Reliable event-based techniques for ASK demodulation in NFC devices

Alexis Rodrigo Iga Jadue

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : NANO ELECTRONIQUE ET NANO TECHNOLOGIES

Arrêté ministériel : 25 mai 2016

Présentée par

Alexis Rodrigo IGA JADUE

Thèse dirigée par **Laurent FESQUET**,Maître de Conférence
Grenoble INP / Phelma, Université Grenoble Alpes
et co-encadrée par **Sylvain ENGELS**, STMicroelectronics

préparée au sein du **Laboratoire Techniques de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Techniques fiables basées sur événements pour la démodulation ASK des dispositifs NFC

Reliable event-based techniques for ASK demodulation in NFC devices

Thèse soutenue publiquement le **24 février 2022**,
devant le jury composé de :

Monsieur Laurent FESQUET
MAITRE DE CONFERENCE HDR, Université Grenoble Alpes, Directeur de thèse
Monsieur Luc HEBRARD
PROFESSEUR DES UNIVERSITES, Université de Strasbourg, Président
Monsieur Pascal NOUET
PROFESSEUR DES UNIVERSITES, Université de Montpellier, Rapporteur
Monsieur Gilles JACQUEMOD
PROFESSEUR DES UNIVERSITES, Université Côte d'Azur, Rapporteur
Monsieur Emil NOVAKOV
PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes, Examinateur

Remerciements

Cette belle aventure du doctorat en territoire français, a commencé bien avant que ce que l'on pourrait imaginer. C'était l'année 1982, et mon frère, âgé de 3 ans, refusait de rentrer faire son examen d'admission en maternelle dans une école anglaise dans notre quartier à Santiago du Chili. Ceci fût un événement clé de cette histoire, car quelques semaines plus tard, il acceptait sans soucis de faire cet examen dans l'école française du même quartier. C'était le début de ma connexion indissoluble avec la culture française, sa langue, ces habitudes et sa vaste communauté installée au Chili.

Toujours passionné par l'électronique, en 2009, je cherchais à faire un master en microélectronique en France, car je parlais déjà la langue, et ça me permettait d'avoir une expérience à l'étranger dans une culture plutôt familiale et habituelle. Parmi les trois options que j'avais entre Montpellier, Lyon et Grenoble, j'ai choisi cette dernière pour ces belles montagnes et sa grande réputation dans le domaine. Pendant la période du master MNE, j'ai eu la chance de rencontrer d'excellentes personnes parmi les enseignants, avec qui j'ai toujours maintenu le contact, et qui m'ont toujours donné leur support pour tous les projets que j'ai entrepris : Olivier Rossetto, Gilles Sicard et Gilbert Vincent. Aussi, pendant la période de stage en 2010, j'ai connu Rodrigo Possamai, qui faisait son doctorat, et avec qui nous avons partagé plusieurs moments ensemble, et avec qui j'ai aussi maintenu le contact.

Quelques années plus tard, en avril 2015, nous avons décidé de visiter l'Europe avec Giselle, ma chère femme, amie, collègue, mais surtout copine de multiples aventures. Quand elle a insisté de visiter Grenoble pour voir une fois de plus notre ville adoptive ainsi que quelques amis, j'ai décidé de contacter Gilles Sicard, Olivier Rossetto et Rodrigo Possamai pour déjeuner ensemble. Dans ce déjeuner, Possamai, qui était déjà devenu membre permanent de l'équipe CIS au laboratoire TIMA, m'a parlé du projet européen THINGS2DO que l'équipe venait d'obtenir. L'équipe avait besoin de quelques collaborateurs, donc il m'a invité à participer. Après quelques semaines de réflexion, nous avons finalement accepté son invitation. Merci beaucoup Rodrigo Possamai pour cette invitation et aussi pour m'avoir fait confiance. Ça a été le début d'une belle

histoire. Je tiens à remercie profondément mon ancien chef, Eduardo Carrasco, qui m'a donné tout son support quand je lui ai transmis que je partais en France, et m'a beaucoup aidé. Muchas gracias Eduardo por tu gran apoyo y ayuda para iniciar esta aventura.

Quelques mois plus tard je connaîtrais, par visioconférence, la personne clé de cette histoire, Laurent Fesquet, qui déjà se battait avec l'administration pour réussir à m'embaucher dans son équipe dans des conditions adéquates. Laurent, je te remercie profondément pour le support, la confiance et l'amitié que tu m'as accordé dans chaque situation, chaque projet, chaque conversation pendant cette aventure, et notamment pendant ces relectures très denses et fatigantes du manuscrit de thèse. Aussi, un grand merci Laurent d'avoir accepté un ingénieur déjà trop mûr (38 ans) parmi tes doctorants, ce n'est pas souvent que l'on voit une situation pareille. Je remercie fortement aussi à Sylvain Engels pour ta générosité, ton ouverture d'esprit par rapport à l'accueil et support des nouvelles idées, pour l'opportunité que tu m'as donné chez STMicroelectronics, pour les moments de partage, les multiples brainstormings et ta capacité de filtrage quand je te bombardais avec plusieurs nouvelles idées.

Merci beaucoup aussi à la communauté brésilienne de notre groupe, avec qui j'ai passé de moments inoubliables : les gros câlins matinales, conversations profondes, nos plus grosses fêtes Grenobloises, les cocktails, les cafés et surtout l'amitié d'Otto, l'humour sérieux de Raphael, les partages avec Leonel (merci particulier pour cette nuit blanche pendant le premier testchip), les heures infinies de travail avec Thiago avec son humour particulier, son amitié, ses danses et ces « franguiçies » sans fin. Aussi merci beaucoup aux membres de la deuxième vague, Matheus pour les conversations, les barbecues, et le partage de grille et Nucléos. Aussi merci à Ricardo pour son amitié, les partages au ski, les « easy-lovers » descendants les pistes, les « bobuns malucos » et les courses au bord de l'Isère. Merci aussi à Renato, pour les conversations, les soirées musicales, les bars, mais surtout ton spécial conseil d'utiliser les comparateurs « Strong-Arm », qu'avant notre conversation, je ne connaissais même pas.

Également, je remercie beaucoup Jean Simatic, pour sa grande amitié, les super brainstormings que nous avons partagés, les conversations, notre partage de connaissance sans fin, les soirées, la dent de Crolles, la lasagne de chocolat

« exquisita » et les shots. Merci à Karim pour les conversations, le partage et les soirées d'équipe chez lui avec cette belle vue depuis Saint-Martin-Le-Vinoux. Je remercie beaucoup aussi Amani, pour son amitié, les conversations profondes, les empanadas et les moments partagés. Je remercie beaucoup aussi Grégoire, pour tout le partage, les conversations techniques ainsi que personnelles, les grands brainstormings, son amitié, sa sympathie et la bonne ambiance de toujours. Aussi, je remercie Sophie pour le temps partagé ensemble, les conversations et la découverte du restaurant libanais avec Adrien. Je remercie Assia pour les conversations, le partage technique et l'ambiance agréable toujours.

Je remercie vivement mes collègues plus récents, Jérémy, Yoan, Mohamed, Olivier, Marco, Hasan, Diana, Xavier, Rosalie, Nils, Ankush, Dayana, Giovanni, Denis, Andrés, Ángel, pour tous les bons moments, les cafés, les restaurants, les bars, les conversations infinies, les brainstormings, les partages, les testchips, les barbecs et tous les types de différents échanges techniques comme personnelles.

Merci beaucoup à tous les membres du TIMA, Anne-Laure, Salvador, Raoul, Skandar, Stéphane, Manuel, Laurence, Youness, Aurore, Mamadou pour tout l'aide qu'il nous a donné, les conversations et son amitié, David, Daniel, Marc et les moments partagés à l'escalade, Ahmed et Fred pour votre support ces années, Florence, Sylvain Bourdel pour tout le support pendant mes CSI, mais notamment pour la confiance que tu m'as accordée.

Je remercie spécialement les membres techniques de nos plateformes de CAO au CIME, Mohamed, Robin, Abdelhamid et Alejandro, pour le magnifique support pendant toutes ces années, l'amitié, les très bons moments, les cafés, les partages de connaissances et bien sûr toutes les conversations. Je remercie beaucoup aussi les membres administratifs du CIME, Lorraine, Olivier, Ahmad, Déborah, pour tous les sourires, les blagues, les conversations et les moments toujours agréables.

Merci aussi à tous les membres du CMP, Christelle, Yann, Jean-François, Kholdoun, Aurélien, Romain, Isabelle, Patricia, François, Nicolas, Chantal, Lyubomir, pour votre support toutes ces années, les conversations et les échanges agréables et intéressants.

Je remercie aussi à tous les membres de l'équipe à STMicroelectronics, Marc, Franck, les Thiery, Alain, Guillaume, Venance, Andrei, Quentin et Sophie.

Sylvain et Laurent, je vous remercie particulièrement pour tous ces bons moments d'échange, de partages d'idées, d'amitié, de convivialité, les cafés, les conversations, l'énorme confiance que vous m'avez accordée (notamment dans ce moment de difficulté bien particulier), et bien sûr tout le support que vous m'avez donné depuis même avant de commencer ce défi.

Je remercie aussi mes chers amis qui m'ont toujours fait passer un excellent moment, pour ainsi maintenir les énergies et surmonter les moments difficiles, Ferenc, Rafael Della Giustina, Elodie, Migue, Margot, Mauricio, Francesco, Julio, Edith, Luana, Tatiana, Munique, Mathieu, Karen, Romain, Rudy, Pascal, et tous les coéquipiers des club de tennis de Saint Nazaire Les Eymes ainsi que du GUC.

Agradezco también a todos mis amigos cercanos que me apoyaron desde la adolescencia, confiados en que llegará un día en que exista la "Unidad Iga", jeje, quién sabe si algún día se pueda hacer realidad. También a los amigos de la Universidad que me apoyaron siempre en esta travesía, confiando en que lo conseguiría. También agradezco a mi familia cercana, que siempre me ha apoyado a cumplir mis sueños, a mi mamá, mi hermano, mi cuñada, mis queridos sobrinitos, Coni y Diego, y mi papá, por todo el amor que siempre me han entregado, aun estando lejos, siempre sentí su enorme apoyo y cariño. A mis abuelos, por habernos ayudado en nuestra educación y darnos un soporte y apoyo en todo momento. Agradezco también a mis tíos y mis primos que siempre me han entregado su gran cariño. Y a la querida familia Andonie-Estay que siempre me han apoyado y aconsejado.

Finalmente, agradezco especialmente a Giselle, mi compañera incondicional, quién ha sido un apoyo fundamental en todas las aventuras que nos hemos propuesto, en todos los momentos favorables como adversos, así como en las noches que pasé en vela intentando resolver alguno de los tantos enigmas que enfrenté durante el doctorado. Gracias infinitas amor por la concretización de esta gran aventura.

¡¡¡Gracias Totales!!!

Table of Contents

List of Figures

List of Tables

List of Equations

1 Introduction

Nowadays, short-range wireless communication circuits are used in our daily life in several applications such as public transport validation, contactless payment cards, RFID product identification, access cards, and even in medical implants like pacemakers.

The most part of these short-range communication devices uses a very popular protocol called *"Near-Field Communication" (NFC)* that is widely fabricated and sold by important actors in the semiconductor industry like STMicroelectronics, NXP Semiconductors, Qualcomm Technologies, Broadcom, among others [1]. In that context, these competitors are constantly enhancing their products and innovating in the field in order to gain market shares. As announced in [2], the main characteristics for developing and improving these circuits are the reduction of the power consumption, the miniaturization, and the electromagnetic spectrum efficiency.

In the NFC ecosystem, the most popular product is the smartcard. Indeed, smartcards are usually battery-less systems, also called NFC passive devices. This means that its supply voltage is internally generated thanks to an internal harvester using the electromagnetic field emitted by the card reader. In these systems, the recovered energy is very limited. Therefore, the internal modules like the harvesting controller, the voltage limiter, the *"Automatic Gain Control" (AGC)*, the demodulator and the digital processor should smartly be designed to avoid any power issue, mainly due to the power consumption peaks.

The NFC protocol integrates many standards, and all of them use the *"Amplitude Shift Keying" (ASK)* modulation with different "*modulation indexes" (MI)*, bitrates and coding. As explained in [3] and [4], *ASK* demodulation is widely used on passive devices because of its simplicity and its low power consumption. As shown in [3] [4] [5], most of the passive *ASK* demodulation devices are designed using custom analog circuits simultaneously obtaining a controlled power consumption and quite good performances. However, these solutions do not benefit from the technology shrink, and hence the cost reduction. Indeed, when the transistor size is reduced, the passive electronic devices required for analog blocks (resistors, capacitors, inductors) will not

benefit from the technology node shrink, maintaining more or less a similar size, and so contributing to a higher fabrication cost.

On the other hand, the most popular digital *ASK* demodulation solutions implemented in literature [6] use a normal *"Analog-to-Digital Converter" (ADC)* with a high sampling rate, generating a large number of samples to process, and hence a high power consumption. For these reasons, these kinds of solutions are only implemented in battery-powered systems, like smartphones, smartwatches, tablets, *etc*.

For the industry, reducing costs and chip area are very challenging, and normally requires an important investment in person-hours and multidisciplinary teams. Nevertheless, advanced technology nodes are an opportunity to lower power consumption, reduce the costs, and enhance robustness and sensitivity of such systems. In battery-less communication systems, when using advanced technology nodes, the supply voltage reduction helps lowering the power consumption and relaxing the design constraints of the supply-voltage harvester. One of the benefits of digitalizing a system is the reusability of the design when the technology is shrunk. In that case, the testbench, the *"Register Transfer Level" (RTL)* simulation, the timing constraints statements and most of the automated design process are reused.

It is well known in literature that the digital asynchronous circuits are robust against supply voltage and process variations, due to their adaptation capabilities and relaxed timing constraints. These asynchronous solutions are responsive to events generated by the environment and remain idle between those events. Thus, they are very suitable for being connected to *"Event-Based Analog-to-Digital Converters" (EB-ADC)* using a *"Level-Crossing Sample Scheme" (LCSS)* [7] [8], because they produce events (samples) only when a level is crossed. When the appropriate levels are chosen, they capture a reduced number of samples compared to a Nyquist *ADC* based on a fixed sampling period.

After the evaluation of the characteristics exposed in the last paragraphs, the work presented in this manuscript is focused on the development of a novel digital *ASK* demodulation solution integrating the two following considerations:

- The use of an *EB-ADC* with a minimal number of levels generating a low number of samples, and including a *"Time-to-Digital Converter" (TDC)* to measure the time between two consecutive level crossing events.

- The implementation of a digital asynchronous *ASK* demodulator that takes the time measurement of each event as input and performs a demodulation algorithm for the modulation symbol recognition.

In other words, the main goal of this proposal is to present a digital *ASK* demodulation alternative that generates a minimal number of samples, uses the latest technology nodes, consequently operating at a reduced voltage, and as a result, reaching a better energy efficiency compared to their analog alternatives. Moreover, these solutions should also enable a cost reduction of these circuits thanks to the use of the most recent technology nodes. In Addition, the event-based asynchronous approach reduces the number of samples to be processed, limits the power consumption, spread in time the current peaks and therefore relax the energy harvesting constraints used in smartcards. Finally, a positive impact of this work is the opening of new perspectives to relax the constraints of the necessary analog modules used in the smartcards, or even completely remove them. Normally, these modules are an *AGC*, an adaptative impedance solution, a voltage-limiter, an energy-harvesting module, etc.

2 State of the Art

2.1 Introduction

In life, all the communications and interactions with the environment around us are analogical. Therefore, the implemented digital systems need a digital representation of the input analog signals before being processed and perform any calculation with it. In this case, an Analog-to-Digital Converter (ADC) is required. Moreover, if the digital system needs to interact and respond to its environment, a Digital-to-Analog Converter is also necessary (DAC).

As explained in the introductory chapter, this thesis is focused on proposing a novel digital ASK demodulation solution. It is known that any ASK demodulation system in smartcards, needs some fundamental analog modules in order to recover energy, avoid overvoltage peaks, adapt the load connected to the antenna and adjust the amplitude of the *"RF input" (RFin)* signal for being used by the demodulation circuit. As shown in Figure 1, if a digital demodulation solution is employed, the analog module must provide the minimum and maximum amplitude levels of *RFin*, and the voltage references that will be used by the ADC module for a correct analog-to-digital conversion. Finally, the symbol recognition is done thanks to a digital demodulation algorithm.

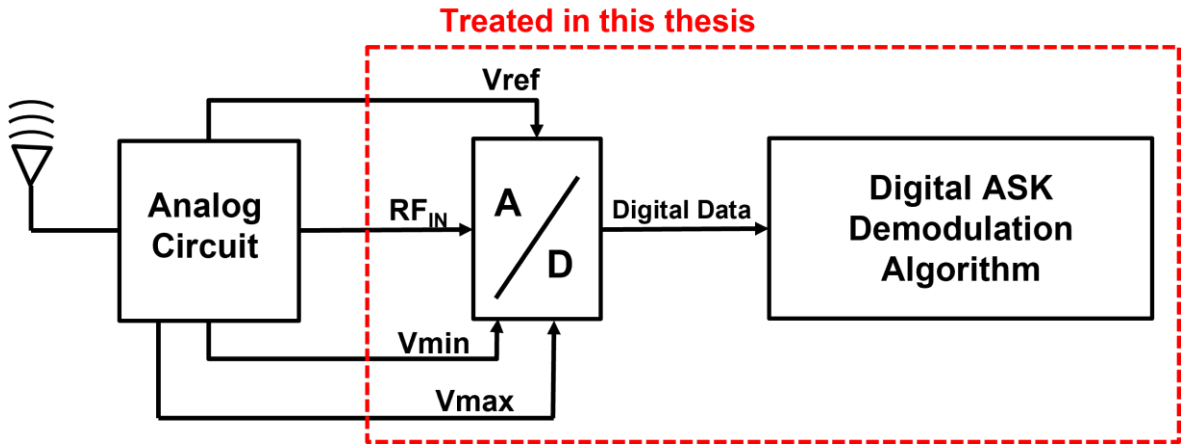


Figure 1: Generic diagram of a digital demodulation system.

The Figure 1 presents a simplified diagram of a digital ASK demodulation system. This thesis is focused on offering a novel ADC solution, using an *EB-ADC* that produces a reduced number of samples, and proposing an adapted digital ASK demodulation solution for processing such samples. The asynchronous circuits are

known to be suited for event-based systems because they are designed to react to any incoming events and stay idle in between.

In this context, the main addressed subjects in this thesis are the development of an **event-based ADC** and a **digital asynchronous ASK demodulator circuit**. The analog circuit parts have not been considered in this thesis. Another team, expert in analog design, should develop this task later.

For the evaluation of the *EB-ADC*, which is composed by a level-crossing detector and a *TDC*, we first review the state of art of the *"Level Crossing Sampling Schemes" (LCSS)* and the basic theory of non-uniform sampling. Then, for the assessment of the TDC, we present some of the related state of art research made by the colleague of the CDSI team at TIMA Laboratory, Assia El Hadbi, during her PhD. The study of the TDC alternatives permits us to find a simple and adapted solution for time stamping data produced by the *EB-ADC*. Then, the state of the art of asynchronous circuits is studied in order to find an event-driven digital technique for executing the demodulation algorithm. Moreover, the basic asynchronous elements have been studied, such as the C-elements (or muller gate) and the linear and non-linear asynchronous flow elements of a controller circuit. The latter allows us to understand how an asynchronous circuit is created. Finally, the ASK modulation and the NFC protocol are revised, showing its basic concepts and solutions, some definitions, and the state of the art of a few digital ASK demodulators.

2.2 Event-based Analog-to-Digital Converter (EB-ADC)

An ADC module is a system providing a digital representation of an analog signal, evaluated on each sample instant. An EB-ADC is a particular type of ADC that captures a sample when a reference voltage level is crossed. The analog signal is digitized on each level crossing, which produces a quantized value representing the time interval between two successive samples. In that particular case, the position and the number of reference voltages should be smartly chosen in order to obtain an appropriate representation of the signal, while generating a minimum number of samples. The next section reviews the theory of non-uniform sampling, the basis of the Level Crossing Sampling Scheme (LCSS) and finally gives some examples.

2.2.1 Non-Uniform Sampling and Level Crossing Sampling Scheme (LCSS)

2.2.1.1 Uniform and non-uniform sampling ADC

As expressed before, when a digital system interacts with an analog environment, the use of an ADC is necessary. Normally, ADC are based on a regular sampling scheme, using a *"Sample and Hold" (S/H)* technique, as shown in Figure 2. In this case, to ensure that all the information of the bandwidth-limited (or band-limited) input signal has been captured by this sampling technique, the sampling rate should follow the Shannon theorem, as presented in Equation 2-1. It establishes that the sampling frequency (f_s) should be more than twice the maximum frequency of the signal (f_{max}).

$$fs \geq 2 \cdot f_{max}$$

Equation 2-1: Shannon theorem

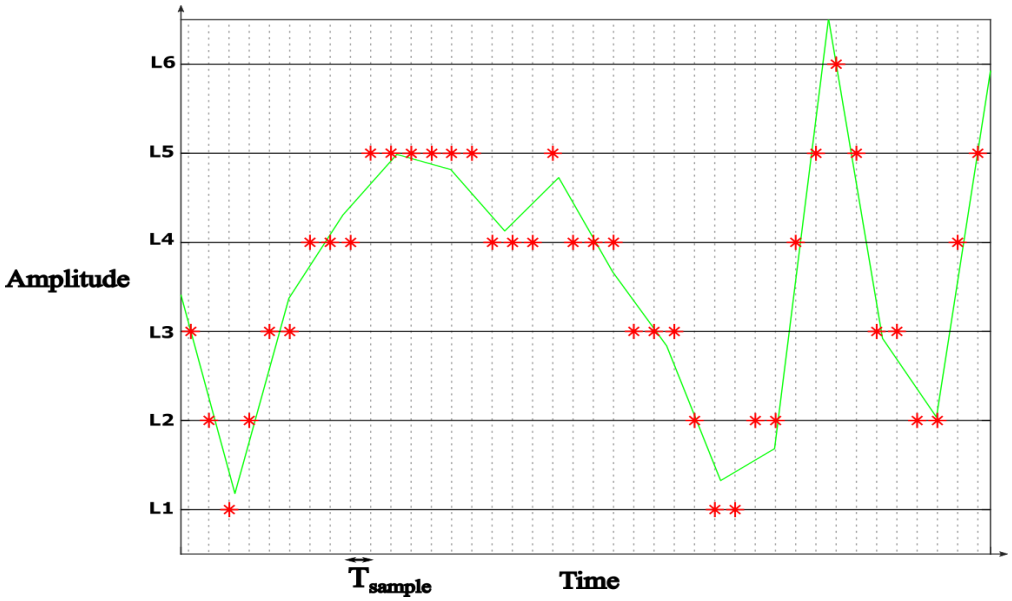


Figure 2: Regular Sampling Scheme Example

If we consider an ideal clock that defines the sampling rate, and an ideal *S/H*, the system will be very precise for getting the time instants of each sample. However, it will incorporate a quantization error due to the noise added during the analog to digital

$$SNR_{dB} = 1{,}76 + 6{,}02 \cdot N$$

Equation 2-2: Signal-to-noise ratio due to quantization error of regular sampled ADC.

conversion. In this case, the signal-to-noise ratio (SNR), defined in Equation 2-2 [9], depends on the effective number of bits N of the converter.

On the other hand, also non-uniform sampling scheme ADC exists. In that kind of ADC, the conversion is triggered at each level crossing, as depicted in Figure 3.

Thus, the system is very precise for obtaining the equivalent voltage level of each sample (L_i). However, there will be a measurement error of the time elapsed between

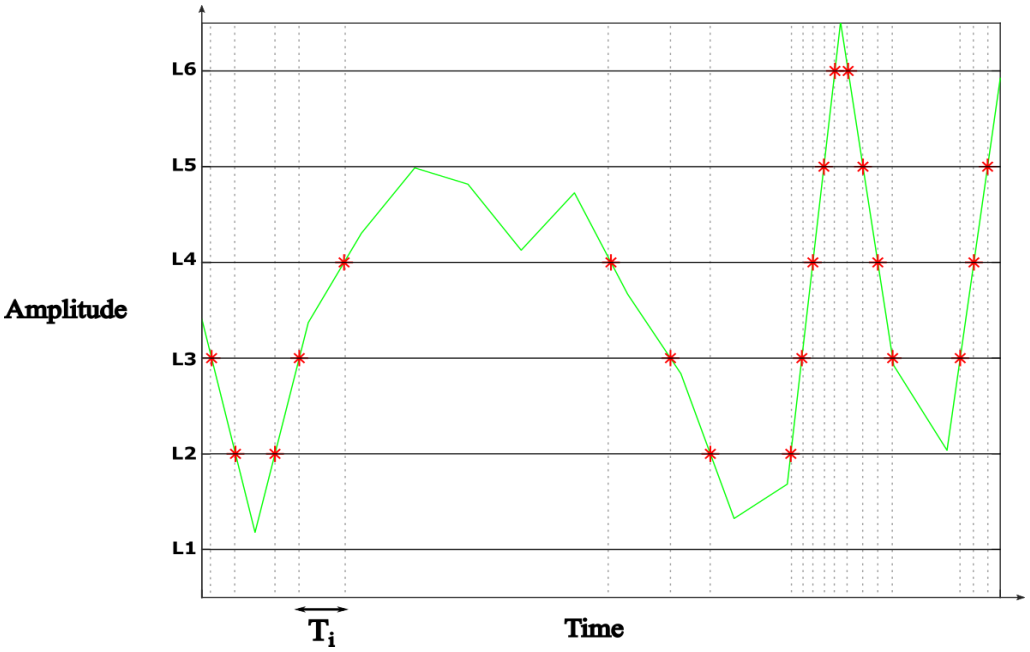


Figure 3: Irregular Sampling Scheme Example

two consecutive samples (T_i), which depends on the precision of the Time-to-Digital Converter (TDC). As stated in [10] - [11], if we define the precision of the TDC as T_c,

each sample (S_i) will be identified with the pair (L_i, T_i), where *Ti* has an associated error Δt, which belongs to the interval [*0,Tc*].

Using the quantization noise power definition $P(\Delta V)$, we can estimate the signal-to-noise ratio (SNR_{dB}) of the quantization process of a non-uniform sampling ADC, stated in the following equations:

$$P(\Delta V) = P\left(\frac{dV_{in}}{dt}\right) \cdot P(\Delta t) \quad \text{(Quantization noise power)}$$

$$SNR_{dB} = 10 \cdot \log\left(\frac{P(V_{in})}{P(\Delta V)}\right) \quad \text{(Generic SNR equation)}$$

$$P(\Delta t) = \frac{T_C^2}{12} \quad \text{(Power of a random variable uniformly distributed across [0, } T_C\text{])}$$

$$SNR_{dB} = 10 \cdot \log\left(\frac{12 \cdot P(V_{in})}{P\left(\frac{dV_{in}}{dt}\right)}\right) + 20 \cdot log\left(\frac{1}{T_C}\right) \quad \text{(Quantization noise power)}$$

Equation 2-3: Equations defining the Signal-to-noise ratio due to quantization of an irregular sampled ADC.

Equation 2-3 was originally proposed in [10] and then was corrected in [12]. Its first term can be defined by statistical properties of the input signal V_{in}, and the second term is only dependent on the timing precision of the TDC (T_C), and does not depend on the voltage quantization as in the uniform sampling case. Then, we can state that the SNR of an EB-ADC can be improved by *6.02 dB* when the TDC precision is improved to the double (i.e.: T_C divided by 2).

The following table shows the main characteristics of a regular sampled ADC versus an irregular sampled ADC.

Table 1: Characteristics of regular v/s irregular sampled ADC.

	Uniform sampling	Non-uniform sampling
Conversion Trigger	clock	level crossing
Amplitude	quantized	exact value
Time	exact value	quantized
SNR dependency	number of bits	TDC precision
Converter output	amplitude	amplitude and time

2.2.1.2 Beutler theorem

For non-uniform sampling ADC systems, another important aspect to consider is the condition for ensuring the correct reconstruction of the original signal. B. J. Beutler demonstrates in [13] that if the average sampling rate $\overline{f_s}$ is greater than twice the signal maximum frequency f_{max} (signal bandwidth), the original signal can be reconstructed.

$$\overline{f_s} > 2 \cdot f_{max}$$

Equation 2-4: Beutler's condition for a correct reconstruction of the original signal.

2.2.1.3 Conventional EB-ADC

As shown in [14] - [10] - [12] and many other research works, the conventional solution for an *"Event-Based ADC" (EB-ADC)* or *"Level-Crossing ADC" (LC-ADC)* is the one presented in Figure 4, that is composed of two comparators, an n-bit digital-

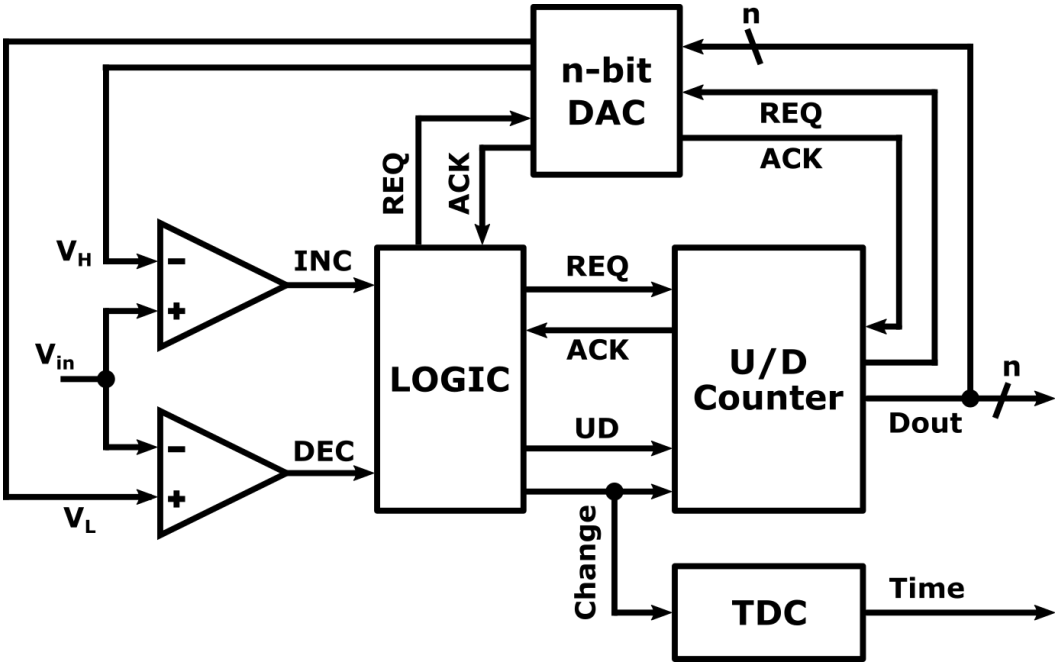


Figure 4: Conventional solution for EB-ADC, sometimes called "floating window Level Crossing-ADC".

to-analog converter (DAC), an up/down counter, a TDC and a control logic. It is also called "floating window LC-ADC". This solution has been proven to correctly work in many articles. However, it can be high power consuming since the comparators consume permanently, and they support a full-scale input dynamic range. Additionally, the DAC contains either resistor chains or capacitor arrays, increasing its consumption for a smaller quantization level, and finally the TDC requires a higher power consumption when improving its precision.

In the floating window LC-ADC of Figure 4, V_H and V_L are two consecutive quantification levels obtained from the DAC, adapting its values for following the analog input signal V_{in}. For a proper operation, V_{in} must be permanently bounded by V_L and

$$q = \frac{V_{range}}{2^M - 1}$$

Equation 2-5: Calculation of the quantification step (or LSB).

V_H. The gap between these values is called the comparison window and is always equal to one *LSB (less significant bit)*, also denoted as q, the quantification step. If V_{range} is the amplitude range of a M-bit resolution LC-ADC, with 2^M comparison levels uniformly distributed in V_{range}, then each comparison level corresponds to q. The following equation presents the calculation of q.

Then, if V_{in} is higher than V_H, $INC = 1$, $DEC = 0$, and $UD = 1$, indicating the up/down counter to count up. In the opposite situation, if V_{in} is lower than V_L, $INC = 0$, $DEC = 1$, and $UD = 0$, indicating the counter to count down. In both cases, the $Change$ signal is generated, indicating the TDC to stop and store the current measurement, and reset the TDC to prepare it for the next measurement. At the same time, the counter output is connected to the n-bit DAC for generating the new values for V_L and V_H, producing a loop feedback for updating the system on each level crossing of V_{in}. The example showed in Figure 4 incorporates asynchronous handshake signals, REQ and ACK interconnecting the communication of LOGIC, U/D Counter and DAC modules, for indicating each other when each stage calculation has finished, when a new data is available and if each module is available or busy. This method avoids the usage of a clock signal and therefore an unnecessary activation of the system.

Another important aspect to consider when designing this kind of solutions, is the so-called *"tracking condition"*. This condition ensures that the V_{in} signal will not cross any quantification level within any loop feedback delay, denoted δ, and particularly the worst case, been δ_{max}. The latter guarantees to capture every level crossing event. The Equation 2-6 presents the tracking condition calculation, considering the dynamic range V_{range}, the signal bandwidth f_{max} and the converter resolution M.

$$\max\left(\frac{\partial V}{\partial t}\right) \cdot \delta_{max} \leq \frac{V_{range}}{2^M - 1} = q$$

$$max\left(\frac{\partial V}{\partial t}\right) = 2 \cdot \pi \cdot f_{max} \Rightarrow \delta_{max} \leq \frac{V_{range}}{2 \cdot \pi \cdot f_{max} \cdot (2^M - 1)}$$

Equation 2-6: Tracking Condition.

2.2.2 Time-to-Digital Converter (TDC)

A time-to-digital converter (TDC) is a device permitting to measure the time elapsed between a $start$ and a $stop$ signals. The TDC are used in many fields where

a precise measure of the time is required, as time-of-flight devices, particle physics systems, tracking and positioning techniques, among others. Its capability of providing a digital representation of the time enables its integration in large digital systems implemented in an ASIC CMOS chip or even being previously tested in a *"Field-Programmable Gate Array" (FPGA)* device.

As stated in the thesis work of our colleague, Dr. Assia El Hadbi [15], a first generation of TDC was mainly analog. The most basic analog technique was the controlled charge of a capacitance, for then using an ADC to measure the final voltage, and hence estimate a digital representation of the time elapsed between the charge activation (start) and deactivation (stop), as depicted in Figure 5.

