end Electronics for Time Projection Chamber Detectors*, PhD thesis, Universidad Politécnica de Valencia, 2012, pp 39-44.

[31] Paul Aspell, Massimiliano De Gaspari,Hugo França, Eduardo García García, Luciano Musa, *Super-Altro 16: A front-end system on chip for DSP based readout of gaseous detectors*, IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC), 2012.

[32] Arild Velure, *Upgrades of the ALICE TPC Front-End Electronicsfor Long Shutdown 1 and 2*, IEEE Transactions on Nuclear Science (Volume: 62, Issue: 3, June 2015).

[33] S. H. I. Barboza, *SAMPA chip: a new ASIC for the ALICE TPC and MCH upgrades*, Journal of Instrumentation, Volume 11, 2016.

[34] J. Adolfsson et al., *SAMPA Chip: the New 32 Channels ASIC for the ALICE TPC and MCH Upgrades*, Journal of Instrumentation, Volume 12, April 2017.

[35] S. M. Mahmood, K. Røed, F. L. Winje, A. Velure on behalf of the ALICE collaboration, *First irradiation test results of the ALICE SAMPA ASIC*, Topical Workshop on Electronics for Particle Physics (TWEPP-17) - Radiation Tolerant Components and Systems, Volume 313, 2018.

[36] DT5780, *Dual Digital Multi Channel Analyzer (HV & Preamplifier PS) - Desktop*, CAEN, https://www.caen.it/products/dt5780/.

[37] ISOLDE, *Radioactive ion beam facility ISOLDE, CERN*, ISOLDE, https://isolde.cern/.

[38] P. Reiter and for the MINIBALL collaboration, *Nuclear-Structure Physics with MINIBALL at HIE-ISOLDE*, Journal of Physics: Conference Series, Volume 966, 12th International Spring Seminar on Nuclear Physics: Current Problems and Prospects for Nuclear Structure 1519 May 2017, Sant'Angelo d'Ischia, Italy.

[39] ADSP-2183, *DSP Microcomputer, ADSP-2183 datasheet*, https://www.analog.com/en/products/adsp-2183.html#product-overview.

[40] Martin Lauer, *Digital Signal Processing for segmented HPGe DetectorsPreprocessing Algorithms and Pulse Shape Analysis*, PhD thesis, University of Heidelberg, 2004, pp 30-41.

[41] Bryan Lizon, *Fundamentals of Precision ADC Noise Analysis, Design tips and tricks to reduce noise with delta-sigma ADCs*, Texas Instruments, e-book, 2020, pp 5-10.

[42] Microchip Technology, *ADC Signal-to-Noise Ratio and Distortion (SINAD)*, https://microchipdeveloper.com/adc:adc-sinad.

[43] Sergio Rapuano, *Preliminary Considerations on ADC Standard Harmonization*, IEEE Transactions on Instrumentation and Measurement (Volume: 57, Issue: 2, Feb. 2008), pp 392-393.

[44] Bonnie C. Baker, *Driving the Analog Inputs of a SAR A/D Converter*, Microchip Technology Inc., white paper, AN246, p. 2.

[45] Wendy M. Middleton, Mac E. Van Valkenburg, *Reference Data for Engineers - Radio, Electronics, Computer, and Communications*, Newnes, 2002, p. 34-8.

[46] Jan Henning Mueller, Sebastian Strache, Laurens Busch, Ralf Wunderlich, Stefan Heinen, *The Impact of Noise and Mismatch on SAR ADCs and a Calibratable Capacitance Array Based Approach for High Resolutions*, International Journal of Electronics and Telecommunications, 2013, vol. 59, n. 2, pp. 161167.

[47] Behzad Razavi, *The Flash ADC [A Circuit for All Seasons]*, IEEE Solid-State Circuits Magazine (Volume: 9, Issue: 3, Summer 2017), pp 9-13.

[48] Sakkarapani Balagopal, Suat U. Ay, *An on-chip ramp generator for single-slope look ahead ramp (SSLAR) ADC*, 2009 52nd IEEE International Midwest Symposium on Circuits and Systems.

[49] Shlomo Engelberg, *Digital Signal Processing - An Experimental Approach*, Springer, Signals and Communication Technology series, 2008, pp 93-97 and pp 29-51.

[50] F. Ciciriello, C. Marzocca, L. Demaria, L. Pacher, F. Rotondo, R. Wheadon, A. Di Salvo, P. Mazzucchelli, *A Rad-Hard 12-bit Auto-Calibrated ADC in CMOS 65nm*, 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC).

[51] Abdulrahman Abumurad, Kyusun Choi, *Increasing the ADC precision with oversampling in a flash ADC*, 2012 IEEE 11th International Conference on Solid-State and Integrated Circuit Technology.

[52] Jong Im Lee, Jong-In Song, *Flash ADC architecture using multiplexers to reduce a preamplifier and comparator count*, 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013).

[53] Ryan Curran, *Exploring different SAR ADC analog input architectures*, Analog Devices, Technical article.