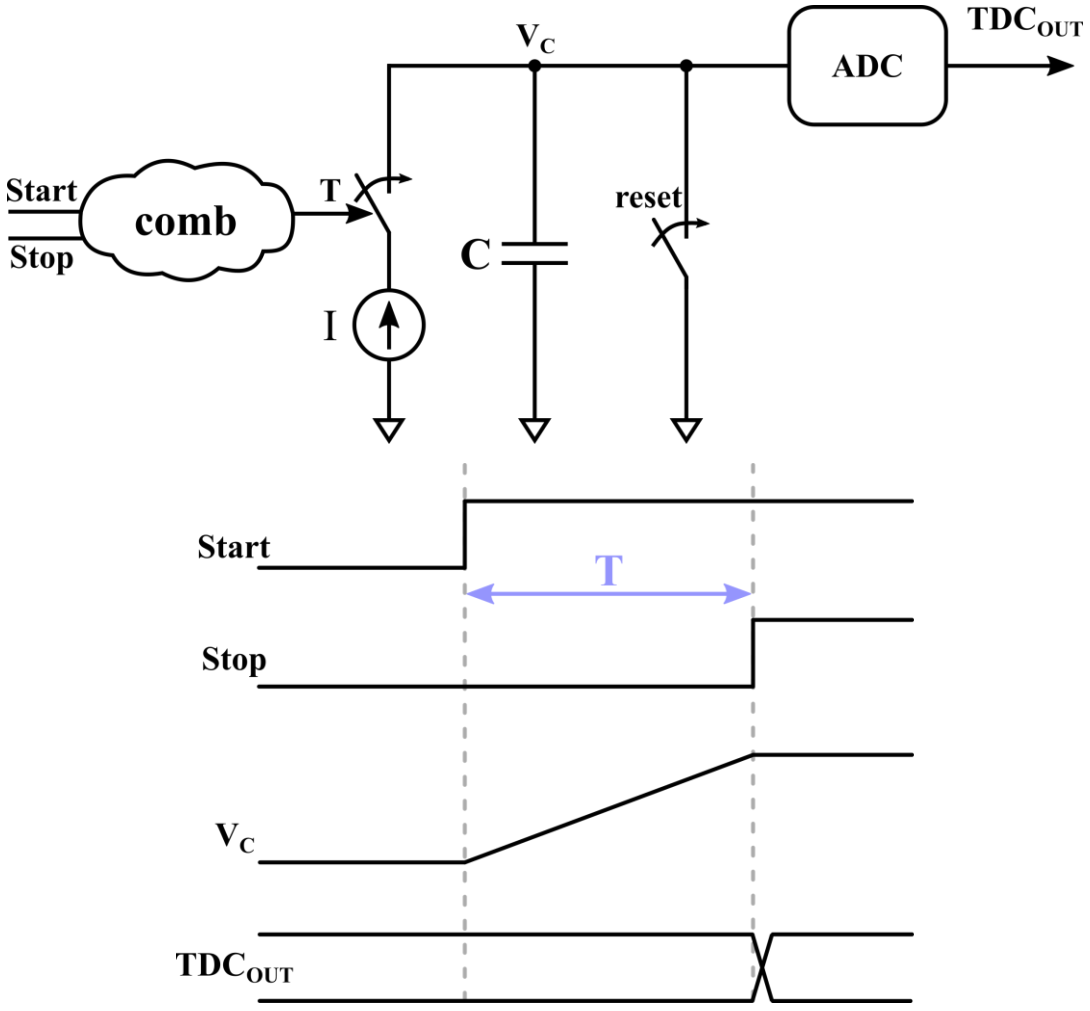


Figure 5: Basic analog TDC technique.

Figure 6 shows another analog solution that improves the latter, for avoiding using an ADC. It uses two current sources, one charging the capacitance, and the second one discharging it. Then, a zero-crossing element is used to detect the full discharge of the capacitance. Finally, a "TDC sampling unit", associated to a stable clock reference, registers the number of cycles during this process.

The analog TDC can reach a time resolution of tens or hundreds of picoseconds. However, they suffer from many issues. They are quite sensitive to temperature drift, not suitable for fast applications and they normally use an averaging procedure for calibration and for improving their time resolution. In addition, these TDC require a large area because of the use of capacitors, hence having more power consumption in comparison to a basic digital TDC solution, and also being less favorable for the technology shrink.

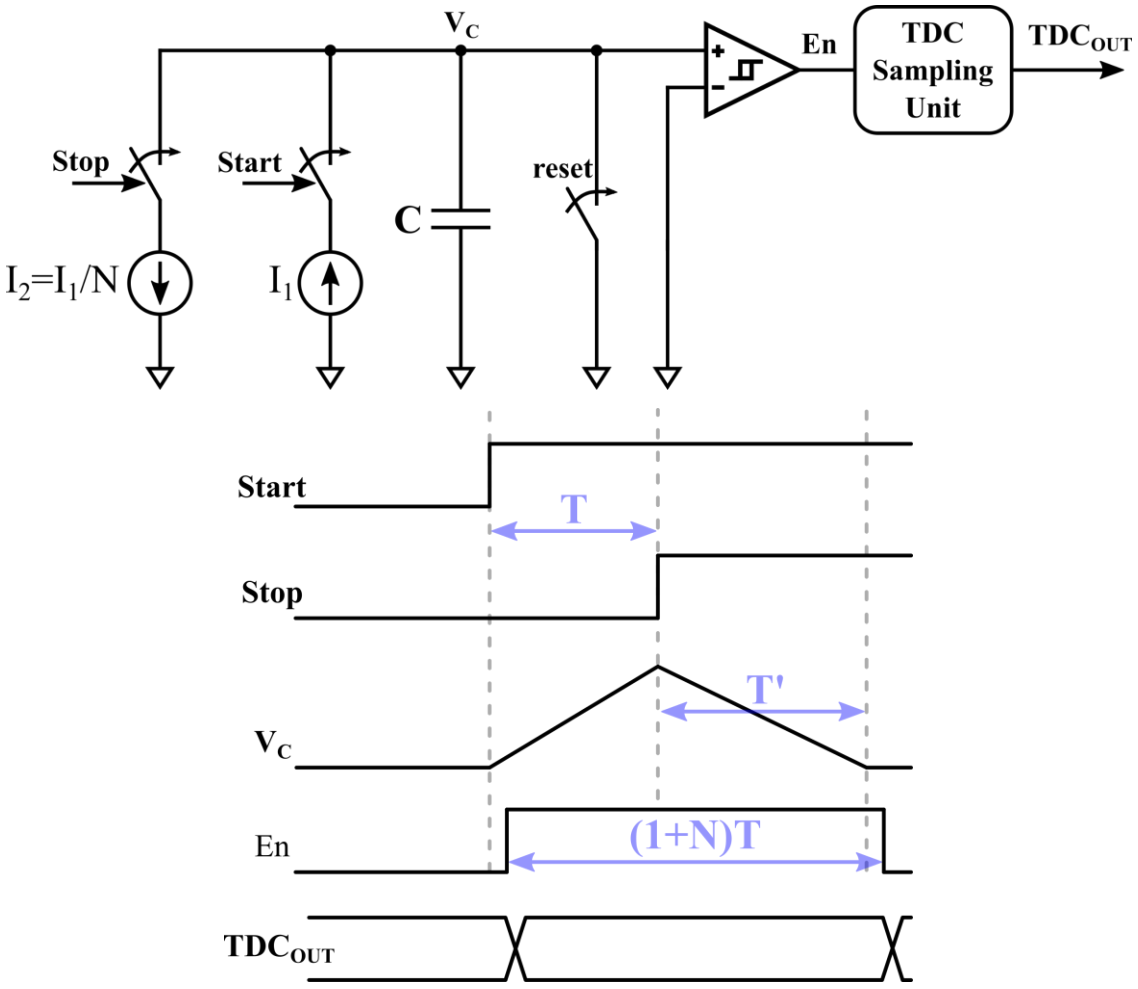


Figure 6: Enhanced analog TDC design.

The second generation of TDC use full-digital techniques. The simplest TDC architecture corresponds to the use of a timer permitting to count the number of rising edge occurrences of a reference clock signal, between the activation of a start and a stop signal, see Figure 7. This solution is simple and compact, compared to the other solutions.

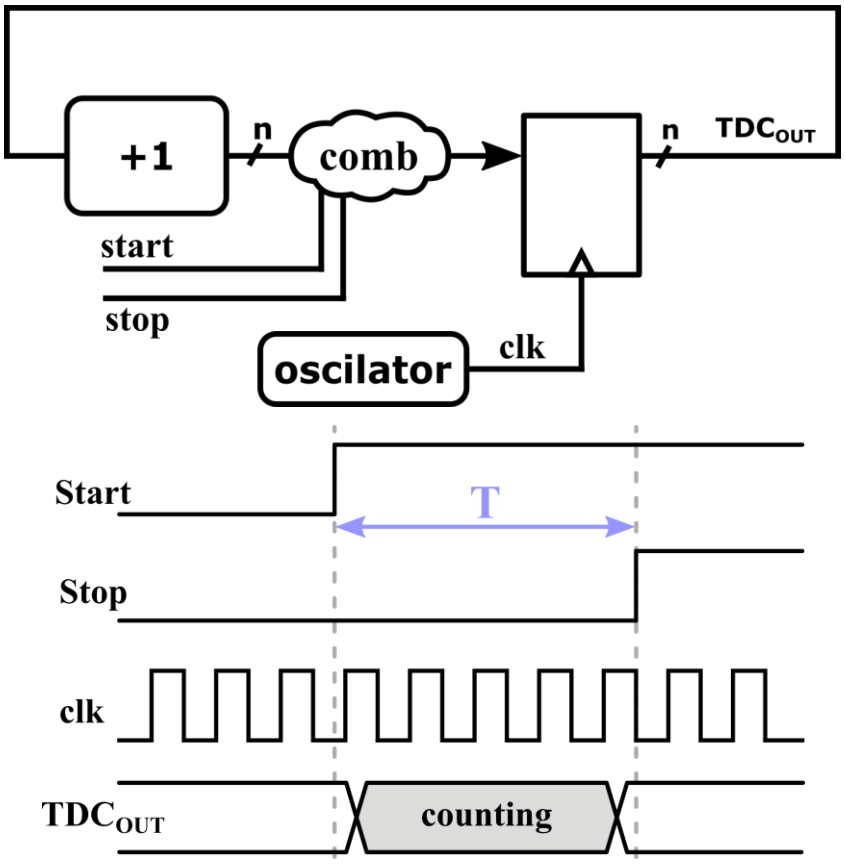


Figure 7: Digital TDC design: Counter TDC.

In order to improve the time measurement precision, another digital solution has been proposed, using an internal Ring-Oscillator *(RO)* as a clock reference to form a similar system to that proposed in Figure 7. Moreover, as depicted in Figure 8, it adds the registration of the final state of the looped signal after propagating through the inverter gates within the *RO*. If the delay of the inverter gates is known, the registered information is used to infer the number of inverters that has been crossed between the start and stop signals and, then, use the inverter gates delay for increasing the precision of this *TDC*. This solution is limited in precision by the delay of the gate used in the RO and changes when switching from one technology node to another one. In addition, this delay will evolve after fabrication due to the aging effect, and will also vary with the *"Process, Voltage and Temperature" (PVT)* conditions.

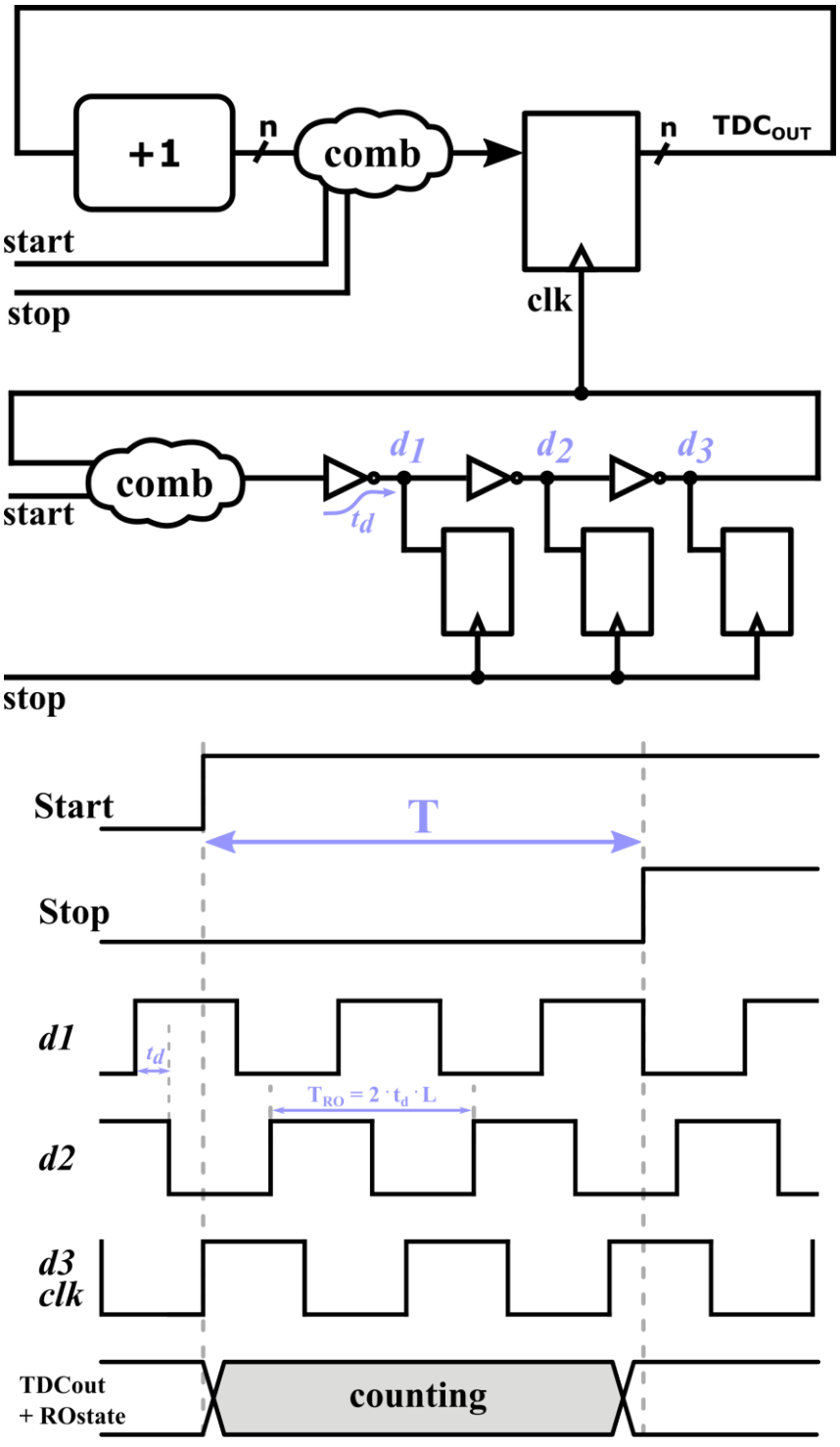


Figure 8: Enhanced digital TDC design.

2.3 Asynchronous Circuits

2.3.1 Synchronous circuits

Digital electronic circuits are devices containing several logic gates executing operations and tasks. Normally, these circuits are composed by several pipeline stages, defining a structured sequence of data transmission. Each pipeline is composed by two main modules, a combinatorial stage and a register stage, as shown in Figure 9. In this figure, we present the synchronous circuit model in the simplest configuration case: the linear pipeline.

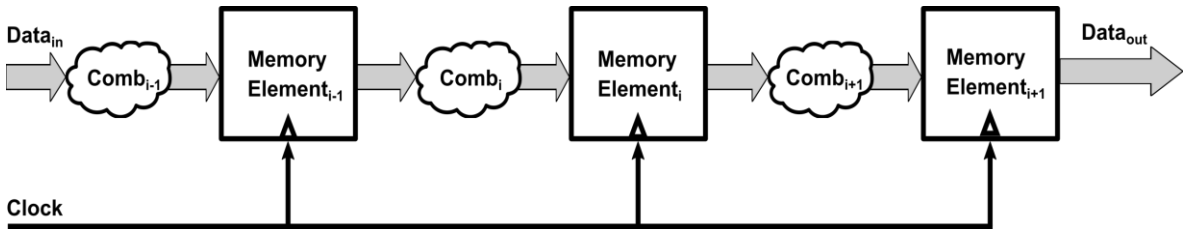


Figure 9: Synchronous linear pipeline example.

2.3.2 Synchronous method

Synchronous circuits are globally synchronized circuits, which use a global reference clock signal for activating all the register stages simultaneously. This clock defines the transmission instant, when the result of each pipeline stage is sent to the combinatorial inputs of the next stages. For the proper operation of this type of systems, all the combinational blocks of the circuit need to complete and stabilize its calculations within the clock period, in order to ensure that the resulting values saved in the registers are correct, and free of metastability or hazard. Due to the usage of a global clock, the clock period must be larger than the slowest combinational path between two registers on the whole circuit, which is called *"the critical path"*.

Then, for ensuring the right transmission and memorization of the data in the circuit, all of its registers must theoretically receive the clock signal at the exact same time. The latency or delay between the reference clock rising edge, and its arrival on the register clock input, is called *"skew"*. For guaranteeing a minimal skew in all the registers of the circuits, the construction of a well-balanced clock-tree, called *"Clock-*

Tree Synthesis" (CTS) is necessary, which is composed by balanced buffers and inverters. The design of a standard *"Clock Tree" (CT)* is shown in Figure 10.

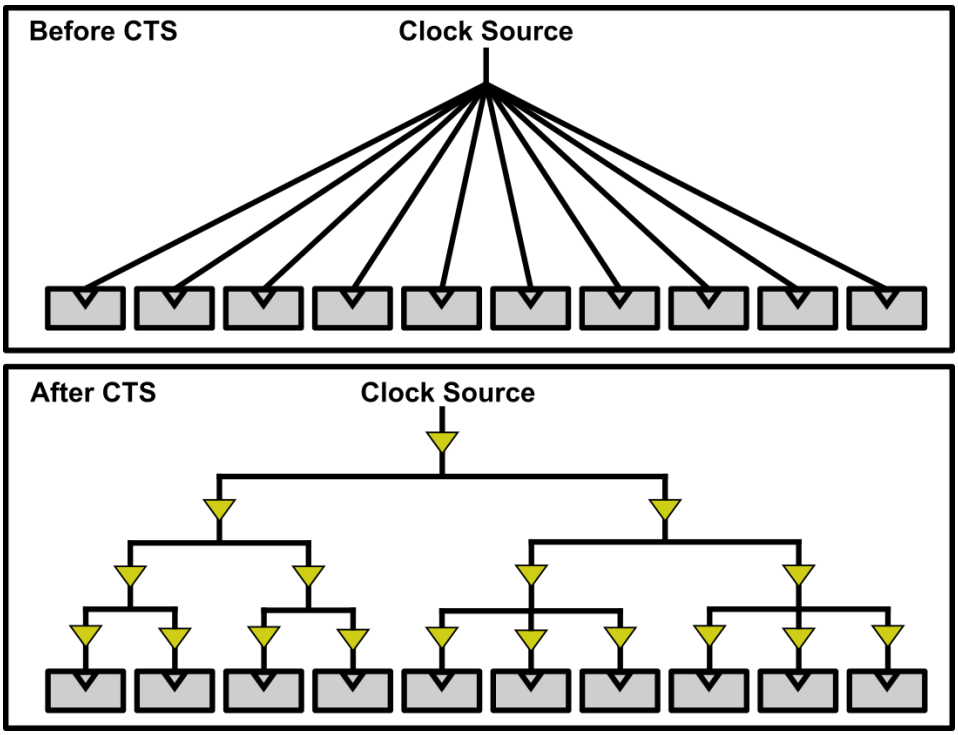


Figure 10: Example of a standard Clock-Tree.

The addition of the logical gates of the *CT* increases the area of the circuit and the complexity of the *"Place and Route" (P&R)* steps in the digital design flow. Moreover, as explained in [16] - [17], the clock propagation contributes between 20% to 45% of the total power consumption of a digital circuit. Additionally, the synchronous circuits require an oscillator (Phase Locked Loops or Delay Locked Loops) for the generation of the global reference clock. This oscillator normally requires the use of a large silicon area, negatively impacting the circuit cost. This constraint might be avoided if using a system that does not need it, like asynchronous circuits. Furthermore, the synchronization of the register activation of an entire circuit produces important electromagnetic noise emissions at the fundamental frequency of the clock, but also at its harmonics. These latter can cause huge interferences in wireless communications. Moreover, when evaluating the I-R drop and the power supply constraints of a circuit, this global register activation can also cause an important power supply reduction in the middle of the power lines, which might affect the correct operation of the digital circuit.

2.3.3 Asynchronous method

For mitigating the issues due to a global synchronization of the digital circuits, the researchers and designers have been working for decades on an alternative based on asynchronous circuits. The asynchronous logic has been introduced by the researchers during the 50's. In some articles, they analyzed the behavior of a circuit combining combinational circuits with the application of some delays between its outputs and its inputs, for establishing "secure" feedback loops [18].

A sequential circuit is called "asynchronous", when a local handshake protocol is used to indicate when the calculation in a pipeline stage is finished, and when valid data at the input of its registers are ready to be transmitted to the inputs of the combinatorial part of the next pipeline stages. This protocol also indicates whether the next stages are available or not to receive new input data to process. If one or more of these stages are busy, the data transmission is stopped, blocking the following circuit stages subsequently.

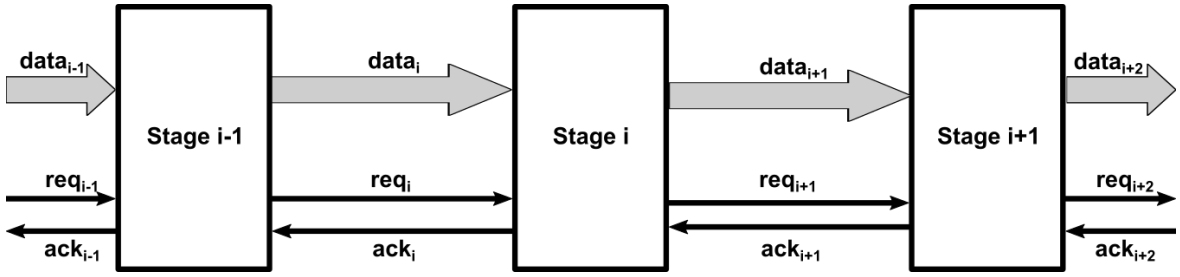


Figure 11: Asynchronous linear pipeline example.

When analyzing the case of Figure 11, *stage$_i$* receives a request, indicating that new valid data are available at the input of its registers. Next, if *stage$_{i+1}$* is available, the *stage$_i$* saves the data in its registers and it sends a request forward to *stage$_{i+1}$* and an acknowledgment backward to *stage$_{i-1}$*, indicating that the data has been processed. It is now ready for a new data transaction. This method activates a stage only if there is new data to process and stays idle otherwise. This technique allows only consuming dynamic power on the active parts of the circuit.

Varying on the signaling and data encoding convention, the asynchronous circuits are categorized in the following circuit classes [19]:

- Huffman circuit.
- Speed Independent circuit (SI).
- Quasi-Delay-Insensitive circuit (QDI).

- Delay Insensitive (DI).
- Micropipeline.

The Figure 12 represent the different types of asynchronous circuits in terms of their timing assumptions and their functional redundancy and robustness.

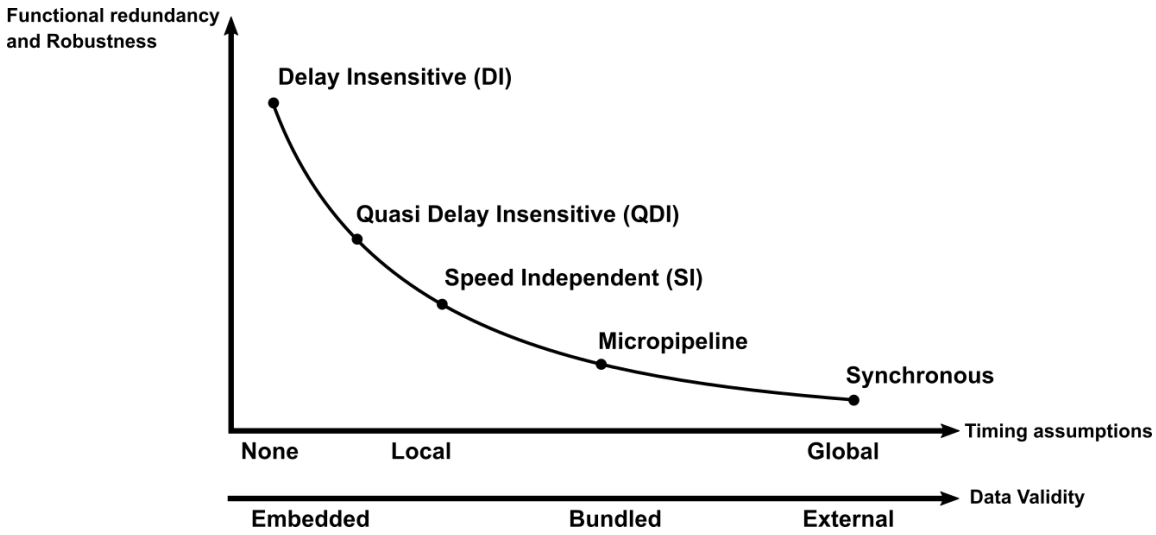


Figure 12: Classification of asynchronous circuits.

In asynchronous circuits the two most commonly used data encoding are bundled-data (micropipeline) and multi-rail encoding (embedded in data - *QDI*). Furthermore, the two most widely used handshake protocol are the two-phase protocol and the four-phase protocol.

In this manuscript, only the four-phase, bundled-data micropipeline asynchronous circuits are used and hence reviewed. Therefore, only these concepts will be studied in the state of the art of this document. For these kinds of circuits, the data path remains similar to synchronous circuits, and only the globally distributed clock signal is replaced by a controller module that generates the clock signal at each pipeline stage as shown in Figure 13. The bundled-data micropipeline circuit used in this thesis is based on four-phase handshake protocol controllers.

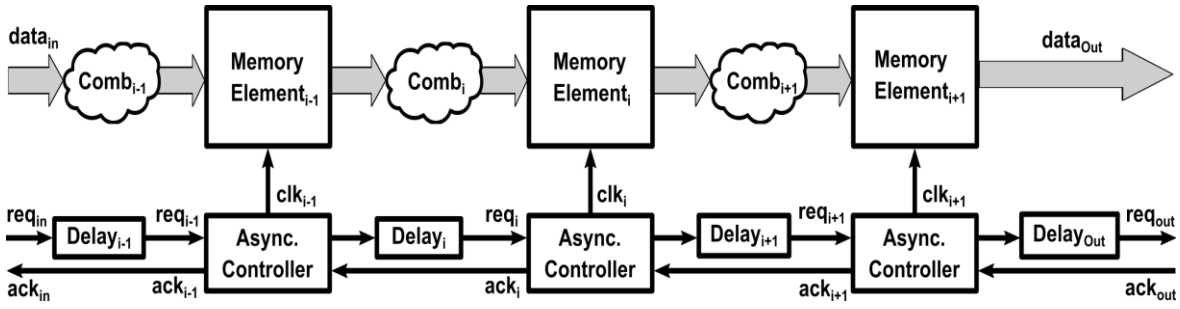


Figure 13: Bundled-data micropipeline linear example.

2.3.4 Bundled-data encoding

As explained before, asynchronous circuits use a local handshake protocol to indicate when a new data has arrived thanks to a request signal, and when the calculation has been done through an acknowledgment signal. For ensuring a correct synchronization between different pipeline stages in the circuit, request signals are associated to data in the system. In this manuscript, we used the bundled-data encoding for this purpose.

As shown in Figure 13, in bundled-data circuits, each pipeline is decomposed into a combinational module and a memorization module, normally composed by registers or latches. Additionally, each bit is binary encoded, and on each pipeline stage, the data validity is signaled by a dedicated request signal. Moreover, the asynchronous controllers are in charge of generating the local clock signals for indicating, to all the memory cells of the pipeline, the instant when capture and save the local data. Therefore, the request on each stage should arrive to the input of its associated asynchronous controller only when the data in the combinational part have been processed. For ensuring that the combinational computations have been treated, a delay module is inserted into the request line with a value larger than the worst combinational delay, commonly called “local critical path”. Furthermore, the acknowledgement signal, normally indicating the end of the computation of the combinational part of the pipeline stage, is generated by the controller just after each switching of the clock signal.

2.3.5 Four-phase handshake protocols

A protocol is the definition of a communication sequence permitting to send data from one point to another point of a circuit, network, bus, or any communication channel. In the case of asynchronous circuits, a handshake protocol set the basis for defining a synchronized approach to propagate data from one pipeline (or memorization stage) to the next one without any metastability or corruption of the data. Two essential characteristics differentiate the handshake protocols: the used data encoding and the data transfer sequence. In literature, several protocol implementations have been proposed [20]. These solutions are classified in two main categories, the two-phase protocols and the four-phase protocols. The first, also called *“Non-Return-to-Zero” (NRZ)* protocol, performs a complete data transfer in two steps.

On the other hand, the latter, also called *"Return-To-Zero" (RTZ)* protocol, needs four steps to achieve a complete data transfer, and is called *RTZ* because the request and acknowledgment signals must be cleared before a new data is allowed to be sent.

The Figure 14 describes an example of the cycles of a four-phase protocol.

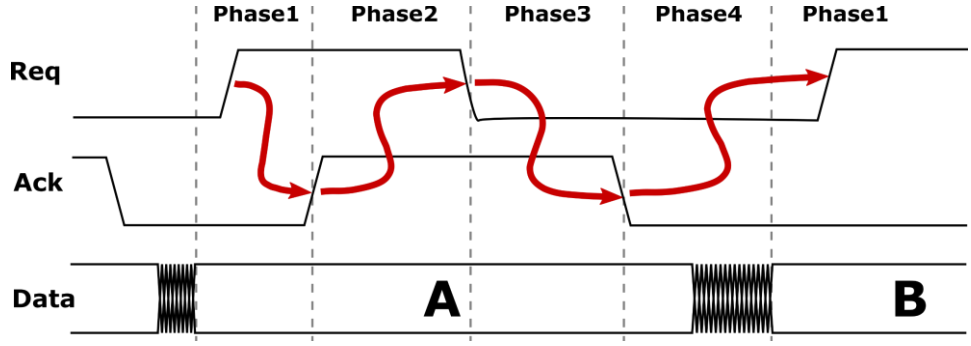


Figure 14: Four-phase handshake protocol example.

- **Phase 1:** It begins when the sender enables a new data on its outputs and sets the request signal for indicating that a valid data *A* is available. Then the request *(req)* signal takes some delay to arrive to the receiver interface.
- **Phase 2:** It starts when the receiver identifies the arrival of the rising edge of the *req* signal, processes the received data and sets the acknowledgment *(ack)* signal, for indicating to the sender that the data has been treated.
- **Phase 3:** It begins when the sender detects the rising edge of the *ack*, and then resets its *req* signal. This phase is called *Return To Zero (RTZ)*.
- **Phase 4:** It starts when the receiver identifies the falling edge of the *req*, then it resets the *ack* for indicating that it is now ready to receive a new data. This phase will stay active until the new data *B* is valid, and a new *req* is sent.

The simplest circuit implementation of a four-phase protocol is the frequently called "Weak-Conditioned Half-Buffers" (*WCHB*). In the case of a bundled-data micropipeline implementation of WCHB, the asynchronous register controller of Figure 13 is only composed by a Muller-gate (or also called C-element) and an inverter, as depicted in Figure 15.

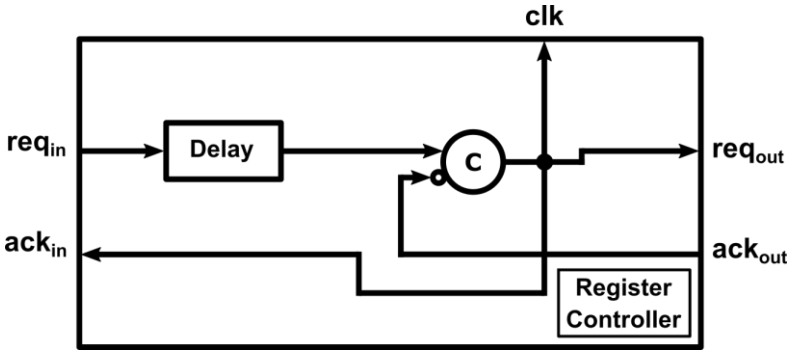


Figure 15: WCHB bundled-data micropipeline asynchronous register controller implementation.

The controller of Figure 15 is called a *register controller* since it controls the activation instant of its associated memory block. From the asynchronous point of view, the register controllers are classified as “linear”, because they allow the control of one data channel at its input, and one data channel at its output. They can be connected in series forming a linear structure, as depicted in Figure 13, for controlling the data transmission of a linear circuit, as for example a FIFO queue.

In a real circuit, many data channel interactions occur. In cases when various input signals enter one module, when one output is connected to several modules, or when conditional channel flow is required, non-linear structure controllers are needed to ensure a correct data transmission and avoid metastability issues. The list of these structures is presented below:

- Fork: is an unconditional asynchronous flow element with one input channel and several output channels. The event token is propagated from its input to all of its outputs.
- Join: is an unconditional asynchronous flow element with several input channels and one output channel. The event token is only propagated to the output after all the input tokens have arrived.
- Split: is a conditional asynchronous flow element with one input channel and various output channels. The event token in its input will only be propagated to one of its selected outputs, according to the selection value, only after the selection token has also arrived.
- Merge: is a conditional asynchronous flow element with multiple input channels and one output channel. After the selection token arrives, the input event token in the selected input channel, according to the selection value, will be transmitted to the output channel.

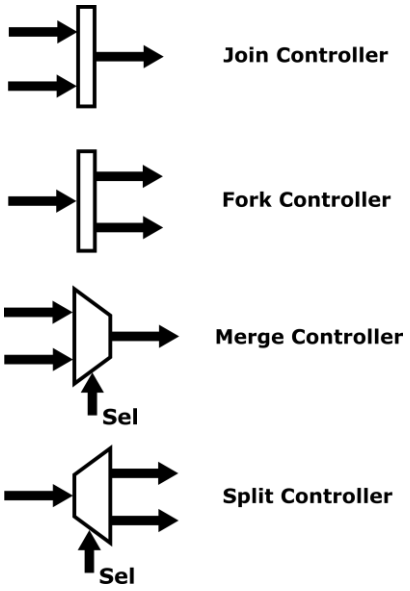


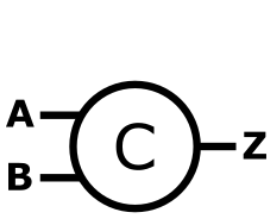
Figure 16: Symbol representation of the most common non-linear asynchronous flow elements.

2.3.6 Muller Gate

The muller gate, also called C-element, has been introduced by David Eugene Muller in 1963 [21]. The C-element is the basic element enabling the implementation of most of the known asynchronous control circuits. This gate switches its output to a value, only if all its inputs have the same value. Otherwise, if its inputs are different, the output is maintained to its previous state. This behavior facilitates the synchronization of its output with some of its inputs, which is the basic operation concept of the handshake protocol in asynchronous circuits.

When muller gates are implemented in a control protocol circuit, the previous behavior permits the validation of the arrival of several signals on its inputs before enabling the transmission of the output validation to the next stages.

The most basic implementation of Muller gate possesses two inputs and one output, as depicted in Figure 17.



A	B	Z
0	0	0
0	1	Z^{-1}
1	0	Z^{-1}
1	1	1

Figure 17: Muller gate symbol and its truth table.

The logic equation representing the behavior of a basic two-input Muller gate is presented in the following equation:

$$Z = A \cdot Z^{-1} + B \cdot Z^{-1} + A \cdot B$$

Equation 2-7: Logic equation representing a Muller gate.

This behavior can be implemented using a transistor level design or with logic gates. The first option is more optimized in terms of size and speed. However, this gate is not usually distributed in commercial design kits (DK), so it must be manually created and characterized for every used technology node. The Figure 18 shows the most basic implementation of a C-element, called “weak-feedback”, introduced by A.J. Martin in 1989 [22]. Other more complex implementations are presented in [23].

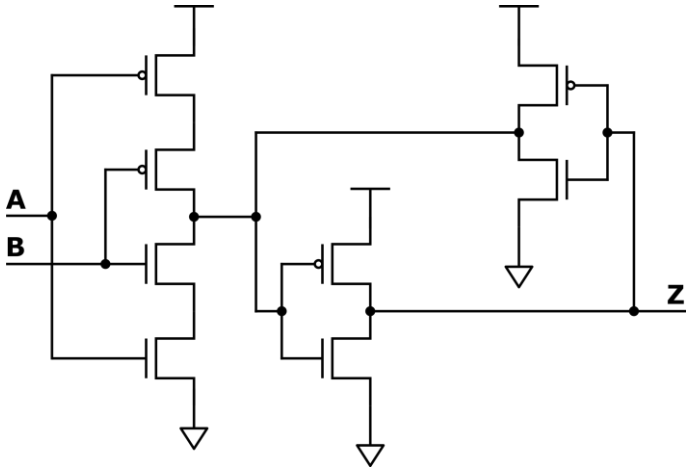


Figure 18: Transistor level design of a weak-feedback muller gate.