[54] Jianwen Li, Xuan Guo, Jian Luan, Danyu Wu, Lei Zhou, Nanxun Wu, Yinkun Huang, Hanbo Jia, Xuqiang Zheng, Jin Wu1, Xinyu Liu, *A 1 GS/s 12-Bit Pipelined/SAR Hybrid ADC in 40 nm CMOS Technology*, 2020, Electronics, 9(2), 375.

[55] Takao Waho, *Non-binary Successive Approximation Analog-to-Digital Converters: A Survey*, 2014, IEEE 44th International Symposium on Multiple-Valued Logic.

[56] Gabriele Manganaro, Dave Robertson, *Analog Dialogue 49-07, July 20151Interleaving ADCs: Unraveling the Mysteries*, 2015, Analog Dialogue 49-07.

[57] Phillip E. Allen, Douglas R. Holberg, *CMOS Analog Circuit Design*, Oxford University Press, 2nd edition, 2002, pp 43-47.

[58] Dai Zhang, Christer Svensson, Atila Alvandpour, *Power consumption bounds for SAR ADCs*, 2011 20th European Conference on Circuit Theory and Design (ECCTD).

[59] Yuefeng Cao, Shumin Zhang, Tianli Zhang, Yongzhen Chen, Yutong Zhao, Chixiao Chen, Fan Ye, Junyan Ren, *A 91.0-dB SFDR Single-Coarse Dual-Fine Pipelined-SAR ADC With Split-Based Background Calibration in 28-nm*

CMOS, IEEE Transactions on Circuits and Systems I: Regular Papers (Volume: 68, Issue: 2, Feb. 2021).

[60] Wenning Jiang, Yan Zhu, Minglei Zhang, Chi-Hang Chan, Rui Paulo Martins, *A Temperature-Stabilized Single-Channel 1-GS/s 60-dB SNDR SAR-Assisted Pipelined ADC With Dynamic Gm-R-Based Amplifier*, IEEE Journal of Solid-State Circuits (Volume: 55, Issue: 2, Feb. 2020, pp 322-332).

[61] Bob Verbruggen, Masao Iriguchi, Manuel de la Guia Solaz, Guy Glorieux, Kazuaki Deguchi, Badr Malki, Jan Craninckx, *A 2.1 mW 11b 410 MS/s dynamic pipelined SAR ADC with background calibration in 28nm digital CMOS*, 2013 Symposium on VLSI Circuits.

[62] Frank van der Goes, Christopher M. Ward, Santosh Astgimath, Han Yan, Jeff Riley, Zeng Zeng, Jan Mulder, Sijia Wang, Klaas Bult, *A 1.5 mW 68 dB SNDR 80 Ms/s 2 Œ Interleaved Pipelined SAR ADC in 28 nm CMOS*, IEEE Journal of Solid-State Circuits (Volume: 49, Issue: 12, Dec. 2014, pp 2835-2845).

[63] Yun Chiu, Wenbo Liu, Pingli Huang, Foti Kacani, Gary Wang, Brian Elies, Yuan Zhou, *Digital Calibration of SAR ADC*, Proceedings of the 10th International Conference on Sampling Theory and Applications, 2013, pp 544-547.

[64] Wenbo Liu, Pingli Huang, Yun Chiu, *A 12-bit 50-MS/s 3.3-mW SAR ADC with background digital calibration*, Proceedings of the IEEE 2012 Custom Integrated Circuits Conference.

[65] Wenbo Liu, Pingli Huang, Yun Chiu, *A 12-bit, 45-MS/s, 3-mW Redundant Successive-Approximation-Register Analog-to-Digital Converter With Digital Calibration*, IEEE Journal of Solid-State Circuits (Volume: 46, Issue: 11, Nov. 2011), pp 2661-2672.

[66] Ming Zhang, Anxue Zhang, *The Superposition Principle of Linear Time-Invariant Systems [Lecture Notes]*, IEEE Signal Processing Magazine (Volume: 36, Issue: 6, Nov. 2019), pp 153-156.

[67] Guanhua Wang, Yun Chiu, *Fast FPGA emulation of background-calibrated SAR ADC with internal redundancy dithering*, Proceedings of the IEEE 2013 Custom Integrated Circuits Conference.

[68] Xian Gu, Xiuju He, Fule Li, *A calibration technique for SAR ADC based on code density test*, 2015 IEEE 11th International Conference on ASIC (ASICON).

[69] Xin Dai, Degang Chen, Randall Geiger, *A cost-effective histogram test-based algorithm for digital calibration of high-precision pipelined ADCs*, 2005 IEEE International Symposium on Circuits and Systems.

[70] Wenbo Liu, Pingli Huang, Yun Chiu, *A 12-bit, 45-MS/s, 3-mW Redundant Successive-Approximation-Register Analog-to-Digital Converter With Digital Calibration*, IEEE Journal of Solid-State Circuits (Volume: 46, Issue: 11, Nov. 2011), pp 2661-2672.

[71] Eric Siragusa, Ian Galton, *A digitally enhanced 1.8-V 15-bit 40-MSample/s CMOS pipelined ADC*, IEEE Journal of Solid-State Circuits (Volume: 39, Issue: 12, Dec. 2004), pp 2126-2138.

[72] Yun-Shiang Shu, Bang-Sup Song, *A 15b-Linear, 20MS/s, 1.5b/Stage Pipelined ADC Digitally Calibrated with Signal-Dependent Dithering*, 2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.

[73] Yuan Zhou, Yun Chiu, *Digital calibration of inter-stage nonlinear errors in pipelined SAR ADC*, 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS).

[74] Gene Frantz, *Where will floating point take us?*, Texas Instruments, White paper SPRY145, 2010.

[75] John Tomarakos, *Relationship of Data Word Size to Dynamic Range and Signal Quality in Digital Audio Processing Applications*, DSP Field Applications, Analog Devices, https://www.analog.com/en/education/education-library/articles/relationship-data-word-size-dynamic-range.html.

[76] Robert Oshana, *DSP Software Development Techniques for Embedded and Real-Time Systems*, Newnes, 2006, chapter 5, pp 125-126 and p 66.

[77] Tomas Fryza, *Introduction to fixed-point multiplication and signal processing application*, 2009 19th International Conference Radioelektronika.

[78] Janne Janhunen, Perttu Salmela, Olli Silvén, Markku Juntti, *Fixed- versus floating-point implementation of MIMO-OFDM detector*, 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[79] Kiran Kintali, *Implementing floating-point algorithms in FPGAs or ASICs*, 2018, online resource: https://www.embedded.com/implementing-floating-point-algorithms-in-fpgas-or-asics/.

[80] Gene Frantz, Ray Simar, *Comparing Fixed- and Floating-Point DSPs*, Texas Instruments, White Paper SPRY061, 2004, p 2.

[81] David M. Buehler, Gregory W. Donohoe, *A Software Tool for Designing Fixed-Point Implementations of Computational Data Paths*, Journal of Aerospace Computing, Information, and Communication, Vol. 3, July 2006.

[82] K. Joseph Hass, *Synthesizing optimal fixed-point arithmetic for embedded signal processing*, 2010 53rd IEEE International Midwest Symposium on Circuits and Systems.

[83] Lizhe Tan, Jean Jiang, *Digital Signal Processing - Fundamentals and Applications*, Academic Press, 2019, pp 13-24, pp 112-115.

[84] Luis Chioye, *Leverage coherent sampling and FFT windows when evaluating SAR ADCs (Part 1)*, Texas Instruments, 2020, https://e2e.ti.com/blogs_/archives/b/precisionhub/posts/leverage-coherent-sampling-and-fft-windows-when-evaluating-sar-adcs-part-1.

[85] National Instruments, *Understanding FFTs and Windowing*, National Instruments, White paper, 2019.

[86] Lars Wanhammar, *DSP Integrated Circuits*, Academic Press, 1999, pp 379-383.

[87] Richard G. Lyons, *Understanding Digital Signal Processing*, Prentice Hall PTR, 1st edition, 1996, pp 12-21.

[88] Winser Alexander, Cranos Williams, *Digital Signal Processing Principles, Algorithms and System Design*, Academic Press, 2017, pp 79-82.

[89] Abhishek Seth, Woon-Seng Gan, *Fixed-point square root*, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[90] P. Kornerup, *Digit selection for SRT division and square root*, IEEE Transactions on Computers (Volume: 54, Issue: 3, March 2005), pp 294-303.

[91] Taek-Jun Kwon, Jeff Draper, *Floating-point division and square root implementation using a Taylor-series expansion algorithm with reduced look-up tables*, 2008 51st Midwest Symposium on Circuits and Systems.

[92] Wanming Chu, Yamin Li, *Cost/performance tradeoff of n-select square root implementations*, Proceedings 5th Australasian Computer Architecture Conference. ACAC 2000 (Cat. No.PR00512).

[93] G. M. Phillips, P. J. Taylor, *Theory and Applications of Numerical Analysis*, Academic Press, 1996, Chapter 1 - Introduction.

[94] Ryan G. McClarren, *Computational Nuclear Engineering and Radiological Science Using Python*, Academic Press, 2018, Chapter 12 - Closed Root Finding Methods.

[95] Sophocles J. Orfanidis, *Introduction to Signal Processing*, Prentice Hall, 2nd. edition, 1995, pp 382-395.

[96] Steven W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*, California Technical Pub, 1997, Chapter 15.

[97] David Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys, Volume 23, Issue 1, 1991.

[98] Li Tan, *Digital Signal Processing Fundamentals and Applications*, Elsevier, 1st edition, 2008, pp 417-419.

[99] X. L. Luo, V. Modamio, J. Nyberg, J. J. Valiente-Dobón, Q. Nishada, G. de Angelis, J. Agramunt, F. J. Egea, M. N. Erduran, S. Ertürk, G. de France, A. Gadea, V. González, A. Goasduff, T. Hüyük, G. Jaworski, M. Moszynski, A. Di Nitto, M. Palacz, P.-A. Söderström, E. Sanchis, A. Triossi, R. Wadsworth, *Pulse pile-up identification and reconstruction for liquid scintillator based neutron detectors*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 897, 21 July 2018, Pages 59-65.