On the other hand, Figure 19 presents one of the possible implementations of a Muller gate using standard cells. Another typical implementation of the C-element is using a three-input Majority gate *(MAJ3)*, with its output feedback connected to one of its inputs.

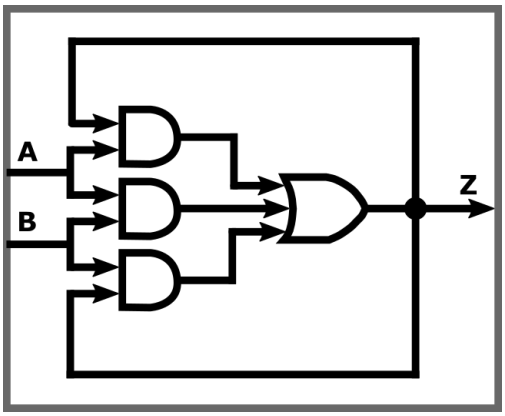


Figure 19: Muller gate implementation using standard cells.

A *set* or *reset* signals are normally necessary to initialize the state of the Muller gate.

2.3.7 The advantages of micropipeline Asynchronous Circuits

As depicted before, micropipeline asynchronous circuits are composed by several pipeline stages connected to each other in the same way as standard synchronous circuits. However, instead of using a global reference clock, asynchronous circuits implement, on each pipeline stage, a local clock ensuring a correct sampling of the data on its flip-flops (registers). The period of each local clock is locally determined by its combinational block delay. Moreover, the implicit behavior of asynchronous circuits could be assimilated to a fine-grained clock gating approach, because the memory elements are only clocked when necessary. The resulting benefits related to the use of asynchronous approach are described in the following chapters.

2.3.7.1 High performance

The average processing delay for an asynchronous circuit varies on the distinct input combinations. Every input combination activates different paths of the systems, therefore varying the input-to-output delays. This occurs because the data propagation on each pipeline stage depends on the local pipeline critical-path, and not on the global critical-path as in synchronous circuits. Finally, it is stated that the circuit operating speed is the average of the local delays, and also, the average of the total delays

depending on the inputs of the system. Normally this results in a higher performance compared to the same synchronous solution.

2.3.7.2 Low power consumption

In synchronous circuits, all the memory elements of the circuit are activated at the same time. For obtaining such a behavior, the synthesis of a Clock-Tree is necessary, as stated in chapter 2.3.2. The construction of that structure requires several synchronization elements, using a large circuit area, but also an important power consumption. On the other hand, when implementing an asynchronous approach, only local *CT* are necessary, reducing the number of synchronization elements, and thus considerably reducing the power consumption while ensuring a correct data transmission synchronization.

In low-power synchronous design, the use of clock-gating technique is very common. In asynchronous circuits, this behavior is already implicit, and is normally more energy efficient than the normal clock-gating method. Indeed, if no new data is arriving to the circuit inputs, it will stay in an idle or inactive state, consuming only static power.

In the context of reducing the power consumption, voltage scaling is a very effective technique that is commonly used in the industry. Reducing the power supply voltage decreases the dynamic power consumption, while lowering also the performance of the circuit. Implementing such a method in synchronous circuits requires an adjustment of the clock frequency for preventing timing violations and metastability. These constraints increase the design complexity, needing supplementary blocks for sensing the voltage changes and then applying the correct frequency adjustments. Conversely, in asynchronous designs this adaptation is naturally done. When the power supply voltage decreases, the propagation delays of the combinational parts increase. Therefore, in bundled-data systems, the delay associated to each pipeline stage will accordingly increase, while maintaining the correct timing constraints.

In some applications, the use of several clock domains is vital to reduce the power consumption. The direct synchronization of the different clocks used in this kind of design is not feasible. In that case, it is essential to use some elements to interface the communication between the different clock domains, augmenting the area usage and

the complexity of the specifications to take into account for avoiding errors in the design.

2.3.7.3 Low electromagnetic emissions

As explained before, synchronous circuits use a global reference clock signal for activating all their memories (flip-flops, latches) at the same time, in order to control a correct data transmission in the circuit. When applying such solutions in wireless communications, a strong source of electromagnetic noise is the circuit itself, hindering the transmission. The synchronized register activation produces high current peaks matching the clock period. This effect results in an electromagnetic noise at the frequency of the clock and its harmonics, affecting the wireless communication performance.

In contrast, an asynchronous circuit activates its memory elements at different moments, depending on the local critical path delay of each pipeline stage. In that case, the current peaks are spread in time, resulting in lower electromagnetic noise emissions. In addition, in the PhD research work of our colleague *Sophie German* at STMicroelectronics and TIMA Laboratory [24], it has been proved that the electromagnetic emissions of a digital circuit can be controlled using the bundled-data asynchronous methodology. This approach allows fitting a frequency mask, which is used to avoid affecting the surrounding wireless communication.

2.4 Conclusion

In this chapter, we described and reviewed the state of the art of the two main blocks treated in this thesis, the EB-ADC and the ASK demodulator.

The EB-ADC is composed by a level-crossing detector and a TDC. It performs a non-uniform sampling, for which we have first reviewed its theoretical basis in order to find a minimum LCSS and, then, we have presented EB-ADC examples found in the literature. For the TDC, we reviewed several implementations found in the literature in order to evaluate which alternative could be best suited for our application.

Then, we checked the characteristics of synchronous and asynchronous circuits, highlighting their differences and checking why asynchronous circuits might be better adapted for the ASK demodulator application in NFC tags. Next, the bundle-data encoding was revised, showing the operation of the four-phase handshake protocol, which is the asynchronous approach we decide implementing for the proposed

solutions. For that methodology, the basic blocks used in the asynchronous control circuits have been presented, including two different possible solutions for the design of the C-element, which is a required gate for creating the different modules of the control circuit.

Finally, we reviewed the advantages associated to the micropipeline asynchronous circuits, highlighting the possibilities to reduce the power consumption when decreasing the number of activations of the circuit, implementing only local clock-trees, enhancing its voltage scaling capabilities and the elimination of the clock oscillator (PLL or DLL). Moreover, asynchronous circuits offer some possibilities to increase the performance of the circuit compared to its synchronous counterpart. Finally, we reviewed its advantages to reduce the electromagnetic emissions when controlling the delays of the pipeline stages. It is even possible to fit an electromagnetic spectral mask.

3 NFC Communications

3.1 NFC demodulation

3.1.1 NFC protocol

The Near-field communication (NFC) is a short-range wireless communication protocol, widely used in our daily life in several applications, as public transport validation, contactless payment cards, access cards, and even in some medical devices like pacemaker prototypes. This technology is based on the existing radio-frequency identification (RFID) standards like ISO/IEC 14443, for type A and type B, and Felica for type F. The operation frequency of NFC is the unlicensed radio frequency ISM band of 13.56 MHz, which is part of ISO/IEC 18000-3 air interface standard. This protocol normally requires a separation of less than 10 cm between two NFC devices, defined as the initiator and the target, to communicate at bitrates going from 106 kbps up to 848 kbps, and beyond for some specific cases. The different NFC standards define the modulation schemes, coding, transfer speeds and frame structure of the RF protocol enabling the communication between NFC devices. Table 2 presents the different types of NFC tags and their characteristics.

Table 2: NFC tag types and their main characteristics.

	Type NFC-A	Type NFC-B	Type NFC-F
Supported Standard	ISO 14443 A	ISO 14443 B	JIS X 6319-4 (Felica)
Carrier Frequency	13.56 MHz ± 7 KHz		
Data Rate	106/212/424/848 Kbps	106/212/424/848 Kbps	212/424/848 Kbps
Modulation (Reader to Tag)	ASK 100%	ASK 10%	ASK 10%
Data Encoding (Reader to Tag)	Modified Miller	NRZ-L	Manchester
Modulation (Tag to Reader)	Load Modulation (ASK) with sub-carrier (± 848 Khz)	Load Modulation (BPSK) with sub-carrier (± 848 Khz)	Load Modulation (ASK) without sub-carrier
Data Encoding (Tag to Reader)	Manchester	NRZ-L	Manchester

Two types of communications are covered by the NFC protocol, the active communications and the passive communications. For active communications, both devices sequentially generate their own magnetic field for transmitting the data. However, in passive communication, one device generates the magnetic field (the reader) and the other one (the passive tag) uses the “load modulation” for transmitting its data.

Several applications use battery-less NFC devices, usually called *“passive NFC devices”*. In that case, the energy needed for computing is recovered from the electromagnetic field of the reader. It supplies power to the different internal modules composing the tag. These modules are:

- A matching network connected to the antenna for acting as an automatic gain control (AGC) module to stabilize the modulated signal. It usually contains a voltage limiter to avoid high voltage levels.
- An energy-harvesting module for recovering and stabilizing the power supply.

- A low-power demodulation system for extracting the modulated symbols and then the data from the modulated signal.
- A low-power microcontroller for processing the transmitted data.

3.1.2 Analog demodulation

In literature, analog and digital ASK demodulation solutions are proposed. As explained in [3], analog solutions are classified in three categories: voltage-mode, current-mode and mixed-mode. All these analog solutions [3] [4] [5] are generally composed by the following modules: an automatic gain control (AGC), an envelope detector, an average detector, and a comparator, as shown in Figure 20.

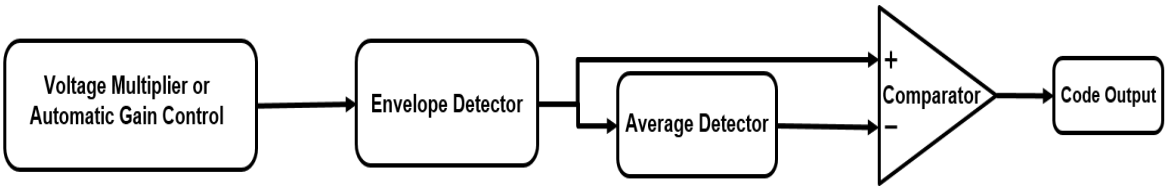


Figure 20: Common structure of an analog ASK demodulation solution.

The AGC adapts the maximal amplitude of the received signal, ideally matching the operational or nominal power supply voltage. The envelope detector extracts the lower frequencies in the modulated signal, recovering the called *"envelope"* of it, as presented in the red line of the example of Figure 21.

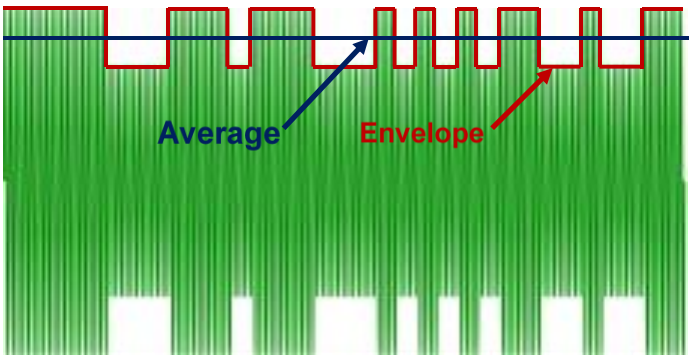


Figure 21: Example of an envelope detector (in red) and an average detector (in blue).

Then, the average detector obtains the mean value of the envelope signal, for generating a reference threshold of comparison. Finally, the envelope and the average

signals are compared using a comparator, for obtaining the digital demodulated output symbols. Some analog solutions in the state of the art also describe the following modules: a rectifier, a voltage limiter, a power on reset, an energy harvesting and even an impedance matching system to adapt the RF input signal (RF_{in}) to the different electromagnetic field conditions. Most of them present simulations or measurements about power consumption, minimal modulation index (*MI*), maximal data rate, circuit area and, in some cases, the minimal carrier power permitting correct system operations. However, few of them only present how noise affects demodulation sensitivity. This criterion is an important parameter when the study is focused on ASK communications, especially for the farther distances.

3.1.3 Digital demodulation

Many digital ASK demodulator solutions are found in the literature. They are usually based on a High-Speed Analog to Digital Converter (ADC) applied directly to the modulated RF input signal and then a digital IQ demodulator allowing to identify the output symbols, as depicted in Figure 22 and in the implementation example of [6]. In that solution, mixers are replaced by digital multipliers, the “in phase” and “in quadrature” signals are digitally generated using preset values and the sampling clock F_S, and the low-pass filters are digitally implemented using FIR filters for obtaining a digital representation of I and Q values in order to recognize the demodulated symbol values.

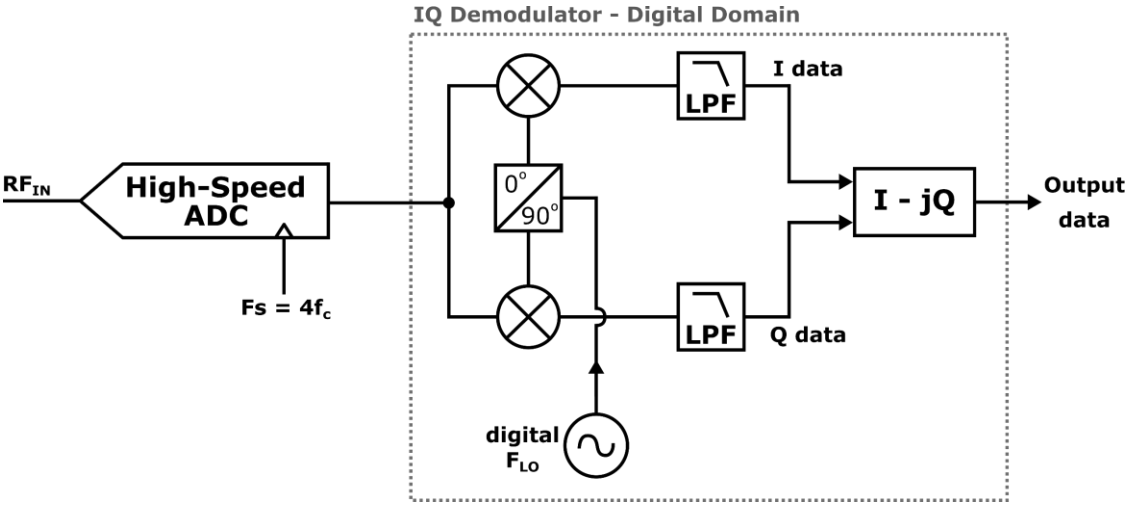


Figure 22: Digital IQ demodulator schematic.

Also, other digital ASK demodulation solutions are proposed in [2] [25] [26]. They directly use comparators on the RF input signal (RF_{in}) to obtain a raw digital representation of it. In order to analyze their demodulation performance in a full digital environment, these solutions assume to have a stable supply voltage and a normalized amplitude of the modulated signal. For all the solutions, their performance is represented by a graphic showing their bit error rate (BER) against their signal to noise ratio (SNR).

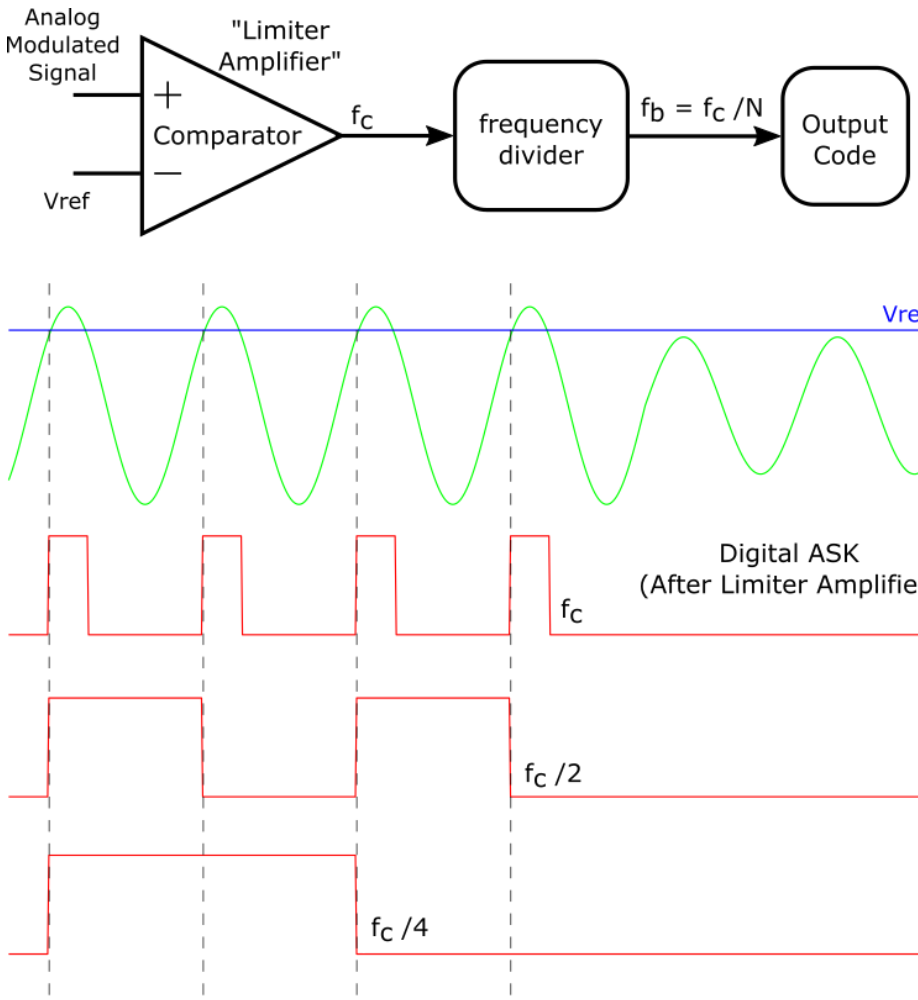


Figure 23: Diagram and signal description of the frequency divider ASK demodulation circuit proposed in [2].

The proposed circuit in [2], depicted in Figure 23, uses a comparator connected to the analog modulated signal (RF_{in}), and to a reference voltage (V_{ref}), which is a voltage level between the minimal and maximal amplitudes of RF_{in}. With this setup, the output of the comparator generates pulses only for high amplitude periods, as shown in the signal description of the last Figure at frequency f_c. Then, a frequency divider is used

to discover when these pulses are present. This technique is very suitable in terms of simplicity and size. However, it is very sensitive to glitches generated by the amplitude noise because, when these unexpected pulses arrive, the frequency divider will end up in an unknown state losing the synchronization of its demodulation logic. Nevertheless, such a technique is quite effective for high modulation index (MI) systems, where those faulty pulses are not generated.

Another digital solution in literature is the demodulation system proposed in [25], which uses a Schmitt Trigger comparator to digitalize RF_{in} regarding V_{ref}, in this case, defined at an arbitrary level below the minimum amplitude of RF_{in}, as represented in Figure 24. This output of the comparator, called $Vraw$, generates narrow pulses when the amplitude of the modulated signal is low, and wide pulses when this amplitude is high. Then, $Vraw$ is connected to a chain of "Pulse Shrinking Elements". On each stage, the original pulse is shrunk to obtain shorter pulses until they disappear. In the communication sequence, this technique permits the identification of carrier periods with shorter or larger pulses, and hence with low or high amplitudes, respectively. The data sequence could be retrieved between the stages when the shorter pulses disappear, while the larger pulses remain, as depicted in Figure 24. The presence or absence of these pulses, representing high or low amplitudes, is stored in a flip-flop synchronized by the inverted signal of $Vraw$. This method is very interesting and clever. However as described in [25], matching the delays between the pulse shrinking path and the delay of the clock path, which is the signal $\overline{Vraw}$ arriving to the clk pin of the register, is usually very sensitive to process, voltage and temperature (PVT) variations. Moreover, it becomes even more difficult when using advanced technology nodes. Thus, the symbol recognition is very sensitive to the choice of the correct shrinking output stage. This choice is a complicated task because it is impacted by many factors, such as noise, the parasitic elements inserted by the *"Place and Route" (PnR)* process, the comparator delay and also the chosen comparison level V_{ref}.

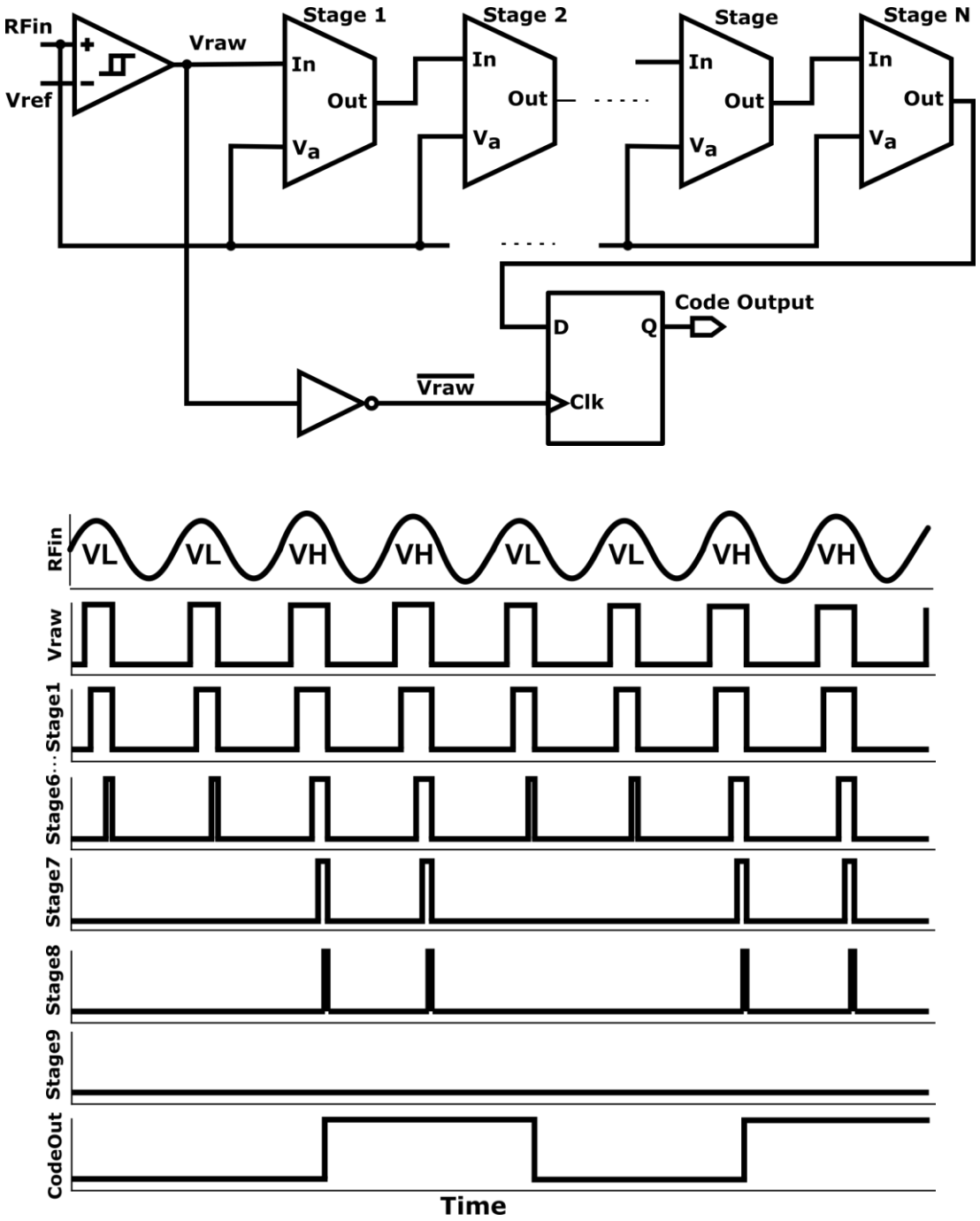


Figure 24: Diagram and signals of the Pulse-Shrinking ASK demodulation circuit implemented in [24].

3.2 Objectives of the proposed solution

The main purpose of this thesis is to enhance the demodulation circuit of a passive tag. For that reason, we are focus on including these two main characteristics on our proposed solution:

- **Event-based solution:** for lowering the number of events to be processed by the demodulation algorithm block and the post-processor block (microcontroller). Also, an *"Event-Based" (EB)* solution should reduce the electromagnetic emissions (also called electromagnetic noise) for improving the communication performance [24].
- **Full digital demodulation method:** in order to use the latest technology nodes, for reducing the power supply voltage and hence the energy-harvesting constraints. This digital solution should be asynchronous for having a better match with the event-based behavior of this solution, and because it is known for being more robust on PVT variations.

3.3 Conclusion

In this chapter, a revision of the main characteristics of the key NFC tag protocols has been presented. Then, a review of the principal ASK demodulation solutions has been shown, including a typical analog solution structure and the three most relevant digital solutions found in the literature: a digital IQ demodulator, a digital frequency divider, and a method using several pulse shrinking modules in series.

The digital IQ demodulator has the advantage of being fully digital. However, it has the disadvantage of requiring an operation and sampling frequency of at least four times the carrier frequency. It also needs a high speed ADC and integrates several large combinational modules for the multiplications and the low pass filter. On the overall, these factors increase its power consumption.

The digital frequency divider demodulator is a very simple and small solution. However, it is extremely sensitive to amplitude noise because it is able to switch to an unknown state if any insertion or deletion error happens due to amplitude noise.

The pulse shrinking solution is very smart and is effective in a RTL simulation. Nevertheless, in a real implementation, it encounters many problems if the shrinking path and the clock path are not balanced. In addition, the PVT variations strongly affects the performance of this system, since these two paths can be unbalanced after fabrication and under different operational conditions. The performance also depends on the choice of the correct pulse shrinking output of the system. The latter is fixed and its choice is affected by the amplitude noise. Overall, the solution is unstable on PVT conditions and the amplitude noise, diminishing considerably its performance.

Finally, the main goals of the proposal solution of this thesis have been described. They have been classified in two main items: finding an event-based solution and proposing a digital demodulation method.

4 ASK demodulator architecture and implementation

4.1 Introduction

The NFC protocol, part of RFID standards, is a short-range wireless technology, requiring a distance of 10 cm or less to allow the communication between two electronic devices. It is composed by a reader, initiating the communication, and a tag replying to it. The tag can be a battery-powered device incorporating a stable power supply with stable operation conditions, or it can be a battery-less appliance with the ability to recover energy from the reader electromagnetic field.

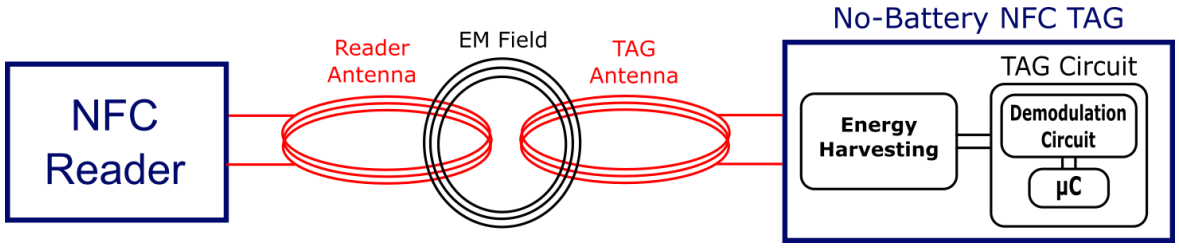


Figure 25: NFC wireless communication diagram.

In the communication process, the user approaches the tag to the reader, incrementing the field caught by the tag. More the user approaches the tag, the more the field is captured by the tag antenna. This effect increases the induced voltage in the antenna terminals. As described in chapter 3.1.2, most of NFC solutions in literature are analog because they offer different solutions for addressing the following problems:

- Voltage limitation: For avoiding high voltages that can damage the NFC tag circuits.
- Voltage gain adaptation: Normally using techniques like an automatic gain controller (AGC) or a variable impedance adaptation. The objective is to stabilize the amplitude voltage of the modulated signal recovered from the antenna's terminals.
- Energy harvesting circuit: Recovers some energy from the antenna terminal for generating the power supply voltages of the tag. It normally integrates a charge-pump, and the impedance adaptation circuits.

- Power-on-reset (POR): Generates a reset pulse for the digital circuits, when the power supply voltages are stabilized.
- Demodulation circuit: In all the solutions proposed in literature different implementations of the same approach was proposed, using an average detector, an envelope detector and a comparator, as shown in Figure 20.

The main objective of this thesis is to push the boundary between analog and digital domain, for minimizing the analog part as much as possible. The goal is to propose a new digital technique for ASK demodulation, enabling the relaxation of the design constraints of the previously described analog modules, by focusing on the following points:

- Diminution of the number of passive elements used in the system by directly applying a simple ADC module to the modulated signal and then a digital demodulation algorithm.
- Use of the latest technology nodes for decreasing the power supply voltage.
- Reduction of the power consumption, by reducing the number of events of the digital system, when using an asynchronous circuit, and taking benefit of the lower power supply.
- Spread of the current peaks of the digital system by implementing an asynchronous circuit solution.

Considering the points exposed before, this work is focused on the development of an EB-ADC, defining the boundary between analog and digital domains, and the digital demodulation algorithm allowing to obtain the demodulated data, as presented in Figure 1. We assumed that proposing this novel digital demodulation solution will open new perspectives for easing the development of the necessary analog modules in future works, leading to considerable improvements in the overall system.

For the digital designs of this chapter and all its simulations, we assume that the RF modulated signal has been normalized to a unitary amplitude, that the power supply voltage is 1 Volt, and that the reference voltages are a percentage of the supply voltage.

4.2 Testbench: Simulation Scenario

4.2.1 Matlab basic functions

As a first step for finding an EB-ADC solution with a minimal number of levels, a set of basic functions has been created in MATLAB programming language, in order to emulate the digital behavior of the registers, muller gates and several logic elements that are not defined by default in the MATLAB environment.

4.2.2 RF Input signal definition

Before finding the first proposal of EB-ADC solution, we focused on generating the RF input signal of the system, using a sinusoidal signal at a frequency of 13.56 MHz, which is the frequency usually used for the NFC protocol. For the Amplitude Shift Keying (ASK) modulation, two amplitudes are defined for *A* and *B*, *A* being the highest amplitude and *B* the lowest amplitude of the modulated signal, as depicted in Figure 27.

In the definition of Amplitude Modulation (AM), the modulation index (MI) is the ratio between the maximum amplitude of the modulating signal $\frac{(A-B)}{2}$ and the maximum amplitude of the carrier signal $\frac{(A+B)}{2}$.

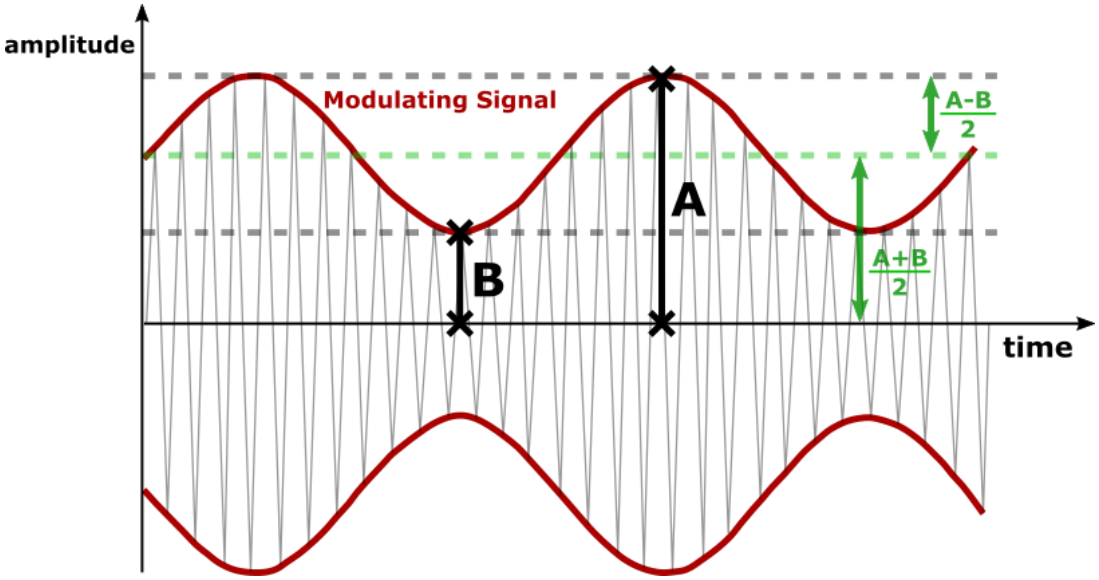


Figure 26: Time domain representation of an AM signal.

The equation of the MI is presented in Equation 4-1, using the amplitudes definitions of Figure 26. In this equation, *MD* is defined as the *Modulation Depth*, which is the ratio between *B*, the smallest amplitude of the modulated signal, and *A*, its largest amplitude.

$$MI = \frac{\frac{(A-B)}{2}}{\frac{(A+B)}{2}} = \frac{(A-B)}{(A+B)} = \frac{\left(1-\frac{B}{A}\right)}{\left(1+\frac{B}{A}\right)} = \frac{1-MD}{1+MD} \; where \; MD = \frac{B}{A}$$

Equation 4-1: Calculation of the Modulation Index (MI) in an Amplitude Modulation.

In the case of ASK modulation, the modulating signal is digital, then the modulated signal switches between amplitude levels A and B, as presented in Figure 27. The modulation encoding defines the symbol amplitude and when the amplitude must change within each symbol period. In the testbench of our solution, the NRZ-L (Non-Return-to-Zero Level) encoding was used. In that encoding, the amplitude changes at the beginning of each symbol period. The voltage level A, which is normalized to 1 Volt, represents a bit 1, and the voltage level B, which depends on the MI, corresponds to a bit 0.

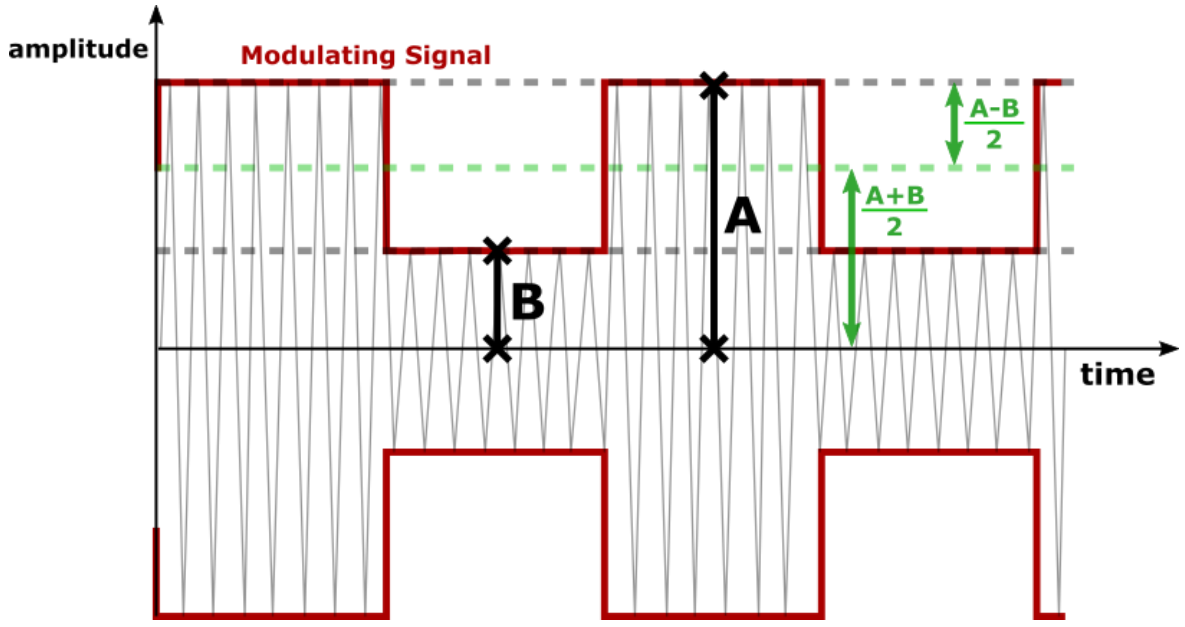


Figure 27: Time domain representation of an ASK modulation signal and NRZ coding.