[100] Mark A. Haidekker, *Linear Feedback Controls The Essentials*, Elsevier, 2020, p 257.

[101] P. Grybos, *Front-end Electronics for Multichannel Semiconductor Detector Systems*, EuCARD Editorial Series on Accelerator Science and Technology, Vol. 8, 2012, pp 31-37.

[102] Flavio Loddo, *Front-end Electronics for Gas Detectors*, Frascati Detector School, Laboratori Nazionali di Frascati (LNF), 21-23 March 2018, p 27.

[103] Huai-Qiang Zhang, Zhuo-Dai Li, Bin Tang, He-Xi Wu, *Optimal parameter choice of CRRC^m digital filter in nuclear pulse processing*, Nuclear Science and Techniques volume 30, Article number: 108 (2019).

[104] M. Nakhostin, *Recursive Algorithms for Real-Time Digital CR(RC)^n Pulse Shaping*, IEEE Transactions on Nuclear Science (Volume: 58, Issue: 5, Oct. 2011), pp 2378-2381.

[105] Xu Hong, Huaiping Wang, Jianbin Zhou, Xiaoyan Yang, Min Wang, Yingjie Ma, Wei Zhou, Yi Liu, Xing Zhu, *Peak tailing cancellation techniques for digital CR-(RC)^n filter*, Applied Radiation and Isotopes, Volume 167, January 2021, 109471.

[106] Yinyu Liu, Jinglong Zhang, Lifang Liu, Shun Li, Rong Zhou, *Implementation of real-time digital shaping filter on FPGA for gamma-ray spectroscopy*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 906, 21 October 2018, Pages 1-9.

[107] C. H. Nowlin, J. L. Blankenship, *Elimination of Undesirable Undershoot in the Operation and Testing of Nuclear Pulse Amplifiers*, AIP Publishing, Review of Scientific Instruments 36, 1830 (1965).

[108] Microsemi, *DSP Design Flows For Microsemi FPGAs*, Application Note AC384.

[109] Valentin T. Jordanov, Glenn F. Knoll, *Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 345, Issue 2, 15 June 1994, Pages 337-345.

[110] M. Dambacher, A. Zwerger, A. Fauler, C. Disch, U. Stöhlker, M. Fiederle, *Measurements with coplanar grid (Cd,Zn)Te detectors and development of the GMCA (Gamma-ray analysis digital filter Multi Channel Analyzer)*, SPIE Optical Engineering + Applications, Proceedings Volume 7805, Hard X-Ray, Gamma-Ray, and Neutron Detector Physics XII; 78051W (2010).

[111] J. Vesic, M. Vencelj, K. Strnisa, D. Savran, *Adaptive triggering for scintillation signals*, Journal of Physics: Conference Series, Volume 599, FAIRNESS 2014.

[112] Wei Liu, Zhi Deng, Fule Li, Xian Gu, Yulan Li, Huirong Qi, *Development of a low power and high integration readout ASIC for time projection chambers in 65 nm CMOS*, Journal of Physics: Conference Series, Volume 1498, Micro-Pattern Gaseous Detectors Conference 2019.

[113] B. J. Kim, K. B. Lee, J. M. Lee, S. H. Hwang, D. H. Heo, K. H. Han, *Design of optimal digital filter and digital signal processing for a CdZnTe high resolution gamma-ray system*, Applied Radiation and Isotopes, Volume 162, August 2020, 109171.

[114] Guoqiang Zeng, Jian Yang, Tianyu Hu, Liangquan Ge, Xiaoping Ouyang, Qingxian Zhang, Yi Gu, *Baseline restoration technique based on symmetrical zero-area trapezoidal pulse shaper*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 858, 21 June 2017, Pages 57-61.

[115] Kim Bjerge, Hans O. U. Fynbo, Jacob G.Johansen, *A system on programmable chip design of a digitizer with improved trapezoidal filter validation*, Microprocessors and Microsystems, Volume 65, March 2019, Pages 7-13.

[116] Vahid Esmaeili-sani, Ali Moussavi-zarandi, Nafiseh Akbar-ashrafi, Behzad Boghrati, Hossein Afarideh, *Neutrongamma discrimination based on bipolar trapezoidal pulse shaping using FPGAs in NE213*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 694, 1 December 2012, Pages 113-118.

[117] B. W. Loo, F. S. Goulding, D. Gao, *Ballistic deficits in pulse shaping amplifiers*, IEEE Transactions on Nuclear Science (Volume: 35, Issue: 1, Feb. 1988), pp 114-118.

[118] Joel Silva, *Digital Signal Analysis for CsI(Tl) Detectors and the Active-Target at R^3B*, PhD thesis, Gutenberg-Universität Mainz, 2016, pp 20-23.

[119] W. Gao, X. Li, J. Yin, C. Li, D. Gao, Y. Hu, *A novel front-end electronic system with full-customized readout ASIC and post digital pulse shaping for CZT-based PET imaging*, 2015 IEEE International Conference on Imaging Systems and Techniques (IST).

[120] CAEN, *WP2081, Digital Pulse Processing in Nuclear Physics*, White paper, August 2011, Rev. 3.

[121] B. Joly, G. Montarou, J. Lecoq, G. Bohner, M. Crouau, M. Brossard, P-E. Vert, *Test and Optimization of Timing Algorithms for PET Detectors with Digital Sampling Front-end*, 2008 Nuclear Science Symposium, Medical Imaging Conference and 16th Room Temperature Semiconductor Detector Workshop, Dresden : Allemagne (2008).

[122] O. Barnabà, Y.B. Chen, G. Musitelli, R. Nardò, G.L. Raselli*, M. Rossella, P. Torre, *A full-integrated pulse-shape discriminator for liquid scintillator counters*, Nuclear Instruments and Methods in Physics Research A 410 (1998) 220228.

[123] Sara Garbolino, G. Aglieri Rinella, V. Carassiti, A. Ceccucci, E. Cortina, J. Daguin, G. Dellacasa, M. Fiorini, P. Jarron, J. Kaplon, A. Kluge, F. Marchetto, E. Martin, A. Mapelli, G. Mazza, M. Morel, M. Noy, G. Nüssle, P. Petagna, L. Perktold, F. Petrucci, A. Cotta Ramusino, P. Riedler, A. Rivetti, R. Wheadon, *Recent Experimental Results from the NA62 Gigatracker Prototypes: an Hybrid Silicon Pixel Detector with 100 ps Time Resolution*, The 2011 Europhysics Conference on High Energy Physics, EPS-HEP 2011, July 21-27, 2011 Grenoble, Rhône-Alpes, France.

[124] Nicola Da Dalt, Ali Sheikholeslami, *Understanding Jitter and Phase Noise - A Circuits and Systems Perspective*, Cambridge University Press, February 2018, pp 1-14.

[125] Eduard Säckinger, *Analysis and Design of Transimpedance Amplifiers for Optical Receivers*, 2018 John Wiley & Sons, pp 421-439.

[126] S Baron, T Mastoridis, J Troska and P Baudrenghien, *Jitter impact on clock distribution in LHC experiments*, Published 20 December 2012, Journal of Instrumentation, Volume 7, December 2012, pp 3-5.

[127] C. Azeredo-Leme, *Clock jitter effects on sampling: a tutorial*, IEEE Circuits and Systems Magazine, 11(3):2637, 2011, p 36.

[128] Glenn F. Knoll, *Radiation Detection and Measurement - 3rd ed*, John Wiley & Sons, Inc., pp 659-665.

[129] Behnam Analui, James F. Buckwalter, Ali Hajimiri, *Data-Dependent Jitter in Serial Communications*, IEEE Transactions on Microwave Theory and Techniques (Volume: 53, Issue: 11, Nov. 2005).

[130] Manuel D. Rolo, Luis N. Alves, Ernesto V. Martins, Angelo Rivetti, Marcelino B. Santos, Joao Varela, *A low-noise CMOS front-end for TOF-PET*, Journal of Instrumentation, Volume 6, September 2011, pp 9-10.

[131] Adriano Lai, *Pixel front-end electronics for high time resolution*, VIII International Course "Detectors and Electronics for High Energy Physics, Astrophysics, Space and Medical Physics", INFN National Laboratory of Legnaro, April 1-5, 2019.

[132] Francesca Carnesecchi, *Experimental study of the time resolution for particle detectors based on MRPC, SiPM and UFSD technologies*, PhD thesis, 2018.

[133] Gianfranco Dalla Betta, *Principles of semiconductor detectors*, VIII International Course "Detectors and Electronics for High Energy Physics, Astrophysics, Space and Medical Physics", INFN National Laboratory of Legnaro, April 1-5, 2019, p 41.

[134] Renesas Electronic Corporation, *The Role of Jitter in Timing Signals*, White paper, 2019, Rev.1.0 Mar 2020, p 11.

[135] Renesas Electronic Corporation, *Understanding Jitter Units*, Application note AN-815, REVISION A, 2014, p 4.

[136] Renesas Electronic Corporation, *What is Phase Jitter? A Brief Tutorial by IDT*, Online tutorial, https://www.renesas.com/us/en/video/what-phase-jitter-brief-tutorial-idt.

[137] Silicon Laboratories, *Calculating total output jitter for PLLs*, Application note AN56, 2012, Rev. 0.3, p 2.

[138] A. Baschirotto, G. Cocciolo, M. De Matteis, A. Giachero, C. Gotti, M. Maino, G. Pessina, *A fast and low noise charge sensitive preamplifier in 90 nm CMOS technology*, Journal of Instrumentation, Volume 7, January 2012.

[139] A. Balakrishnan, *On the problem of time jitter in sampling*, IRE Transactions on Information Theory (Volume: 8, Issue: 3, April 1962), pp 226-236.

[140] NumPy, *The Role of Jitter in Timing Signals*, https://numpy.org/doc/stable/reference/random/generated/numpy.random.randn.html.

[141] Yan Duan, Hsinho Wu, Masashi Shimanouchi, Mike Peng Li, Degang Chen, *A Low-Cost Comparator-Based Method for Accurate Decomposition of Deterministic Jitter in High-Speed Links*, IEEE Transactions on Electromagnetic Compatibility (Volume: 61, Issue: 2, April 2019), pp 521-531.