The testbench programmed in Matlab generates a settable number of 32 random bits packets. Then, the 13.56 MHz carrier signal is modulated using the generated

packets for simulating the transmission of these random bits using the NRZ-L encoding.

Subsequently, the *awgn*[1] function was applied to the base modulated signal, for adding a *white gaussian noise*, and hence emulating a communication channel, where a Signal-to-Noise Ratio (SNR) value must be set up for defining the level of noise in this channel. Lower values of SNR correspond to higher levels of noise in the channel.

The result of this function is an ASK modulated signal, used as the RF input signal of our proposed ASK demodulation system, at which we can set the level of ***SNR***, the level of ***MI*** and the number of ***packets*** to send.

4.2.3 Level-Crossing Sample Scheme Selection

In an EB-ADC solution, as a first step, a LCSS must be selected in order to be specifically adapted for the application we are facing. In the case of an ASK demodulation system, the main challenge is to detect the amplitude change occurrences of the modulated signal and identify its change instants.

Several LCSS have been simulated with different numbers and positions of the reference levels. The first trial used only one level, set between amplitudes *A* and *B*, in order to identify the presence or absence of high amplitude periods. This method was almost correct for this purpose. It normally works in an ideal environment with ideal comparators and without communication noise. However, in a real case, the amplitude noise provokes glitches in the level crossing detection, and hence several unnecessary input events for the demodulation system. Therefore, adding a second reference level is necessary in order to obtain a stable high-level amplitude identification. The new level (called $L1$) has been set near the ground in order to detect each carrier period as a zero-crossing detector. The second reference level (called $L2$) is set near the maximum nominal amplitude (normalized voltage) of the modulated signal, as depicted in Figure 28, for avoiding generating faulty events on low amplitude periods (event insertion error). Using these two levels as inputs of a Muller gate has been proved, by simulation, to be a good method for stabilizing the recognition of the high amplitude level crossing, even in presence of amplitude noise.

[1] Function belonging to Matlab's Communications Toolbox.

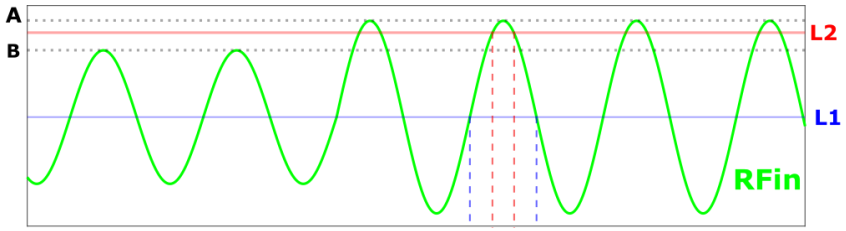


Figure 28: Chosen Level-Crossing Sampling Scheme.

4.2.4 First Proposed Solution

The first proposal of this thesis, presented in Figure 29, is a simplified EB-ADC solution. It uses the LCSS defined in Chapter 4.2.3. For addressing the level crossings detection of $L1$ and $L2$, two comparators have been applied directly to the RF input signal (modulated signal), one for each reference voltage level. Then, a C-element is applied to both comparator outputs to obtain a stable pulse, on each positive carrier period when the $L2$ level is crossed, going from the first rising-edge of *L2Sig* signal to the falling-edge of $L1Sig$ signal. This element allows to synchronize the falling-edge of $COut$ signal with the falling edge of the "*clock extraction*" signal $L1Sig$. With the objective of registering the presence or absence of the pulse at $COut$ on each carrier period, this technique takes advantage of that synchronization, slightly delaying the $COut$ signal, for then activating a D-type flip-flop using the inverted signal $L1SigInv$. This method is very simple, it uses a small area because it contains only few digital elements, and is also low-power consuming. However, this method is very sensitive to amplitude noise, since it can easily produce two kinds of error: "deletion errors" ($\boldsymbol{D_{err}}$), which means the absence of a $L2$ level crossing in high amplitude periods, or "insertion errors" ($\boldsymbol{I_{err}}$), which means the occurrence of a $L2$ level crossing in low amplitude periods. Thus, this technique is only useful for systems with low noise or using a high *MI*. In addition, one of the drawbacks of this technique relies on the presence of $L1Sig$ signal, then it will not work for *MI* equal to 100%, when $L1Sig$ is absent on low amplitude periods.

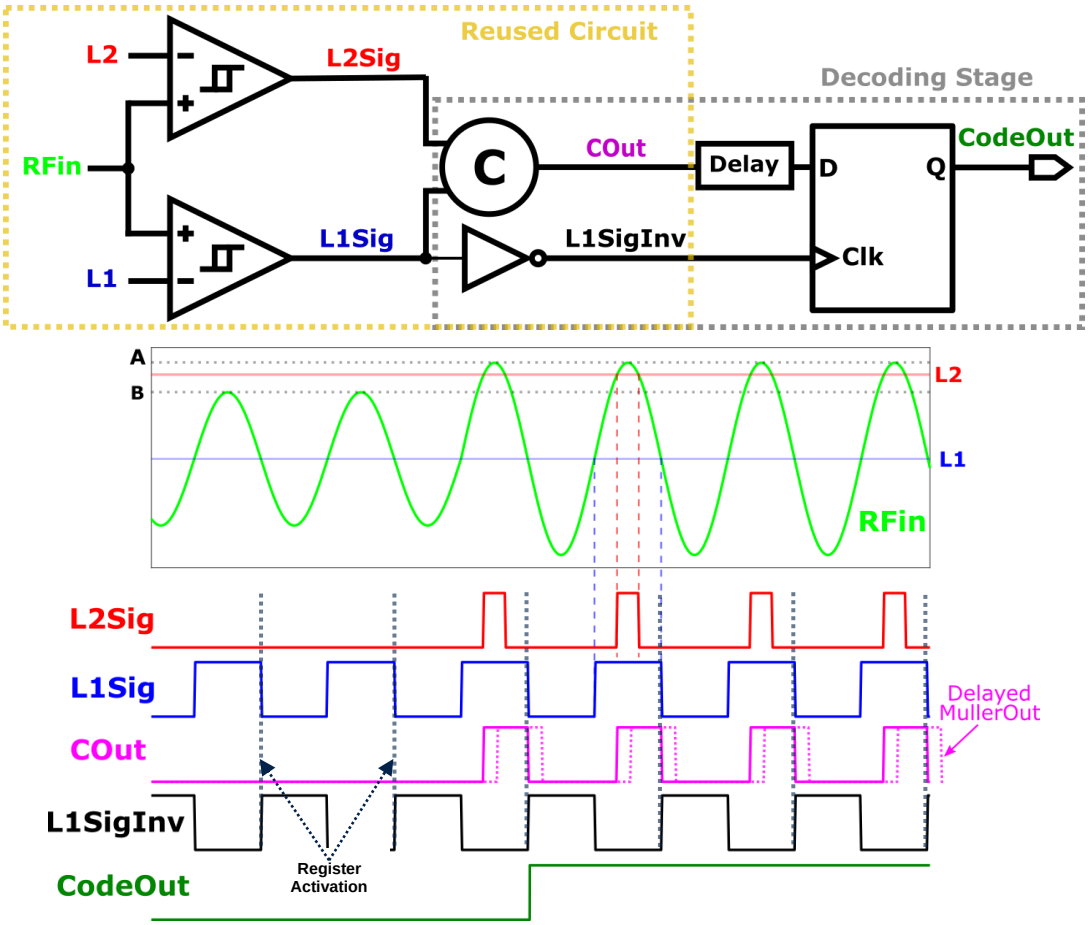


Figure 29: Schematic and signals of the circuit presented in the conference EBCCSP 2019 [26].

4.2.4.1 L2 Reference level Selection

For the $L2$ reference level selection, an empirical test has been made for choosing the level generating the best *"Bit Error Rate v/s Signal-to-Noise Ratio"* (BER v/s SNR). A set of simulations have been done using the demodulation system presented in Figure 29, to identify the best position for the $L2$ reference voltage. For these simulations we fixed the modulation index *(MI)* to 10%, which is the standard *MI* used in NFC-B, and the symbol period to $128T_C$, which is equivalent to the minimum bitrate. Then, the $L2$ reference voltage has been modified from $0.9\ V$ to $1.0\ V$ with $0.01\ V$ step. Then, we identified for which SNR level the BER equal to 10^{-4} have been reached in order to obtain the graph depicted in Figure 30.

It is important to highlight that the simulations have been executed using a testbench programmed in Matlab, where the input signal has been created as described in Chapter 4.2.2 and the logical modules has been emulated as defined in Chapter 4.2.1. The voltage $Vdd = 1\ V$ has been defined as the normalization voltage

of the maximal amplitude for the modulated signal. Thus, the reference level presenting the best BER v/s SNR results is $0.96\,V$ ($96\%\ of\ Vdd$).

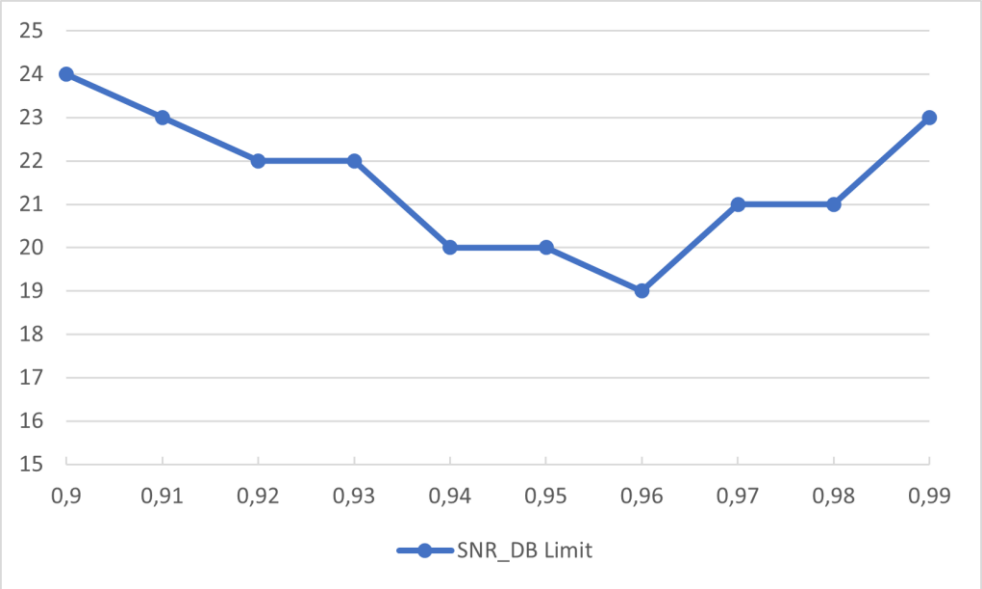


Figure 30: Graphic showing the simulation result used for the L2 reference level selection.

4.2.4.2 First Tests in Matlab

Figure 31 presents one example of the Matlab simulation testbench. The RF input signal was generated with a MI of 3% and a SNR of 28 dB, and a random 32-bits communication packet. The blue line represents the input modulating code, the pink line corresponds to the output code identified by the system at the output of the register of Figure 29 ($CodeOut$ signal), and the black stars denotes the code identification capture instants, taking a snap value of the output code exactly in the middle of each symbol period.

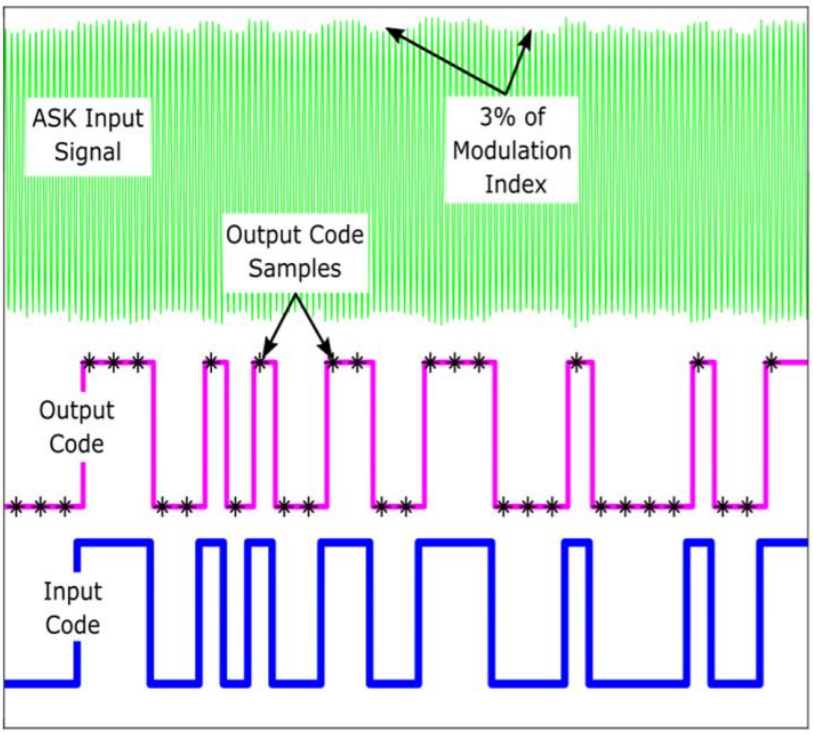


Figure 31: Simulation Result of the first proposed EB-ASK demodulation technique for SNR=28 dB, MI=3% and using a NRZ-L encoding.

In this case, the solution proposed in Chapter 4.2.4 correctly identifies the 32 bits modulating the signal of this communication example.

We can recognize that the output signal of the system, $Output\_Code$, correctly reconstructs the input code. For obtaining the 32-bit output, in order to perform a comparison between the input bits and the output bits, this output signal was sampled at the symbol frequency ($Bitrate$) exactly in the middle of each symbol period (T_{symbol}). This comparison allows the calculation of the "Bit Error Rate" (BER) of our system for the simulation of each 32-bit packet.

For obtaining the response of our demodulation system, the same simulation was repeated 500 times (500 x 32 bits = 16000 bits) for each value of *SNR* and *MI*. The result of this full simulation testbench is depicted in Figure 32. These first results only simulate the demodulation of a fixed number of bits.

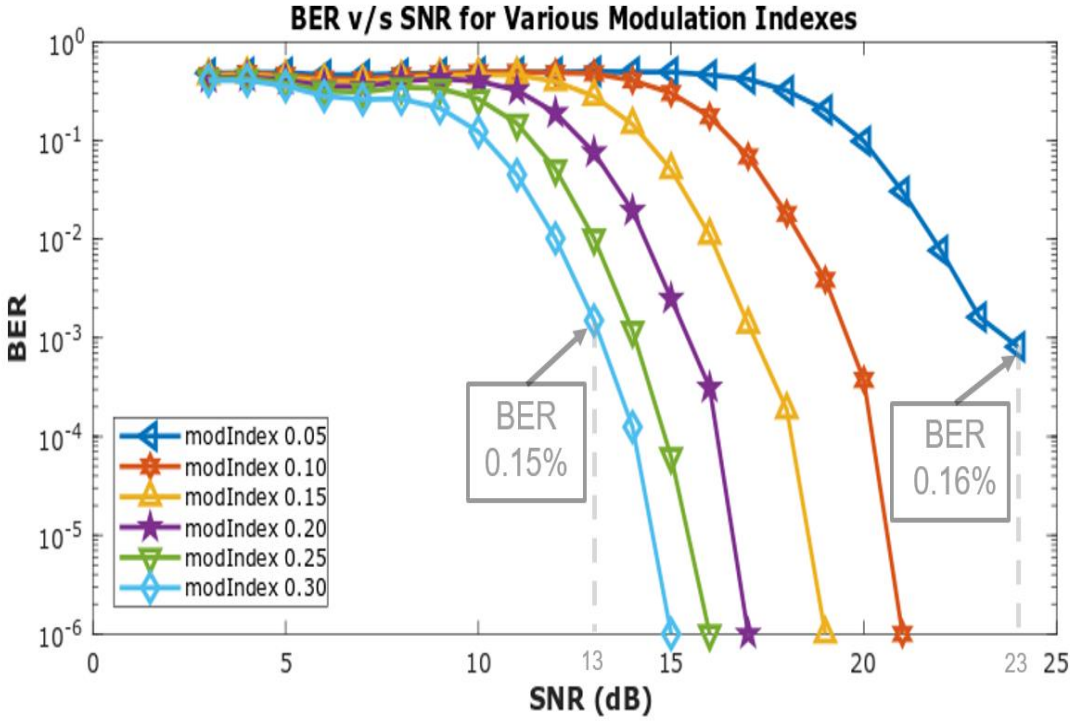


Figure 32: BER v/s SNR simulation for various modulation indexes.

After reviewing these results with experienced colleagues at STMicroelectronics, they suggested to use a more reliable technique for obtaining a more realistic value of BER. This technique relies on simulating, for each SNR value, the demodulation proposal until a fixed number of demodulation error occurs in the system. Particularly, they suggested stopping each simulation when 20 demodulation errors are reached.

4.2.4.3 Testbench adaptation to SystemVerilog

In order to obtain a more reliable and realistic BER value of the solution proposed in Chapter 4.2.4, a simulation with a longer bitstream of several millions of transmitted bits has to be done. The Matlab simulations presented in the previous chapter were too slow to achieve this objective in a reasonable time because it contains some inevitable loops in the program code.

For accelerating the simulations and validating them in a digital integrated circuit simulation environment, a *SystemVerilog* testbench has been implemented and then simulated using the software *QuestaSim (Siemens EDA)*. This testbench integrates several parameters and details permitting to emulate a real communication environment for an ASK modulation using a NRZ-L encoding.

This testbench produces 32-bit random packets, which are modulated using a carrier at 13.56 MHz. Each random packet is obtained using the *urandom* function, which provides a random unsigned 32 bits integer as a result.

In ASK modulation with NRZ-L encoding, a one is represented by a high amplitude A and a zero is represented by a low amplitude B. For each simulation, the *MI* is calculated using the Equation 4-1. This equation permits to infer that when the *MI* is close to zero, the two amplitudes A and B tend to be similar. In that case, in presence of amplitude noise, the system has more difficulties to accurately demodulate the RF signal.

In order to emulate a communication channel, the "*Additive Gaussian White Noise*" (AWGN) technique has been used, by replicating the *awgn* function of *commpy* communications library of Python [27]. This function has been reproduced, using the *SystemVerilog (SV)* language, for been integrated in a RTL digital simulation environment. The latter permits to add a white noise, expressed as SNR in dB, to a RF modulated signal.

For generating the RF signal (including the noise), a *SV* module has been created. This module receives the following listed input parameters for emulating the different communication scenarios:

- The size of each packet to be sent (default value: 32 bits).
- The noise level, expressed in dB (SNR).
- The modulation index (MI).
- The relation between the carrier frequency (also called bitrate) and the symbol frequency $\left(f_{symbol\_mult} = 128 \cdot \frac{f_{symbol}}{f_c}\right)$.

With the purpose of being compatible with the RF testbenches and the laboratory measurements of real RF signals obtained by the NFC team at STMicroelectronics, the following values have been adopted for the simulation steps and the generation of the RF signal:

- A sample period (t_s) of 576,16 ps.
- A carrier period (t_c) equivalent to $128 \cdot t_s$.
- A symbol period (t_{symbol}) represented by $t_{symbol} = \frac{128 \cdot t_c}{f_{symbol\_mult}}$.

The resulting module, called *signal_gen*, allows generating any ASK modulated signals, including amplitude noise, in a professional digital simulation environment for

being applied at the input of any digital demodulation solutions under test and comparison.

Another module, called *comparator*, has been created in *SV* to imitate the behavior of a Strong-Arm comparator (*SAC*) described in [28], by executing an ideal comparison of the two input analog values, and by applying a comparison period of $3\,ns$ for the comparator at $L1$ (near *gnd*), and $2\,ns$ for the comparator at $L2$ (at $Vref = 0.96 \cdot Vdd$). These comparison periods were validated using an analog design of these two *SAC* on the *Cadence Virtuoso* tool, employing electrical simulations.

For all the simulation results obtained using SystemVerilog in this work, the value of $Vref$ has been fixed to 96% of Vdd, as suggested in Chapter 4.2.4.1.

Then, the main testbench module includes several options, defined as key simulation parameters that allow running independent simulations for each value of *MI*, each *SNR* level and each f_{symbol_mult} number. These simulations can be launched on a graphical interface (gui) or on a tcsh terminal, as desired, and also integrate three parameter options for stopping the simulations:

- By the simulation time limit (simTime).
- By the number of erroneous demodulated bits (bitErrorLimit).
- By limiting the total number of transmitted bits (simTotalBits).

A "*make"* script was created to control the parameter values given for each simulation at a high level, as well as defining the design stage of the simulated solution: Register Transfer Level (RTL), Synthesis (Synth) or Place and Route (P&R).

Finally, a "*python*" script was created to call the "*make*" script for each simulation alternative, and thus being able to run several simultaneous simulations in the UNIX STMicroelectronics environment, which has hundreds of "*QuestaSim*" available servers and licenses.

Finally, a *python* script is used to aggregate all these individual results, in order to create the BER v/s SNR result chart for each solution.

4.3 Results obtained with the SystemVerilog testbench

4.3.1 First proposal results

The first proposed demodulator solution, depicted in Figure 29, was simulated using the ASK modulation environment, created with the SystemVerilog language, and using the professional digital simulation tool "*QuestaSim*" from "*Siemens EDA solutions*", as described in Chapter 4.2.4.3.

This demodulator was simulated for the following cases:

- SNR from 4 to 31 dB.
- Symbol period (T_{symbol}) from $2T_C$ to $128T_C$, equivalent to f_{code_mult} from 64 to 1.
- MI of 10%.
- Vref equals 96% of Vdd.
- simTime of 40 s.
- bitErrorLimit of 20 bits.
- simTotalBits of 3,000,000 bits.

The simulation results are presented in Figure 33. This graphic clearly shows the limits of the proposed demodulator against amplitude noise. In this case, the *BER* mainly depend on the rate of "deletion errors"[2] ($\boldsymbol{D_{err}}$) caused by the amplitude noise on high amplitude periods. This design directly catches the level crossing events on $L2$, and does not integrate any methodology to mitigate the "deletion or insertion errors". This infers that the results are independent of the symbol period because all the curves are almost superposed. That corresponds with the simplicity of the solution.

[2] Level crossing not occurring on $L2$, during high amplitude periods. An example is depicted in Figure 34.

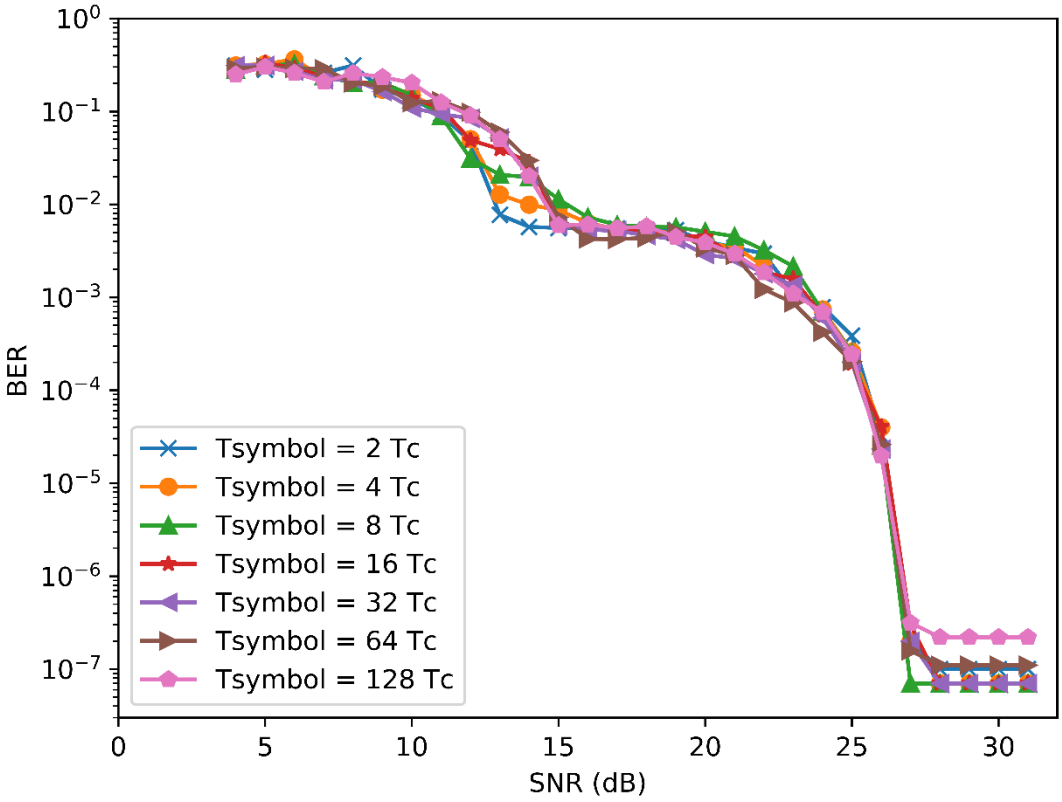


Figure 33: Results of the 1st proposed demodulation solution, for an MI of 10%, using the digital SystemVerilog testbench.

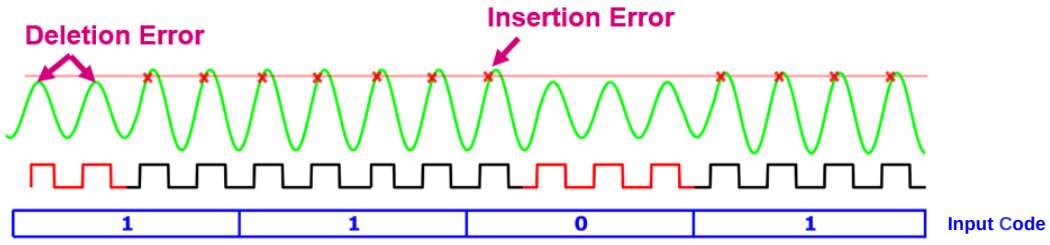


Figure 34: Example of insertion and deletion errors

It is important to specify that for all the simulations presented in this manuscript, the calculation of the *"Bit Error Rate"* (BER) is the ratio between the total erroneous demodulated bits (*errorBits*) and the total transmitted bits (*totalBits*), as presented in Equation 4-2. On each $BER\ v/s\ SNR$ graphics, the BER axis is represented in a logarithmic scale. For that reason, in the cases when the $errorBits$ value is 0 after a simulation of the transmission of millions of bits has finished, the BER has been calculated using an arbitrary value of 0.2 $errorBits$, just for obtaining a valid visualization value for the graphic. We assume that if we go further with the simulation, the system will generate an error at some point.

$$BER = \frac{errorBits}{totalBits}$$

Equation 4-2: Calculation of the Bit Error Rate (BER) in the ASK demodulator simulations.

In the case of $T_{symbol} = 128Tc$ and $T_{symbol} = 64Tc$, the transmission of the total number of bits could not be reached before the end of the simulation was attained by the simulation time limit (simTime). For that reason, the minimum BER limit for these two cases of T_{symbol} is different than the other cases.

4.4 Conclusion

This chapter describes firstly the structure of a NFC tag device, which is composed by different modules, for recovering the energy from the electromagnetic field, for avoiding overvoltage risk, adapting the antenna load and stabilizing the signal inputted to the demodulation circuit. In this thesis only the demodulation circuit is addressed, so a stable and normalized modulated signal is considered for testing the proposed solutions. The objective of this thesis is to propose a new digital demodulation method with a stable and improved performance compared to the existing digital solutions found in the literature. Additionally, an event-based solution has been proposed for generating a minimal number of events in order to reduce the power consumption. Asynchronous circuits are well suited for event-based techniques, reducing the power consumption and increasing the robustness to the PVT variations.

A testbench environment has been set for generating the input RF signals, the additive white gaussian noise, processing the demodulated symbol and comparing it with the input code. First, this test simulation environment was programmed in Matlab, and, then, has been adapted to SystemVerilog for enabling its use in all the steps of the integrated circuits design flow.

A first demodulation proposal has been presented, using only a few digital gates. The demodulation capabilities of this solution are correct and its timing constraints are controlled. However, its performance results are quite sensitive to the amplitude noise when the modulation index is reduced. As expected, the demodulation performance of this solution is not affected by the bitrate because the technique is sensitive to the generation of insertion or deletion errors, regardless of the bitrate.

In order to enhance the performance of this solution, two improvements of this approach are presented in the following chapter.

5 Enhanced ASK demodulators

5.1 Second Proposal results

5.1.1 Introduction

As depicted in Figure 35, the new demodulation system described further in this manuscript maintain the same LCSS as the first proposal. As described before, it defines only 2 crossing thresholds, a first level ($L1$) near the reference of the system (gnd), and a second level ($L2$) near the maximum amplitude of the modulated signal (Vdd). The whole solution is divided in two main blocks, an EB-ADC and a digital decoding stage.

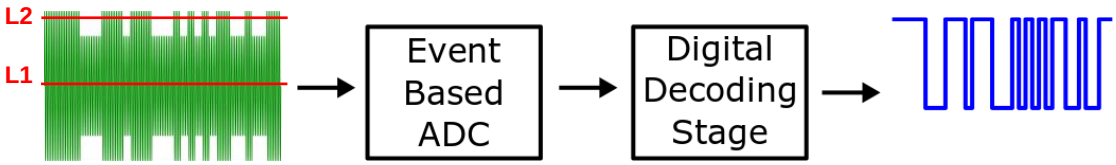


Figure 35: High-level diagram of the demodulation system and the LCSS.

The EB-ADC is composed of two main blocks, the *level crossing detector* and the *TDC*. In this case, the level crossing detector will maintain the same structure of the left part of the first proposal, integrating the two comparators and the Muller-gate as presented in Figure 36. As already stated before, this structure permits to obtain a sensitive and stable $L2$ crossing that will be used as an input event indication (*request*) for the asynchronous control circuit of the "digital decoding stage" solution.

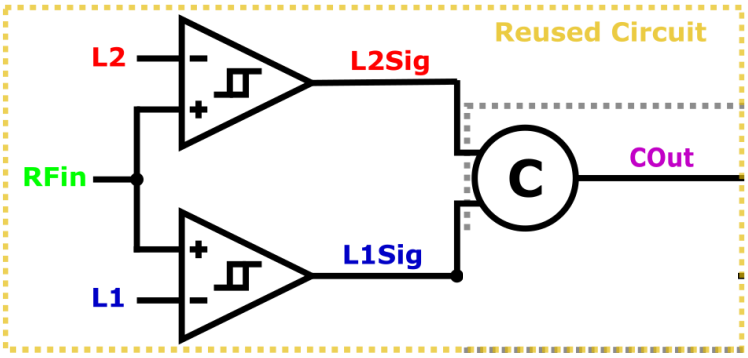


Figure 36: Module Performing the Level Crossing Detection.

Then, a TDC measures the time between consecutive events. Nevertheless, usually the TDC are intended to perform one measurement between a *Start* and a *Stop* signal. In this case, a novel method must be designed to enable a continuous time

measurement, where the end of one measure is the start of the next one, as depicted in Figure 37.

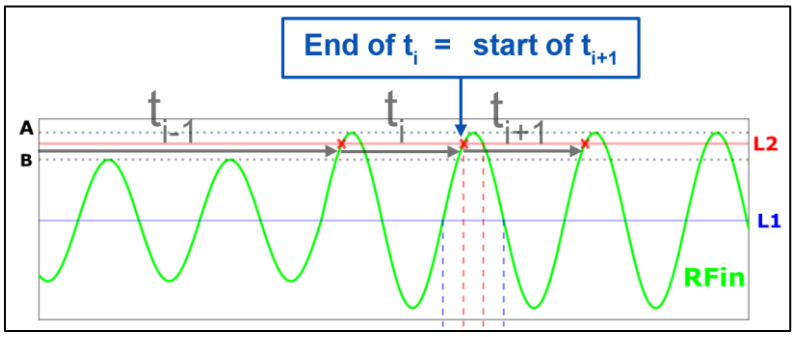


Figure 37: High-level description of the Continuous measurement TDC.

For the initial purpose of validating the full ASK demodulation method, the first RTL simulations do not use a real TDC solution. Instead, the time measurement was done by directly using the simulation time obtained by the internal variable *"$time"*, and comparing it with the time already saved in the "*lastTime*" variable corresponding to the time of the last event (level crossing) iteration. The challenge of adapting a normal TDC, for the implementation of a real continuous time measurement TDC, is described further in this document.

Finally, a digital demodulation algorithm module, using the time measurement already obtained from the EB-ADC, is implemented to improve the robustness of the system against amplitude noise.

5.1.2 Second Proposed Solution

The first proposed demodulation technique, presented in the previous chapter, offers a minimal LCSS and permits a correct and stable detection of the high amplitude level crossing ($L2$ level). This event-based solution properly uses the level crossing events without taking care of the time between events.

For improving the results obtained with this first proposal, the system must also integrate the time measurement between consecutive level crossing events. Then, this time measurement is applied in a digital demodulation algorithm for making the demodulation more robust to the amplitude noise.

In this chapter, a new solution is proposed, always based on the events as the previous solution, but also using the time measurement between two events. Thus, this technique uses an "amplitude change detector" digital algorithm, for making it possible to identify the moment of symbol change (end of each symbol periods

T_{symbol}). Identifying this exact moment is a key requirement for the ASK communication standards, and it is normally the most difficult and area consuming task to achieve when using an analog demodulation solution.

As already stated in this document and in the literature, it is known that asynchronous circuits are especially well-suited to event-based solutions. They naturally react to events, and they stay idle between them. The top hierarchy of this kind of solutions has two main blocks: the **logic** circuit and the **control** circuit.

The logic circuit is composed by several interconnected pipelines, which are composed of combinational circuits (calculations) and an associated memorizing circuit (memory blocks) with a local clock signal. The control circuit is in charge of managing the activation of the local clock signal for all the pipeline stages in the circuit, as the example presented in Figure 13.

For this solution, only the “High Level Crossings” (*HLC*) will generate system input events (control circuit input requests). Thus, HLC are processed at the time of each event, while “Low Level Crossings” (*LLC*) are processed and inferred, using a time measure, in the first HLC event following the LLCs. This characteristic allows the demodulation to be performed even in the case of a 100% ASK modulation, where the carrier is absent (no signal) during the symbol periods where the amplitude is low.