[142] Nelson Ou, Touraj Farahmand, Andy Kuo, Sassan Tabatabaei, André Ivanov, *Jitter models for the design and test of Gbps-speed serial interconnects*, IEEE Design & Test of Computers (Volume: 21, Issue: 4, July-Aug. 2004), pp 302-313.

[143] Mike Tranchemontagne (Tektronix), *Jitter Basics, Advanced, and Noise Analysis*, Presentation on IEEE volunteer tools (vtools) site, 2016, p 6.

[144] A. Telba, J.M. Noras, M. Abou El Ela, B. Almashary, *Simulation technique for noise and timing jitter in phase locked loop*, Proceedings. The 16th International Conference on Microelectronics, 2004. ICM 2004, pp 501-504.

[145] ORTEC, *Fast-Timing Discriminator Introduction*, Online resource: https://www.ortec-online.com/-/media/ametekortec/other/fast-timing-discriminator-introduction.pdf?la=en.

[146] Mark A. Nelson, Brian D. Rooney, Derek R. Dinwiddie, Glen S. Brunson, *Analysis of digital timing methods with BaF_2 scintillators*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 505, Issues 12, 1 June 2003, pp 324-327.

[147] F. Loddo, *Basic electronic concepts for particle physics and beyond*, International Workshop on HEP, Tirana 2012, slide 47.

[148] B. Vojnovic, *Error minimization of sensor pulse signal delay-time measurements*, 2002 23rd International Conference on Microelectronics. Proceedings (Cat. No.02TH8595).

[149] T. J. Paulus, *Timing Electronics and Fast Timing Methods with Scintillation Detectors*, IEEE Transactions on Nuclear Science (Volume: 32, Issue: 3, June 1985), 1242-1249.

[150] Suvendu Bose, Chandi Charan Dey, Bedanta Kumar Sinha, Rangalal Bhattacharya, *Performance of extrapolated leading edge timing for fast coincidence with a large-volume HPGe detector*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 295, Issues 12, 1 October 1990, Pages 219-223.

[151] Jean-Francois Genat, Gary Varner, Fukun Tang, Henry Frisch, *Signal processing for picosecond resolution timing measurements*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 607, Issue 2, 11 August 2009, Pages 387-393.

[152] Junwei Du, Jeffrey P. Schmall, Martin S. Judenhofer, Kun Di, Yongfeng Yang, Simon R. Cherry, *A Time-Walk Correction Method for PET Detectors Based on Leading Edge Discriminators*, IEEE Transactions on Radiation and Plasma Medical Sciences (Volume: 1, Issue: 5, Sept. 2017), 385-390.

[153] S. V. Paulauskas, M. Madurga, R. Grzywacz, D. Miller, S. Padgett, H. Tan, *A digital data acquisition framework for the Versatile Array of Neutron Detectors at Low Energy (VANDLE)*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 737, 11 February 2014, Pages 22-28.

[154] D. Breton, V. De Cacqueray, E. Delagnes, H. Grabas, J. Maalmi, N. Minafra, C. Royon, M. Saimpert, *Measurements of timing resolution of ultra-fast silicon detectors with the SAMPIC waveform digitizer)*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 835, 1 November 2016, Pages 51-60.

[155] D. A. Gedcke, W. J. McDonald, *A constant fraction of pulse height trigger for optimum time resolution*, Nuclear Instruments and Methods Volume 55, 1967, pp 377-380.

[156] M. R. Maier, P. Sperr, *On the construction of a fast constant fraction trigger with integrated circuits and application to various photomultipliertubes*, Nuclear Instruments and Methods Volume 87, Issue 1, 1 October 1970, pp 13-18.

[157] L. Bardelli, G. Poggi, M. Bini, G. Pasquali, N. Taccetti, *Time measurements by means of digital sampling techniques: a study case of 100 ps FWHM time resolution with a 100 MSample/s, 12bit digitizer*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 521, Issues 23, 1 April 2004, Pages 480-492.

[158] Pedro Guerra, Juan E. Ortuño, George Kontaxakis, Maria J. Ledesma, Juan J. Vaquero, Manuel Desco, Andres Santos, *Digital timing in positron emission tomography*, 2006 IEEE Nuclear Science Symposium Conference Record.

[159] A. Fallu-Labruyere, H. Tan, W. Hennig, W.K. Warburton, *Time Resolution Studies using Digital Constant Fraction Discrimination*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 579, Issue 1, 21 August 2007, Pages 247-251.

[160] CAEN, *Time Measurements with CAEN Waveform Digitizers*, Application Note AN3251, 2015.

[161] C. J. Prokop, S. N. Liddick, N. R. Larson, S. Suchyta, J. R. Tompkins, *Optimization of the National Superconducting Cyclotron Laboratory Digital Data Acquisition System for use with fast scintillator detectors*, Nuclear Instruments

and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 792, 21 August 2015, Pages 81-88.