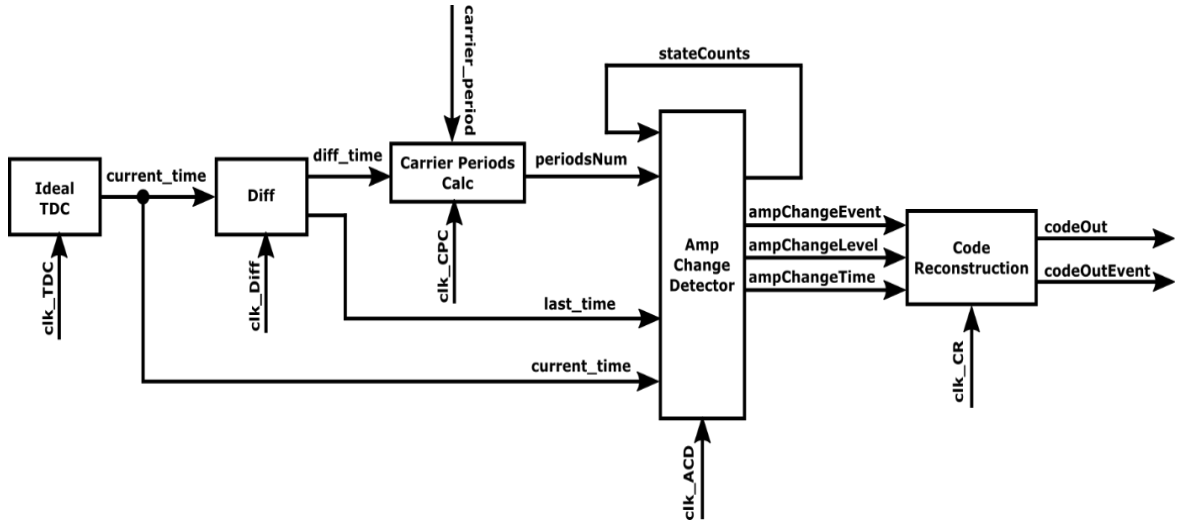


Figure 38: Logic Circuit description of the 2nd proposal (the control circuit is omitted by simplicity).

This 2nd demodulation implementation, depicted in Figure 38, is composed by the digital modules described in the following chapters.

5.1.3 Ideal TDC Module

This block was only used in the validation phase of this solution. It retrieves the simulation time (obtained by SystemVerilog internal function *$time*) on each *clk_TDC* signal rising-egde arrival. The result is available at *current_time* signal output. This module is not synthesizable.

5.1.4 Diff Module

This module compares, on each clock rising-edge arrival, the current time with the last event time, and outputs their difference (*diff_time* signal) and the latter (*last_time* signal).

5.1.5 Carrier Periods Calc Module

This block takes the time difference between the two last consecutive level crossing events and calculates the time equivalence expressed in number of carrier periods ($n \cdot T_C \Leftrightarrow periodsNum$). It needs an input, called *carrier_period*, with the TDC equivalent value of one carrier period. It performs the division $\frac{diff\_time}{carrier\_period}$ and outputs an integer number.

5.1.6 Amp Change Detector (ACD)

This module implements an algorithm that aims to detect the moment of the amplitude change of the RF modulated signal. In this algorithm, the *HLC* are detected and counted (when $periodsNum = 1$), and the *LLC* are inferred and counted (when $periodsNum > 1$). This method uses counters for the LLC and HLC events, for deducing the amplitude change moments of the modulated signal, without synchronizing the whole system with the communication sequence.

In this method, the system is always in a state where it has detected a high amplitude symbol period, also called an “Amplitude Change Event High” (*ACE-H*) or where it has detected a low amplitude symbol period, also called “Amplitude Change Event Low” (*ACE-L*). Then, the time of the first opposite event is saved in memory. This means that, if the system is already an *ACE-H state*, the time of the first *LLC* is

saved as a potential initial point for the next “Amplitude Change Event Low” (*ACE-L*). That low symbol period initial point is only confirmed if the counter of the LLC events reaches the *“sensitivity number” (SN)*. This number is set to 3 by default but it is a parameter of the system. If *SN* is smaller, the system is more sensitive to wrong symbol period changes and, if it is larger, it is more stable. The number 3 has been chosen to allow the system to reach a bitrate of $4Tc$, while keeping some stability. Another behavior of the systems is that, if two consecutive *HLC* occur while been in an *ACE-H* state, the *LLC* counter and the potential time for an *ACE-L* are reset. This same behavior is valid for the opposite situation, when the system is in an *ACE-L* state and a first *HLC* is detected as a potential initial point for a new *ACE-H* period.

This method analyzes and detects a tendency change in the amplitude of the modulated signal. In the example of Figure 39, the sensitivity number is 3. The output indicating a change of symbol, *"ACE-L"*, is generated at the time of the 1st *HLC* after the *LLC* events have reached the *SN* (orange ellipse) because the system is activated only when *HLC* event occurs. *LLC* are inferred and integrated to the system only at the *HLC* instants. The time associated with the “*ACE-L”* is the first *LLC* detected (purple cross).

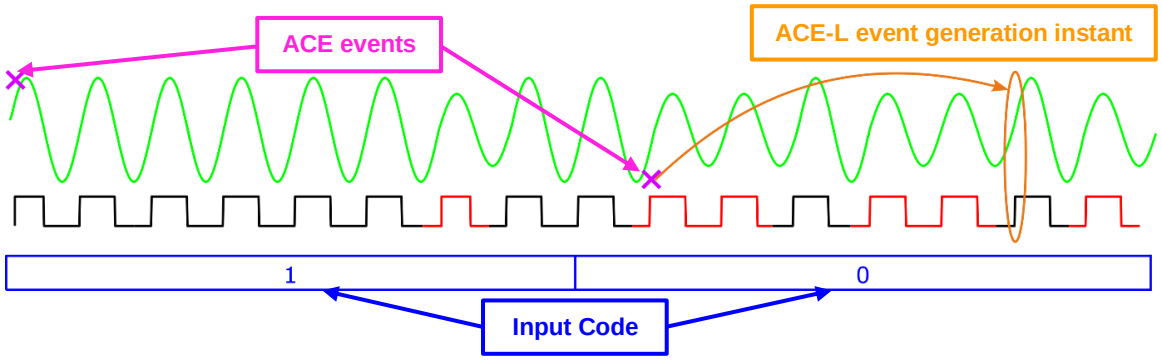


Figure 39: Example of the implemented algorithm in the « Amp Change Detector » block.

5.1.7 Code Reconstruction Module

The function of the *“Code Reconstruction”* module is to collect each demodulated bit from the *“Amplitude Change Detector” (ACD)* module and group them in 32-bit packets to be outputted to the simulation techbench environment. The behavior of this block is very simple. It implements a shifter in an internal 32-bit register, filling each bit with a *“one”* or a *“zero”* depending on the input value coming from the *ACD*. When the bit 32 is reached, the full 32-bits are outputted using isolated registers and the internal

ones are reset. This module is only activated when a new demodulated symbol is identified and is idle otherwise.

5.1.8 Simulation Results

This proposal was simulated using the testbench described in chapter 4.2.4.3. The BER v/s SNR graph's result is shown in Figure 40.

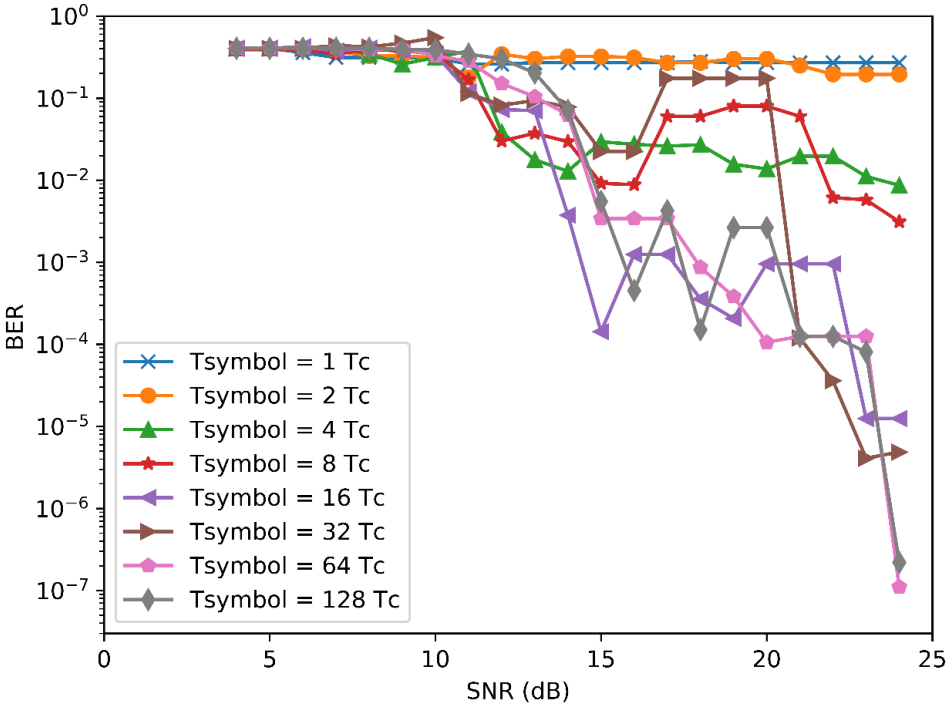


Figure 40: BER v/s SNR results of the 2nd proposed solution, using a MI of 10%.

From last Figure, it can be determined that this technique enables a robustness improvement against amplitude noise, compared to the 1st proposal. In this case, the BER response is reduced around 22-24 dB, for cases when T_{symbol} is greater than or equal to $16T_C$. The last proposal BER improvement was around 26 dB for the cases when T_{symbol} is equal to $4T_C$ and $8T_C$. In the cases when T_{symbol} is T_C and $2T_C$, the system does not respond at all, mainly because of the chosen *SN* number.

The second proposed solution allows the identification of an approximated symbol period end without requiring the synchronization of the demodulator regarding the communication sequence, which can be seen as an advantage. However, that identification is not enough accurate and the BER improvement is not as good as expected. The effectiveness of this technique depends too much on the number of the generated insertion and deletion errors and if they are generated in the middle of the

symbol period or near its transition. Thus, the result is quite hazardous, and it is not stable, as it can be clearly deduced from Figure 40.

For correcting these problems, a new solution has been developed, in order to obtain a better BER v/s SNR response and a precise period symbol end identification. This new proposal is presented in Chapter 5.2 .

5.2 Third Demodulation Solution

After analyzing the results obtained for the second proposal, it can be deduced that the tendency change detection is too sensitive to noise. Indeed, in presence of a high level of noise, it can generate additional output bits and dephasing the symbol change instant detection.

During different tests performed on the second proposal, after analyzing the modulated signal as well as the output signals of the EB-ADC, we observe that performing a majority calculation of the *HLC* or *LLC*, mainly during large symbol period T_{symbol}, improve the correct detection of the output symbol, even with a strong noise level.

Consequently, the third proposal is an enhanced adaptation of the last proposal, which takes benefit of that previous observation. The new system maintains the capacity of measuring the time between two consecutives *HLC* events, using the same TDC module employed in the previous solution. However, in this case, it calculates the end time of each symbol period, and then the majority of *HLC* or *LLC* events during each period. That method defines the output bit value of each of these symbols.

The diagram of the complete system, including the logic circuit and the control circuit, is shown in Figure 41.

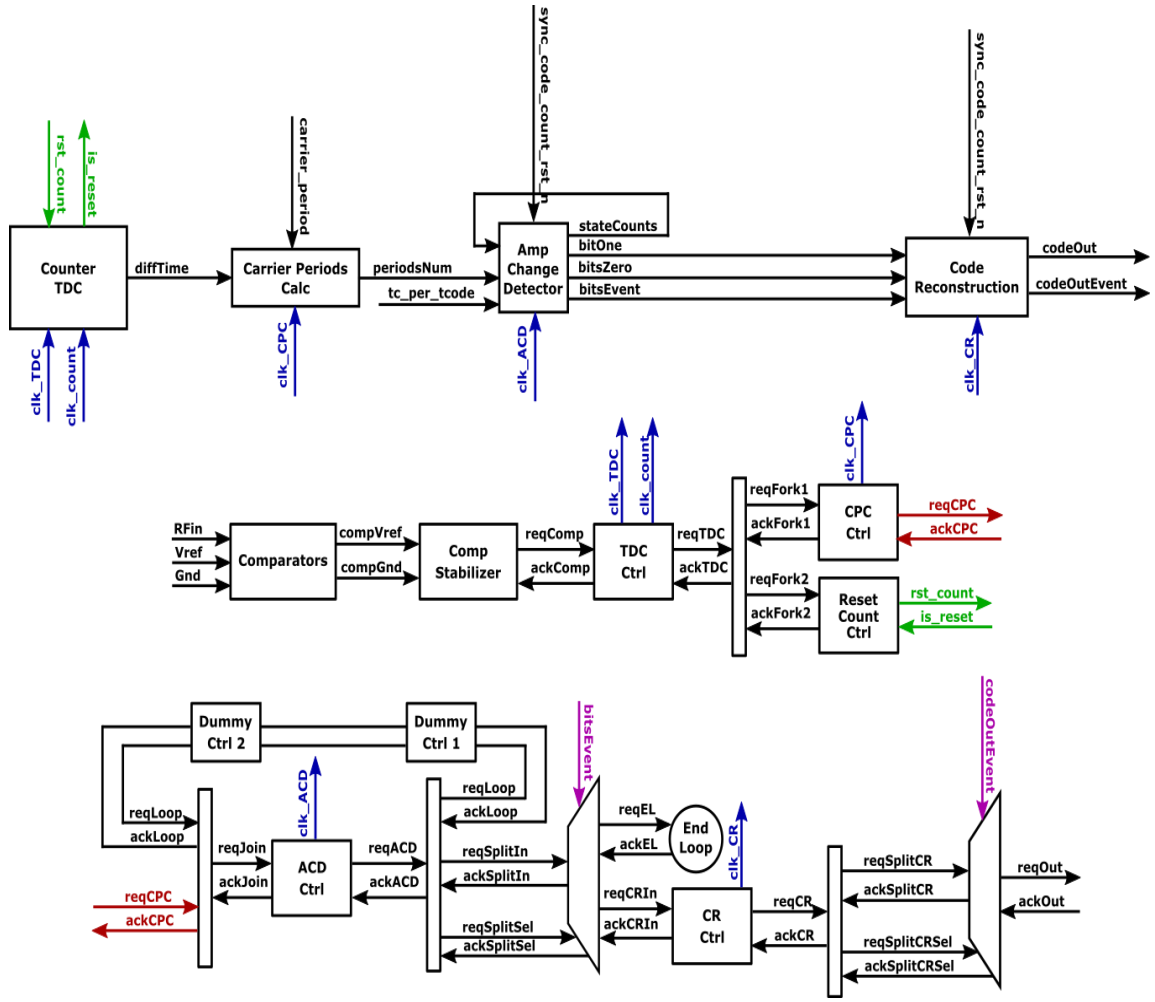


Figure 41: Diagram of the 3rd proposed solution.

5.2.1 "Amp Change Detector" module's logic

The main module is the *"Amplitude Change Detector"* (*ACD*) module. It enables the calculation of the end of each symbol period and computes the majority of the *HLC* or *LLC* events during that time.

The following figure shows a sequence of system input signals. The modulated RF signal is marked in green, HLCs are marked in black, LLCs in red, and finally the modulated input bits (or symbols) are marked in blue.

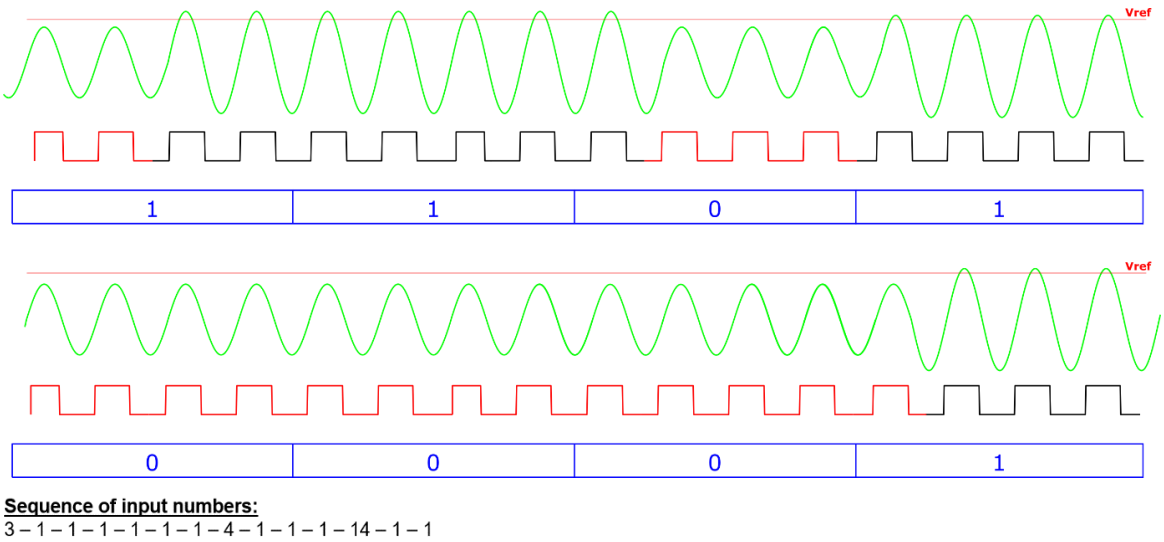


Figure 42: Example of the demodulation technique sequence.

The algorithm performed by this block is divided into 3 sections according to its input value *"Periods_num"*:

- $Periods\_num = 0$*:*
  - *Nothing to do.*
- $Periods\_num = 1$*:*
  - *Increment the total counter.*
  - *Increment the HLC counter.*
  - *Evaluate if the symbol period has been reached.*
    - *Evaluate the majority of HLC or LLC in this symbol period.*
      - *Generate an output event and indicate the majority result with a 0 or a 1 output bit.*
- $Periods\_num \neq 1$*:*
  - *Increment the total counter.*
  - *Increment the LLC counter.*
  - *Evaluate if the symbol period has been reached.*
    - *Process the majority operation between HLC and LLC until reaching the end of the first* T_{symbol}*.*
      - *A value 1 or 0 is saved, depending on the majority result.*
    - *Evaluate the number of reached extra* T_{symbol} *after the several consecutive LLC.*
      - *Save an extra 0 for each extra* T_{symbol}*.*
    - *Update the LLC and HLC counters, equivalent to their remaining values after the last processed* T_{symbol}*.*
  - *Generate an output event and indicate a result with the number of calculated 0's bits and 1 bit.*

5.2.2 TDC Implementation

In order to obtain a fully synthesizable solution of the proposed ASK demodulator, it is imperative to replace the TDC, which uses the simulation time, described in Chapter 5.1.3. For this purpose, a new TDC has been designed, using an internal ring oscillator (*RO*), producing an internal clock, which is required for the counter logic in

order to calculate the number of rising-edge occurrences of this clock and hence measuring the elapsed time. The TDC implementation for an event-based ASK demodulation should perform a continuous time measurement, where the stop of one measurement is the start of the next one, as described in Figure 37. This behavior differs from a normal TDC. Therefore, some modifications and adaptations have to be made for this application.

With this TDC proposal, three main problems have been encountered:

1. Create a circuit for resetting the TDC counter at each new event arrival.
2. Avoid the possible metastability issues caused by using an internal *RO* and a random input request instant arrival.
3. Ensure a safe current time measurement registration before the TDC resets.

The following sections describe the implemented methods for addressing each of these challenges.

5.2.2.1 TDC counter reset module

The first issue has been solved by using the TDC reset circuit depicted in Figure 43. This method mixes a 4-phase WCHB protocol, employed in our ASK demodulation solution (reqIn and ackIn handshake signals input) and a 2-phase handshake CLICK register controller modified and shaped for the reset process. This solution uses the phase-decoupled version presented in [29]. The combination with this 2-phase controller element permits to send a reset command in the first half-phase of the 4-phase protocol (*reset_counter* signal activation), while blocking the whole system until the reset application has been checked. During the second half-phase, the TDC control block waits for the reset confirmation signal "*isReset*", established by an *or_tree* module. This is done before switching back the input handshake signal "*ackIn*" to its initial state in order to unlock the system, thus accepting a new incoming request event and, hence, enabling the start of a new time measurement.

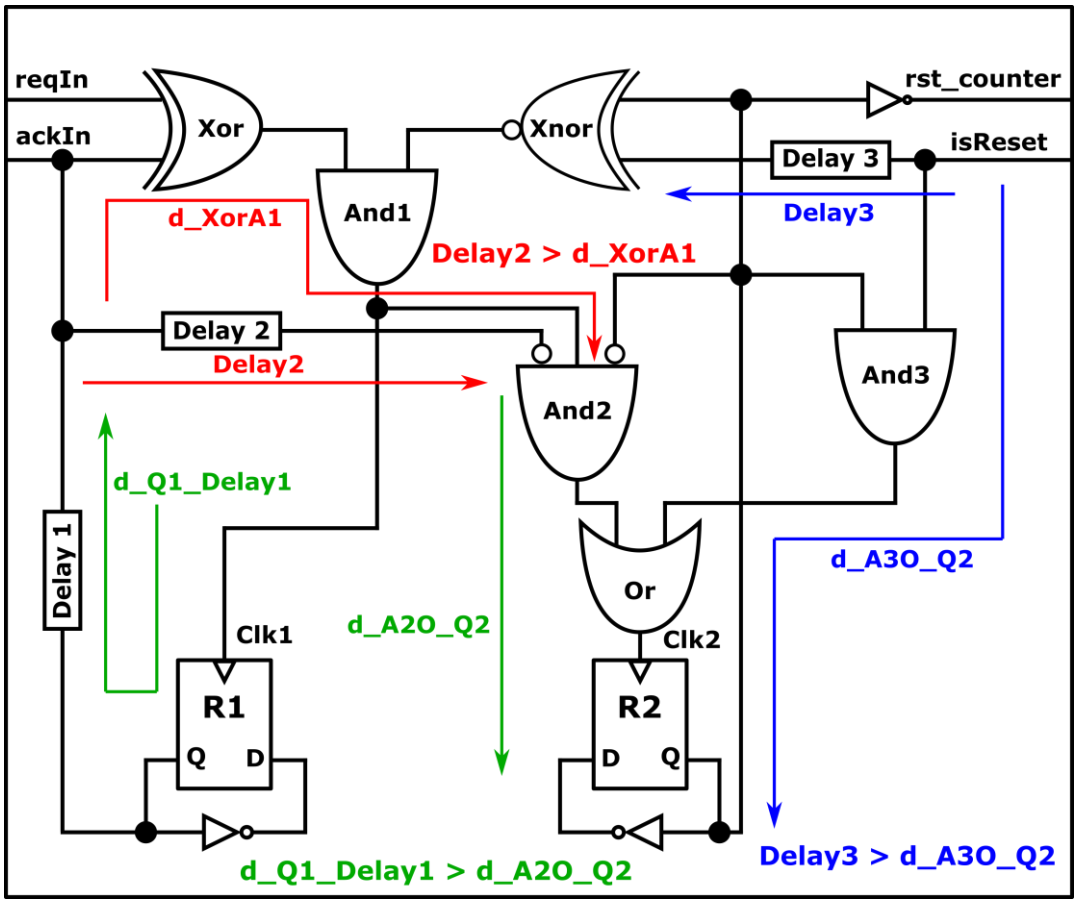


Figure 43: Diagram of the TDC reset circuit permitting to reset the TDC counter safely.

Three timing assumptions must be considered for a correct operation of this circuit, which are highlighted in colors in Figure 43. The first timing assumption to consider, using *Delay1,* is when a new input request arrives, then the rising edge of *Clk1* produces a switching state in *R1, and also in R2 in that particular case. Thus,* if the environment response is quicker than the switching state of *R2*, it can generate a wrong rising edge on *Clk1*. The second timing assumption is when the register *R1* returns to its initial state (value 0), then we must ensure that the propagation of the new initial state of *R1* through the *Xor* and the *And1* gates, and hence the return to zero of *Clk1 (d_XorA1)*, is shorter than its propagation through *Delay2* for avoiding an erroneous rising edge on *Clk2*. The third timing assumption to take into account, using *Delay3*, is when the counter has been reset (*isReset* is 1). Then we need ensuring to switch the state of the register *R2* to the "*value 0"* and propagate it to the input of the *Xnor* gate, before the signal *isReset* arrives to it, for avoiding a wrong rising edge on *Clk1*. An example of the time diagram describing the operation of this module is depicted in Figure 44.

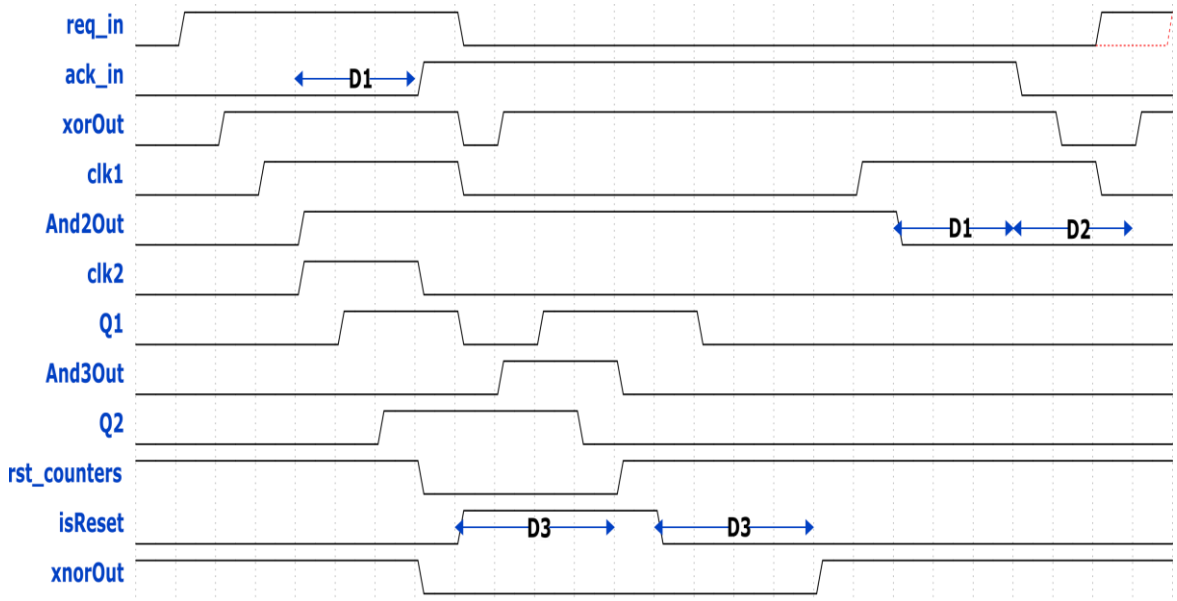


Figure 44: Timing diagram of the TDC reset circuit.

5.2.2.2 TDC Synchronization Module for Metastability Avoidance

It is well known that when using an internal RO and an external asynchronous input signal, some metastability issues can happen in a register, when the rising-edge of the RO clock signal coincides with the external signal arrival. In the industry, a well know method to confront this problem is to use two or more D flip-flops in series, for having a latency greater than the metastability settling time.

In the case of this TDC implementation, depicted in Figure 45, two C-elements have been used in combination with an inverter, for synchronizing the rising-edge of the input request signal (*reqInTDC*) with the rising-edge of the RO clock signal.

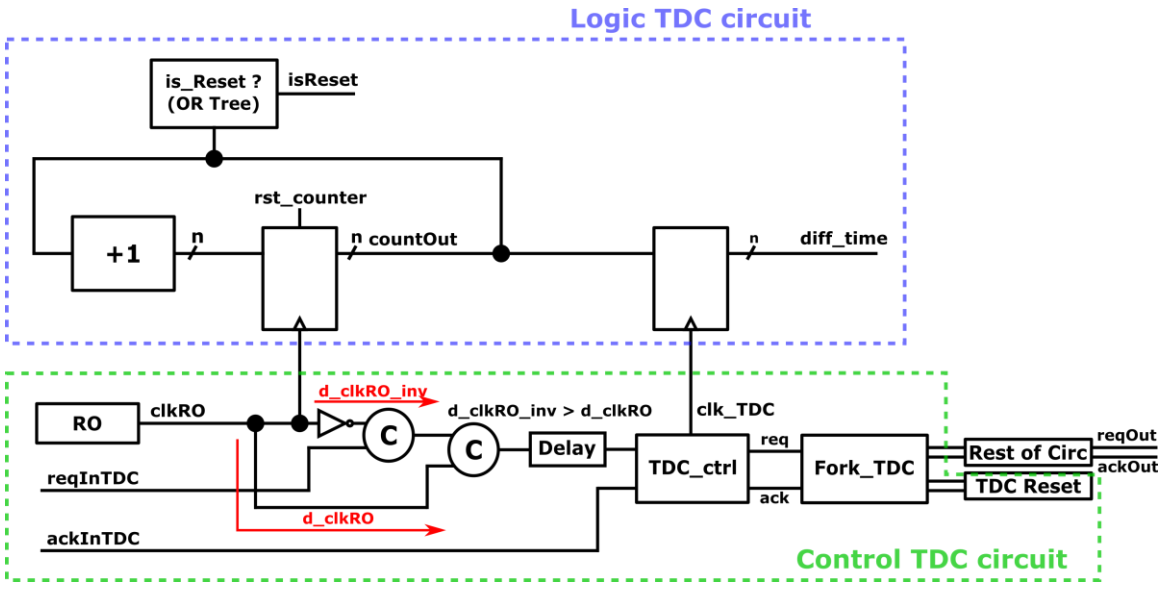


Figure 45: Diagram of the counter TDC circuit, integrating the synchronization block.

Four possible cases can occur when combining the RO clock with the arrival of the asynchronous input request *reqInTDC*, which are shown and analyzed in Figure 46 to Figure 49, indicating boundary situations when the metastability resolves to *gnd* or to *vdd*.

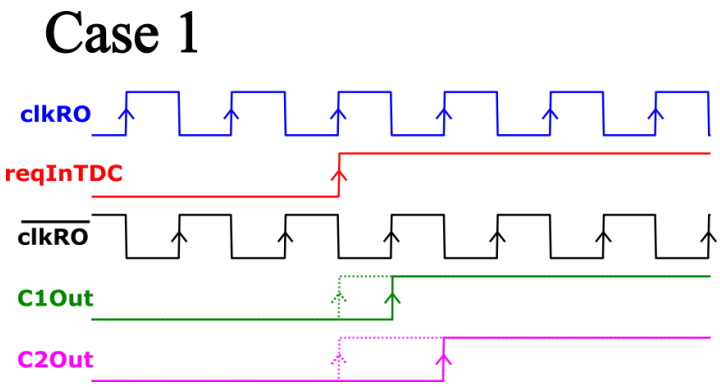


Figure 46: Datagram indicating the input TDC request arriving in synchronization with the rising edge of clkRO.

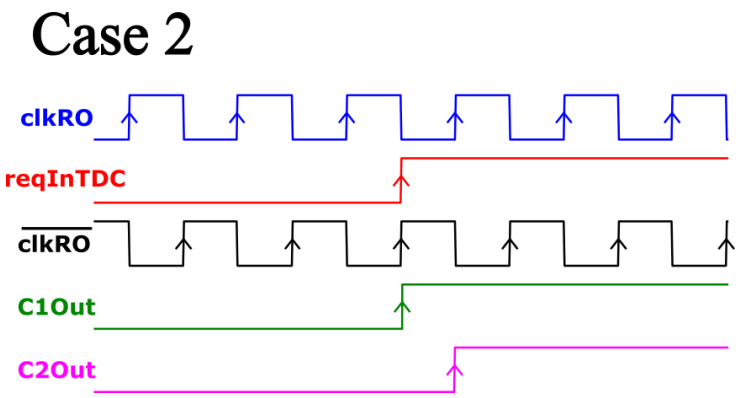


Figure 47: Datagram indicating the input TDC request arriving in synchronization with the falling edge of clkRO.

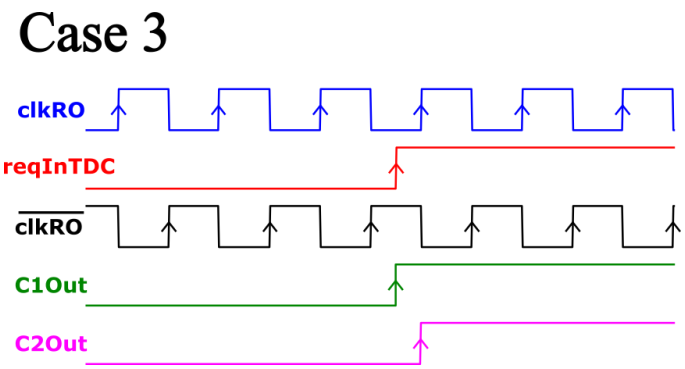


Figure 48: Datagram indicating the input TDC request arriving before the rising edge of clkRO.

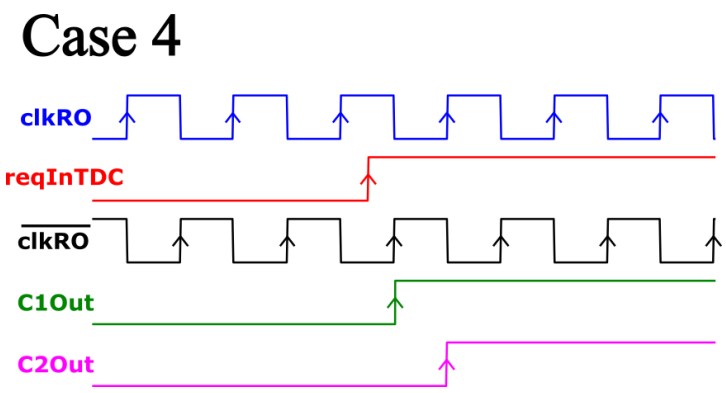


Figure 49: Datagram indicating the input TDC request arriving after the rising edge of clkRO.

The C-elements integrate an internal feedback logic for maintaining the current state of the cell, when the two inputs are unequal. Such a characteristic, combined with the use of two C-elements in series, ensures the stabilization of the possible metastability to *gnd* or *vdd* before arriving to the *TDC_Crl* element.

In all the cases, the rising-edge output of the second C-element, *C2Out*, is synchronized with the rising-edge of the RO clock, *clkRo*.

5.2.2.3 Safe Current Time Measurement Registration

The synchronization between the rising-edge of the input request at the output of the second C-element, with the rising-edge of the RO clock, lets the system ensuring a correct time measurement registration when the resulting request arrives to the *TDC_Ctrl* module. The delay module between the C-element and this module allows waiting for the correct instant to activate the registers at the output of the TDC for delivering the *diff_time* signal.

5.2.3 RTL Simulation results

The same testbench described in section 4.2.4.3 has been applied to simulate the performance of the demodulation circuit described in this chapter (3rd proposal).

The results are depicted in the BER v/s SNR graph below.

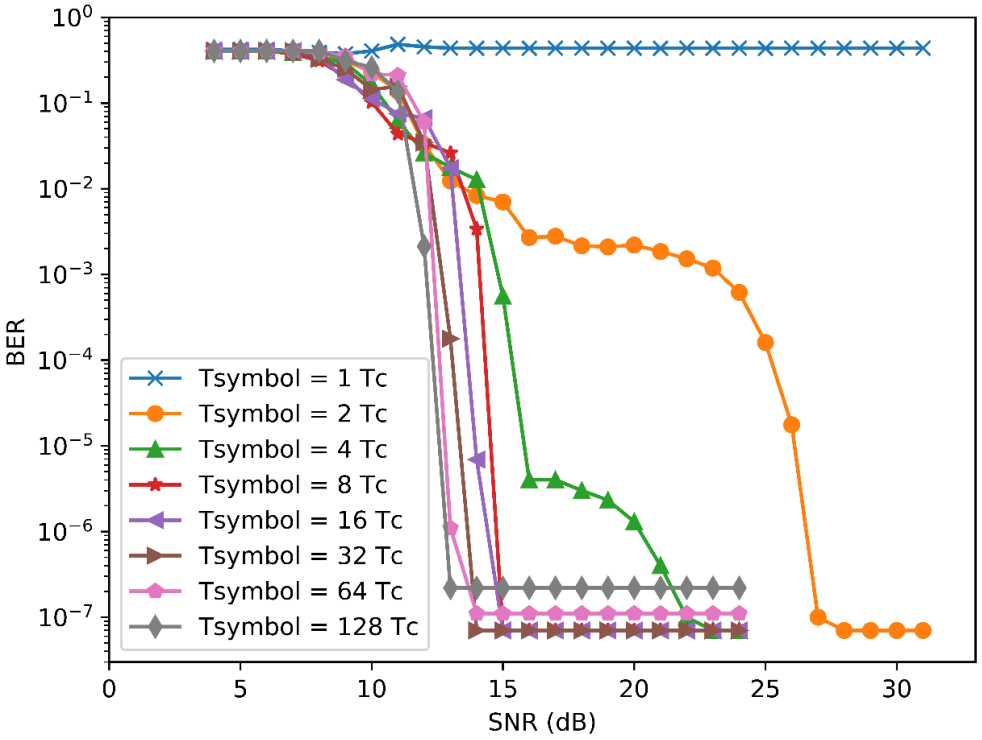


Figure 50: Results of the 3rd proposed solution, using the same SystemVerilog testbench.

From Figure 50, we can state that for a T_{symbol} equals to $2T_C$, the system has a response very similar to the results obtained in the 1st proposal (Figure 33). In that case, the majority calculation does not help to confront insertion or deletion errors caused by amplitude noise. So, noise sensitivity is almost maintained compared to the 1st proposal. It is noticeable that noise sensitivity is drastically reduced, thanks to the majority operation, when T_{symbol} increases.

5.3 Comparison with Solutions of the State-of-the-Art

5.3.1 Introduction

As explained earlier in this document, the focus of this work is to find a digital solution for ASK demodulation, compliant with the NFC standard. One of the main goals is to shift the frontier between analog and digital modules in order to reduce the analog part to the minimum. For that purpose, only a few necessary analog blocks should be kept, in order to control the received power from the electromagnetic fields and to stabilize the modulated signal inputted to the system. For instance, these

modules are typically the energy harvester, the impedance adaptation and the *AGC (Automatic Gain Control),* that are not discussed in this work. Therefore, this thesis is only focused on finding a digital solution addressing the ASK demodulation of a stable RF input modulated signal. Other implicit contribution which could be developed in future works, is that some of these analog blocks can be optimized or even removed when adapted to the new digital conditions related to the solution proposed in this thesis.

As described and analyzed in chapter 3.1.3, three full digital ASK demodulation solutions have been found in literature, one using a frequency divider, another employing pulse shrinking elements and the last one using a digital IQ demodulation technique.

In the following sections, we present a performance comparison of these solutions. For obtaining a fair comparison, the same testbench, used in 4.2.4.3, has been applied for each of them. All these solutions assume a stable supply voltage and a stable RF input signal amplitude, and only analyze their demodulation performances, represented by graphs showing the bit error rate (BER) against the signal to noise ratio (SNR).

5.3.2 Frequency Divider Technique

Figure 51 presents simulation results of the ASK demodulation performance of the solution using a frequency divider. This graph denotes an important sensitivity to amplitude noise. Indeed, the system abruptly improves its BER v/s SNR performance around 26 dBs, which is the limit where the amplitude noise stops producing faulty level crossings on V_{ref} for a MI of 10%. For the design of the frequency divider, a typical solution has been implemented using several D flip-flops in series, connecting the inverted output of one flip-flop to its own input and also to the clock pin of the next flip-flop. The number of flip-flops integrated in the frequency divider depends on the number of symbol periods. If $T_{Symbol} = NT_C$ then $n^{\underline{o}}\, of\ FF = \log_2(N)$. For example, if $T_{Symbol} = 4T_C$, then 2 flip-flops are necessary. With this method, the system instantly loses its synchronization with the communication sequence and, hence, its frequency division, when only one erroneous sample is added or lost. This is respectively equivalent to an insertion error or a deletion error.

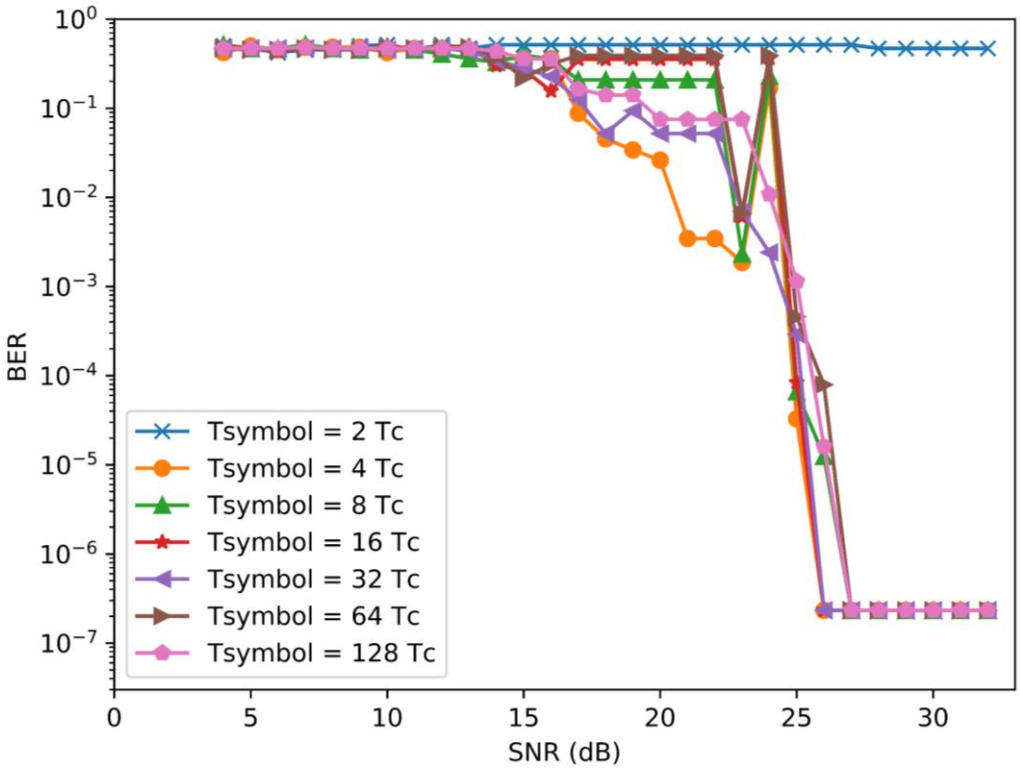


Figure 51: Results of the frequency divider demodulation technique, using the same SystemVerilog testbench.

5.3.3 Pulse Shrinking Method

The result of the pulse shrinking demodulation method is depicted in Figure 52. In this case, the system is extremely disturbed by pulse width variations due to large amplitude noise. Some high amplitude pulses unexpectedly disappear even for short pulse width variations (low noise), making the system loses its demodulation accuracy. For that reason, the improvement of *BER* only occurs after 30 dB, when the noise has almost disappeared. Thus, the choice of the output stage is particularly sensitive and even to worsen this negative effect, as explained previously in this document. Also, the *PVT* variations affect the matching between the pulse shrinking path and the clock path. It is important to highlight that the *"pulse shrinking element" (PSE)* has been simplified as a 2-input *"AND"* gate receiving a pulse and the delayed pulse in order to obtain the shrunk pulse. This method is easily integrated in a digital design since it avoids the use of a custom *PSE* module. The schematic of the implemented *PSE* is presented in the Figure 53.

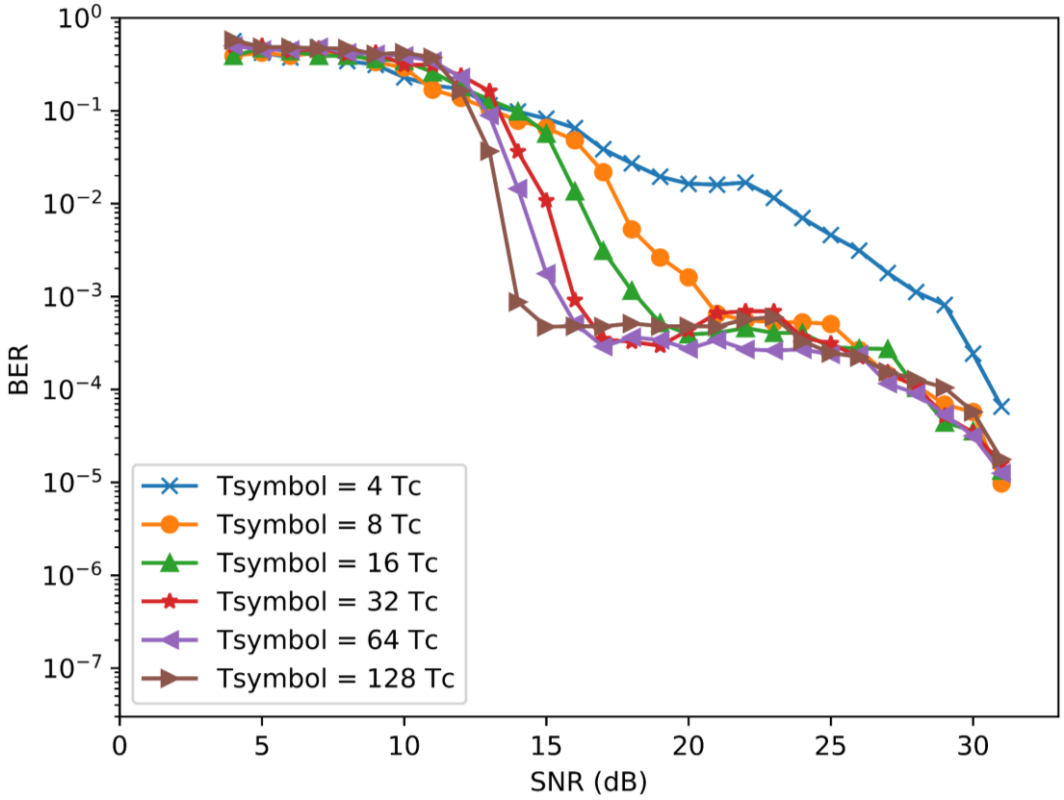


Figure 52: Results of the pulse shrinking demodulation solution, using the same SystemVerilog testbench.

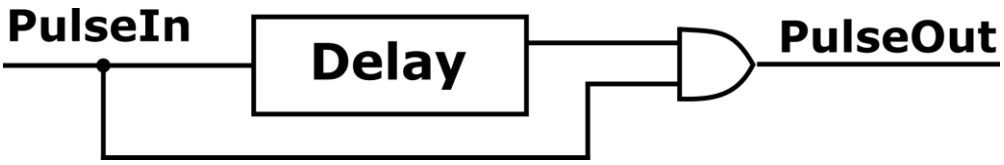


Figure 53: Schematic of the implemented pulse shrinking element.

5.3.4 Digital IQ Demodulation

The same SystemVerilog testbench has been applied to the digital IQ demodulator. The high-speed ADC has been implemented in SystemVerilog, by simply taking the value of the RF input signal on each clock rising edge, at four times the carrier frequency ($4 \cdot f_C$), and assigning a digital representation of it as output. Of course, that part of the design is not synthesizable.

Then, on the digital domain, the *in-phase* and *in-quadrature* signals has been generated using four fixed digital values that are cyclically outputted in sequence. The mixers have been replaced by digital multipliers. Then the low-pass filters have been

digitally implemented using coefficients arbitrarily obtained using the "*Filter Designer*" tool in Matlab.

The performance simulation results are shown in Figure 54. It can be noticed that the BER performance of the IQ demodulator drastically increases near 23-24 dB. This result is a slightly better than the results of the 1st proposal of Figure 33, where the BER decreases near 25-26 db. It represents a gain of 2 to 3 dB in some cases, which is mainly due to the filtering operation, which deals with some of the faulty level crossings occurring when the amplitude noise increases.

Notice that the number of coefficients of the digital FIR filter also affects the maximal bitrate of operation of this type of demodulation. If the number of coefficients increases, at high bitrate, the system will filter the symbol change, and hence will not recognize the symbols. However, if the number of its coefficients grow, the demodulator will be more robust to insertion or deletion errors, produced by the amplitude noise.

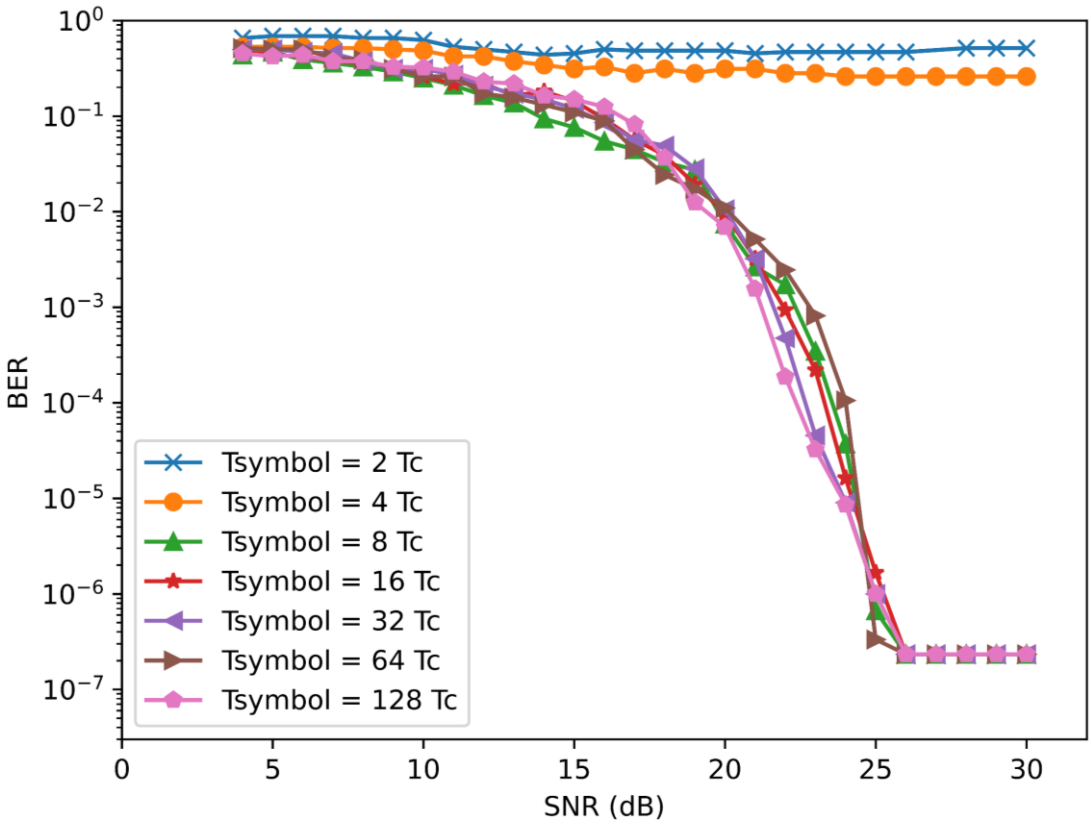


Figure 54: Results of the digital IQ demodulation, using the same SystemVerilog testbench.

For comparison purpose, a logic synthesis has been made for this IQ demodulator in order to estimate its area and its power consumption. The post-synthesis results for the IQ demodulator will be compared to the post-synthesis results of our proposal. The synthesized area usage for this IQ demodulation system is 15429 µm2. Also, its

estimated power consumption is $668.6\ \mu W$. For these results, it is important to highlight that we are not considering the addition of the area and power consumption of the external oscillator (PLL/DLL) and the high-speed ADC, which are not at all negligible. For comparison considerations, it is also important to note that this solution is operating at four times the carrier frequency ($4 \cdot f_C$), while the proposed technique is limited to only one event every two carrier periods ($2 \cdot T_C$). This normally adds power consumption when comparing two equivalent circuits.

5.4 Conclusion

This chapter describes two demodulation solutions, which combine an EB-ADC and a digital demodulation algorithm. For each solution, the EB-ADC is composed firstly by the structure of the first proposal as their level crossing detector, and then a TDC in order to measure the time between two consecutive events. This time information is later used by the digital demodulation algorithm for improving its performance compared to the first proposal. In both cases, the modules performing the demodulation algorithm are described in detail. Basically, they estimate the equivalence of the time measurement in terms of the number of symbol periods. With that information, based on different strategies, they infer the end of the symbol period and the symbol value.

In the second proposal, the employed method does not need any synchronization with the communication sequence. However, it has some difficulties for detecting the right instant when the symbol period ends. In some cases, it can even produce extra output symbols and delay the symbol switching moment and hence obtaining a poor demodulation performance.

Contrarily, the demodulation algorithm of the third proposal requires a strict synchronization with the communication sequence. It has the ability to precisely identify the symbol switching instants. Thanks to this, after each elapsed symbol period, a majority calculation improves the confidence in the symbol value identification. Therefore, the proposed system performance is enhanced when a longer symbol period is used.

Assuming that a certain level of amplitude noise generates a fixed rate of modulation errors, represented by insertion or deletion errors in the modulated signal. Then, the majority calculation applied to a long symbol period makes the system more robust against amplitude noise. The case when $T_{symbol} = 2T_C$ does not generate any

performance improvement compared to the first proposal. In that border case, no modulation error in the system can be masked by the majority calculation since no better decision can be taken with only two carrier periods for each symbol.

Finally, the digital demodulation solutions of the literature have been simulated using the same RTL level testbench and then compared to the results of the third proposal of this thesis. In the three cases, the last proposal of this thesis shows a better BER v/s SNR performance, identifying a gain of around 9 to 11 dB when comparing the limit where the BER decreases abruptly in the BER v/s SNR graphs.

This novel solution is very promising and could be a digital alternative for increasing the robustness against amplitude noise, reducing the power consumption and relaxing the power constraints on the energy harvesting modules. In the following chapter, its physical implementation is described.

6 Physical Implementation: Testchip

6.1 Introduction

Chapters 4 and 5 present the three ASK demodulation designs proposed in this thesis, which describe the evolution of this work for arriving to the final proposal, the third design. The simulation results have been performed using the SystemVerilog testbench explained on chapter 4.2.4.3. In chapter 5, some digital ASK demodulators found in the literature have been tested with the same testbench environment.

After analyzing the BER v/s SNR results of these different approaches, we could state that the third proposal of this work presented a large improvement in its performance, compared to all the tested alternatives, **reducing about 10 dB** the SNR tipping point when the BER abruptly drops.

This new proposal also complies with all the main objectives established in this thesis:

- Design a novel digital ASK demodulation method.
- Minimize the generated events of the system (minimum LCSS).
- Use an EB-ADC.
- Propose an asynchronous solution taking benefit of the event-based ADC, and adding robustness to PVT variations.

Due to these previous reasons, the third proposal has been chosen for a physical implementation. Given the NFC carrier frequency of 13.56 MHz, no particular constraint was considered for choosing the technology node for this implementation. However, after an analysis with my PhD director, we decided to share and collaborate in the same testchip with some colleagues of the "Circuits, Devices and System Integration" (CDSI) group at TIMA Laboratory, participating at the *OCEAN12*[3] project, intended to develop the *"Opportunity to Carry European Autonomous driviNg further with FDSOI technology up to 12nm node"*. Their project was already related to the technology node *"FDSOI 28nm STMicroelectronics"*, so it was the chosen node for our physical implementation.

[3] OCEAN12 project - https://cordis.europa.eu/project/id/783127/fr

For a correct physical implementation, the third proposal presented in Figure 41 has received some modifications to be adapted to a real testing environment, and for facing the possible issues we could encounter in the test laboratory. This system was designed during the year 2020, sent to fabrication in December 2020 and finally received in August 2021. Figure 55 shows a simplified diagram of the manufactured ASK demodulator. All the implemented modules are explained in detail in the following chapters of this section:

- Comparators
- Comparison Stabilizer
- ASK Demodulation Logic
- ASK Demodulation Control
- SPI Slave and Bank of Registers

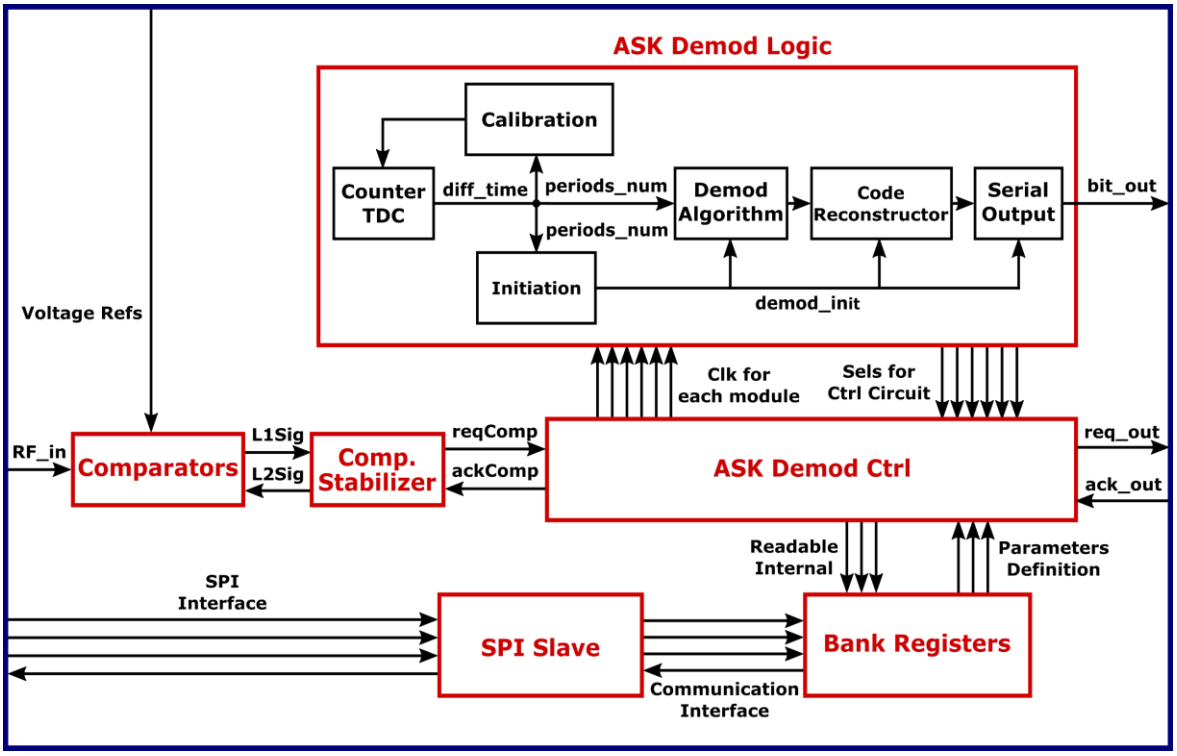


Figure 55: Diagram of the fabricated circuit, in the node FDSOI 28nm STMicroelectronics.

6.2 Physical implementation of the comparators

This thesis presents an ASK demodulation technique using an EB-ADC with a 2-level sampling scheme for demodulating the raw ASK RF signal. This EB-ADC is

composed by a *level crossing module (LCM)* detecting when the ASK signal crosses each level and a *Time-to-Digital converter (TDC)* measuring the time between two consecutive events. The Figure 56 describes the *LCM* used in this design, which is composed by two comparators detecting when the signal crosses the levels. Then L1Sig and L2Sig signals are inputted to the comparison stabilizer module for obtaining a stable request signal that is employed as the input request of the control circuit of our solution.

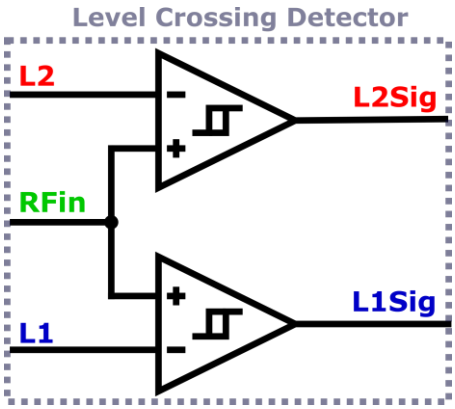


Figure 56: Module Performing the Level Crossing Detection.

These comparators must be very efficient to compare the input signal to the two reference voltages, which are close to Vdd and to Gnd (this is not an easy task). A state-of-the-art was done for finding an idoneous comparator complying with our operation needs.

The most interesting comparator found, called *"Strong-Arm Comparator" (SAC)* [28], has been chosen for the following advantages. It uses the rising-edge of a clock signal to establish each comparison instant. They only have a dynamic consumption during the clock rising-edge. Their static consumption is related to leakage and behave like standard logic gates. According to the literature, the *SAC* are known for having a fast comparison response and a reduced power consumption. These characteristics permit to set fixed and stable comparison instants that better match with the digital simulation models created for these comparators. The SAC also filter most of the glitches produced by the amplitude noise.

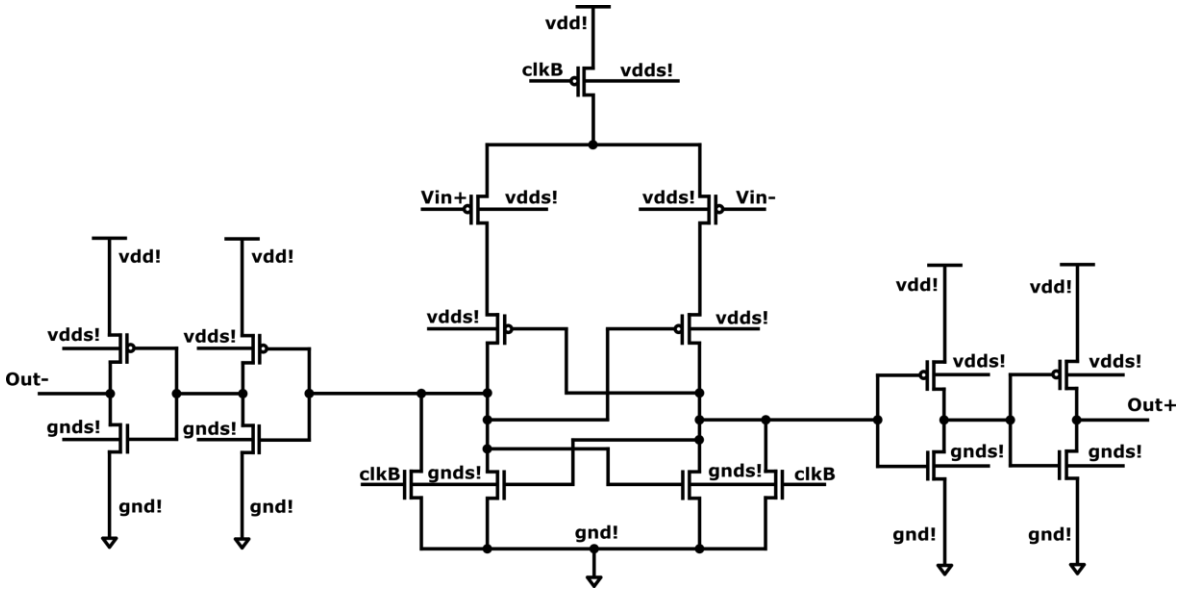


Figure 57: Schematic of the Strong-Arm comparator adapted for signals near Gnd (SAC-Gnd).

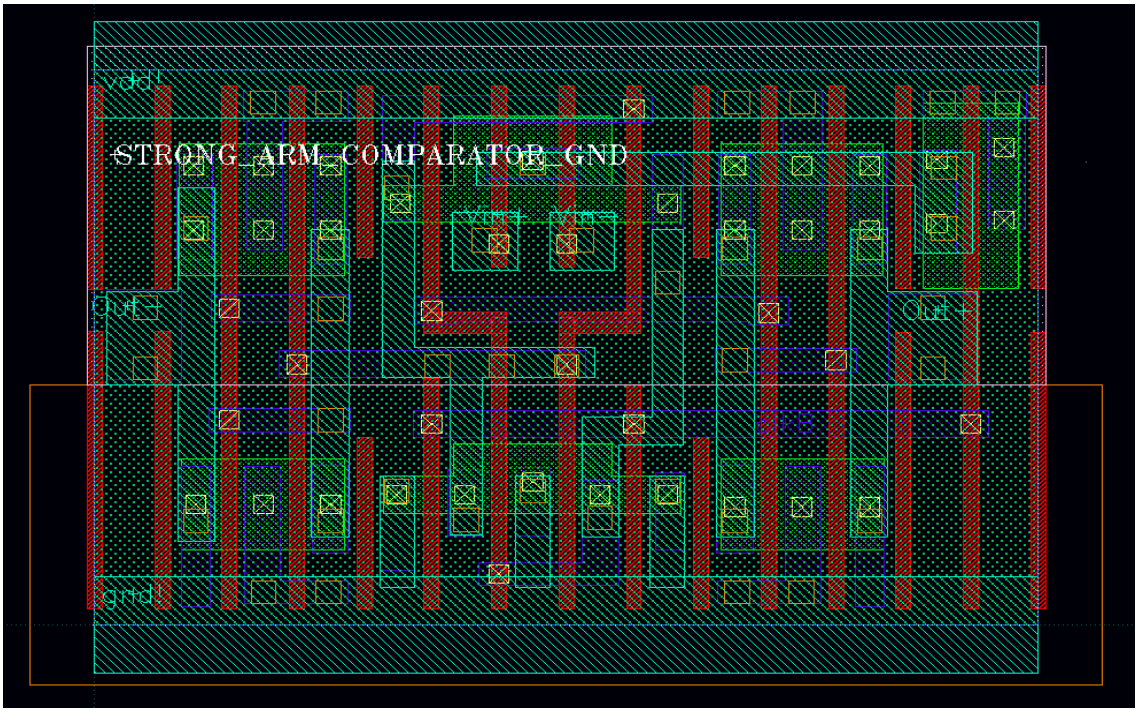


Figure 58: Layout of the implemented SAC-Gnd.

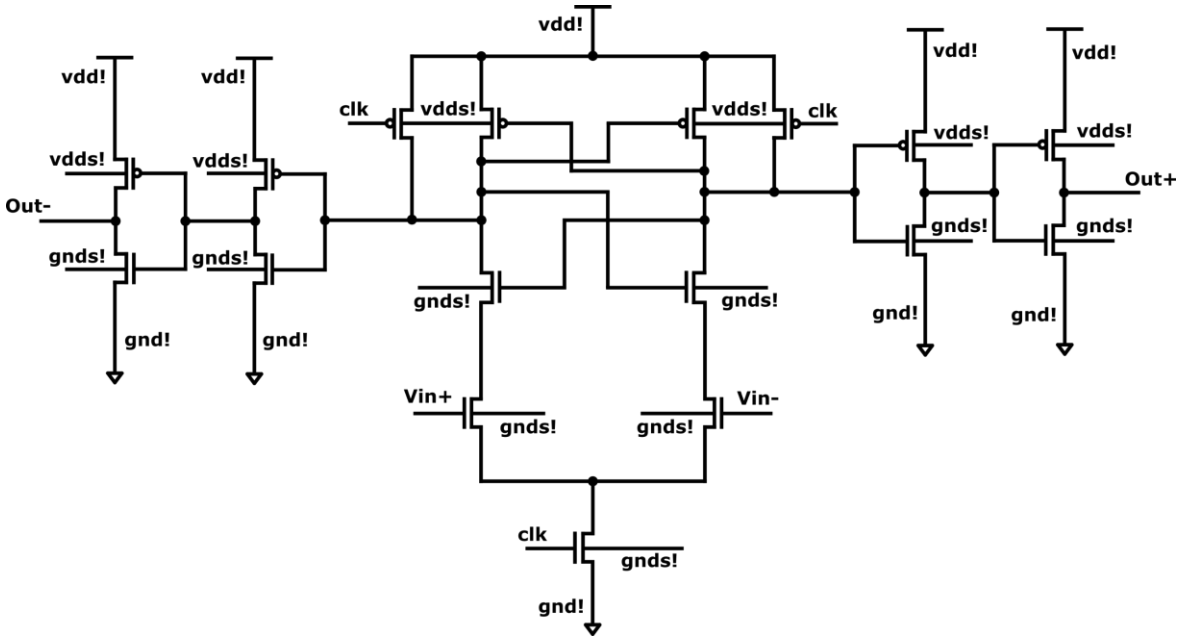


Figure 59: Schematic of the Strong-Arm comparator adapted for signals near Vdd (SAC-Vdd).

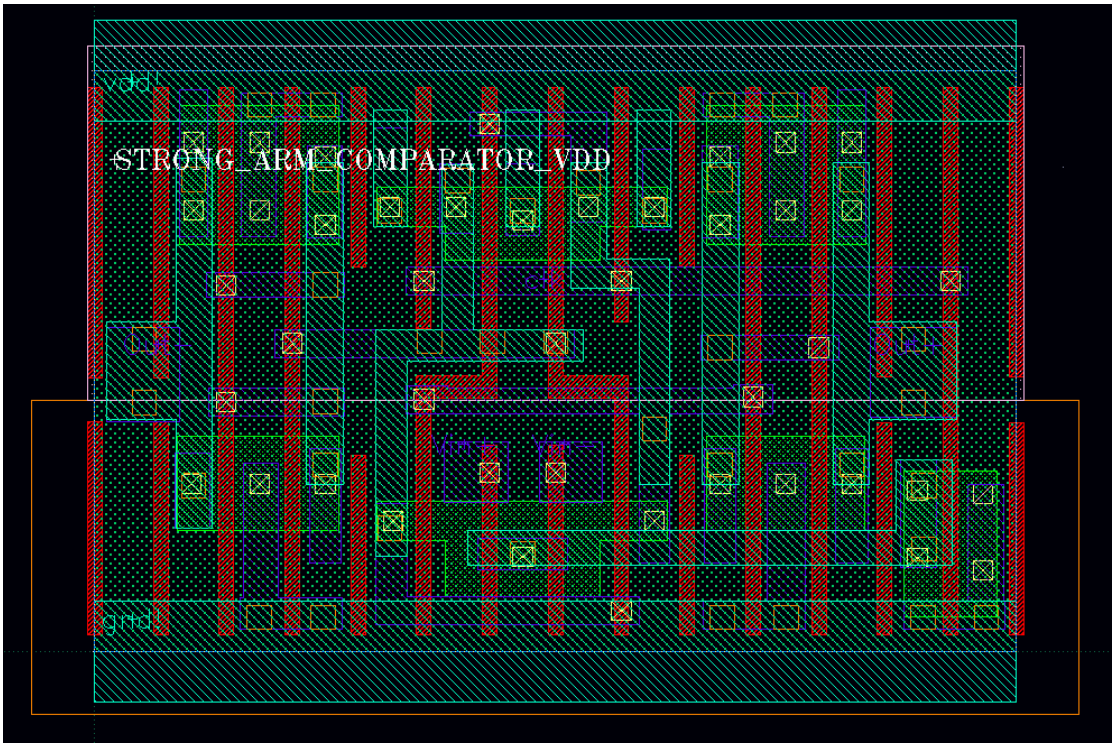


Figure 60: Layout of the implemented SAC-Vdd.

As observed in [28], in Figure 57 and in Figure 59, a Strong-Arm comparator contains only few transistors, resulting in a small, fast, and easily integrable design. In our application, these SAC have been designed as standard cells of the "*FDSOI 28nm STMicroelectronics technology"* and integrated in the commercial digital design flow.

In a SAC, each comparison starts on the rising edge of a clock. In the circuit implementation, a programmable delay is employed to generate a settable internal clock period. A high-level description of the complete *SAC* design is depicted in Figure 61.

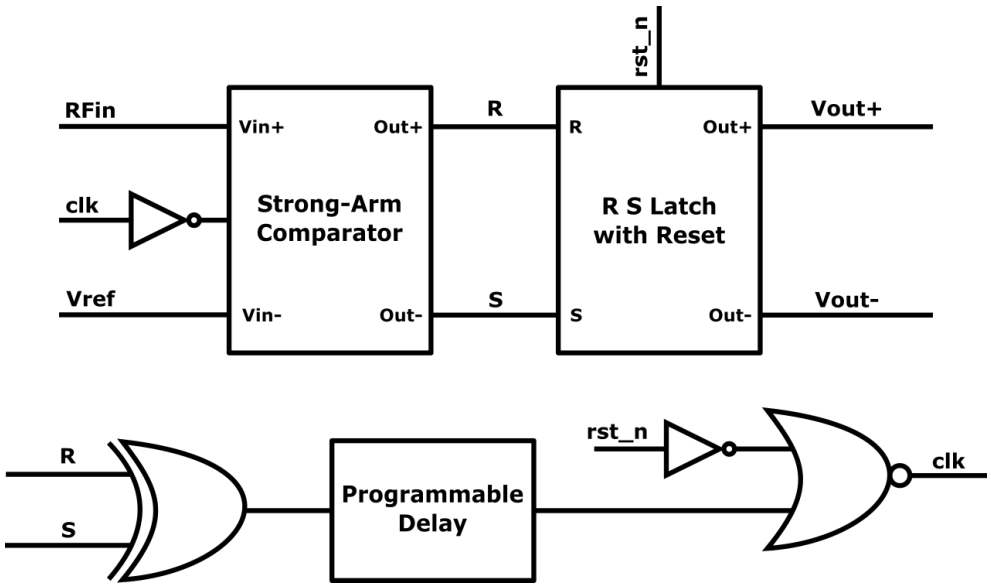


Figure 61: Self-generation of the comparation clock for a SAC-Gnd - High-level diagram.

This design employs the R and S signals to detect when each comparison cycle ends. These signals are used to generate a new clock rising-edge with a specific period, set with the programmable delay module and only after ensuring that the previous comparison correctly ended. Two types of *SAC* have been designed, one for comparisons near *Vdd (SAC-Vdd)*, where the "current source" is connected to *Gnd*, as shown in Figure 59, and another for comparisons near *Gnd*, with the "current source" connected to *Vdd*, as depicted in Figure 57. As noticed, these two designs use a symmetrical topology, interchanging the *pmos* and *nmos* transistors for each type of SAC. For ensuring the correct behavior and good performance of the complete comparison module, its whole layout has been manually designed with the tool *"Cadence Virtuoso Layout XL"* and electrically analyzed with the simulator *"Spectre"* from the *"Cadence Virtuoso ADE"* framework.

6.3 Comparison Stabilizer

The "Comparison Stabilizer" module is used for filtering the glitches caused by the amplitude noise to the comparator detecting the *"L2"* level crossing. Additionally, its

second function is to manage the handshake communication with the “ASK demodulation control” circuit. It is composed by two C-elements, one for each function.

The first muller gate is connected to each comparator output, one comparing RFin to *“L1”* and the other comparing RFin to *“L2”*, replicating the structure used for the first proposal. These reference levels are distant, one is near *“Gnd”* and the other close to *“Vdd”*. Therefore, the possible glitches caused by the amplitude noise to each of these comparison outputs do not occur simultaneously. In addition, the number of glitches produced by the comparators can be controlled by their programmable delays, reducing them if the comparison period is large. These facts allow the muller gate to obtain a stabilized output indicating the “L2” crossing detection since its output only switches if both inputs signal have the same digital level. A stable signal representing an event is absolutely necessary if it is used as an input request for an asynchronous circuit.

The second C-element is connected as a WCHB register controller, in order to be able to receive the acknowledgement from the system and transmit the input request only if the system is open to receive a new event. The description of the connection of both muller gate is depicted in Figure 62.

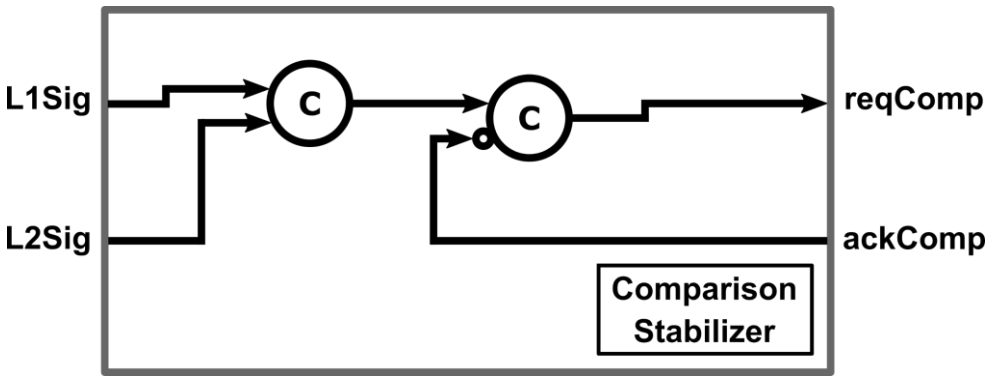


Figure 62: Diagram of the comparison stabilizer module.

6.4 ASK Demodulation Logic

As depicted in Figure 55, the *“ASK Demodulation Logic” (ASKDL)* module is one of the main blocks of the demodulation system. It contains all the pipelines of the circuit, integrating each combinational and sequential parts of the circuit. In this module, each pipeline receives one separate clock signal from the environment, in order to activate

each pipeline independently, as requested by the control circuit, described in chapter 6.5 . The *ASKDL* module is also responsible for generating the *"selection signals"* for indicating to the control circuit in which phase of operation the circuit should be and which individual parts of the circuit must be activated. Figure 63 depicts the different operation phases we find in the proposed demodulation process. Three main phases are identified: calibration, initiation and communication.

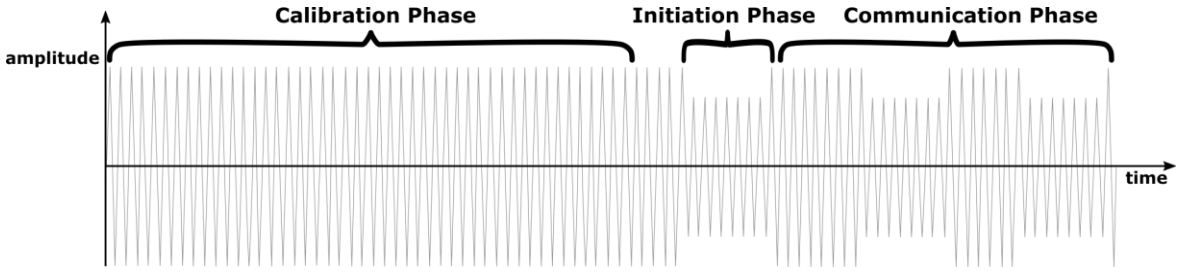


Figure 63: Operation phases of the ASK demodulator.

6.4.1 Counter TDC

As specified in chapter 5.2.2, the counter TDC is responsible for measuring the time between two consecutive level crossing events. The logical part contains two main blocks. The first part, on the left of Figure 64, is a counter dependent of the *"Ring-Oscillator Clock" (clkRO)* that counts the number of times the rising-edge of *clkRO* occurs before being reset just after a new event arrives. The second part, on the right of this figure saves the measurement value when a new event arises.

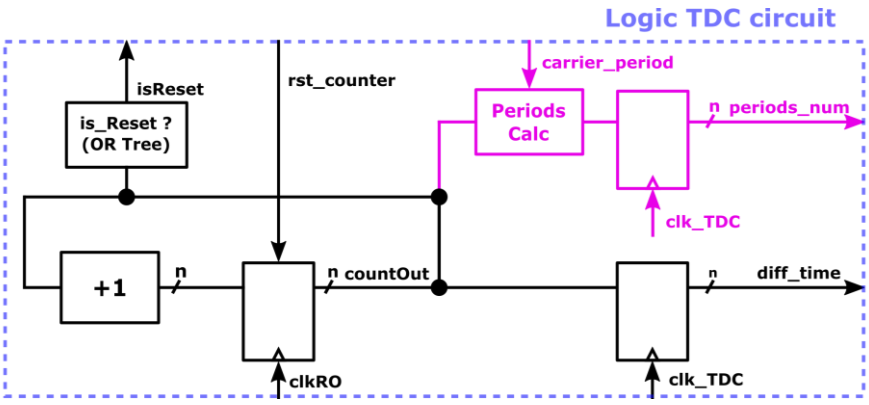


Figure 64: The TDC logic circuit implemented in the fabricated testchip.

For the physical implementation, an additional counter, highlighted in pink, has been integrated in the TDC module, called *"Period Calc"*, enabling an "on-the-fly" calculation of the number of carrier periods (T_C) elapsed between two consecutive events designated as *"periods_num"*. This technique allows eliminating the former *"Period Calc"* module that normally used a divider for this calculation with the *"diff_time"* and *"carrier_period"* values. This choice will reduce the area during the synthesis and the layout of the resulting circuit.

6.4.2 Calibration

In a fabricated circuit, some variations can affect the correct operation behavior. The three most important causes are the following. The first is related to all the variabilities of the fabrication procedure, normally called *"the process"*. The second is related to the operating temperature conditions of the circuit. Finally, the third one is associated to the power supply voltage of operation. These variations are called "*PVT*" as previously specified in this work.

The proposed circuit uses a *Ring Oscillator (RO)* inside the implemented TDC and the *RO* frequency can be altered by the *PVT* variations. In this ASK demodulation application, the most important step is to obtain an accurate number of T_C measurement after each event arrival. For that reason, a calibration module has been added in order to calculate the average of the TDC measurement for one T_C, after 32768 (2^{15}) iterations. The calculated mean value represents an approximation of the TDC measurement for one T_C at the operating conditions of the circuit when tested or used.

During the calibration phase, the reader (or transmitter) must maintain a high amplitude communication for at least 32768 carrier periods, in order to let the demodulation circuit to accomplish this phase. Once the calibration phase is finished, the tag (or receiver) must indicate to the transmitter that it is ready to start the communication.

6.4.3 Initiation

Once the calibration phase has ended, the demodulator is ready to start the communication. However, for enabling a communication, it must accurately identify the

beginning of the first transmitted symbol. This is called the initiation phase. The transmitter must send one low amplitude symbol followed by one high amplitude symbol (first communication bit), in order to synchronize the end of the first low amplitude symbol as the beginning of the communication. The first communication bit must always represent a high amplitude, in order to permit the system to identify the start of the transmission.

6.4.4 Demodulation Algorithm

The *“Demod Algorithm”* module is equivalent to the *"Amplitude Change Detector"* (*ACD*) module that has already been described in the chapter 5.2.1. In this physical implementation, no modification has been done compared to the former module.

6.4.5 Code Reconstruction

The *“Code Reconstruction”* module has the same behavior as described in chapter 5.1.7. However, in this case, it can receive more than one demodulated bit on each activation. For this purpose, the module has been adapted to save all these bits always shifting the internal register until the 32 bit is reached. If a larger number is reached in one iteration, the first 32 bits are outputted towards the serial output module, and the remaining bits are kept safe in the internal 32-bit register for receiving new bits in the next iteration.

6.4.6 Serial Output

The *“Serial Output”* module takes each 32-bit packet, grouped by the *CR* module, and outputs the packet bit per bit using a register associated to *“bit_out”*, which is one of the outputs of the system. The activation of the register is associated to a basic asynchronous protocol with a request *(req_out)* and an acknowledgement *(ack_out)* connected to the circuit output environment, as shown in Figure 55. The whole control circuit, *“ASK Demod Ctrl”*, is described in the section 6.5 .

6.5 ASK Demodulation Control

6.5.1 Introduction

The implemented demodulation system is based on a digital asynchronous bundled-data circuit, which operates with a 4-phase protocol. One of the most important parts of this solution is the "Control Circuit" that has been called *"ASK Demod Ctrl"*. This module generates the clock rising edges for each pipeline stage of the *"ASK Demod Logic"* circuit, for each new sampled event arrival. It is important to emphasize that, for all the pipeline stages, the control circuit must comply with local timing assumptions associated to each combinatorial logic. In other words, let's imagine two interdependent blocks, where outputs of the first block are inputs of the second. In that case, the control circuit ensures that the clock rising edge of the first block is delayed before reaching the second. This delay must be large enough to allow the completion of the combinational calculation of the second block before generating the rising clock edge for the second block. This corresponds to the *"setup time constraint"*, as depicted in the example of Figure 65.

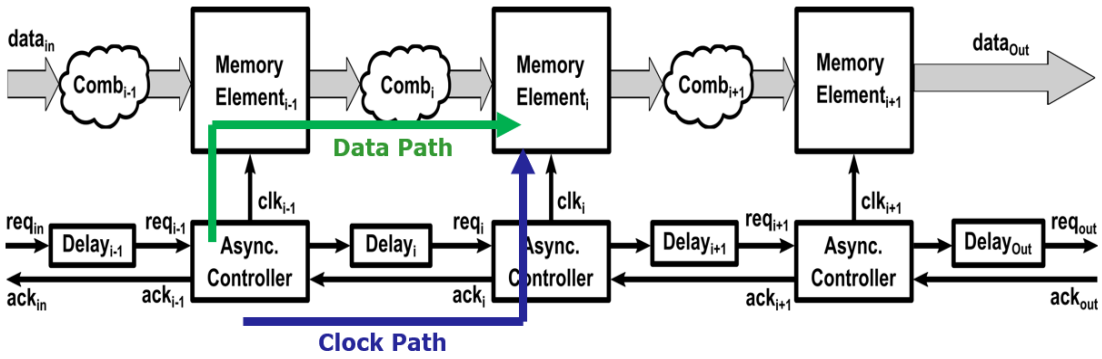


Figure 65: Example of the setup timing constraint verification.

Additionally, the "hold time constraint" must be checked. This corresponds to compare two delays. The first is the delay taken by the clock signal going from its associated controller to the pin clock of the registers, called the *"hold clock path delay"*. The second is the related acknowledgement going to the previous controllers and hence activating their connected registers for modifying the input data of the current pipeline, called the *"hold data path delay"*. For the hold timing verification, the clock

path delay must be shorter than the data path delay, in order to ensure that the clock activates the registers for saving the current data before a new input data arrives. A simplified example of this timing constraint is described in Figure 66.

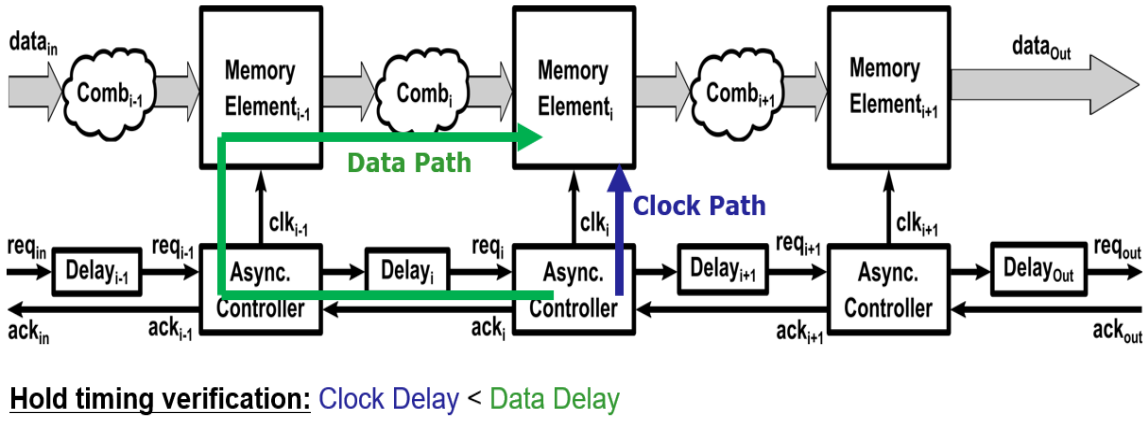


Figure 66: Example of the hold timing constraint verification.

In this implementation, the frequency aimed by our application is only 13.56 MHz, which is equivalent to 73.746 nanoseconds (ns). This information should be considered when designing the *"delay lines" (DL)* inside each *"register controller" (RC)*, which are the modules responsible of managing the activation of the pipeline stage registers. Specifically, for each pipeline, the delay of the *DL* of its *RC* must be larger than the delay of the critical path of its associated combinational stage and must be shorter than 73.746 ns, which is equivalent to the carrier period "T_C". In this analysis, the setup and hold delays associated to the registers of the technology node used in this implementation (T_{setup} and T_{hold}), must be also considered together with all the delays related to the gates used in the control circuit.

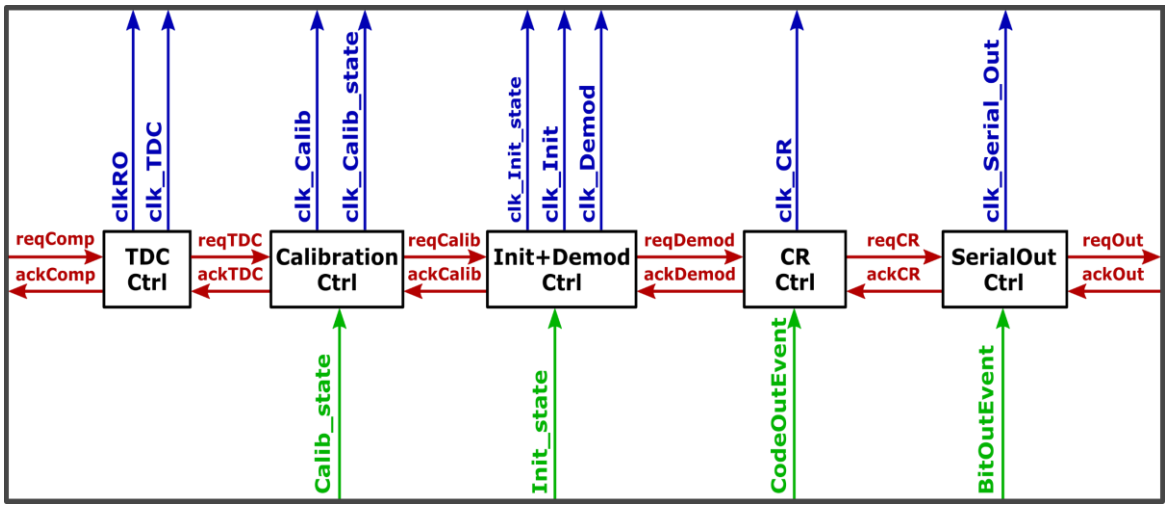


Figure 67: Diagram of the asynchronous control circuit, "ASK Demod Ctrl", of the fabricated testchip.

Figure 67 shows the diagram of the designed *"ASK Demod Ctrl"* block, that contains five different internal control blocks. Each block is reviewed in the following paragraphs of this section. In the diagrams of this chapter, the signals specified in blue are *clocks*, the one in green are *selections*, and the ones in red are the handshake protocol signals between the controllers and the environment. The legend employed in the diagrams is given in Figure 68. The basic control elements implemented in this design are reviewed in the next section.

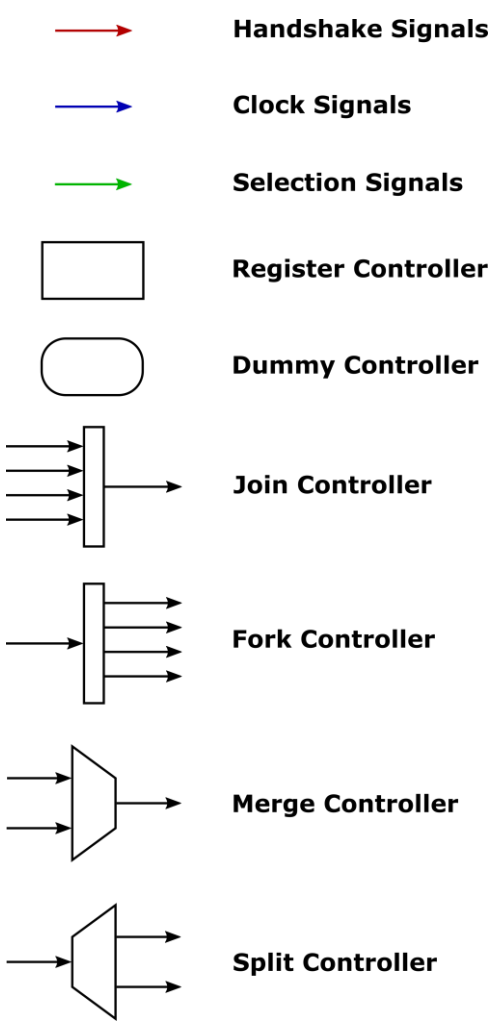


Figure 68: Legend of the elements used in the controller's diagram description.

6.5.2 Basic Elements of the Control Circuit

The design description of the basic elements used in the implemented control circuit are presented in the sequel.

6.5.2.1 C-element

The C-element (or Muller gate) used in the whole circuit has been implemented with standard cells of the technology employed in this testchip. Its schematic is shown in Figure 69. *A*, *B* and *rst_n* are inputs, *Z* is an output and *rst_value* is a parameter value defined directly in SystemVerilog for choosing the reset value of the C-element gate.

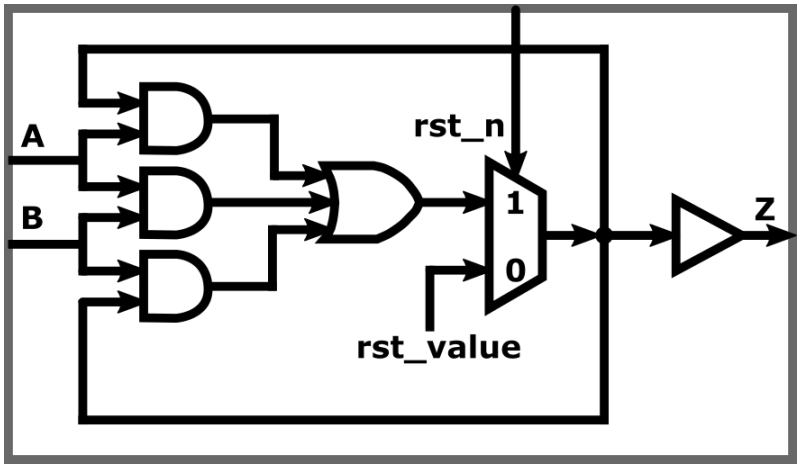


Figure 69: Diagram of the implemented C-element.

6.5.2.2 Register Controller integrating the delay line

The register controller has been designed using a fixed delay-line, a C-element, an inverter and three buffers, as depicted in Figure 70. The delay line has been implemented with several buffers in series, as described in Figure 71. The number of buffers is defined using a parameter in SystemVerilog and the *"set_don't_touch"* flag during the synthesis and place and route process in order to avoid a deletion by the tool. For all the *"register controllers"* used in the *"ASK_demod_Ctrl"*, the delay size has been taken wide (with a large margin) to ensure the timing assumptions related to the combinational part of each pipeline stage. As previously explained, the additional constraint to consider is that delays cannot exceed the carrier period "T_C", which is an easy task in this fast technology node.

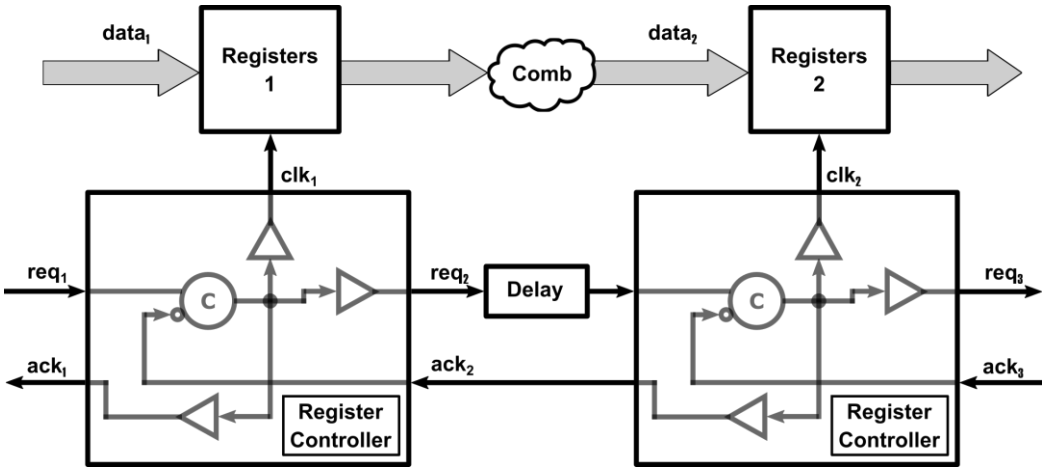


Figure 70: Diagram of the used "Register Controller" and its connection with a consecutive stage.

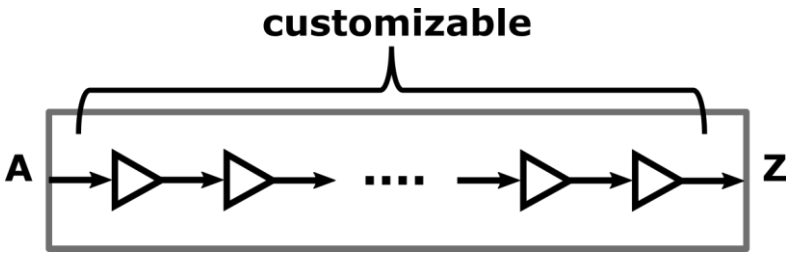


Figure 71: Diagram of the implemented "Delay Line".

6.5.2.3 Fork Controller

The implementation of the fork controllers is described in Figure 72. If the fork controller connects more than two other controller blocks, a tree structure of C-element must be used for the acknowledgment path.

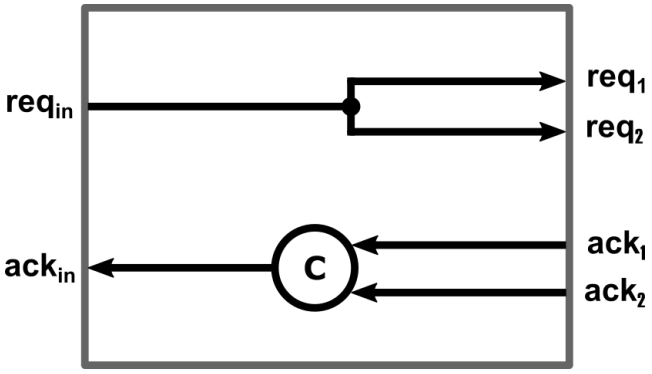


Figure 72: Diagram of the implemented "Fork Controller".

6.5.2.4 Join Controller

The join controller used in the designed circuit is depicted in Figure 73. If the join controller concentrates more than two other controller blocks, a tree structure of C-element must be used in the request path.

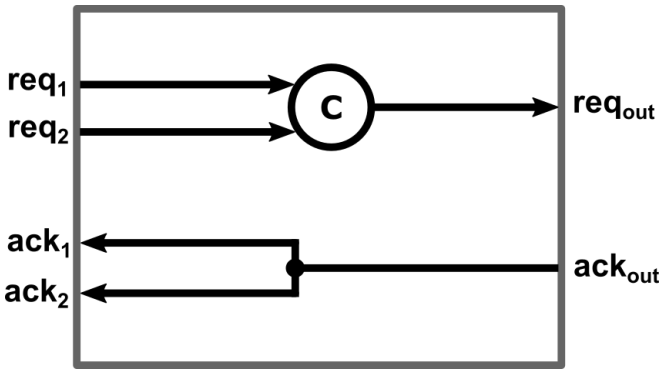


Figure 73: Diagram of the implemented "Join Controller".

6.5.2.5 Split Controller

The split controller design employed in the control circuit is presented in Figure 74.

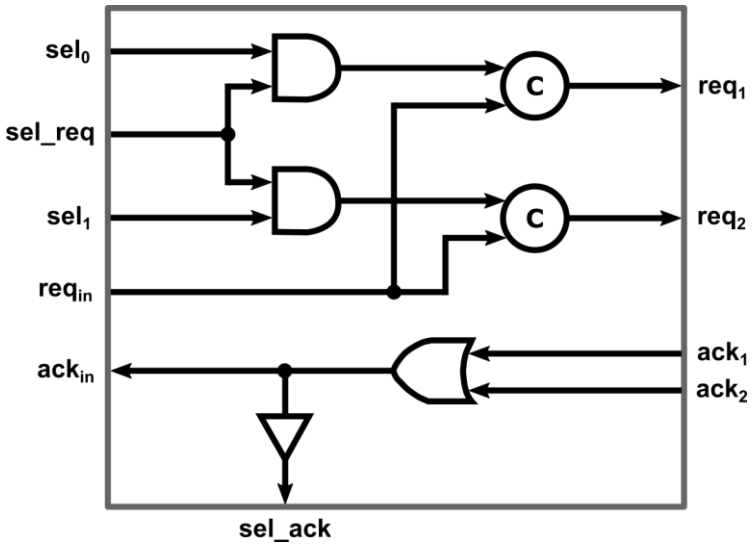


Figure 74: Diagram of the implemented "Split Controller".

6.5.2.6 Merge Controller

The merge controller design used in this implementation is depicted in Figure 75.

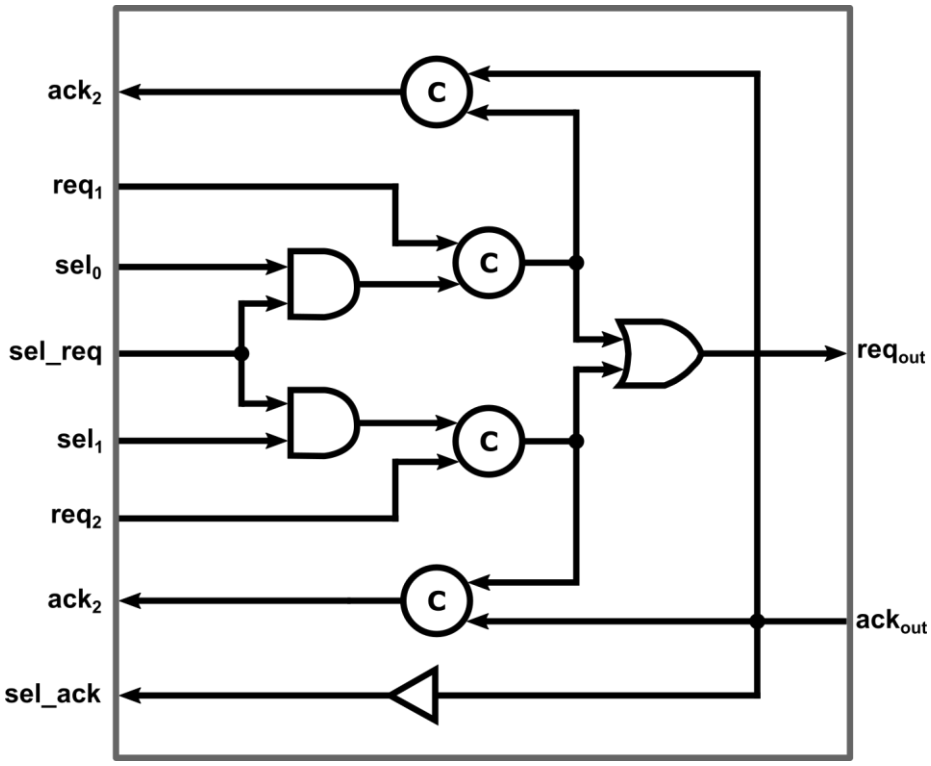


Figure 75: Diagram of the implemented "Merge Controller".

6.5.2.7 *Dummy Controller*

The dummy controllers used in this circuit is shown in Figure 77.

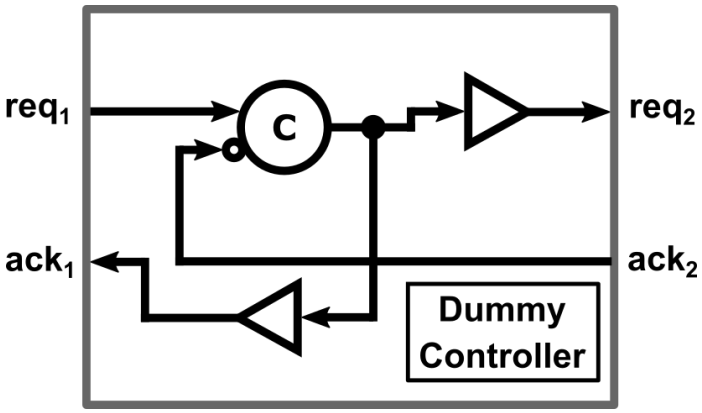


Figure 76: Diagram of the implemented "Dummy Controller".

6.5.3 TDC Control Circuit

The *"TDC control circuit" (TDCcc)* has already been described in Chapter 5.2.2. Its structure and functionality are the same as the ones used for the RTL simulations. However, the diagram describing the TDC control circuit has been reshaped to match

the legend definitions of this section. Figure 77 presents the updated *"TDC Controller"* circuits, which contains two *C-elements*, one *Inverter*, a *RO*, a *Register Controller* and the *"TDC Reset"* Controller, already presented in chapter 5.2.2.1. The latter is a custom module that allows identifying a new measurement cycle, using the asynchronous 4-phase handshake protocol signals. Indeed, it generates the *"rst_counter"* signal, which is used to send a reset command to the TDC counters. In addition, this module receives the *"isReset"* signal that indicates when the counters are already reset, preparing it for a new measurement cycle. These signals interact with the "TDC Logic" circuit reviewed in chapter 6.4.1. Once the *"isReset"* signal is validated, the *"TDC Reset"* module releases the TDC counter and let the control circuit continuing its operations.

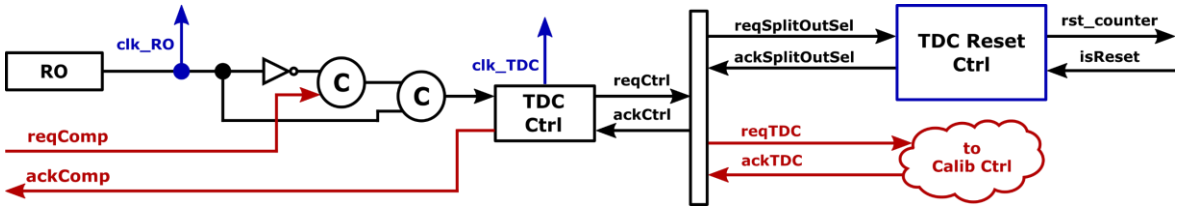


Figure 77: Diagram of the implemented TDC Controller.

Another custom module of the TDC block is the ring oscillator. It has been designed using only standard cells of the FDSOI 28nm STMicroelectronics technology node. The *RO* is composed by *two Inverter gates*, one *Nand gate*, three *Delay gates*, one *two-inputs Multiplexer* and *one Buffer*. The diagram describing the *RO* is presented in Figure 78. In this configuration, the multiplexer permits bypassing one of the delay gates, in order to accelerate the RO. This feature allows the TDC improving its measurement resolution when required.

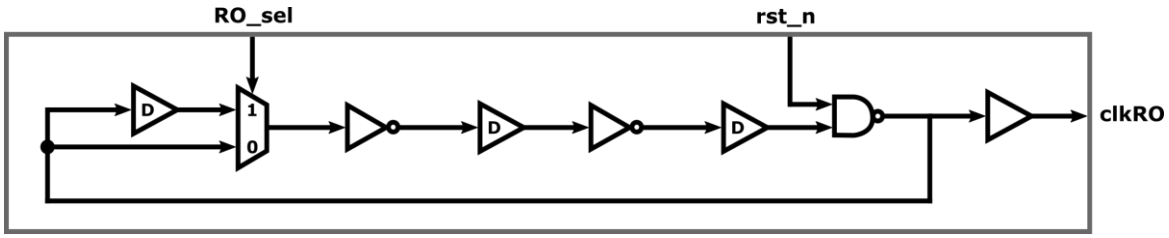


Figure 78: Diagram of the implemented Ring Oscillator.