[162] Baptiste Joly, Gérard Montarou, Jacques Lecoq, Gérard Bohner, Michel Crouau, Michel Brossard, Pierre-Etienne Vert, *An Optimal Filter Based Algorithm for PET Detectors With Digital Sampling Front-End*, IEEE Transactions on Nuclear Science (Volume: 57, Issue: 1, Feb. 2010).

[163] Hicham Semmaoui, Marc-André Tetrault, Roger Lecomte, Réjean Fontaine, *Signal Deconvolution Concept Combined With Cubic Spline Interpolation to Improve Timing With Phoswich PET Detectors*, IEEE Transactions on Nuclear Science (Volume: 56, Issue: 3, June 2009), pp 581-587.

[164] J. Torres, A. Aguilar, R. Garcia-Olcina, P. A. Martinez, J. Martos, J. Soret, J. M. Benlloch, P. Conde, A. J. Gonzalez, F. Sanchez, *Time of Flight measurements in PET systems using FPGAs*, 2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC).

[165] Hicham Semmaoui, Nicolas Viscogliosi, François Bélanger, J.-B. Michaud, Catherine M. Pepin,Roger Lecomte, Réjean Fontaine, *Crystal Identification Based on Recursive-Least-Squares and Least-Mean-Squares Auto-Regressive Models for Small Animal Pet*, IEEE Transactions on Nuclear Science (Volume: 55, Issue: 5, Oct. 2008), pp 2450-2454.

[166] Réjean Fontaine, François Lemieux, Nicolas Viscogliosi, Marc-André Tétrault, Mélanie Bergeron, Joel Riendeau, Philippe Bérard, Jules Cadorette, Roger Lecomte, *Timing Improvement by Low-Pass Filtering and Linear Interpolation for the LabPET Scanner*, IEEE Transactions on Nuclear Science (Volume: 55, Issue: 1, Feb. 2008), pp 34-39.

[167] John R. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, 1996, 2nd. edition, pp 182-199.

[168] Timmy Siauw, Alexandre Bayen, *An Introduction to MATLABő Programming and Numerical Methods for Engineers*, Academic Press, 2015.

[169] S.F. Obermann; M.J. Flynn, *Division algorithms and implementations*, IEEE Transactions on Computers (Volume: 46, Issue: 8, Aug 1997), pp 833-854.

[170] M.J. Flynn, *Division*, Online Stanford lectures: https://web.stanford.edu/-class/ee486/doc/chap5.pdf.

[171] S.F. Oberman, M.J. Flynn, *Design issues in division and other floating-point operations*, IEEE Transactions on Computers (Volume: 46, Issue: 2, Feb 1997), pp 154-161.

[172] Robert E. Goldschmidt, *Applications of division by convergence*, Thesis (M.S.), Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1964.

[173] Inwook Kong; Earl E. Swartzlander, *A Rounding Method to Reduce the Required Multiplier Precision for Goldschmidt Division*, IEEE Transactions on Computers (Volume: 59, Issue: 12, Dec. 2010), pp 1703-1708.

[174] Peter Malík, *High Throughput Floating-Point Dividers Implemented in FPGA*, 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems.

[175] IEEE standard, *1800-2017 - IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language*, 2018, IEEE.

[176] Stefan Haenzsche, Sebastian Höppner, Georg Ellguth, Rene Schüffny, *12-b 4-MS/s SAR ADC With Configurable Redundancy in 28-nm CMOS Technology*, IEEE Transactions on circuits and systemsII: Express Briefs, VOL. 61, NO. 11, November 2014.

[177] Chuan-Ping Yan, Guang-Jun Li, Qiang Li, *A Fast Correlation Based Background Digital Calibration for Pipelined ADCs*, 2012 IEEE Asia Pacific Conference on Circuits and Systems.

[178] Alma Delić-Ibukić, Donald M. Hummels, *Continuous Digital Calibration of Pipeline A/D Converters*, 2005 IEEE Instrumentationand Measurement Technology Conference Proceedings.

[179] Bei Peng, Hao Li, Pingfen Lin, Yun Chiu, *An Offset Double Conversion Technique for Digital Calibration of Pipelined ADCs*, IEEE Transactions on Circuits and Systems II: Express Briefs (Volume: 57, Issue: 12, Dec. 2010).

[180] Andrea Di Salvo, *Design of a 12-bit SAR ADC with digital self-calibration for radiation detectors front-ends*, 2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME).

[181] Texas Instruments, *CMOS Power Consumptionand C_{pd} Calculation*, 1997, Texas Instruments.

[182] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, *Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits*, Proceedings of the IEEE (Volume: 91, Issue: 2, Feb. 2003).

[183] Qink K. Zhu, *Power distribution network design for VLSI*, 2004, John Wiley & Sons, Inc., Chapters 4 and 5.

[184] Lane Brooks, Hae-Seung Lee, *Background Calibration of Pipelined ADCs Via Decision Boundary Gap Estimation*, IEEE Transactions on Circuits and Systems I: Regular Papers (Volume: 55, Issue: 10, Nov. 2008).

[185] Altera, *Introduction to Single-Event Upsets*, WP-01206-1.0, White Paper, 2013, Altera Corporation.