6.5.4 Calibration Control Circuit

The *"calibration control circuit" (CalibCc)*, described in Figure 79, allows controlling the operations of the *"calibration logic circuit" (CalibLc)*, which performs a calibration sequence of 32768 T_C cycles before letting the system switching to the next operation phase. Initially the *"Calib_state"* value is *"zero"*, activating the upper control path between the split and the merge controllers. This path activates the *"CalibLc"* for one extra calibration cycle and, hence, checking if the calibration phase is finished before updating the *"Calib_state"* signal value. In this design, the "Calib Loop" register controller has been used to control the activation of a unique register managing the *"Calib_state"* value. This choice has been taken for two reasons. The first is to stabilize the *"Calib_state"* signal before applying it to the split controllers. The second is to use the delay inside the calibration loop controller for ensuring the *"Split Out Calib"* split controller to finish its current cycle task before changing the *"Calib_state"* to its new value.

Once the calibration cycle is finished, the value of *"Calib_state"* switches to *"one"* and the calibration controller uses the lower path between the Split and the Merge, until a new reset occurs.

It is essential to highlight that it should normally be necessary to add a delay *"Calib Loop"* controller and the *"Split In Calib"* selection request pin, in order to guarantee that the selection signal *"Calib_state"* arrives before its associated request. In our physical implementation, the insertion of this delay is not necessary. Indeed, if a clock is declared in the timing constraints in the *"reqTDC"* path, the Synthesis and Place and Route tools automatically identify a clock gating behavior in the path and insert the necessary delays ensuring that the selection signal arrives before the selection request rising edge.

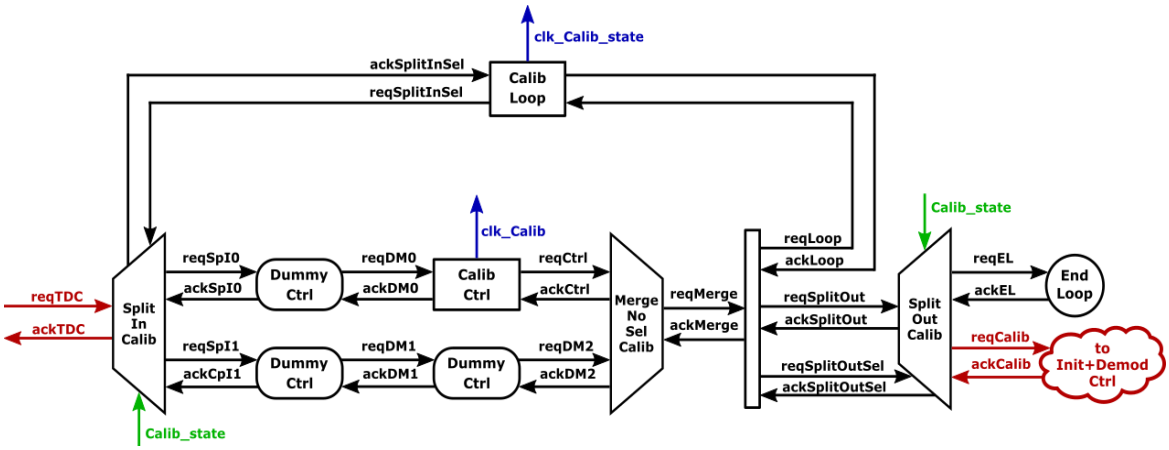


Figure 79: Diagram of the implemented Calibration controller.

6.5.5 Initiation and Demodulation Control Circuit

Once the calibration phase is completed, the system enters the initiation phase. During this phase, the system waits for a first low amplitude symbol indicating the initiation of the communication sequence. In this phase, the behavior is similar to the calibration controller. Initially, the *"Init_state"* value is set to *"zero"* and the upper path between the split and the merge controllers is activated. Once the first low amplitude symbol has been detected, the *"Init_state"* value is set to *"one"*, signaling the start of the demodulation phase. As for the case of the Calibration Controller, the *"Init Loop"* register controller activates a unique register that provides a stable *"Init_state"* signal to the controller. The delay inside this controller also permits the activation of the *"Split Out Init"* controller before changing the *"Init_state"* value. No delay has been added between the *"Init Loop"* and the selection request of the *"Split In Init"* controller, for the same reason as explained for the calibration controller. The "initiation and demodulation" controller is presented in Figure 80.

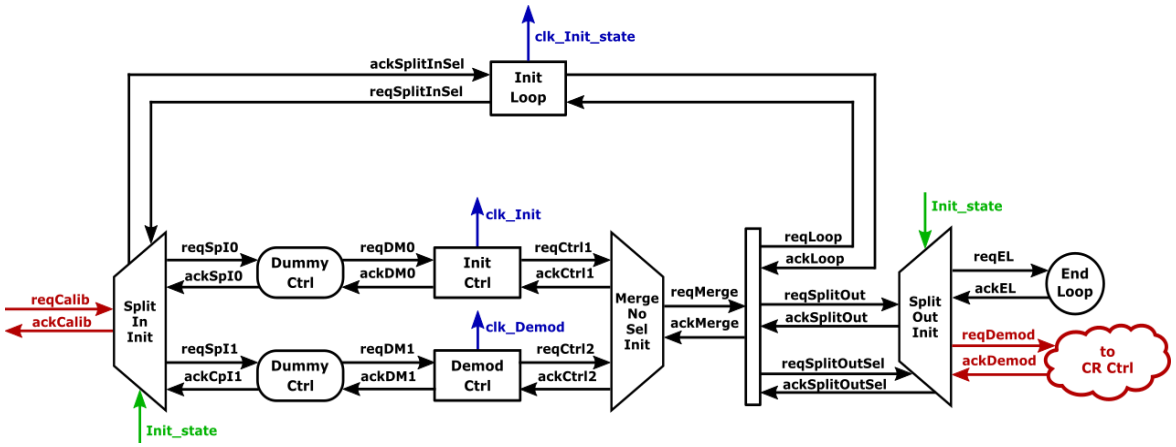


Figure 80: Diagram of the implemented Initiation and Demodulation controller.

6.5.6 Code Reconstruction Control Circuit

In this section, the code reconstruction control circuit, depicted in Figure 81, is described and explained. The demodulation controller generates a new request only when new output symbols have been identified by the demodulation algorithm. In that case, the code reconstruction logic circuit is activated in order to save this new output codes in its internal registers. When 32 new output bits have been collected, the *"codeOutEvent"* selection signal switches to "one" and activates the next controller, the "Serial Out" controller. Otherwise, it maintains the value "zero" for using the *"End Loop"* module in order to stop the activity request and for responding to the necessary acknowledgments. Once again, no delay has been added between the "CR Ctrl" and the "Split Out CR" controllers for ensuring that the "CodeOutEvent" signal arrives before the corresponding selection request *"reqSplitOutSel"* on account of the clock gating features of the synthesis and place and route tools.

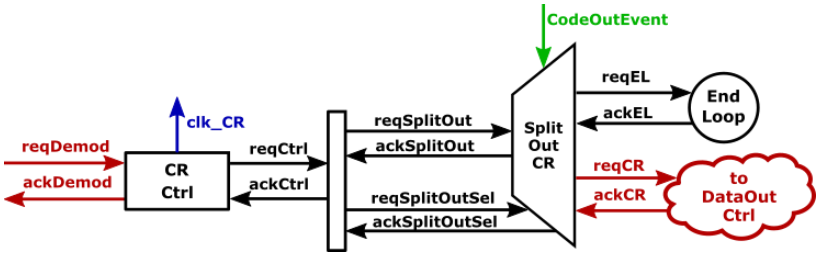


Figure 81: Diagram of the implemented Code Reconstruction controller.

6.5.7 Serial Output Control Circuit

This chapter describes the serial output control circuit presented in Figure 82. After a new 32-bit packet has been grouped by the *"Code Reconstruction"* module, it generates an input request for the *"Serial Output"* controller. Initially, the *"BitOutEvent"* value is "zero" allowing the reception of a new input request for activating the output serialization of 32 new bits through a simple asynchronous 4-phase handshake protocol, using *"reqOut"* and *"ackOut"* signals. When the *"SerialOut Ctrl"* register controller is first activated, the *"BitOutEvent"* value changes to "one", enabling the loop through the *"Split Out Serial"* controller. This loop is activated until the last bit of the packet has been outputted. At that moment, the *"BitOutEvent"* value returns to its initial state activating the "End Loop" controller, and waiting for a new input request arriving from the *"Code Reconstruction"* controller.

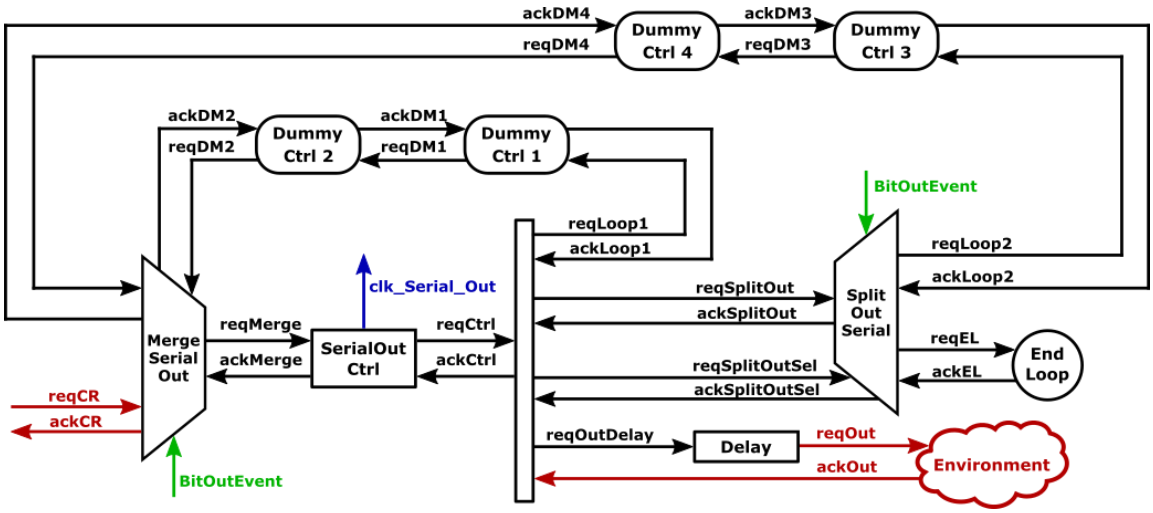


Figure 82: Diagram of the implemented Serial Output controller.

6.6 SPI Slave and Bank of Registers

The SPI module has been designed to communicate with a register bank containing 16 8-bit registers. Some of these register outputs are directly connected to internal inputs of the demodulation system in order to set internal parameters and performing configurations. Some of these registers are directly connected to internal outputs of the demodulation system in order to perform debugging when necessary.

6.7 Results

The proposed solution has been designed using the FDSOI 28 nm from STMicroelectronics and sent to fabrication in November 2020. The fabricated testchip has been received in August 2021, however, it has not been tested yet. Therefore, in this chapter, we present the area usage of the synthetized circuit, the post-synthesis power consumption and the post-layout simulation results of its performances. The power consumption estimation and the post-layout simulations have been executed using the same digital testbench environment in SystemVerilog, as those described in chapter 4.2.4.3.

6.7.1 Synthetized area and power consumption estimation

In order to compare the proposed solution with the synthetized IQ demodulator, the result of the area usage and power consumption of our ASK demodulator are also presented after synthesis. These results are depicted in the following table.

Table 3: Synthetized area and power consumption results comparison.

	Digital IQ Demodulator[1]	Proposed Demodulator
Synthetized Area	15429 μm^2	5138 μm^2
Power consumption after Synthesis	668.6 μW	861.4 μW

These results show that the synthetized area of the proposed solution is three times smaller than the one used by the IQ demodulator. For the latter, this number does not consider the area employed by the High-speed ADC and the PLL needed for the operation of the IQ solution. Usually, a common ADC needs several passive devices adding a considerable area to that alternative.

The power consumption results have been both obtained using *"Synopsys Primetime PX"* tool, after providing a switching activity file acquired from the testbench simulation with *"Mentor Graphics QuestaSim"* tool from Siemens EDA. The proposed demodulation solution consumes 29% more power compared to the IQ demodulator. However, this value does not include the power consumption of the ADC working with

a clock frequency equivalent to four times the carrier frequency ($4f_C$). Moreover, it does not consider the power consumption associated to a PLL for generating this "high-speed" clock. As shown in Table 3, the average power consumption of the proposed system is 861.4 μW. Thus, when $T_{symbol} = 2T_C$, which is the maximal bitrate of the system, the energy consumption per symbol is 127 pJ.

6.7.2 Post-layout Performance Results

For the simulations with the *SV* testbench, the comparators integrating the programmable delay described in Figure 61 have been characterized using electrical simulations and then adapted to a SystemVerilog model. The electrical simulations integrate the extraction of the parasitic elements from the layout obtained after the complete comparator circuits have been custom designed. The performance result of the implemented testchip is shown in Figure 83.

The first result to highlight is the fact that the proposed system does not work for a bit rate equal to $T_{symbol} = 128T_C$. In Figure 41, this issue does not appear with the RTL design of the system since the TDC is ideal and uses the *$time* SystemVerilog function, thus obtaining an exact time measurement value using a *real* variable. Indeed, the value representing one carrier period (T_C) in terms of the TDC measurement was also precise, using a *real* value. In that case, the "carrier_periods_calc" module calculates an accurate number of equivalent carrier periods while using an exact division calculation.

In the synthesizable implementation, an integer value is applied to represent the T_C value arriving from the calibration module. For the case $T_{symbol} = 128T_C$, when two or more consecutive *zeros* are modulated, the module calculating the equivalent carrier periods accumulates a decimal error that produces an erroneous output "carrier_periods" value, which shifts the system in terms of following the communication sequence. This problem is not produced for lower values of T_{symbol}, where the decimal error does not produce an erroneous carrier periods calculation.

Both cases when $T_{symbol} = 128T_C$ and $T_{symbol} = 1T_C$ have been deleted from the graph.

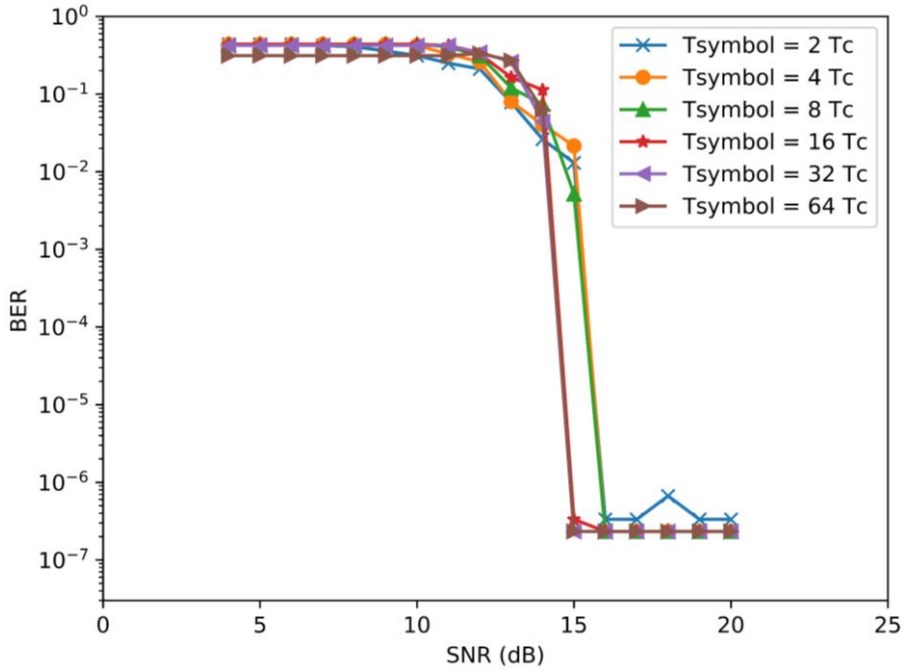


Figure 83: Graph of the SNR v/s BER performance of the fabricated testchip.

From Figure 83, we can infer that the performance of the fabricated system has the same behavior of the RTL design, obtaining an abrupt *BER* decrease around $15\ dB$ of *SNR* for the three lower bitrates and $16\ dB$ of *SNR* for the three higher bitrates of the graph. In the simulation where $T_{symbol} = 2T_C$, a *BER* of $3.33 \cdot 10^{-7}$ is obtained for $SNR \geq 16dB$. For all the other values of T_{symbol}, we reach the limit of 3,000,000 of sent bits without any error. In these cases, the *BER* is less than $3.33 \cdot 10^{-7}$. This performance improvement, compared to other solutions from the literature, is mainly due to the majority operator, making the system more reliable to unexpected pulses produced by the amplitude noise.

The layout of the fabricated testchip IP is presented in Figure 84. The dimensions of the entire testchip are 1649 μm x 2085 μm and those from the ASK demodulator IP are 182.92 μm x 114 μm.

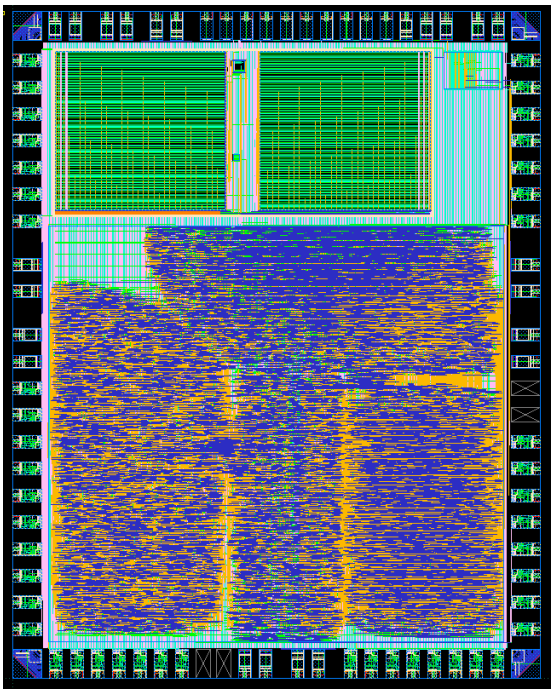


Figure 84: Layout TOP of the fabricated Testchip (1649 μm x 2085 μm).

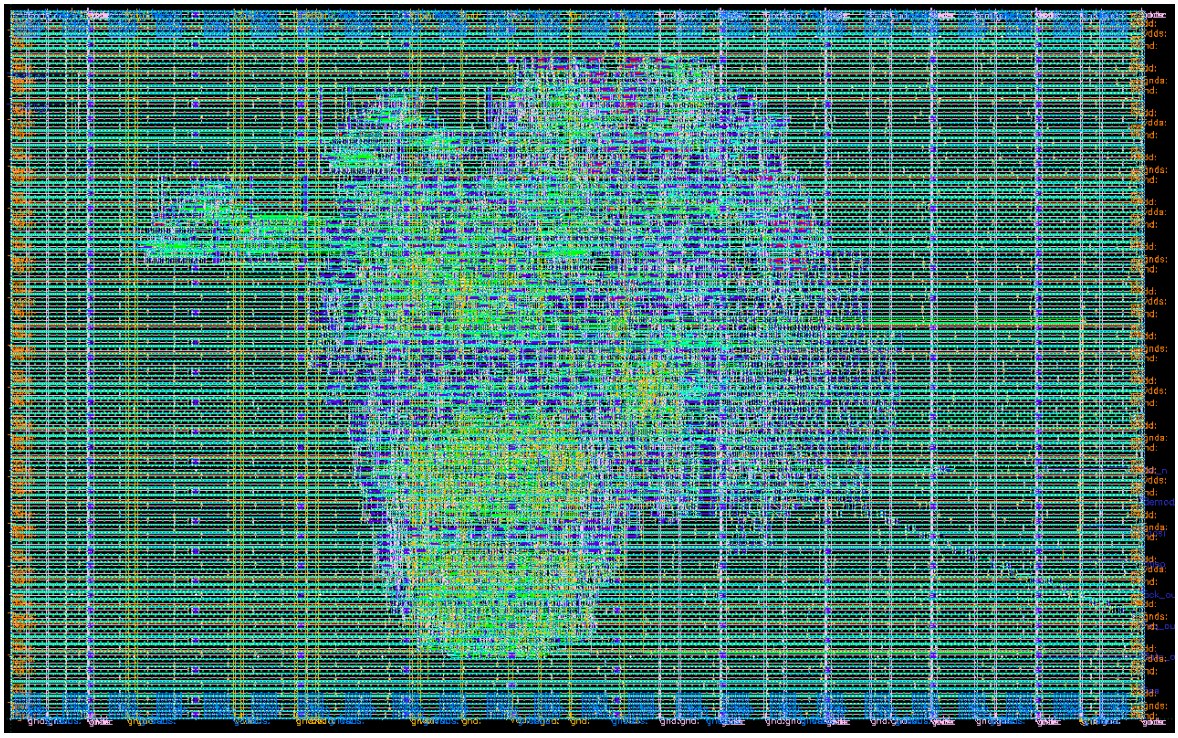


Figure 85: Layout of the fabricated ASK demodulator IP (182.92 μm x 114 μm.).

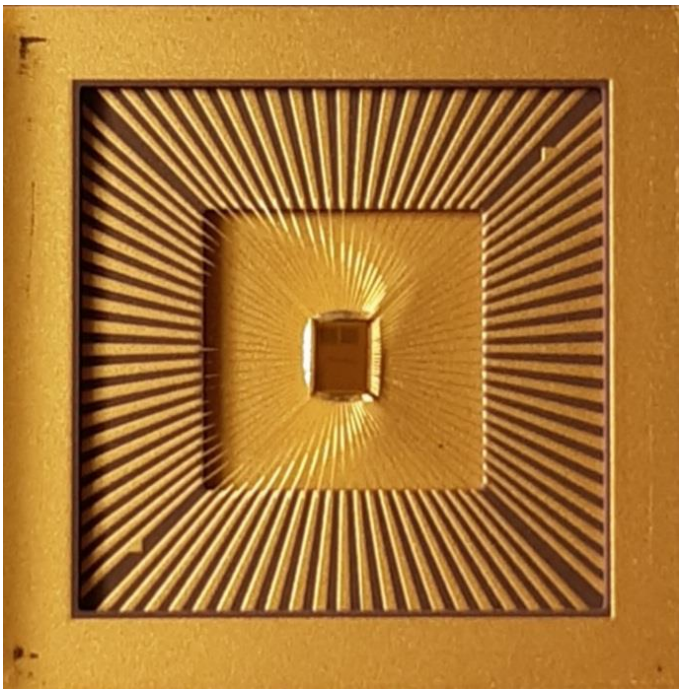


Figure 86: Picture of the package of the fabricated Testchip.

6.8 Conclusion

This chapter presents the physical implementation of the third proposal of this thesis, which obtained the best performance results and fulfilled all the objectives established for this work. For the physical design, some modifications have been made in order to be compliant with the laboratory tests. These adaptations are briefly described as follows.

A SPI communication interface has been designed along with a register bank allowing the configuration of internal parameters and the reading of internal outputs. Additionally, an asynchronous serial interface has been designed for outputting the 32-bit words to the testing environment. This allows comparing the input code used for the modulation with the output code obtained from the demodulation. Then, a calibration module has been implemented to internally calculate the average of the TDC measurements equivalent to one carrier period. Thank to this calibration phase, the system remains reliable whatever the PVT variations. Furthermore, an initiation module has been designed to automatically synchronize the demodulation system with the communication sequence. This task is very important since the entire demodulation method relies on the precise identification of the end of each symbol period. Moreover, the *"carrier_periods_calc"* block has been integrated in the TDC module for an "on-the-

fly” calculation of the equivalent number of carrier periods of each time measurement. This permits to reduce the synthetized area of the circuit since it avoids using a division for performing that calculation. Then, two different “Strong-Arm” comparators have been custom designed and simulated. One is accurate in the vicinity of “Gnd” and the other is adapted to operate close to “Vdd”. These comparators use a clock signal to define the comparison instant. A self-generated clock logic has been implemented using a programmable delay. This method allows controlling the glitches generated by the comparators.

In addition, a comparison between the IQ demodulator and the third proposal has been done, where the synthetized area and power consumption are analyzed. The proposed solution divides by three the area of the IQ demodulator without taking into account the required additional blocks. However, the power consumption is increased by 29%. In both cases as already pointed, the IQ demodulator does not consider the PLL and the ADC because they have not been designed and, hence, integrated in this comparison. It is important to highlight that the energy consumption per symbol of our system is 127 pJ when $T_{symbol} = 2T_C$.

Finally, a post-layout simulation is shown, permitting to verify the same performance encountered in the RTL simulation of chapter 5.2 with a complete designed circuit. In the simulation results, the BER abruptly decreases when the SNR is greater than 15-16 dB, and its values are in the order of 10^{-7}.

The size of the fabricated ASK demodulator IP is 182.92 μm x 114 μm with a layout utilization of around 40-50%, as it can be determined from Figure 85.

7 Conclusions and Perspectives

7.1 Thesis contributions

Nowadays, NFC devices are extensively used in many applications covering a massive market. This technology serves for many functions, such as transport validation, access badges and contactless payment. One of the most sensitive blocks of those systems is the demodulation module, which requires a full ASK demodulation solution with a limited harvested power. Most of the implementations of these demodulators employ analog circuits, which are custom solutions allowing a perfect control on the power consumption and the system performance. However, if newer technology nodes are used, these analog solutions are not well suited when the technology node shrinks.

In this work, we presented three novel digital event-based ASK demodulation strategies in order to be better compliant with the technology shrink.

The first solution presents a technique allowing to detect and to register the occurrence of a high-level crossing for each carrier period. This detection is particularly stable when combining with a Muller gate this high-level detection with a second level crossing detection near *"Gnd"*. This allows masking the glitches produced by the amplitude noise. Therefore, a minimal level crossing sampling scheme is proposed, with only two levels. Each level crossing detection is performed with a comparator connected to the modulated signal and each reference level. This method is only accurate if no insertion or deletion error occurs. These latter correspond to high-level crossings in low amplitude symbol periods or to an absence of high-level crossings during high amplitude periods. For that reason, this technique is only recommended for high modulation indexes. Moreover, the demodulation circuit is really small, since it uses only a few gates.

This previous method does not use any time measurements. The second proposal introduces the same circuit as the first proposal for identifying the high-level crossings and adds a TDC to measure the time between two consecutive samples. Internally, this measurement is processed into an equivalent number of carrier periods. If the samples are separated by $1T_C$, the system infers that two consecutive high amplitude periods happen. Otherwise, the system deduces that low amplitude periods occur. This

technique analyzes the tendency of the last N samples to high or low amplitudes and deduces the instant of a symbol change. This solution is more accurate than the first proposal. However, it is sensitive to crossing errors near the edge between two different symbols.

A third solution has been proposed, taking profit of the collected time information of the second proposal, for keeping the system always synchronized with the communication sequence. This method identifies the end of each symbol period with the information of the number of carrier period elapsed between consecutive samples. At that instant, the system calculates the majority of occurrence between high or low amplitude within each symbol period. It is noticeable that the majority operation enhances the reliability of the symbol recognition for large symbol periods. Indeed, the relative probability of false level detections is reduced when the symbol period lengthens. Another contribution was the design of the reset mechanism used by the TDC, which ensures the realization of continuous measurements.

The last two solutions use a bundled-data asynchronous circuit for processing the events generated by an EB-ADC. The third method is our most performing solution. The chosen EB-ADC operates with only two reference levels and uses a TDC measuring the time between two consecutive events. This time measurement is processed by a digital algorithm performing the ASK demodulation. Thanks to the digital implementation of the ASK demodulator, this makes not only the circuit able to easily support the technology node shrink but also less consuming. Indeed, the combination of an asynchronous logic with an EB-ADC only activates the demodulation circuit when a new event occurs (i.e. a new sample is captured) and lets the circuit in a standby mode otherwise. This helps a lot reducing the power consumption of the whole system. Therefore, the low activity system produced by the EB-ADC drastically reduces the power consumption compared to a standard Nyquist ADC but also saves area and power due to the absence of peripherals such as a PLL. Moreover, our proposal also allows relaxing the constraints of the energy harvesters since the power supply voltage is reduced with technology shrinking and the current peaks are spread with the asynchronous method. Furthermore, the BER rate is also improved thanks to the demodulation algorithm that proves being robust to the amplitude noise. The results are very promising, as a BER of $3.33 \cdot 10^{-7}$ is obtained for a SNR=16dB and a bit rate at 6.78 Mbps. At this bitrate, the energy consumption per symbol is only 127 pJ.

The validation of our approach has been done from the system level to the gate level. Finally, a circuit has been designed and fabricated in FDSOI 28nm from STMicroelectronics and is currently under test.

7.2 Perspectives

The demodulation technique used in the proposed solution performs time measurements approximated to the equivalent carrier periods. However, the implemented TDC achieve very precise time measurements, in the order of hundreds of picoseconds. This precision enhances the demodulation possibilities using the same technique. This TDC might be also used for estimating the phase shifting of a PSK *(Phase Shifting Key)* modulated signal. It can be done employing the same *L1Sig* signal, which is near *"Gnd"* for mimicking a zero-crossing detection. An important feature of the proposed technique is that the PSK and ASK demodulations can be accomplished independently. In that case, the future works could address the demodulation of polar QAM constellations.

From a general standpoint, RF Communications are a permanent growing technology field with the adoption of 5G and in a near future 6G communications. With these upcoming technologies, demodulating a RF signal at very low energy is today mandatory. RF communications due to their nature need a carrier, which produces a useless activity on the receiver side, leading to extra power consumption. Therefore, low-energy receivers are very challenging to design. Furthermore, due to the 5G and 6G specifications, the receivers require agility for demodulating several standards and complex modulation schemes. Only digital circuits are able to achieve these capabilities. Indeed, they offer the required agility, configurability and a certain ability to support more easily the technology shrink. Moreover, digital receivers allow faster developments, which is a mandatory requirement to be on the telecommunication market.

In order to address these challenges, smart receivers and low-energy, the envisioned solutions, discussed inside our team, target event-based approaches and asynchronous digital designs. Indeed, such strategies are able to limit the receiver activity by reducing the number of processed data. This leads to a reduced power consumption and lower electromagnetic emissions, which is a plus for RF

communications. Finally, this thesis is the premises of a wider approach for the RF systems, which already require agility and very low power consumption.

References

[1] "The Cowboy Channel," [Online]. Available: https://www.thecowboychannel.com/story/43210384/global-near-field-communication-nfc-chips-market-2021-covid-19-impact-on-top-countries-data-industry-trends-share-size-demand-growth-opportunities. [Accessed 04 06 2021].

[2] M. R. Yuce and A. Tekin, "Ultra Low-Power Digital Demodulators for Short Range Applications," in *2006 IEEE 63rd Vehicular Technology Conference*, Melbourne, Vic., Australia, 2006.

[3] F. Yuan, "Design techniques for ASK demodulators of passive wireless microsystems: a state-of-the-art review," *Analog Integrated Circuits and Signal Processing,* vol. 63, no. 1, pp. 33-45, 2010.

[4] M. L. Navaii, M. Jalali and H. Sadjedi, "A 34-pJ/bit Area-Efficient ASK Demodulator Based on Switching-Mode Signal Shaping," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 64, no. 6, pp. 640-644, 2017.

[5] S. Baik, G. K. Arup, J. Minkyu and L. Junghyup, "A 1 V 2.5 μW fully-differential ASK demodulator with 12.5 pJ/bit FOM for ultra-low power biomedical applications," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boston, MA, USA, 2017.

[6] C. D. Ziomek and P. L. Corredoura, "Digital I/Q demodulator," *Proceedings Particle Accelerator Conference,* vol. 4, pp. 2663-2665, 1995.

[7] B. Bidegaray-Fesquet and L. Fesquet, "Levels, peaks, slopes… which sampling for which purpose?," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, Krakow, Poland, 2016.

[8] K. Han, Y. Wei and X. Ma, "An efficient non-uniform filtering method for level-crossing sampling," in *IEEE International Conference on Digital Signal Processing (DSP)*, Beijing, China, 2016.

[9] P. G. Jespers, Integrated Converters : D to A and A to D Architectures, Analysis and Simulation, Oxford University Press, 2001.

[10] E. Allier, G. Sicard, L. Fesquet and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," in *Ninth International Symposium on Asynchronous Circuits and Systems*, Vancouver, BC, Canada, 2003.

[11] J. Mark and T. Terence, "A Nonuniform Sampling Approach to Data Compression," *IEEE Transactions on Communications,* vol. 29, no. 1, pp. 24-32, 1981.

[12] M. TLILI, A. MAALEJ, M. BEN-ROMDHANE, M. C. BALI, F. RIVET, D. DALLET and C. REBAI, "Level-Crossing ADC Modeling for Wireless Electrocardiogram Signal Acquisition System," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Taipei, Taiwan, 2016.

[13] F. J. Beutler, "Error-Free Recovery of Signals from Irregularly Spaced Samples," *SIAM Review,* vol. 8, no. 3, p. 328–335, 1966.

[14] Y. Li, D. Zhao and W. A. Serdijn, "A Sub-Microwatt Asynchronous Level-Crossing ADC for Biomedical Applications," *IEEE Transactions on Biomedical Circuits and Systems,* vol. 7, no. 2, pp. 149-157, 2013.

[15] A. El Hadbi, "Time-to-Digital Conversion based on a Self-Timed Ring Oscillator (Doctoral dissertation)," Available at Archives Ouvertes HAL (HAL Id : tel-03132550), Grenoble, France, 2019.

[16] C. Sitik, W. Liu, B. Taskin and E. Salman, "Design Methodology for Voltage-Scaled Clock Distribution Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 24, no. 10, pp. 3080-3093, 2016.

[17] H. Kawaguchi and T. Sakurai, "IEEE Journal of Solid-State Circuits," *IEEE Journal of Solid-State Circuits,* vol. 33, no. 5, pp. 807-811, 1998.

[18] S. H. Unger, "A Study of Asynchronous Logical Feedback Networks," Research Laboratory of Electronics - Massachusetts Institute of Technology, Cambridge, Massachusetts, 1957.

[19] A. Kondratyev and K. Lwin, "Design of asynchronous circuits by synchronous CAD tools," in *Proceedings 2002 Design Automation Conference*, New Orleans, USA, 2002.

[20] J. Simatic, "Flot de conception pour l'ultra faible consommation : échantillonnage non-uniforme et électronique asynchrone.," Micro et nanotechnologies/Microélectronique. Université Grenoble Alpes, Grenoble - France, 2017.

[21] D. E. Muller, "Asynchronous logics and application to information processing," pp. 289-297.

[22] A. J. Martin, "Formal program transformations for VLSI circuit synthesis," in *Formal Development Programs and Proofs*, 1989, pp. 59-80.

[23] M. Shams, J. Ebergen and M. Elmasry, "Modeling and comparing CMOS implementations of the C-element," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 6, no. 4, pp. 563-567, 1998.

[24] S. Germain, "Electromagnetic spectrum control of asynchronous digital circuits," Available at Archives Ouvertes HAL (HAL Id : tel-02523608), Grenoble - France, 2019.

[25] M. K. Kangi, M. Maymandi-Nejad and M. Nasserian, "A fully digital ASK demodulator with digital calibration for bioimplantable devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 23, no. 8, pp. 1557-1561, 2015.

[26] A. R. Iga Jadue, S. Engels and L. Fesquet, "An Event-Based Strategy for ASK demodulation," in *2019 5th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*, Vienna, Austria, 2019.

[27] V. Taranalli, "Awgn function from CommPy Library. Release 0.3.0," 24 May 2019. [Online]. Available: https://github.com/veeresht/CommPy/blob/master/commpy/channels.py. [Accessed 20 April 2021].

[28] A. Almansouri, A. Alturki, A. Alshehri, T. Al-Attar and H. Fariborzi, "Improved StrongARM latch comparator: Design, analysis and performance evaluation," in *13th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Giardini Naxos - Taormina, Italy, 2017.

[29] A. Mardari, Z. Jelčicová and J. Sparsø, "Design and FPGA-implementation of Asynchronous Circuits Using Two-Phase Handshaking," in *25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, Hirosaki, Japan, 2019.