[186] Szymon Kulis, *Digital synthesis for rad-hard components, Single Event Upsets mitigation techniques with TMRG tool*, Ecole de Microélectronique IN2P3, 2017, Bénodet.

[187] Fan Wang, Vishwani D. Agrawal, *Single Event Upset: An Embedded Tutorial*, 21st International Conference on VLSI Design (VLSID 2008).

[188] G. C. Messenger, *Collection of Charge on Junction Nodes from Ion Tracks*, IEEE Transactions on Nuclear Science (Volume: 29, Issue: 6, Dec. 1982).

[189] G. R. Srinivasan, P. C. Murley, H. K. Tang, *Accurate, predictive modeling of soft error rate due to cosmic rays and chip alpha radiation*, Proceedings of 1994 IEEE International Reliability Physics Symposium.

[190] C. M. Hsieh, P. C. Murley, R. R. O'Brien, *Dynamics of Charge Collection from Alpha-Particle Tracks in Integrated Circuits*, 19th International Reliability Physics Symposium (1981).

[191] Jonathan Swingler, *Reliability Characterisation of Electrical and Electronic Systems*, 2015, Woodhead Publishing, 1st Edition, p 132.

[192] Paul E. Dodd, Lloyd W. Massengill, *Basic mechanisms and modeling of single-event upset in digital microelectronics*, IEEE Transactions on Nuclear Science (Volume: 50, Issue: 3, June 2003) .

[193] Fernanda Lima Kastensmidt, Luigi Carro, Ricardo Reis, *Fault-Tolerance Techniques for SRAM-Based FPGAs*, 2006, Springer US.

[194] F.L. Kastensmidt, L. Sterpone, L. Carro, M.S. Reorda, *On the optimal design of triple modular redundancy logic for SRAM-based FPGAs*, Design, Automation and Test in Europe, 2005.

[195] W. G. Brown; J. Tierney; R. Wasserman, *Improvement of Electronic-Computer Reliability through the Use of Redundancy*, IRE Transactions on Electronic Computers (Volume: EC-10, Issue: 3, Sept. 1961), pp 1-3.

[196] Himanshu Bansal, *Bounds in Placement*, White paper, https://www.design-reuse.com/articles/48441/bounds-in-placement.html.

[197] Synopsys, *IC Compiler^{TM} II Implementation User Guide*, Version L-2016.03-SP4, 2016, pp 86-89.

[198] Mentor Graphics Corporation, *ModelSimő User's Manual*, Software Version 10.4c, p 481.

[199] David Salomon, *Data Compression - The Complete Reference*, Springer-Verlag London, Fourth Edition, 2007.

[200] Christian Patauner, Alessandro Marchioro, Sandro Bonacini, Attiq Ur Rehman, Wolfgang Pribyl, *A lossless data compression system for a real-time application in HEP data acquisition*, 2010 17th IEEE-NPSS Real Time Conference.

[201] David A. Huffman, *A Method for the Construction of Minimum-Redundancy Codes*, Proceedings of the IRE (Volume: 40, Issue: 9, Sept. 1952).

[202] Ida Mengyi Pu, *Fundamental Data Compression*, Butterworth-Heinemann, 2005.

[203] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann, fifth edition, 2017.

[204] Cornell University, Department of Mathematics, *English Letter Frequency (based on a sample of 40,000 words)*, http://pi.math.cornell.edu/ mec/2003-2004/cryptography/subs/frequencies.html.

[205] Doron Gottlieb, Steven A. Hagerth, Philippe G. H. Lehot, Henry S. Rabinowitz, *A classification of compression methods and their usefulness for a large data processing center*, AFIPS, National Computer Conference, 1975.

[206] CMS, *Compact Muon Solenoid*, https://cms.cern/detector.

[207] W. Adam et al, *The DAQ and Control System for the CMS Phase-1 Pixel Detector*, Journal of Instrumentation, Volume 14, 2019.

[208] W. Adam et al, *Beam Test Performance of Prototype Silicon Detectors for the Outer Tracker for the Phase-2 Upgrade of CMS*, Journal of Instrumentation, Volume 15, 2020.

[209] W. Adam et al., *Experimental Study of Different Silicon Sensor Options for the Upgrade of the CMS Outer Tracker*, Journal of Instrumentation, Volume 15, 2020.

[210] RD53A, *The RD53A Integrated Circuit*, https://cds.cern.ch/record/2287593, CERN-RD53-PUB-17-001, Version 3.51, 2019.

[211] Stamatios Poulios, Konstantin Androsov, Massimo Minuti, Fabrizio Palla, *Lossless data compression for the HL-LHC silicon pixel detector readout*, 2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST).

[212] Ming Yang Kao (editor), *Encyclopedia of Algorithms*, Springer-Verlag US, 2008, pp 145-150.

[213] Giuseppe Baruffa, Pisana Placidi, Andrea Di Salvo, Sara Marconi, Andrea Paternò, *An Improved Algorithm for On-Chip Clustering and Lossless Data Compression of HL-LHC Pixel Hits*, 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC).