UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Highly Programmable Low-Power RF Front Ends for Adaptive Long-Range Internet-of-Things Receivers

Marco Antonio Gonzalez Gonzalez

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. David Bol	*Supervisor*	UCLouvain, Belgium
Prof. Christophe Craeye	*President*	UCLouvain, Belgium
Prof. Denis Flandre	*Secretary*	UCLouvain, Belgium
Dr. Jan Craninckx		imec, Belgium
Prof. Georges Gielen		KU Leuven, Belgium
Dr. Dominique Morche		CEA, France

January 2026

Abstract

Despite outstanding advances in energy efficiency in Internet-of-Things (IoT) devices, wireless communication often remains one of the most significant contributors to their overall power consumption. The contribution from radio-frequency (RF) receivers is mainly due to three reasons. First, the fact that the input signal has a small amplitude, needing great amplification and a low input-referred noise. Second, the high-frequency operation because of the carrier, usually at least $10\times$ higher than clock frequencies used by IoT nodes in the tens of MHz. Finally, the amplitude of the input signal is comparable to that of the noise, which makes its demodulation more difficult. Its low input power is due to the high channel attenuation linked to communicating over the air, especially over long distances. The stochastic variability of this channel attenuation is an aggravating factor. Shadowing from obstacles in the path between objects or movement in their surroundings causes a stochastic attenuation of the signal of interest, and other RF transmissions close to the objects also cause dynamic interference. To overcome this, transceivers for long-range IoT communications are conventionally designed to operate correctly under certain worst-case conditions. However, these worst-case conditions are rarely encountered in practice. Operating under this hypothesis causes a non-negligible power overhead on RF receivers because the circuits' performance in terms of noise, linearity, and interference immunity is at its best continuously. Reducing the power consumption of IoT devices is especially important for battery-powered or energy-harvesting ones, as it is key to maximizing their lifetime.

In this thesis, we explore a different design approach in which the receiver adapts its performance to the state of the wireless channel. The goal is to minimize its power consumption while maintaining an acceptable bit

error rate (BER), usually fixed at 0.1 %. Inside the low-power wide-area networks (LPWANs), LoRaWAN already features such adaptation capacity in its physical layer LoRa, through the configuration of the spreading factor (SF). The SF determines the bit rate and the necessary signal-to-noise ratio (SNR) for demodulation because it determines the number of bits sent per symbol and the symbol duration. The SF choice being discrete and limited between 6 and 12, we aim in this thesis to extend the adaptation principle to the circuit level. With programmable circuits in the RF front end of the receiver that can dynamically trade off noise for power, we can have further power savings for any SF. The design of such circuits is the main focus of this thesis, and we tackle it in three steps that are presented in three parts of this thesis: (i) methodology, (ii) prototyping, and (iii) specific use case where the noise-power trade-off is pushed to its limit.

In the first part of the thesis, we study how to design programmable analog circuits that dynamically exploit their noise-power trade-off. In particular, the problem of how to evaluate the function to optimize is analyzed in two different situations. In the first case of adding programmability to low-complexity blocks like amplifiers, simulations are sufficiently fast for this evaluation. However, in the case of large and complex mixed-signal blocks like phase-locked loops (PLLs), noise simulations are both time-consuming and computation-intensive due to the different timescales present in the circuit. Therefore, they cannot be used for design space exploration. We present a novel optimization approach for PLLs in which measurements of the fabricated circuit replace simulations to find the optimal sizing. The analog blocks must be digitally programmable to tune the design parameters after fabrication. Analytic expressions are nevertheless used to choose the parameters' tuning ranges during design time.

In the second part of the thesis, we present two prototypes of programmable RF receivers to highlight the possible power savings of channel-adaptive receivers. This is done progressively, as only the RF front end is present in the first prototype, whereas the second one includes the whole receiver chain. The first RF front-end prototype in 22-nm fully depleted silicon-on-insulator (FD-SOI) technology features a programmable low-noise amplifier (LNA) able to trade off a noise figure between 4 and 2.4 dB for a corresponding power consumption between 0.88 and 2.72 mW. It also includes (i) in-phase and quadrature mixers consuming 281 µW each with a 24.8-dB noise figure, and (ii) a fractional-N PLL consuming 370 µW. Using models for the missing blocks in the receiver chain, we estimate 46-% power consumption savings at the full receiver level, by adaptive tuning of the

LNA, if the channel attenuation improves by 2 dB. The second prototype in 65-nm bulk CMOS technology features a complete highly programmable LoRa receiver with: (i) programmable LNA and mixers, (ii) a duty-cyclable programmable all-digital PLL, (iii) low-power filter and ADC, and (iv) a digital baseband processor (DBB) with an FFT hardware accelerator. It consumes between 1.71 and 4.57 mW of for a noise figure between 13.2 and 4.8 dB, respectively. With an estimated sensitivity of −138 dBm, we reach the same sensitivity as commercial LoRa modules, but reducing power consumption by more than 3.5×.

Finally, the third part of the thesis explores wake-up receivers. Wake-up receivers also fall into the noise-power trade-off of RF receivers previously studied, being at the extreme case where the power consumption is at its lowest. This type of receiver senses the wireless channel continuously, looking to detect activity in the channel before waking up the main receiver, avoiding constant use of costly demodulation steps such as synchronization. As the wake-up receiver is continuously active, minimizing its power consumption is the main objective. Noise performance is less critical for two reasons: (i) the SNR requirement for detecting a signal is relaxed compared to a complete demodulation, and (ii) the DBB is optimized to cope with the performance loss in favor of ultra-low power operation. A prototype of a LoRa wake-up receiver was implemented in parallel with the main LoRa receiver described in the second part of the thesis in 65-nm technology. Long-range wake-up receivers with high sensitivity are rare in the literature, and for the moment, none use the LoRa modulation. The receiver can reduce its power consumption down to 1.5 mW in this mode, for a noise figure of 13.95 dB. This translates into a sensitivity of −126 dBm for a wake-up latency of 32 ms. Having a sensitivity in wake-up mode close to that of the main receiver enables further power savings in long-range networks because wake-up receivers could replace synchronization between nodes in the network. This could significantly increase the lifetime of IoT devices used in wireless sensor networks.

Acknowledgments

The work presented in this thesis is the product of more than six years of work that passed by more quickly than expected at first. It was a challenging road filled with individual efforts, but made possible by the help, support, and guidance from many people whom I want to thank now. Those who know me well will understand how hard this exercise of putting these words in writing will be, but the closing of this chapter of my life gives me this unmissable opportunity to thank them.

First, I want to thank my supervisor, Prof. David Bol, for his guidance and support from my Master's thesis, for proposing to go on this journey together, and for introducing me to this fascinating world of chip design. His suggestions and ideas, his encouragements, and his constructive comments were key to the success of this project. I think I will never forget the countless times he asked me during a meeting "*Est-ce que ça prendrait beaucoup de temps de ...*" ("Would it take a long time to ...") followed by a very relevant suggestion to enhance my work. I would always answer yes in my head, but I was always happy to do it because it brought a great added value to the work.

I am also grateful to my PhD committee and jury for guiding and reviewing this work. Thanks to Prof. Denis Flandre and Prof. Jérôme Louveaux from my PhD committee for the fruitful discussions we had on several occasions, especially during the confirmation. Their suggestions, coming from different technical backgrounds at this checkpoint halfway down the road, shaped how I conducted my research thereafter. Thanks to Prof. Christophe Craeye for agreeing to preside over my thesis jury, and to Dr. Jan Craninckx, Prof. Georges Gielen, Dr. Dominique Morche, and again Prof. Denis Flandre for taking the time to review my thesis, for the discussions during the private defense, and for the valuable comments they gave me to greatly

improve this manuscript.

In addition, I want to thank UCLouvain for the financial support and for giving me the opportunity to be a teaching assistant, which allowed me to continue to pursue my passion for teaching. Even if it sometimes took a lot of time, I truly enjoyed interacting with students, so it was time well spent. On the professional side, this experience helped me gain a deeper understanding of electronic circuits in preparation for the sometimes unexpected questions from students, and on a more personal side, my French definitely improved at a greater speed. Thank you particularly to those who shared this experience with me during many years in different courses. Martin Lefebvre with the coordination meetings and the rushed shift to online teaching in ELEC1370. Léopold Van Brandt with the reconstruction of the ELEC2650 course, and Sylvain Favresse, who contributed too to this after Léopold finished his thesis. And also Nicolas Roisin for the flipped classes and friendly discussions over the distribution of the practical sessions in ELEC2532.

I, of course, must thank all the people who helped create the great working environment I enjoyed during all these years. Starting with the people that welcomed me to the Electronic Circuits and Systems (ECS) group: Maxime Schramme, Pengcheng Xu, Roghayeh Saeidi, Rémi Dekimpe, Martin Lefebvre, Ludovic Moreau, and Mathieu Xhonneux, including those who joined the group at the same time as me: Adrian Kneip and Thibault Pirson, also the people that joined the group after: Julien Verecken, Pol Maistriaux, Louis Golard, Nicolas Chauvaux, Augustin Wattiez, Sylvain Favresse, Robin Dethienne, and Thomas Antoine, and the latest additions of Prof. Martin Andraud and his researchers Artuur Astaes and Gaurav Singh. Thank you all for the conversations, shared laughs, and in general the time spent together during coffee and lunch breaks, social events, online gaming during the COVID lockdown, great trips to Milan and Bruges, and even during the painful late nights spent during tape-outs. I will keep fond memories of all this time. I also appreciate the supportive atmosphere we have in the group, from a simple catch-up during lunch to the general willingness to help in any needed way. Two colleagues particularly helped me with the technical content of this thesis. First, thanks to Maxime Schramme for his support during my Master's thesis, during my first months in the group as my sponsoring researcher, and during my first tape-out. We had great and fruitful discussions over PLLs being the only PLL guys in the group. Also thanks to Pol Maistriaux, who found the way to change his research topic and create a very nice and nurturing collaboration between our theses,

which resulted in the co-authorship of most of my scientific articles. He eventually finished integrating the group of PLL guys.

I would like to thank more in general all the people in ELEN who also contributed to the great working environment. This includes all the people in the A1 hall where my office was, as well as all the technical and administrative staff, in particular Brigitte Dupont, Anne Gillet, Viviane Sauvage, Nathalie Ponet, and Isabelle Dargent, who were always available to help. Special thanks to Souley Djadjandi and Benoît Hubert for helping me solder all the PCBs and for coming up with creative ways to solve some design problems, as well as to Pierre Gérard for helping automate tests in WELCOME.

I don't forget my friends from my school and university years, but I must make a special mention to my two lifelong friends Fabiana and Adriana, who stayed close friends even after we all moved to different countries in Europe. What wonderful trips have we made together, which I hope will continue for many years to come. I also want to thank my volleyball teammates for helping me get rid of stress when playing with you.

Last, but certainly not least, I want to thank my family. My mom and dad for their endless love and support, even at a distance, but certainly compensating when we are together. My big sister Adriana and her husband Mohieddine for their guidance, from advice when moving to Belgium to professional advice as colleagues, and so many other things. My nieces and nephew, Mimi, Milo, and Rori. I hope they one day understand how much their daily photos and their voices warm my heart. My little sister Veronica, or should I say roommate and best friend, thank you for everything and your everyday support. Also to her (recently) fiancé, Deven, my fellow volleyball supporter, for your humor and support. Without you all, I would definitely not be here writing these words at the end of this journey. Thank you so much.

Contents

List of Publications

Related journal papers

JP1. M. Gonzalez, P. Maistriaux, P. Harpe, J. Louveaux, and D. Bol, "Highly programmable RF receiver," (in writing process).

Related conference papers

CP1. M. Gonzalez and D. Bol, "Post-Silicon Optimization of a Highly Programmable 64-MHz PLL Achieving 2.7-5.7 µW," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Antwerp, Belgium: IEEE, Apr. 2023.
DOI: 10.23919/DATE56975.2023.10137143.

CP2. M. Gonzalez, P. Maistriaux, and D. Bol, "A Narrowband RF Front End in 22-nm FD-SOI Featuring a Programmable Low-Noise Amplifier with a Configurable Noise-Power Trade-Off," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, Singapore, Singapore: IEEE, May 2024.
DOI: 10.1109/ISCAS58744.2024.10557855.

CP3. P. Maistriaux, M. Gonzalez, J. Louveaux, and D. Bol, "Leveraging a Digital Chirp Spread Spectrum Detector for LPWAN Wake-Up Receivers," in *2024 14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Rome, Italy: IEEE, Jul. 2024, pp. 638–643.
DOI: 10.1109/CSNDSP60683.2024.10636391.

Unrelated journal papers

UJP1. M. Gonzalez, P. Xu, R. Dekimpe, M. Schramme, I. Stupia, T. Pirson, and D. Bol, "Technical and Ecological Limits of 2.45-GHz Wireless Power Transfer for Battery-Less Sensors," *IEEE Internet Things Journal*, vol. 10, no. 17, pp. 15431–15442, Sep. 2023, DOI: 10.1109/JIOT.2023.3263976.

Unrelated conference papers

UCP1. R. Dekimpe, M. Schramme, M. Lefebvre, A. Kneip, R. Saeidi, M. Xhonneux, L. Moreau, M. Gonzalez, T. Pirson, and D. Bol, "SleepRider: a 5.5µW/MHz Cortex-M4 MCU in 28nm FD-SOI with ULP SRAM, Biomedical AFE and Fully-Integrated Power, Clock and Back-Bias Management," in *2021 Symposium on VLSI Circuits*, Kyoto, Japan: IEEE, Jun. 2021. DOI: 10.23919/VLSICircuits52068.2021.9492365.

Acronyms

AAF	Anti-aliasing filter
ADC	Analog-to-digital converter
ADPLL	All-digital phase-locked loop
AWGN	Additive white Gaussian noise
BBCO	Back-bias-controlled oscillator
BBN	NMOS back-bias voltage
BBP	PMOS back-bias voltage
BER	Bit error rate
BFSK	Binary frequency-shift keying
CDF	Cumulative distribution function
CFO	Carrier-frequency offset
CMFB	Common-mode feedback
CMOS	Complementary metal-oxide-semiconductor
CP	Charge pump
CPU	Central processing unit
CSRO	Current-starved ring oscillator
CSS	Chirp spread spectrum
DAC	Digital-to-analog converter
DBB	Digital baseband
DCO	Digitally controlled oscillator
DDSM	Digital delta-sigma modulator
DMEM	Data memory
F	Noise factor
FAR	False-alarm rate

FBB	Forward back-bias
FD-SOI	Fully depleted silicon-on-insulator
FF	Fast nMOS and fast pMOS process corner
FFT	Fast Fourier Transform
FIR	Finite impulse response
FoM	Figure of merit
FSM	Finite-state machine
GF	Global Foundries
Gm	Transconductance
GPIB	General-purpose interface bus
GPIO	General-purpose I/O
HVT	High-threshold-voltage
I	In-phase
IIP3	Input third-order intercept point
IIR	Infinite impulse response
IoT	Internet of Things
LC	Inductor-capacitor
LNA	Low-noise amplifier
LP	Loop filter
LPF	Low-pass filter
LPWAN	Low-power wide-area network
LS	Level shifter
LSB	Least-significant bit
LT	Long-term
LVT	Low-threshold-voltage
MAC	Medium access control
MASH	Multistage noise shaping
MCU	Microcontroller unit
MIM	Metal-insulator-metal
MMD	Multi-modulus divider
MOM	Metal-oxide-metal
MSB	Most-significant bit
NB-IoT	Narrowband IoT
NF	Noise figure
NSGA-II	Non-dominated sorting genetic algorithm

NTF	Noise transfer function
P2D	Phase-to-digital converter
PCB	Printed circuit board
PER	Packet error rate
PFD	Phase-frequency detector
PLL	Phase-locked loop
PMEM	Program memory
PSD	Power spectral density
PVT	Process, voltage, and temperature
Q	Quadrature
RBB	Reverse back-bias
RF	Radio frequency
RMS	Root mean square
RVT	Regular-threshold-voltage
SF	Spreading factor
SLVT	Super-low-threshold-voltage
SNR	Signal-to-noise ratio
SOI	Silicon on insulator
SQNR	Signal-to-quantization-noise ratio
SRAM	Static random-access memory
SS	Slow nMOS and slow pMOS process corner
STF	Signal transfer function
STO	Sampling-time offset
SVT	Standard-threshold-voltage
TDC	Time-to-digital converter
TIA	Transimpedance amplifier
TSMC	Taiwan Semiconductor Manufacturing Company
TT	Typical nMOS and typical pMOS process corner
TX	Transmitter
UHVT	Ultra-high-threshold-voltage
ULP	Ultra-low-power
VCO	Voltage-controlled oscillator
WuS	Wake-up signal

1

Introduction

THE idea behind the Internet of Things (IoT), at its beginnings, intended to give objects the ability to collect data for the Internet. Data collected by objects without human intervention to help humans keep track of information such as expiration dates, available stock, or repair needs, and hopefully reduce costs, loss, and waste [1]. Over the last decade, the IoT has greatly diversified [2] to include applications such as connected homes, which currently hold the largest share of connected devices [3], and connected cars, the fastest-growing application [3]. But also smart wearables [4], smart agriculture [5], environmental monitoring [6, 7], and industrial applications [8], to give some examples. In terms of devices connected to the Internet, the IoT has expanded to include an estimated number between 15.7 and 29.3 billion connected devices in 2023 [3, 9], and is expected to more than double in the next ten years [9].

Fabricating such a huge amount of devices inevitably comes with direct environmental impacts. All activities involved in the fabrication of electronics, from the extraction, refining, and transport of materials to the fabrication itself, have multiple impacts on the environment, such as resource depletion, pollution, freshwater usage, and the emission of greenhouse gases. Following the trend of connected devices reported in [3], the production of IoT devices after 2027 could emit 180 $MtCO_2$−eq per year [10], and this quantifies greenhouse gas emissions only. Moreover, this is unfortunately only part of the impacts of IoT devices because the other stages in their

life cycle, i.e., transport, use, and end of life, also have direct environmental impacts [11]. However, it has been shown that the carbon footprint of battery-powered consumer electronics such as smartphones and tablets is dominated by their fabrication because of their limited lifetime [12]. As engineers, we can work on extending the lifetime of IoT devices and also on different architectures of IoT devices that might be more suitable for reuse or repair. This can amortize their embodied impacts linked to their fabrication, and can also enable their reuse or repurposing to hopefully decrease the amount of fabricated devices in the future.

Most IoT devices are battery-powered or run on energy harvesting because they are often placed in locations where wiring them to the power grid would be either impossible or expensive. Therefore, minimizing their power consumption is key to reducing their environmental impacts by, e.g., increasing their lifetime, reducing the amount of battery replacements or even the capacity of the batteries for battery-powered devices [UJP1], or reducing the size of the photovoltaic panels for devices harvesting solar energy [13]. To identify how the power consumption can be reduced, we must go deeper into the technical details of the design of IoT sensor nodes.

1.1 IoT Sensor Nodes

Objects in the IoT are equipped with sensor nodes with the general architecture shown in Fig. 1.1. These sensor nodes give the object the ability to sense physical stimuli from its environment, process this information, and transmit it to the Internet wirelessly. Due to the heterogeneity of IoT devices [10], it is difficult to give a general power breakdown for them as it

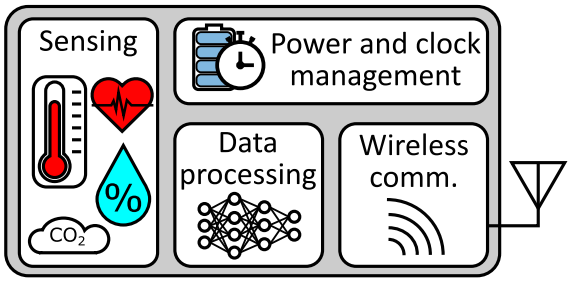

Fig. 1.1 Example of IoT node.

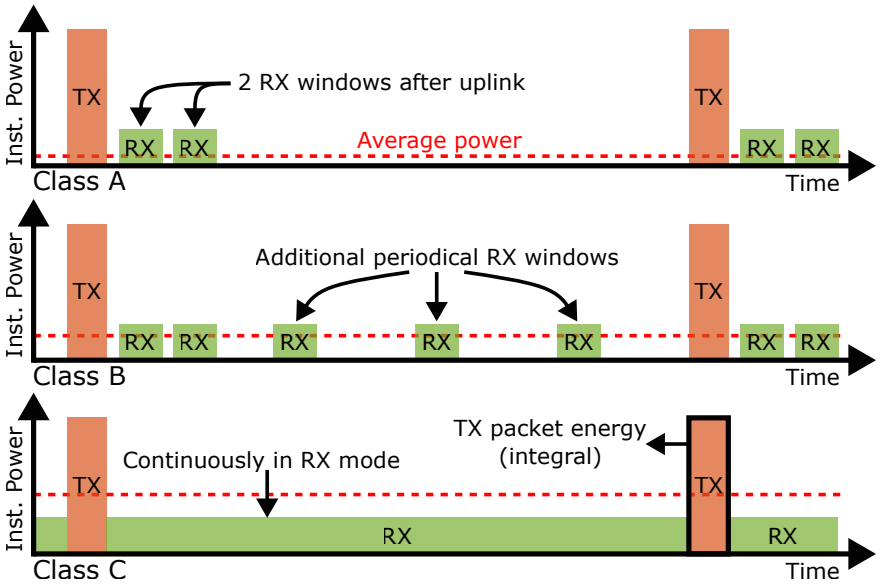

Fig. 1.2 Power profile of LoRaWAN devices representing the instantaneous power consumption over time. The average power consumption and the transmitted (TX) packet energy are highlighted.

depends strongly on the application. In environmental monitoring applications, data loggers typically require very low processing capabilities at the node level because they simply record parameters like temperature or humidity and transmit them to the cloud. In such nodes, power consumption is often dominated by the wireless communication [14], even though they use low-power wide-area networks (LPWANs) to achieve lower power consumption than cellular protocols by sacrificing data rate. LPWANs benefit from a longer communication range to monitor a larger area.

The study in [14] analyzes the power consumption of a data logger using LoRaWAN, one of the most common LPWAN protocols, using the proprietary LoRa modulation. For the case where a device transmits 16-byte packets once an hour in the longest-range configuration, the radio-frequency (RF) transceiver consumes more than 90 % of the instantaneous power consumption and almost 50 % of the average power consumption. In LoRaWAN, device classes are defined according to how often they use their receivers, as illustrated in Fig. 1.2. Class-A devices have only two short receive windows for downlink messages scheduled after an uplink packet is

transmitted [15, 16], resulting in a high link asymmetry; while class-B and C devices are in receive mode periodically and continuously, respectively. The link asymmetry of class-A devices does not allow, for example, the update of the nodes' firmware over the air, which is a useful technique to increase the lifetime of the nodes by limiting the risk of software obsolescence [17]. The device analyzed in [14] belongs to class A. Therefore, the contribution of the receiver to the average power in devices of classes B and C can increase significantly. These devices, however, are key to enabling more downlink capabilities, which can then be leveraged in environmental monitoring applications.

Types of Network

Star networks, as the one shown in Fig. 1.3(a), are the most commonly used network topology in LPWANs. They are unable to cover large areas in remote locations such as forest environments because the deployment of relatively large and relatively complex base stations is only feasible at the edges of the forest. The coverage of star networks, composed of devices communicating exclusively with the base station, is thus limited to the communication range of the latter. In LoRaWAN, this can go up to 10 km in rural areas [18], which is not enough to cover big forest areas, especially because the presence of trees affects the communication quality [19] and thus reduces this range.

Mesh networks, on the other hand, are usually composed of devices that are periodically in receive mode to relay packets in the network until reaching the base station, as shown in Fig. 1.3(b). These networks are

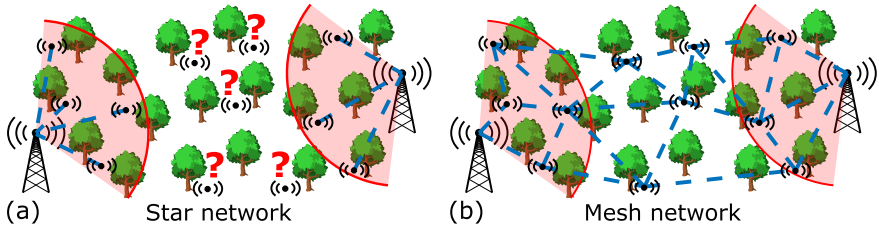

Fig. 1.3 Example of environmental monitoring application to highlight the differences between (a) star and (b) mesh wireless networks. The red circles represent the coverage of the base stations, and the dashed blue lines represent the available links between sensor nodes.

thus able to cover remote areas of any size. However, the fact that nodes enter receive mode periodically implies a synchronization of the network to compensate for possible drifts in the real-time clock of the nodes. Packets received for synchronization purposes could be more frequent than information packets in simple monitoring applications with low data rate, significantly increasing the power consumed by the transceiver. Enabling a more widespread use of mesh networks requires removing the need for synchronizing the network so the nodes use their transceivers only when an information packet needs to be relayed. Wake-up receivers have been proposed as a solution for this. Usually in parallel with the main receiver, wake-up receivers are continuously active to sense the wireless channel. When a packet is detected, the main receiver is enabled so it can demodulate the incoming packet. Wake-up receivers are designed to consume very little power so they can be left on continuously. To do so, either their performance is degraded, or their architecture is simplified, which is why they are not able to demodulate themselves the incoming packet. Nevertheless, the sensitivity needs to be close to that of the main receiver to maintain the communication range. Long-range wake-up receivers with high sensitivity are rare in the literature, and none use the LoRa modulation at the moment.

Reducing the power overhead of mesh networks compared to their star counterparts requires both optimizing the main receiver due to a more frequent use, and designing wake-up receivers with very low power consumption but similar sensitivity to the main receiver.

Reducing Power Consumption in RF Receivers

The high power consumption of the RF transceiver is linked to the high attenuation incurred in communicating through a wireless channel. In the case of the receiver, it must be able to demodulate the incoming low-power signal, targeting sensitivities for LoRa below −120 dBm to reach ranges between hundreds of meters to a few kilometers. Several impairments in the wireless channel can make this task more difficult, as they can introduce a stochastic variation of the channel attenuation due to shadowing from obstacles or dynamic interferences due to the presence or absence of blockers. Because of these stochastic behaviors, receivers are usually designed to guarantee a certain performance, e.g., a packet error rate (PER) of 1 % in the case of LoRa, in the worst-case channel conditions. However, these worst-case conditions are only encountered on rare occasions. In a forest

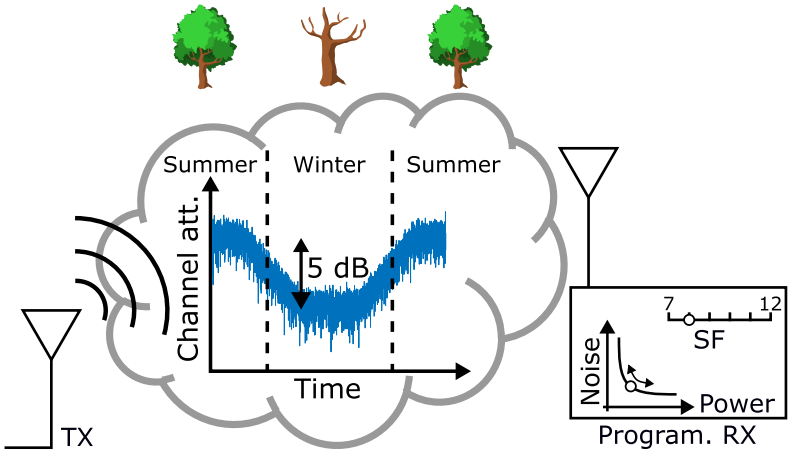

Fig. 1.4 Wireless channel conditions in a forest environment through the seasons. A programmable receiver (RX) is proposed to adapt to such conditions and avoid power overheads.

environment, for example, the channel attenuation strongly depends on the season, as depicted in Fig. 1.4. The presence of foliage on trees could add as much as 5 dB of attenuation [19]. Even during the same season, the channel attenuation still changes significantly due to the other impairments. Therefore, designing a receiver for operation in the rarely present worst-case conditions results in a power penalty because the analog performance of the RF front end, such as noise, linearity, and interference immunity, must be at its best continuously.

This power penalty can be avoided if the receiver adapts its performance in terms of its noise-power trade-off to the conditions of the wireless channel, as shown in Fig. 1.4. The conventional adaptive scheme in wireless communications consists in changing the modulation complexity of the signal to be more energy-efficient when good channel conditions are experienced [20]. LoRa includes such adaptability in the form of the spreading factor (SF), which determines the number of bits sent per symbol, as well as the duration of this symbol. A small SF results in a short symbol which needs a high signal-to-noise ratio (SNR) to be demodulated correctly. As the SF is incremented by one, the required SNR is relaxed by 2.5 dB [21], but the data rate decreases. By tuning the SF as a function of the received signal power, the transmitted packet energy can be adapted to the channel conditions. However, the power consumption of the receiver remains un-

changed, which is especially problematic for devices relying heavily on the receive mode, e.g., class B and C devices. Significant energy savings can only be obtained by reducing this power consumption [22]. Programmable receivers can result in instantaneous power savings of a factor 2× compared to static designs [23], or in average power savings of 30 to 75 % in the case of wake-up receivers [24]. With this goal, several programmable receivers or sub-blocks have been proposed, but they have either a power consumption in the tens of milliwatts that is prohibitive for IoT nodes using LPWANs [23, 25, 26], or a limited tuning range of only a few modes of operation [27]. Recently, it has been shown that achieving both low power consumption and a wide range of configurability is possible in mixers [28].

1.2 Research Questions and Thesis Outline

In this thesis, we explore how to convert the analog and mixed-signal circuits that compose the RF front end of LoRa receivers into highly programmable designs and the potential power savings this change can provide to an adaptive receiver by addressing the following research questions:

Question 1: *How can we design highly configurable circuits for IoT receivers that can dynamically tune their noise-power trade-off to adapt to the wireless channel conditions, thereby minimizing power consumption while preserving communication quality?*

Question 2: *Can we exploit the extremes of the noise-power trade-off of RF receivers to enable wake-up receivers able to detect signals with a degraded noise performance but with an ultra-low power consumption?*

This thesis contributes to addressing them at a level in between the system level and transistor-level circuits with the following outline. **Chapter 2** starts by introducing the basics of LoRa modulation and relevant characteristics of LoRaWAN. Specifications for LoRa receivers are given, and the proposed wake-up receiver scheme is detailed. Parts of this chapter's content have been presented in [CP3].

In **Chapter 3**, we give an overview of existing hierarchical design methodologies for analog and mixed-signal circuits to select the multi-objective

bottom-up methodology as the best fitting for an RF receiver. This methodology is based on optimizing the system based on Pareto fronts extracted from each of its sub-blocks. For extracting the Pareto fronts, we rely on an optimization of the target circuit architecture with the non-dominated sorting genetic algorithm (NSGA-II) [29]. We show how this algorithm can be used to design a highly programmable low-noise amplifier (LNA) from the initial design-space exploration to the final selection of optimal values and sizes for the components. However, following such a methodology for complex mixed-signal systems like phase-locked loops (PLLs) is not possible due to the prohibitively long time it takes to simulate them using SPICE [30]. Instead, we propose a post-silicon "hardware-in-the-loop" optimization method to solve multi-objective problems in mixed-signal systems with numerous degrees of freedom. We use this method to optimize the noise-power trade-off of a 64-MHz PLL based on a back-bias-controlled ring oscillator. The NSGA-II algorithm was run based on measurements of a 22-nm fully depleted silicon-on-insulator (FD-SOI) PLL prototype to find the Pareto-optimal configurations in terms of power and long-term jitter. The obtained Pareto front gives a range of power consumption between 2.7 and 5.7 µW, corresponding to a root mean square (RMS) long-term jitter between 88 and 45 ns. Whereas the simulation-based optimization would require more than a year using the genetic algorithm based on SPICE simulations, we conducted the post-silicon optimization in only 17 h. Parts of this chapter's content have been presented in [CP1].

Chapter 4 presents the first RF front-end prototype in 22-nm FD-SOI technology featuring the programmable LNA whose design is detailed in Chapter 3. The LNA is able to trade off a noise figure between 4 and 2.4 dB for a corresponding power consumption between 0.88 and 2.72 mW. It also includes (i) in-phase and quadrature mixers consuming 281 µW each with a 24.8-dB noise figure, and (ii) a fractional-N PLL consuming 370 µW. Using models for the missing blocks in the receiver chain, we estimate 46-% power consumption savings at the full receiver level, by adaptive tuning of the LNA, if the channel attenuation improves by 2 dB. Parts of this chapter's content have been presented in [CP2].

A second prototype is presented in **Chapter 5**. Implemented in 65-nm bulk CMOS technology, it features a complete highly programmable LoRa receiver with: (i) programmable LNA and mixers, (ii) a duty-cyclable programmable all-digital PLL, (iii) low-power filter and analog-to-digital converter (ADC), and (iv) a digital baseband (DBB) with a Fast Fourier Transform (FFT) hardware accelerator. Post-layout simulations exhibit be-

tween 1.4 and 3.6 mW of power consumption for a noise figure between 13.23 and 4.83 dB, respectively. With an estimated sensitivity of −138 dBm, we reach the same sensitivity as commercial LoRa modules, but reducing power consumption by more than 4×. The measured results are not available due to an error in the prototype, whose origin is also discussed in this chapter. This chapter's content is intended for publication [JP1].

Chapter 6 completes the description of the second prototype, presenting the wake-up receiver working in parallel with the programmable main receiver. Post-layout simulations show that the receiver can reduce its power consumption down to 1.5 mW in this mode, for a noise figure of 13.95 dB. This translates into a sensitivity of −126 dBm for a wake-up latency of 32 ms. This chapter's content is intended for publication [JP1].

Finally, **Chapter 7** concludes this thesis, discussing its main contributions and remaining challenges.

2

LoRaWAN: Introduction and Specifications for Radio Receivers

ALTERNATIVES to cellular standards for the IoT have emerged in the form of LPWAN standards. They aim to increase the range of communication and to reduce power consumption on the end devices by shifting complexity to the base stations at the cost of a decreased data rate. There are diverse ways to achieve this, and they depend on the particular LPWAN standard. The most popular are narrowband IoT (NB-IoT), LoRaWAN, and Sigfox [31, 32, 33]. NB-IoT is based on the long-term evolution (LTE) cellular protocol, but without some power-hungry features such as the monitoring of the wireless channel quality or the possibility of connecting to two base stations [31]. Moreover, the uplink is done over a single carrier with a narrow band of either 180 or 200 kHz instead of the orthogonal frequency-division multiplexing (OFDM) modulation used for downlinks [34]. Sigfox relies instead on communicating over an extremely narrow bandwidth of 100 Hz. This limits the noise power, allowing long ranges of communication, but also limits the data rate to 100 bps [31]. Finally, LoRa, the proprietary physical layer of LoRaWAN, utilizes the chirp spread spectrum (CSS) modulation, which uniformly spreads the signal energy across the entire bandwidth. During demodulation, the signal

is concentrated to a single frequency, while noise and interfering signals are spread over the whole bandwidth. This allows a correct demodulation at negative SNR and robustness against interference and multi-path propagation [31].

Even though NB-IoT outperforms Sigfox and LoRa in quality of service and latency [31], mostly because it operates on licensed bands with less interference, its higher power consumption due to a more complex modulation and its shorter range limit its usage on many IoT applications [31]. The choice between Sigfox and LoRaWAN is less clear and may thus depend on the application. It has been shown that LoRaWAN has a better coverage than Sigfox, but it is less energy efficient per transmitted bit of data because LoRaWAN devices must transmit additional packets to connect to base stations [33]. However, in the specific case of this thesis, LoRaWAN has the extra advantage that it already features the capability of adapting to the conditions of the wireless channel at the medium access control (MAC) layer level. The proposed adaptive receivers can thus further exploit the benefits of this adaptability. Furthermore, it is challenging to reconfigure a Sigfox network into a mesh topology due to the asymmetry between uplink and downlink, unlike LoRa.

In the rest of this chapter, we discuss in detail LoRaWAN and LoRa, as well as the specifications that radio receivers using this standard must fulfill.

2.1 Introduction to LoRaWAN

LoRaWAN is a MAC layer protocol defined by an open standard by the LoRa Alliance [15]. It is built on top of the patented LoRa physical layer owned by Semtech [35, 36]. For this reason, its details and specifications are not available. However, many studies have been performed in order to reverse engineer the mathematical description and possible implementation of the LoRa physical layer [37, 38, 39, 40]. In the following sections, we summarize these findings to provide an overview of LoRa.

2.1.1 LoRaWAN Frequency Plan

LoRaWAN operates on the industrial, scientific, and medical (ISM) unlicensed frequency bands. The exact frequency depends on the country; it

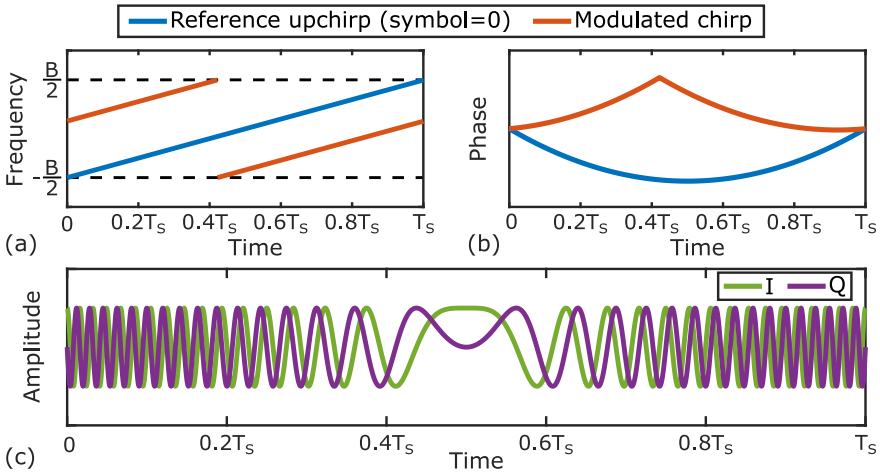

Fig. 2.1 (a) Instant frequency for 2 symbols in baseband. (b) Instant phase for 2 symbols. (c) In-phase and quadrature components of the reference upchirp.

operates around 868 MHz in Europe and India, around 915 MHz in the United States, Australia, and Asia, and around 490 MHz in China [41].

In Europe in particular, eight channels are defined with a separation of 200 kHz. They support communication using a bandwidth of 125 kHz, with a single one allowing a 250-kHz bandwidth. In the United States and Australia, downlink channels allow a bandwidth of 500 kHz [41].

2.1.2 The LoRa modulation

The LoRa physical layer uses the CSS modulation, which is based on linear chirps. These are signals whose instantaneous frequency changes linearly, as shown in blue in Fig. 2.1(a), sweeping a total frequency range equal to the bandwidth B. In LoRa, mostly upchirps are used, with increasing frequency over time, although downchirps are also needed for synchronization purposes. As can be seen in the in-phase and quadrature components of an upchirp shown in Fig. 2.1(c), CSS is a constant-envelope modulation.

The duration of the frequency sweep, i.e., the symbol period T_S, depends both on the bandwidth and on the spreading factor (SF) through the following expression:

$$T_S = \frac{2^{\mathrm{SF}}}{B}. \tag{2.1}$$

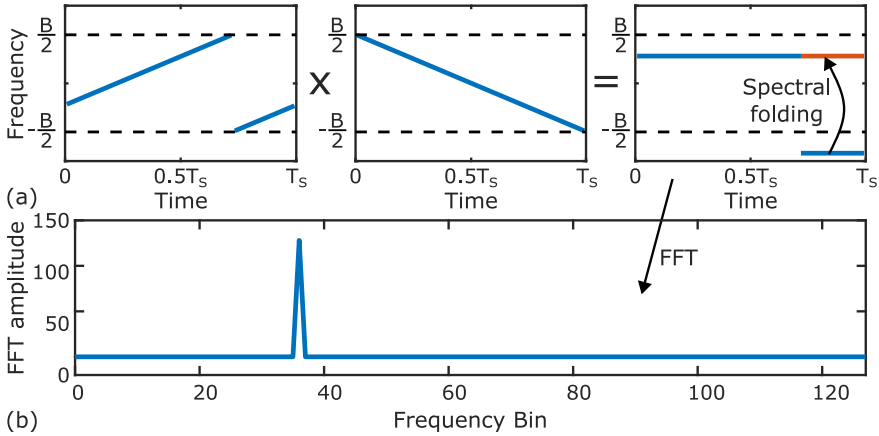

Fig. 2.2 (a) Illustration of the LoRa demodulation process in SF7. (b) 128-point FFT of the dechirped signal.

The SF, which can take integer values between 7 and 12, also corresponds to the number of bits encoded in the symbol. Hence, the symbol period is longer as more bits are transmitted with the same bandwidth. Different symbols differ by the starting frequency in the sweep. In baseband, once the instantaneous frequency reaches $B/2$, it continues the sweep at $-B/2$, as can be seen in Fig. 2.1(a). This is possible because the phase of the signal remains continuous even after drastic changes in the instantaneous frequency, as shown in Fig. 2.1(b). As the spreading factor increases, a longer communication distance is possible because the spreading gain is higher, but at the cost of longer transmission time and thus lower data rate.

The demodulation process of a LoRa symbol is illustrated in Fig. 2.2(a) for an SF of 7. The received signal, sampled at a frequency equal to its bandwidth, is dechirped by multiplying it with an unmodulated downchirp. This operation produces a signal containing two frequency components separated by B, caused by the shift from $B/2$ to $-B/2$ in the modulated upchirp. As the sampling frequency is B, spectral folding in the discrete-time domain turns this into a constant-frequency signal. The resulting frequency corresponds to the starting frequency of the modulated upchirp, which allows the transmitted symbol to be easily identified using an FFT, as shown in Fig. 2.2(b).

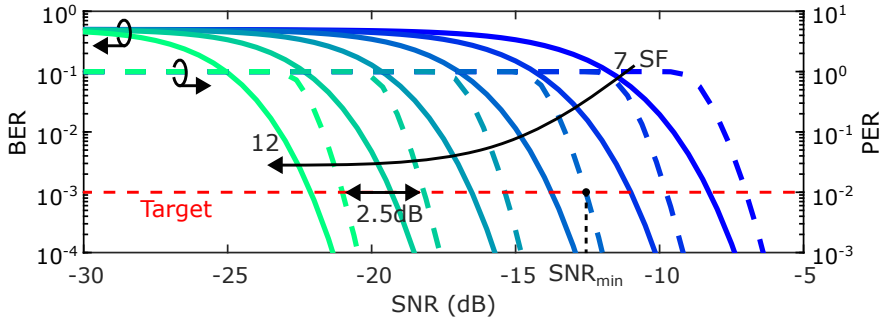

Fig. 2.3 Bit error rate (BER) (solid lines) and packet error rate (PER) (dashed lines) as a function of signal-to-noise ratio (SNR) for the LoRa modulation, considering an AWGN channel. The plot is shown for the six possible values of the spreading factor (SF).

2.1.3 Error Rates

Considering a wireless channel with additive white Gaussian noise (AWGN), an approximation of the theoretical bit error rate (BER) for the LoRa modulation as a function of the SNR was derived in [42]. The expression is not given here as it is out of the scope of this thesis. From the BER, the packet error rate (PER) can be derived with the following expression [43]:

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^{\mathrm{PL}}, \tag{2.2}$$

where PL is the payload length in bits.

Fig. 2.3 illustrates both the BER and the PER for a 59-byte payload as a function of the SNR, considering the six different spreading factors. As expected, the spread spectrum technique used in LoRa enables a correct demodulation at negative SNRs. The requirement for good-quality communication is usually to have a BER of 0.1 %. However, LoRa transceivers from Semtech define in their datasheets a requirement in terms of PER of 1 % [21] because it represents a stricter requirement on the minimum SNR than the BER, as depicted in Fig. 2.3. Table 2.1 summarizes the $\mathrm{SNR}_{\mathrm{min}}$ for different SFs. A packet with a 59-byte payload was considered because it is the maximum payload allowed for the three largest SFs [44].

Table 2.1 also shows the minimum required SNR for the commercial SX1276 transceiver developed by Semtech [21]. The theoretical approximation has an error below 1 dB, but, in both cases, the gain for increasing the

Table 2.1 Minimum required SNR for correct demodulation, both theoretical and for the SX1276 Semtech transceiver.

SF		12	11	10	9	8	7
Min. SNR [dB]	Theoretical	−20.8	−18	−15.2	−12.4	−9.6	−6.8
	SX1276 [21]	−20	−17.5	−15	−12.5	−10	−7.5

SF by one remains close to 2.5 dB. The fact that the minimum SNR improves when increasing the SF means that a higher path loss can be tolerated, and thus that the communication range can be extended.

2.2 Motivation for LoRa Programmable Receivers

The main trade-off in the LoRa modulation involving transmission time and data rate is controlled through the SF, as already discussed before. This trade-off has a direct impact on the energy consumption of transmitters, as it is proportional to the transmission time. When the wireless channel introduces fewer losses, the SF can be decreased to profit from a higher data rate and a lower energy consumption at the transmitter for a fixed communication range. However, this type of adaptation to the channel conditions brings only limited energy savings in classical non-programmable receivers. Designing programmable receivers, able to exploit their noise-power trade-off, would allow them to adapt to the channel conditions and also reduce their energy consumption when the channel introduces fewer losses by decreasing their power consumption.

Exploiting the power-noise trade-off requires modifying its noise figure (NF). The NF is an input-referred metric quantifying the degradation of the SNR due to the noise introduced by the receiver. It directly affects its sensitivity, i.e., the minimum input signal power required for correct demodulation of the signal. As discussed in the previous section, the requirement for a correct demodulation in LoRa is defined by the minimum SNR resulting in a PER of 1 %, that we denote here $\mathrm{SNR}_{\mathrm{min}}$. We can then calculate the sensitivity with the following expression:

$$\text{Sensitivity [dBm]} = 10\log(kTB) + NF + SNR_{\mathrm{min}}, \tag{2.3}$$

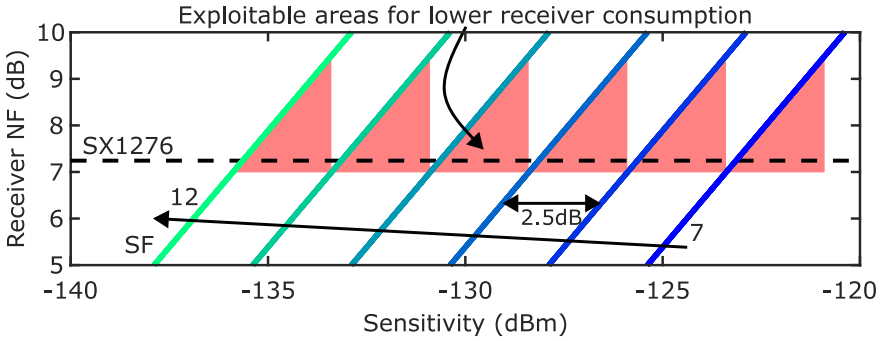

Fig. 2.4 Noise figure (NF) required at the receiver to attain the sensitivity of LoRa as a function of the spreading factor (SF), considering a 125-kHz bandwidth. The dotted line represents the mean NF of the SX1276 chip from Semtech.

where k is Boltzmann's constant, T is the absolute temperature in Kelvin, and B is the bandwidth of the signal of interest. Fig. 2.4 shows the NF required to reach the sensitivity of LoRa at the most-used bandwidth in Europe of 125 kHz. The dotted line indicates the mean NF for all SFs of the SX1276 chip [21]. This NF was calculated from the reported sensitivities using (2.3) and the SNR_{min} reported in Table 2.1.

As the SF can only be decreased when the SNR improves by 2.5 dB, any reduction in the channel losses within this range does not allow for saving energy on the transmitter side. On the receiver side, this translates into a power overhead, as the NF becomes lower than needed. Relaxing the NF in such situations would result in power savings at the receiver. This is highlighted in Fig. 2.4. Even though the transmitter could decrease the transmit power when the SNR improves within 2.5 dB, it does not have the information on channel quality that the receiver has. Any adaptation on the transmitter side depends on a specific packet exchange with the receiver, while the receiver can adapt its performance independently without the need to inform the transmitter. With the prototypes described in Chapter 4 and Chapter 5, we will show how we propose to dynamically change the NF of the receiver and quantify the power that can be saved by implementing programmable receivers.

2.2.1 LoRa Wake-Up Receiver

The proposed wake-up receiver scheme was conceived in collaboration with Pol Maistriaux (ICTEAM - UCLouvain), who is the main author of the resulting conference paper [CP3].

The idea of a programmable receiver able to dynamically change its noise-power trade-off could be further extended to enable long-range wake-up receivers leveraging the spread spectrum technique used in LoRa (CSS).

Wake-up receivers are ultra-low-power receivers designed to be continuously in reception mode and able to simply detect the presence of an incoming packet or wake-up signal (WuS) at the same sensitivity as the main receiver to achieve the same communication range. The main receiver, usually in parallel, is then enabled to demodulate the packet at a higher power consumption. We propose instead to merge both into a single programmable receiver, able to push its power-noise trade-off towards the minimum in power consumption. This comes at the cost of an increased NF that limits the achievable range in wake-up mode.

The digital baseband (DBB) processor at the end of the receiver chain provides a processing gain to compensate for this increased NF by integrating the signal over time, effectively trading off latency for range. Conventional DBB processors are usually kept as simple as possible to avoid power consumption overheads [45, 46], limiting the processing gain. To achieve the same sensitivity as the main receiver, the processing gain can be increased at the cost of a more complex DBB processor. This could be the key to enabling long-range wake-up receivers, currently rare in the literature.

Conventional DBB processors of current wake-up receivers in the literature rely on the correlation of the incoming signal with the expected WuS. However, to avoid the resource-intensive multiplication operations, the correlation is performed on a single bit so that trivial exclusive-or (XOR) operations can be used instead [45]. This correlator can be referred to as an M-out-of-N (MoN) detector since M bits out of the N-bit WuS must match to declare a detection. This method works well for simple modulations like binary frequency-shift keying (BFSK) [47, 48], where the transmission of zeros and ones is done at two distinct frequencies. The RF front end, in this case, can have a simple architecture based on envelope detectors that extract the energy at these two frequencies. A comparator decides which frequency has more energy to demodulate the sent bit. As the two compared values are affected by independent noise sources, there is a 3-dB penalty on the minimum SNR needed to detect the sent bit correctly because the noise

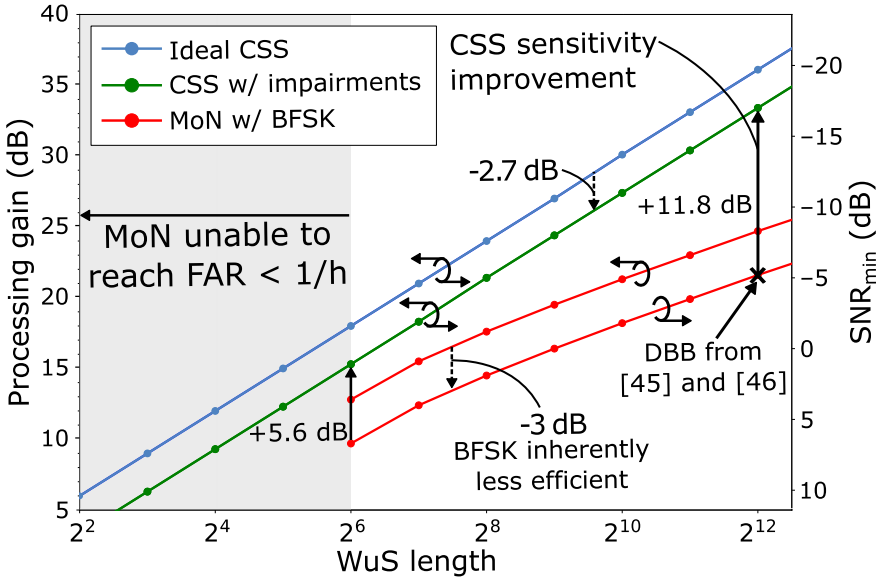

Fig. 2.5 Processing gain and SNR_{min} with respect to the WuS length for a false-alarm rate (FAR) of one per hour.

power is doubled. In the context of wake-up receivers, the SNR_{min} is no longer defined by the PER but by the false-alarm rate (FAR) of less than one per hour. Fig. 2.5 shows the processing gain and the SNR_{min} for different WuS lengths of the one-bit correlator with a BFSK modulation. We can see that it is not possible to reach the same range as LoRa with this correlator, as the SNR_{min} is several decibels below the ideal CSS modulation due to the simplicity of the used correlation.

Another alternative to avoid the resource-intensive multiplication operations needed to compute the correlation is to perform it in the frequency domain. While most modulation schemes require two FFTs to move to and from the frequency domain, the CSS modulation used in LoRa only requires a single FFT. Since the dechirping operation done in the LoRa demodulation results in a single tone, it can be integrated with a single FFT. Examples of the FFT output are shown in the illustration of the proposed wake-up receiver scheme depicted in Fig. 2.6 for input chirps with different SNRs and random sampling-time offsets (STOs) to simulate a lack of synchronization. It can be observed that this lack of synchronization only shifts the FFT maximum to a different bin. This means that CSS inherently performs a

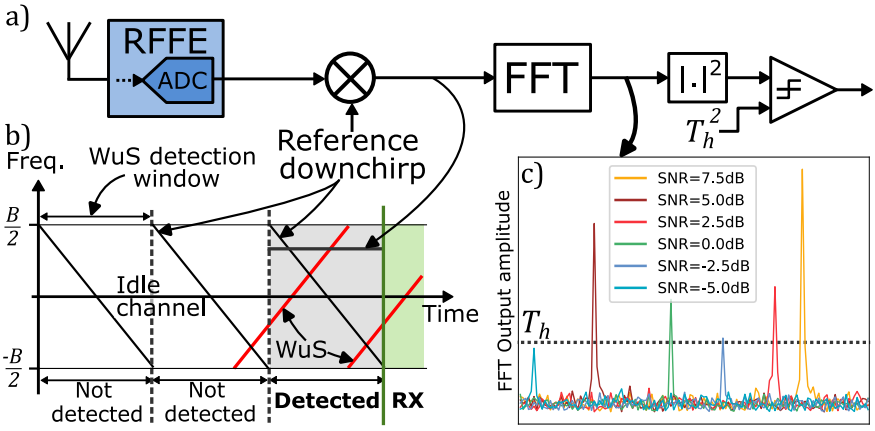

Fig. 2.6 (a) Wake-up receiver scheme using a detection based on the CSS modulation with a fixed threshold. (b) Overview of baseband signals at different points of the wake-up receiver, up to the dechirped signal. The red signal represents the incoming signal. (c) Output amplitude of the FFT for different SNRs and with random sampling-time offsets (STOs).

cyclic correlation. Therefore, a cyclic shift of the reference chirp, used here as a WuS, by any number of samples does not affect the magnitude of the FFT outputs. Even though the CSS detection scheme works on a buffer of N samples to perform the cyclic correlation rather than continuously, repeating the WuS successively at the transmission ensures a complete WuS will be acquired, as illustrated in red in Fig. 2.6(b). This repetition is often already required by modern wake-up receivers, which use duty-cycling to reduce their average power consumption while increasing their latency. Moreover, the cyclic correlation inherent to CSS only requires the acquisition of N samples with the RF front end. Its duty-cycling scheme is thus twice as efficient as that of a linear correlator, which must acquire 2N samples [47]. The resulting latency for this wake-up receiver is thus the time it takes to capture these N samples, corresponding to the symbol period in (2.1).

Fig. 2.5 shows the major sensitivity improvement of CSS, with an $\mathrm{SNR}_{\mathrm{min}}$ of −19.7 dB for a WuS length of 4096 compared to −5.2 dB for MoN. This 14.5-dB gain corresponds to a range in free-space path loss that is five times longer. Two points must be noted. First, due to STOs and carrier-frequency offsets (CFOs), the dechirped signal can be located in between two FFT bins, distributing its energy between them. When subject to the worst-case CFO and STO, a 2.7-dB degradation is observed, reducing the $\mathrm{SNR}_{\mathrm{min}}$ improve-

ment to 11.8 dB. Second, since the WuS is the reference chirp, common to all receivers, the addressing capabilities are lost, even though further work could exploit multiple CSS symbols to encode an address. Nevertheless, the processing gain of the CSS detection scheme reaches SNR_{min} values close to those of the LoRa demodulation, which means that the resulting sensitivity of both receivers will be close. Further details on the prototype of a wake-up receiver will be provided in Chapter 6.

2.3 Specifications for Frequency Synthesizers in LoRa Receivers

We have already defined most of the specifications needed to design a LoRa receiver, e.g., carrier frequency, bandwidth, SNR. However, the specifications for the frequency synthesizer have not been discussed yet.

One of the important things to specify for frequency synthesizers is the needed precision of the generated carrier frequency. Commercial LoRa transceivers from Semtech report no sensitivity degradation for a maximum tolerated frequency offset between transmitter and receiver, i.e., CFO, of 25 % of the signal bandwidth [21]. This is in part confirmed by the low-complexity LoRa synchronization algorithm proposed in [40]. Their algorithm shows that a CFO up to 16.67 % of the signal bandwidth has no significant impact on the receiver sensitivity, but between 16.67 % and 25 % of the bandwidth, a 1-dB impact is observed. This happens because large values of CFO lower the accuracy of the STO estimation, which happens earlier than the CFO estimation in their proposed algorithm. A wrong estimation of these offsets results in a synchronization error, after which the receiver is unable to demodulate the packet payload.

The other important specification relates to the noise around this carrier frequency. This is quantified by the phase noise ($\mathcal{L}(\Delta f)$), defined as the ratio of the signal power in a 1-Hz bandwidth at an offset Δf from the carrier, divided by the total power of the carrier [49]. The phase noise is measured in dBc/Hz. Wireless standards define stringent requirements on the phase noise, usually needing inductor-capacitor (LC) oscillators [50]. These oscillators, however, are hard to integrate because of the large area occupied by inductors with the high inductance needed at sub-GHz RF frequencies. Ring oscillators are often preferred instead because of their simplicity, compactness, and low power [51], but they exhibit significantly higher phase noise. Nevertheless, it has been proven that a phase noise

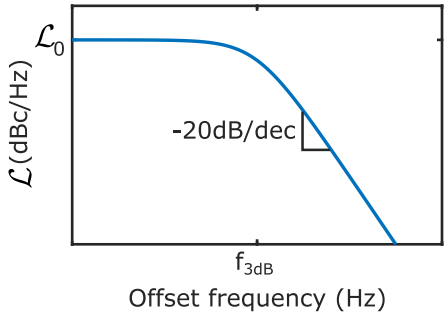

Fig. 2.7 Phase noise profile of a PLL.

as high as −78 dBc/Hz, corresponding to a typical performance of ring oscillators, is good enough for Bluetooth radios [50], which use a simple frequency-shift keying modulation.

LoRa being a proprietary physical layer, there exists no official requirement for the phase noise. The SX1276 transceiver from Semtech includes a PLL with a phase noise of −122 dBc/Hz at a 1-MHz offset [21]. Knowing that ring oscillators can reach a minimum of −165 dBc/Hz of the following figure of merit (FoM) [51]:

$$\text{FoM} = \mathcal{L}(\Delta f) + 20\log_{10}\left(\frac{\Delta f}{f_0}\right) + 10\log_{10}\left(\frac{\text{Power}}{1\,\text{mW}}\right), \tag{2.4}$$

a ring oscillator would need to consume at least 37.76 mW to reach such a low phase noise at 868 MHz. This is evidently too high for an IoT node. What we propose instead is to evaluate the impact of phase noise on the LoRa demodulation to verify if ring oscillators can be used, as is the case for Bluetooth.

For this evaluation, we modified the MATLAB testbench developed by Mathieu Xhonneux, a colleague who conceived the low-complexity LoRa synchronization algorithm in [40], to include the phase noise of the receiver's local oscillator. This phase noise is modeled using the typical phase noise profile of a PLL [49] shown in Fig. 2.7. It features a flat profile until its bandwidth (f_{3dB}) and then a −20-dB/decade slope corresponding exclusively to thermal noise. Flicker noise is not considered. Considering a 32-MHz reference frequency for the PLL, a typical value for its bandwidth is 1 MHz because it must be at least 10× lower than the reference frequency for stability issues [52].

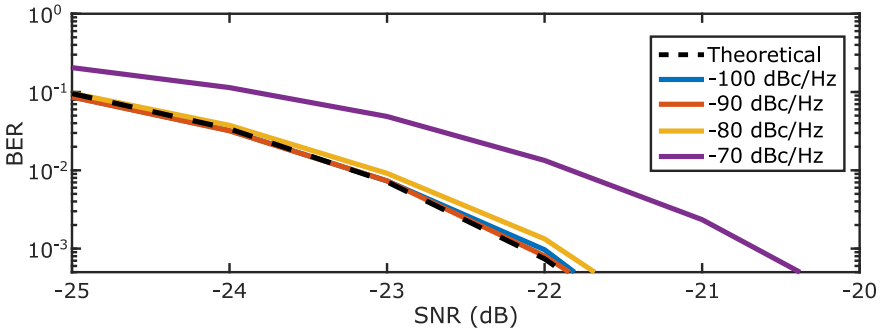

Fig. 2.8 Impact of phase noise on the BER of LoRa in SF12 and a 125-kHz bandwidth. The shown phase noise values correspond to an offset of 1 MHz. The phase noise profile of a PLL with a 1-MHz bandwidth was used.

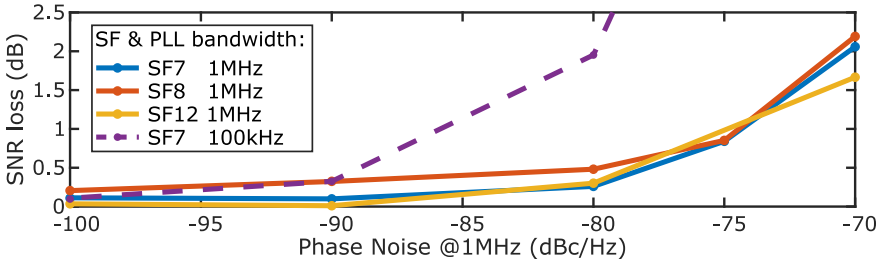

Fig. 2.9 Loss in SNR due to a PLL-like phase noise profile. Multiple SFs were tested for a 1-MHz PLL bandwidth. Two different PLL bandwidths were tested in SF7.

The testbench uses Monte-Carlo simulations to evaluate the error rates of the communication. We modified it by using the existing MATLAB system object *comm.PhaseNoise*, included in the communications toolbox, to add phase noise to the baseband signals. The impact of phase noise from a PLL with a 1-MHz bandwidth on the BER is shown in Fig. 2.8 for 5000 packets, an SF of 12, and a bandwidth of 125 kHz. Values of phase noise up to −90 dBc/Hz at a 1-MHz offset have a negligible impact on the BER compared to the theoretical curve. Significant impacts appear somewhere between −80 and −70 dBc/Hz.

To better quantify the impact of phase noise, we define the SNR loss as the additional SNR needed at the receiver to obtain a 0.1-% BER when phase noise is added, compared to the theoretical curve. This way, the impact of both the SF and the PLL bandwidth can be studied more easily. These

results are shown in Fig. 2.9 for 5000 packets again. We can see that the SF does not significantly impact the results. The slight increase in SNR loss at SF8 could be explained by numerical errors due to a limited amount of simulated packets. For the three simulated SFs, we see again that the SNR loss starts to increase after −80 dBc/Hz for a PLL with a 1-MHz bandwidth, confirming the trend observed in Fig. 2.8.

The choice of PLL bandwidth has a higher impact on the BER, as the SNR loss starts to increase for phase noise levels at 1 MHz which are 10 dB lower when the bandwidth is decreased 10×. As the phase noise decreases with a 20-dB/decade slope, the PLL with lower bandwidth has a close-by phase noise, i.e., at low offset frequency, which is 10 dB higher. This shows that phase noise closer to the carrier is the parameter with the most impact.

As the minimum theoretical power of ring oscillators running at 868 MHz is only 2.38 and 23.83 µW to achieve a phase noise of −80 and −90 dBc/Hz, respectively, we can conclude that we can benefit from the compactness and lower power of ring oscillators to implement a LoRa modulation in IoT nodes. These oscillators are included in the prototypes presented in Chapters 4 and 5.

2.4 Conclusions

This chapter introduced the LoRaWAN protocol and gave an overview of the advances in the literature for reverse engineering its proprietary physical layer LoRa. This provides the specifications for the prototypes that will be presented in Chapters 4 and 5. In this sense, we provided an important missing specification in terms of the required phase noise at the receiver. We showed that ring oscillators are theoretically able to provide the required phase noise under an acceptable power budget, allowing us to use them instead of LC oscillators to benefit from their compactness and lower power consumption.

We also illustrated how a programmable receiver exploiting its noise-power trade-off could offer an extra layer of adaptation to the wireless channel conditions in addition to the adaptation already implemented in LoRaWAN with the spreading factors. The methodology followed to design such a receiver is presented in Chapter 3, while the prototypes later presented in this thesis will quantify the potential power savings of implementing the adaptation described here.

Following the same idea of reducing the power consumption of RF receivers, we explained the working principle of the first LoRa-compatible long-range wake-up receiver. By implementing a more complex DBB processor in the receiver with an acceptable power overhead, we can leverage the CSS modulation to obtain a significantly higher processing gain compared to the simplified correlation performed in most state-of-the-art wake-up receivers. This increased processing gain could be exploited to allow a deterioration of the noise performance of the receiver, hoping to reduce its power consumption without affecting its sensitivity. Chapter 6 will present an implementation of this wake-up receiver.

3

Analog and Mixed-Signal System Optimization

The design of complex analog and mixed-signal systems has historically relied on hierarchical optimization methods to simplify the design process. As the name suggests, these methods are based on subdividing the system into smaller functional blocks across multiple levels of hierarchy. Each block is designed independently before being integrated with the other blocks.

This subdivision is mainly done because of the highly accurate but computationally intensive SPICE simulations used during the design of the system. While SPICE simulations are suitable for small designs, like the individual sub-blocks, they become impractical for the full system due to the significantly long simulation time required. Instead, mixed-signal simulations are used to reduce simulation time at the cost of decreased accuracy. These simulators use models written in languages such as Verilog-A or Verilog-AMS, which reproduce the behavior of the sub-blocks at the system level.

In the rest of this chapter, we explore the different existing hierarchical design methods to choose the best fit for the programmable receiver. Then, the chosen approaches are described in detail.

3.1 Analog and Mixed-Signal Design Methodology

3.1.1 Hierarchical design methodologies

Existing hierarchical methods can be sorted into three categories depending on how they use behavioral models during the design. First, bottom-up methods design the individual sub-blocks at the transistor level before assembling them and verifying the system-level performance. It is at this last stage that the behavioral models replace the flat simulations of the complete system [53, 54], but they can lead to non-functional or sub-optimal systems because they do not consider system-level aspects in the design of the sub-blocks. Second, top-down methods derive the requirements of each sub-block from the system specifications to ensure its functionality [55, 56, 57]. The behavioral models are used to make this translation, which can sometimes lead to unachievable specifications for certain sub-blocks or overconstraints on others, leading to additional power consumption. Finally, hybrid methods, like the concurrent constraint-driven methodology [58], look to solve the limitations of the previous two approaches. In this method, the block-level parameters are optimized simultaneously with the system-level specifications for each sub-block. A numerical optimizer minimizes a cost function that not only includes the desired system-level performance metrics but also the difference between the modeled and the simulated behavior of the sub-blocks.

Close to the bottom-up approach, the MUlti-Objective Bottom-Up (MUBU) methodology [53] is particularly useful for designing the adaptive receiver introduced in Chapter 2. This methodology is based on the extraction of Pareto fronts of each individual sub-block. A Pareto front represents a set of non-dominated solutions to a multi-objective optimization problem. A solution is considered non-dominated if no other solution exists that improves one objective without degrading at least one other. These fronts approximate the trade-offs inherent in each sub-block and can be exploited in system-level behavioral models to identify the feasible specifications that enable the system to meet its desired performance. The system-level optimization can then combine different sub-block designs to achieve a system-level trade-off. By implementing all the designs represented in the sub-blocks' Pareto fronts, we can obtain a system able to adapt to the run-time conditions by dynamically exploiting this system-level trade-off.

3.1.2 Multi-Objective Genetic Optimization

The NSGA-II [29] was selected as the multi-objective optimization algorithm needed in the described methodology for designing an adaptive system to extract the Pareto fronts of each sub-block. This algorithm was first applied to the sizing of an analog circuit by Rémi Dekimpe, a colleague who showed that the algorithm performed well for this task [59]. Some of the reasons that make this algorithm successful in analog sizing are that it does not rely on the derivative of the function we want to optimize, nor does it try to estimate it, and that it can handle circuit parameters that are continuous or discrete. Additionally, since it can handle design constraints, the algorithm can verify performance metrics that do not require optimization but must satisfy certain limits to ensure functionality.

Fig. 3.1 shows the block diagram of the genetic algorithm along with a simplified explanation of each step using colored circles as an analogy. The algorithm is based on the evolution of a population of individuals throughout an arbitrary number of generations. In the case of the sizing of an analog circuit, an individual represents the circuit implementation with a certain set of parameters. These parameters can be continuous, like the resistance or capacitance of a component, or the size of a transistor, and they can also be discrete, like the number of fingers of a transistor, or the number of parallel circuit elements. In the colored-circles analogy, these parameters are presented by a color in one of the three sections of an individual. A set of individuals is what we call a population.

For the algorithm initialization, an initial population is usually created randomly. Not every individual can be a part of a population, as it must respect a set of constraints. These constraints are circuit performance metrics, but rather than being minimized as the optimization objectives, they must simply respect a condition set by the designer. Therefore, the algorithm iterates over this initialization step until at least three valid individuals are found among the randomly created ones. The designer can also provide an initial population.

The second step is the fitness evaluation. It is during this step that the circuit performance metrics are extracted, usually by means of SPICE simulations. This is done during the initialization to find valid individuals for starting the algorithm, and then at each generation to evaluate the objectives and constraints of the new population. To save time, simulations are done sequentially, and individuals violating any constraint are directly discarded. Simple and thus fast simulations are done first, so individuals

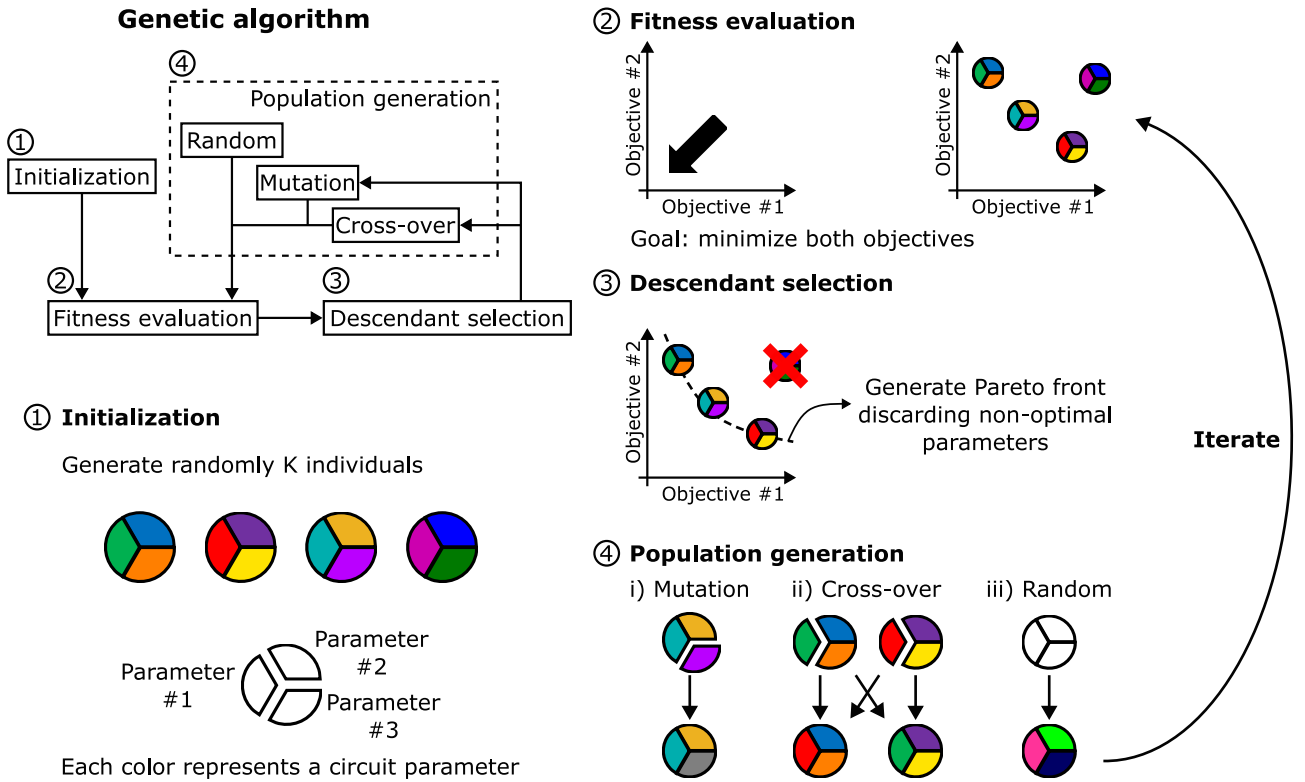

Fig. 3.1 Block diagram of the genetic algorithm and description of each step. An analogy using colored circles explains the steps of the algorithm.

violating the most basic constraints can be discarded early and avoid including them in more complex and time-consuming simulations for the more advanced constraints. Taking an amplifier, for example, calculating the gain first and discarding individuals with low or negative gain, saves time later when extracting the offset voltage, needing Monte-Carlo simulations for the transistor mismatch.

Then comes the descendant selection. In this step, only the best-performing individuals respecting all constraints are selected, following an elitist approach. Individuals are compared not only to the members of their generation, but also to the Pareto front of the previous generation to generate the new Pareto front so that good individuals are kept across generations. More precisely, individuals are sorted according to two attributes: their non-domination rank and their local crowding distance [29]. To obtain the non-dominated rank, the individuals are separated into non-dominated fronts. A non-dominated front contains individuals that perform better than the rest in at least one of the optimization objectives. The front of the lowest rank is extracted from the whole population, and the front of the next rank is extracted from the remaining individuals. The other attribute, called the crowding distance, measures the largest cuboid enclosing the individual without including any other. The population is then sorted first according to the non-dominated rank to prioritize the optimal points, and then according to the crowding distance to keep points in regions with a lesser number of individuals. From this sorted list, the first N individuals are chosen to generate the Pareto front of the current generation.

Finally, for creating the next generation, three independent processes are used: i) mutation, where a few parameters of an individual are modified to create a new one; ii) cross-over, where two individuals mix their parameters to create an offspring; and iii) random generation. The first two processes allow the search for optimal points in the vicinity of already found points, while the random generation helps reduce the risk of falling into local minima.

This new generation then goes to the fitness evaluation step, and the algorithm iterates until a stopping criterion is met. Quantitative stopping criteria exist for this algorithm, like the hypervolume metric [59], but for simplicity, we stop the algorithm after a certain number of generations have passed.

3.1.3 Challenges of hierarchical design methodologies for complex mixed-signal systems

The main disadvantage common to all existing hierarchical design methodologies is that they rely on behavioral models of the different sub-blocks to decrease the simulation time compared to a transistor-level SPICE simulation. These models, however, can fail to reproduce precisely the behavior of complex blocks. The presence of circuit non-idealities and their complex interactions forces the models to be complex. Additionally, some non-idealities cannot be known exactly at design time and are thus not modeled, such as the process, voltage, and temperature (PVT) variations.

An alternative to avoid the long SPICE simulation times when designing the system would be to rely on measurements of the fabricated circuit, by doing a final post-silicon optimization. Post-silicon tuning is already a common technique used for analog circuits like current references [60, 61, 62] or relaxation oscillators [63, 64]. The main objective is to achieve the desired performance despite the fabrication process variations. The optimization of such circuits consists in trimming a component to calibrate the nominal current or frequency, or to minimize their temperature dependence. Both objectives are usually reached by independently modifying two degrees of freedom of the circuit [60, 63, 64]. Therefore, simple optimization algorithms, e.g., an exhaustive search, are sufficient for this task.

Post-silicon tuning of more complex mixed-signal systems like PLLs has also been proposed in [65], but the authors focus on improving the jitter of a PLL optimized at the design stage by compensating for a single non-ideality due to process variations. Generalizing these methods to multi-objective problems in such complex systems is anything but trivial. For instance, the jitter and the power consumption of a PLL must be co-optimized because these metrics are in direct conflict. Although the jitter of PLLs can be modeled relatively accurately and therefore optimized [66, 67], the trade-off between power and jitter is less straightforward, especially in the presence of circuit non-idealities [52], and it depends on numerous degrees of freedom. This makes an exhaustive search for finding the optimal points very time-consuming.

For extracting the Pareto front of the PLL, the key block in the RF receiver introduced in Chapter 2, we will use the same genetic optimization described in Section 3.1.2, but the SPICE simulations used in the performance evaluation will be replaced by measurements of the fabricated prototype.

In the rest of this chapter, we will detail the methodology used to design

the prototypes of adaptive receivers presented in Chapters 4 and 5. For this, two blocks will be taken as examples. First, an LNA to illustrate how we can extract the Pareto front of a simple analog block using SPICE simulations exclusively. Then, a PLL will be used to detail the novel methodology of a genetic optimization algorithm with the hardware in the loop.

3.2 SPICE-Based Genetic Optimization

In this section, we are going to show how the multi-objective genetic optimization can be used to size a configurable analog circuit. An RF LNA will be used as an example because of its simple architecture, which limits both the number of parameters in the optimization and the simulation time.

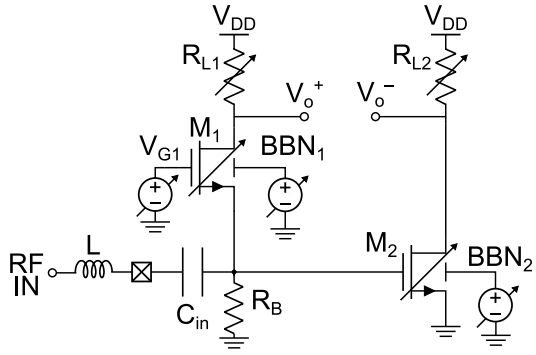

Fig. 3.2 Schematic of the balun noise-canceling LNA, highlighting its degrees of freedom. It is implemented using LVT RF transistors of the Global Foundries 22-nm FD-SOI technology and supplied at 0.8 V.

The schematic of the LNA is shown in Fig. 3.2. It is going to be implemented with low-threshold-voltage (LVT) RF transistors of the Global Foundries (GF) 22-nm FD-SOI technology and supplied at 0.8 V. The architecture is based on the balun noise-canceling architecture [68], modified with an off-chip inductance to reduce its bandwidth. It integrates the balun function, meaning the single-ended input signal is transformed into a differential output with the common-gate branch generating the positive output and the common-source branch generating the negative one. It is also called a noise-canceling architecture because transistor M_2 is sized to eliminate the noise introduced by M_1. More details about the architecture are left for the

next chapter. Instead, we will focus on how to transform this circuit into a programmable one able to change configurations dynamically at run time.

Fig. 3.2 highlights the degrees of freedom available in the sizing of the LNA. They include the load resistance of both branches, the nMOS back-bias voltage (BBN) of both transistors, the gate voltage of M_1, and the width of both transistors. There are three ways to change the width of the special RF transistors available in this technology, which have a minimum of eight fingers. We can thus change the width and number of fingers, as well as their multiplicity, i.e., the amount of transistors connected in parallel. The parameters we chose to keep fixed are: i) the length of the transistors, kept to their minimum value of 18 nm because in practice it is less straightforward to vary the length with switches than the width; ii) the input passive components, limited by commercially available inductors to 15.5 nH resonating at 868 MHz with a 2.17-pF metal-oxide-metal (MOM) capacitor; and iii) the bias resistor R_B to reduce the number of switches connected to the input signal. Its resistance was fixed to 415 Ω, a value found by preliminary simulations of the LNA. Given the relatively low resistance, it is implemented with an unsilicided N+ diffusion resistor in the thin active SOI film.

Having identified the degrees of freedom, we can begin using the genetic algorithm for sizing the circuit. The optimization objectives are the minimization of the power consumption and the NF, and the maximization of the linearity, measured with the input third-order intercept point (IIP3). Four performance constraints were imposed, as shown in Table 3.1. The gain, gain imbalance, and phase imbalance check for a correct DC point and AC response, as well as a correct balun behavior, and the S_{11} checks for a correct impedance matching. As described in [59], these constraints were

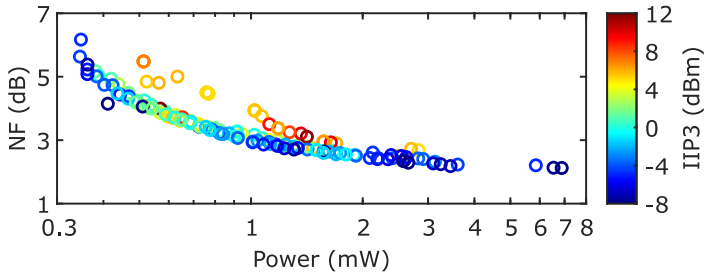

Fig. 3.3 Pareto front of the balun noise-canceling LNA after 50 generations of the genetic optimization algorithm.

Table 3.1 Performance constraints for the LNA.

Performance	Constraint
Gain	$\geq 15\,\text{dB}$
S_{11}	$\leq -10\,\text{dB}$
Gain Imbalance	$\leq 2\,\text{dB}$
Phase Imbalance	$\leq 10°$

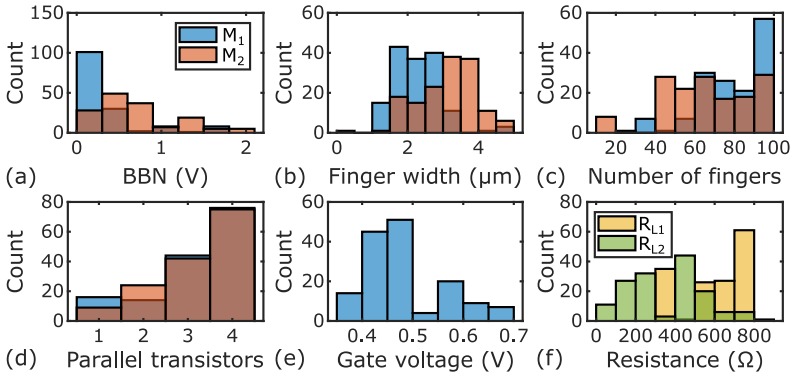

Fig. 3.4 Histograms of the optimal circuit parameters of the LNA.

verified in separate SPICE simulations, ordered from the fastest to the most time-consuming. Directly rejecting sets of parameters that violate one of these constraints helps to significantly reduce the total optimization time.

The Pareto front obtained after 50 generations is shown in Fig. 3.3. The individuals extend over a large range for each of the three optimization objectives. It must be noted that the Pareto front for the IIP3 is not as smooth as the noise-power Pareto front because of how this metric is handled in the optimization algorithm. To help in its convergence, this third metric is handled as a constraint. The algorithm is run several times, and the IIP3 constraint becomes stricter from one run to another. The individuals plotted in Fig. 3.3 thus correspond to the superposition of several populations. The range of the optimal circuit parameters is shown in Fig. 3.4. At this stage of exploration, it is important to check that the parameters are not limited by the range imposed by the designer, except if physical limits are reached. In this example, we can see in Fig. 3.4(e)-(f) that the gate voltage of M_1 and the load resistances are well distributed in their ranges, and the most used values are not at the limits. However, this is not the case,

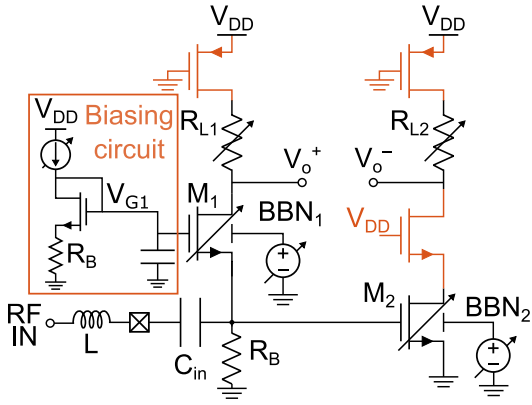

Fig. 3.5 Schematic of the balun noise-canceling LNA with the switches needed for making it configurable. The degrees of freedom are highlighted, and the biasing circuit is included.

for example, for the number of fingers of M_1. The technology limits this number to 100, corresponding to the most used value for this parameter in the individuals of the Pareto front. Unfortunately, there is nothing we can do about this, apart from adjusting other parameters to avoid being limited by this technology constraint.

Having identified the range of the circuit parameters allowing to have a large range of performance for the LNA, we can continue with the implementation of the tuning knobs that will change the configuration of the circuit. For this, we first study the impact of the three needed switches, shown in Fig. 3.5, as well as the biasing circuit that generates the gate voltage of M_1. It consists of a replica of the left branch of the LNA, downsized to save power, and biased with a current source that will later be a current digital-to-analog converter (DAC).

Several iterations of the genetic algorithm were used to find the optimal size of the switches. The optimal width of the pMOS switches, implemented with RF high-threshold-voltage (HVT) transistors, converged to 40 µm. However, the width of the nMOS switches, implemented with RF super-low-threshold-voltage (SLVT) transistor, varies depending on the width of M_2. Other parameters were also fixed to ease the implementation of the LNA. The finger width and multiplicity of M_1 and M_2 were fixed to 1.85 µm and four, and 2 µm and 3, respectively. The width of these transistors can now be changed by the number of fingers only. The choice of

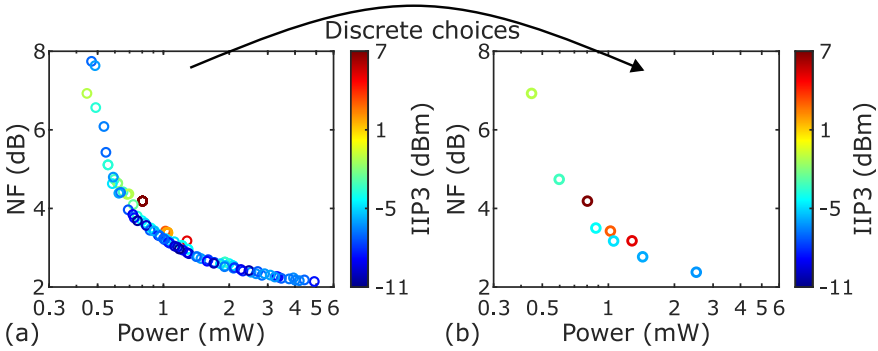

Fig. 3.6 (a) Pareto front of the balun noise-canceling LNA after 100 generations of the genetic optimization algorithm, including switches. (b) Individuals selected to implement.

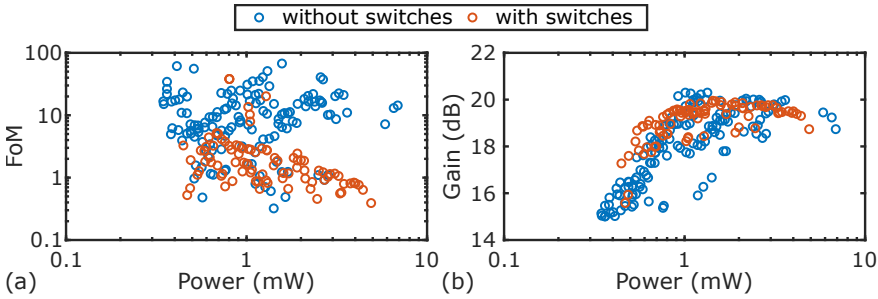

Fig. 3.7 Impact of the addition of switches to the LNA in terms of (a) FoM, and (b) voltage gain.

back-bias voltage was discretized; the need for a complex DAC is avoided by restricting it to zero or one of the supply voltages available in the chip (0.8 and 1.8 V).

The Pareto front resulting from 100 generations with these new conditions is shown in Fig. 3.6(a). The power consumption and NF ranges were not significantly impacted by the addition of switches, unlike the IIP3, whose range was reduced by 1 dB and shifted down by 3 dB. This comes as no surprise as more non-linear components are introduced in the signal path, and the voltage headroom is slightly decreased.

We can further analyze the impact of the addition of switches to the

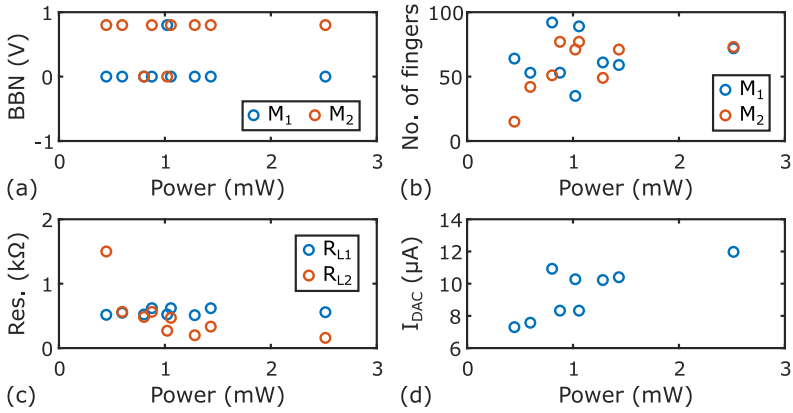

Fig. 3.8 Optimal circuit parameters of the chosen subset of individuals of the LNA including switches.

LNA with the following FoM:

$$\text{FoM} = \frac{\text{Gain}\left[\frac{\text{V}}{\text{V}}\right] \times \text{IIP3}\,[\text{mW}]}{(F-1) \times \text{Power}\,[\text{mW}]}, \tag{3.1}$$

where F is the noise factor, i.e., the NF in linear scale. Although this FoM is commonly used in the literature to compare LNAs, it has the disadvantage of being dominated by the IIP3 because this metric is used in a linear scale, while it can span several tens of dBm depending on the design. Fig. 3.7(a) shows the FoM of the Pareto fronts in Figs. 3.3 and 3.6 to illustrate the impact of the addition of switches to the LNA. The LNA with switches tends to have a lower FoM because of the decrease of around 3 dB discussed before. The power and NF are not significantly impacted, and neither is the gain, as shown in Fig. 3.7(b), further explaining the decrease in the FoM due to the switches.

From this population, we chose to implement the subset of individuals shown in Fig. 3.6(b). We chose a subset of six equally distanced non-dominated individuals with respect to the power-noise trade-off and three others that excel in linearity. From their circuit parameters, shown in Fig. 3.8, we extract the ranges of interest. First, we notice that two choices of back-biasing are sufficient, so a simple inverter can be used as a DAC for this tuning knob. The values for the number of fingers and resistances were taken directly from those of the chosen individuals, except for values close

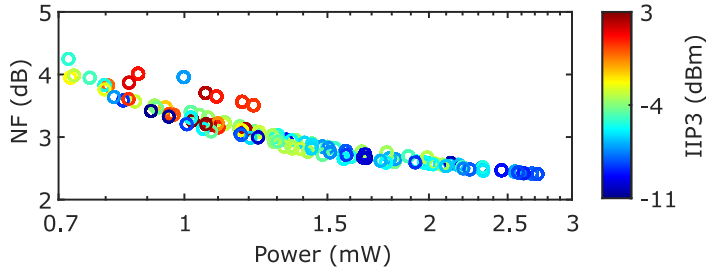

Fig. 3.9 Pareto front of the configurable balun noise-canceling LNA after 50 generations of the genetic optimization algorithm. In this design, a discrete subset of individuals from previous algorithm runs are implemented in parallel.

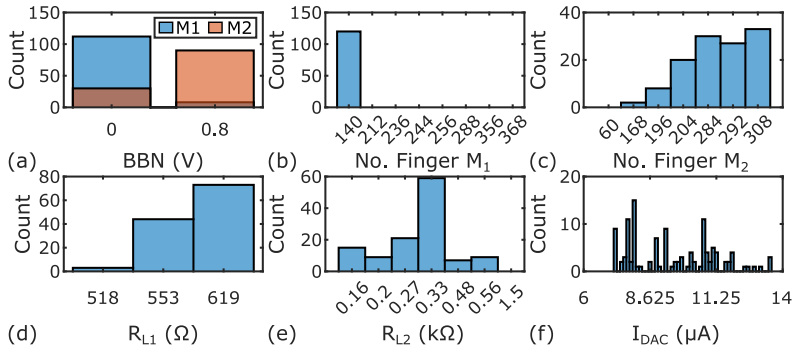

Fig. 3.10 Histograms of the optimal circuit parameters of the configurable LNA.

together, for which a mean was taken instead. Finally, the range of the current DAC was fixed between 6 and 14 µA. The value for the least-significant bit (LSB) was fixed at 125 nA because it is easy to generate from the 1-µA current reference available in the chip. This gives a total of 6 bits for the current DAC.

After connecting the different configurations in parallel, we can still use the genetic algorithm implementation to verify the performance of the previously chosen individuals and to hopefully complete the Pareto front with combinations of the chosen parameters that were not tested before. The random initialization of the genetic optimization algorithm can be skipped by providing the chosen individuals as an initial population to save time. The resulting Pareto front after 50 generations is shown in Fig. 3.9.

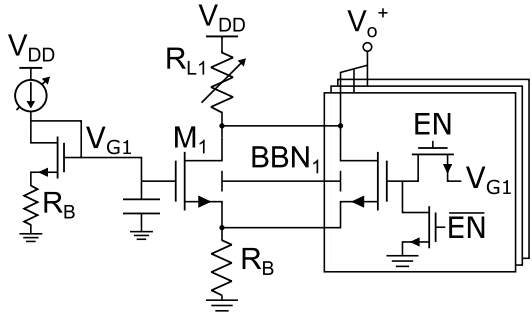

Fig. 3.11 Part of the schematic of the configurable balun noise-canceling LNA showing how switches are inserted to allow for a configuration of the width of M_1.

There are two important things to notice. The two chosen individuals with the lowest power consumption do not appear in this Pareto front, and the maximum achievable IIP3 was reduced by 4 dB. The additional parasitic capacitance linked to the unused configurations can cause previously optimal individuals to violate one of the constraints and thus be discarded in the optimization, or can also have a significant impact on one of the optimization objectives, so that the individual is no longer optimal. To verify this, we can see in Fig. 3.10 that some previously optimal parameters are no longer used by the individuals in the Pareto front. Specifically, only one value for the number of fingers of M_1 is used. This suggests that the switches inserted to connect the gate of the different parallel M_1 transistors either to the bias voltage or to ground introduce unwanted effects.

Fig. 3.11 shows how the switches were inserted in the LNA to modify the width of M_1. There are seven transistors in parallel to the main one, which is directly connected to the biasing circuit. The first problem with this implementation is that a pMOS transistor should have also been used so high bias voltages could be copied correctly on the gate of the parallel transistors. The second is that the sizing of the switches was close to minimal to save area. A large capacitor is needed at the gate of the common-gate transistor to ensure a clean bias voltage. Inserting a relatively high resistance between the two will inevitably increase the voltage ripple on the gate of the transistor coming from capacitive coupling to the source. The gain of the LNA is one of the most affected performance metrics of the LNA by a bad biasing of the common-gate transistor. At the time of this design, we chose to remove this tuning knob instead of optimizing the switch size due

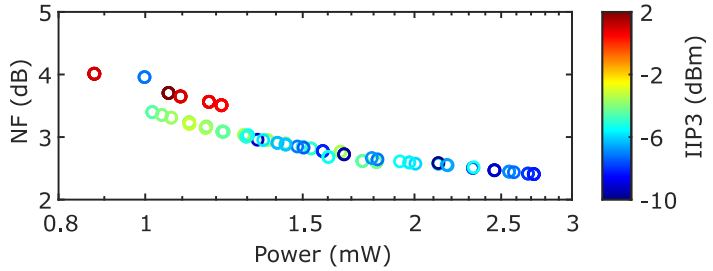

Fig. 3.12 Pareto front of the configurable balun noise-canceling LNA with optimized parameters after 50 generations of the genetic optimization algorithm.

to a lack of time before the tape-out date.

The consequence of not having access to larger widths for M_1 shows directly on the low-power configurations. We can see in Fig. 3.9 that the minimum power consumption increased from 0.4 to 0.7 mW. Only the gate voltage as a degree of freedom is not enough to sufficiently decrease the inversion level of M_1 to reduce the bias current of the left branch while maintaining an acceptable gain and impedance matching, which depends on the transconductance of M_1.

Removing the unused values for the number of fingers of M_1 and the load resistance of the right branch (R_{L2}), we obtain the Pareto front shown in Fig. 3.12. The IIP3 increased by 1 dB compared to the previous Pareto front shown in Fig. 3.9, while the power and NF were not significantly changed. The unused transistors were thus introducing non-linearity to the LNA, even after disabling them.

The optimal parameters of the final implementation of the LNA are shown in Fig. 3.13 as a function of the power consumption of the LNA to analyze the trends. Except for a couple of individuals, the back-bias voltage is constant. Similarly, the load resistor of the left branch R_{L1} maintains a relatively constant resistance. This is because, as the g_m of transistor M_1 must remain constant to guarantee a 50-Ω input impedance, the load resistance does not change to keep the same voltage gain. The other three parameters do have a visible trend. The gate voltage of transistor M_1 and the width of transistor M_2 are proportional to the power consumption, and the load resistance of the right branch is inversely proportional. These trends are linked together to result in a reduced NF at a higher power consumption. They interact to increase the transconductance of transistor M_2 to reduce its

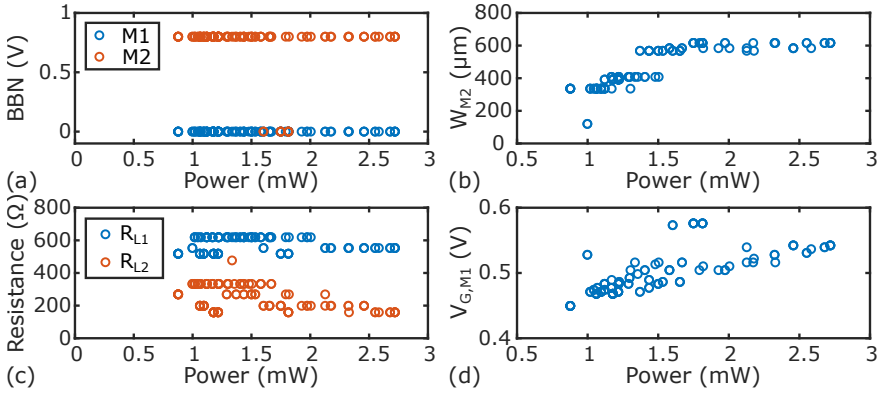

Fig. 3.13 Circuit parameters corresponding to the final implementation of the LNA as a function of the power consumption. (a) Back-bias voltage, (b) width of transistor M_2, (c) load resistance, and (d) gate voltage of transistor M_1.

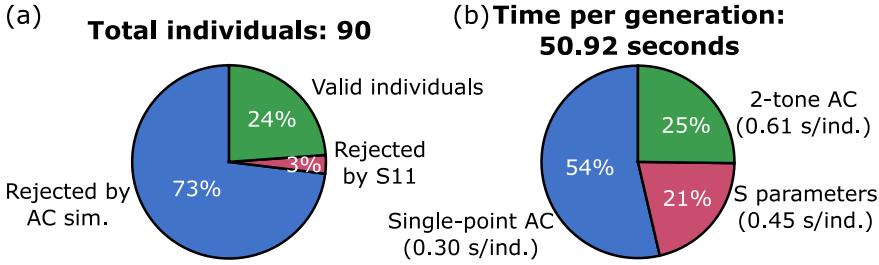

Fig. 3.14 (a) Breakdown of the individuals rejected per simulation. (b) Time breakdown per generation.

intrinsic noise while maintaining the noise-canceling property of the LNA at a similar voltage gain as the left branch for a correct balun operation.

This finalizes the schematic design of the configurable LNA. More details on the architecture and design choices are given in the next chapter, along with the post-layout simulation results and comparison to the state of the art.

In terms of simulation time, a 100-generation run of the genetic optimization algorithm during the exploratory phase of the design of the LNA takes approximately 1.5 hours in a single processor core running at 2.4 GHz. In this example, 90 individuals are created per generation to select a Pareto front of 30 individuals. At each generation, three SPICE simulations are

needed to evaluate the four constraints and the three optimization objectives: i) single-point AC simulation, the quickest one at 0.3 s/individual, for gain, power consumption, and gain and phase imbalance evaluation; ii) S-parameter extraction, the second quickest at 0.45 s/individual, for S_{11} and NF evaluation; and iii) two-tone AC simulation, the longest one at 0.61 s/individual, for IIP3 extraction. They are performed in this order to save time because, as shown in Fig. 3.14(a), most invalid individuals are discarded for violating one of the three constraints evaluated in the single-point AC simulation. Time is saved by avoiding running longer simulations on invalid individuals. Fig. 3.14(b) shows the breakdown of the mean time taken per simulation for a single generation, neglecting the time taken by the algorithm to create and sort individuals.

3.3 Post-Silicon Genetic Optimization

The use of the genetic optimization algorithm for exploring the design space of simple analog circuits as LNAs, selecting the optimal sizes, and obtaining the optimal configuration points is only possible thanks to the short time it takes to extract the performance metrics using SPICE simulations. As the circuits become more complex, with more design parameters and with significantly longer simulation times, relying only on the genetic algorithm is not viable. PLLs are a great example of this kind of circuit. In this section, we will give an overview of PLLs to show why it is not possible to design them with the same methodology as LNAs. Instead, we propose an alternative and novel genetic optimization methodology for PLLs relying on circuit measurements. As an example, we design a 64-MHz PLL to generate the main clock of a microcontroller unit (MCU) implemented in Global Foundries 22-nm FD-SOI technology. The PLL has a 32-kHz clock reference and two supply voltages at 0.65 and 1.8 V.

3.3.1 PLL basics

PLLs are mixed-signal control systems used in frequency synthesizers to generate a stable high-frequency clock signal from a precise low-frequency reference. Fig. 3.15 shows the block diagram of a charge-pump PLL featuring five sub-blocks. The voltage-controlled oscillator (VCO), as it name suggests, is an oscillator whose frequency depends on the input voltage

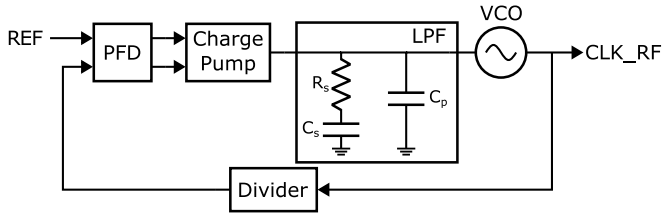

Fig. 3.15 Block diagram of an integer PLL.

and generates the high-frequency clock; a frequency divider, which ideally divides the frequency of the high-frequency clock to match that of the reference clock; the phase-frequency detector (PFD), which generates an error signal based on the frequency and phase difference between the reference and the divided clocks; and the charge pump (CP) and low-pass filter (LPF), which jointly transform this error signal into a filtered control voltage for the VCO.

The main performance metrics of PLLs involve the noise of the generated clock signal, which is quantified through the jitter in the time domain, or the phase noise in the frequency domain. Jitter has various representations that thus need to be clarified. Period jitter is the random variation of the duration of a clock cycle. In free-running oscillators, it accumulates over time, but PLLs effectively stop the accumulation to keep the output clock synchronized with the reference clock. The accumulation of jitter is commonly characterized by the N-period jitter, defined as the deviation of N clock cycles from the duration of N nominal periods [49, 66]. Fig. 3.16(a) depicts the N-period jitter profile for a PLL. As the N-period jitter stabilizes for large N values, we can define the long-term (LT) jitter as the N-period jitter when N tends to infinity.

The choice of the PLL bandwidth is not trivial because it affects noise coming from different sources differently. Noise coming from the VCO will be high-pass filtered by the loop, while noise coming from any other block in the PLL can all be referred to the reference and will be low-pass filtered by the loop. Increasing the bandwidth will filter more of the noise from the VCO but less of the reference-referred noise, and the other way around [66]. This choice directly affects the period jitter of the PLL [49]. Moreover, a lower bandwidth allows more jitter accumulation over time for a fixed period jitter, resulting in a higher LT jitter [49]. In Fig. 3.16(b), we see that there is a clear relationship between the LT jitter and the square root of

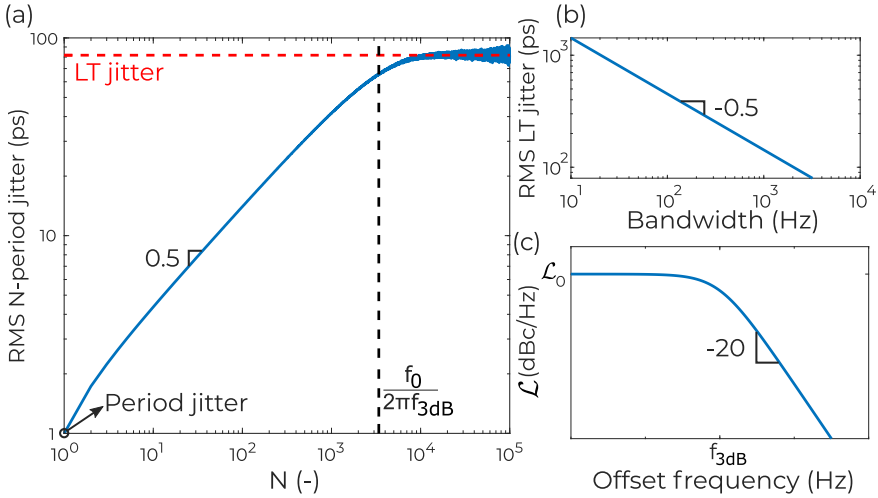

Fig. 3.16 Behavioral simulations: (a) N-period jitter profile for a 64-MHz PLL with a 3-kHz bandwidth and (b) effect of the 3-dB PLL bandwidth on the LT jitter. (c) Phase noise profile.

the bandwidth, evidenced by the slope of the line in log-log scale. In any case, the bandwidth must remain well below the reference frequency so the LPF can filter it correctly; a factor of ten is usually taken between the two [52].

The simplified phase noise ($\mathcal{L}$) profile of a PLL is shown in Fig. 3.16(c). It is linked to the LT jitter (σ_{LT}) through the following expression [49]:

$$\sigma_{\mathrm{LT}} = \lim_{N\to\infty} \sqrt{\frac{\mathcal{L}_0 f_{3\mathrm{dB}}}{2\pi f_0^2}\left(1 - e^{-2\pi f_{3\mathrm{dB}} N/f_0}\right)} = \sqrt{\frac{\mathcal{L}_0 f_{3\mathrm{dB}}}{2\pi f_0^2}}, \tag{3.2}$$

where f_0 is the operating frequency and $\mathcal{L}_0$ is the phase noise level at low frequencies. This expression shows that minimizing the LT jitter is equivalent to minimizing the phase noise.

The noise-power trade-off of a charge-pump PLL depends on complex interactions between five degrees of freedom: the driving strength of the VCO, the CP current, and the values of the three passive components of the LPF. These complex interactions complicate the theoretical prediction of which combinations reach the optimal noise-power trade-offs of the PLL. Therefore, an optimization algorithm is useful for finding these combinations. However, the time required for a pure numerical optimization using

Table 3.2 Optimization time.

	Single performance evaluation	Exhaustive search of the optimal solution	Genetic optimization
Simulation in the loop*	≈ 4 days	3.7×10^5 years	1.37 years
Hardware in the loop	20 s	21 years	17 h + fab. time

* SpectreX post-layout transient noise simulation on 8 cores at 2.4 GHz

a genetic optimization algorithm with SPICE transient noise simulations is prohibitive, as shown in Table 3.2. The reasons for this is that we need a precise simulation with a small time step to capture correctly the high-frequency clock signal, but at the same time, due to the low-frequency reference, a very large amount of clock periods must be simulated to estimate the phase noise at low frequencies or the LT jitter, as shown in Fig. 3.16(a). Noise transient simulations are used as the system complexity makes the convergence of a steady-state simulation difficult. Using behavioral models for accelerating the simulations is an option [69], but because these complex interactions depend on the circuit non-idealities, the behavioral models need to be accurate enough and thus complex as well. Alternatively, we propose to perform the optimization post-silicon with the hardware in the loop by using the measurements from the fabricated prototype. For that, the PLL was made highly programmable by implementing a large range of values for the five degrees of freedom. Table 3.2 shows that this approach significantly reduces the optimization time. However, this means that the genetic optimization cannot be used to find the appropriate ranges for the degrees of freedom. Instead, analytic expressions will be used to size the programmable PLL to ensure that the optimal parameters are included. This highlights that expert knowledge cannot be disregarded. Even when numerically optimizing a circuit, the input from the designer is key for an optimal result.

3.3.2 Implementation

Fig. 3.17(a) shows the general flow of the multi-objective genetic optimization algorithm based on the NSGA-II algorithm [29, 59] with the proposed modifications. In the evaluation step of the algorithm, SPICE simulations

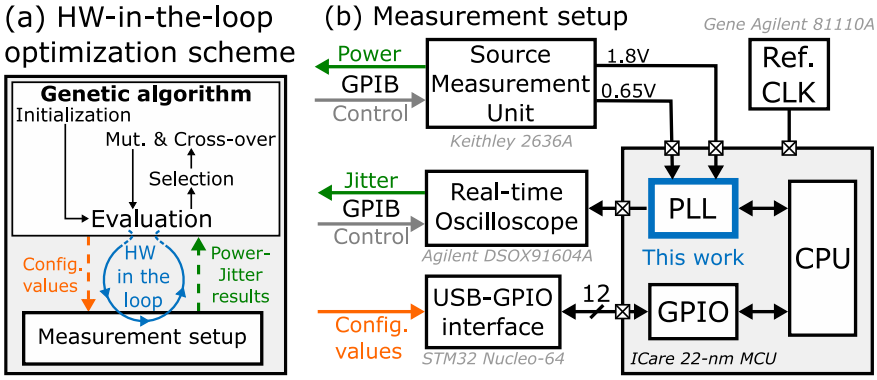

Fig. 3.17 (a) Hardware-in-the-loop optimization scheme based on a genetic algorithm. (b) Automatic measurement setup.

are replaced by measurements of the prototype. This requires the automatic measurement setup shown in Fig. 3.17(b). To measure the objectives, i.e., power consumption and LT jitter, of each individual set of design parameters in a given generation of the genetic algorithm, the configuration values are communicated to the chip under test one at a time with an ad hoc protocol using the general-purpose I/Os (GPIOs). Each configuration value corresponds to a 25-bit word containing the values of the five degrees of freedom. An STM32 Nucleo-64 board is used as the interface between the serial communication with the computer running the optimization and the GPIOs of the chip. For the power measurements, the 2636A source measurement unit from Keithley provides both supply voltages needed in the PLL (i.e., 0.65 and 1.8 V) and measures the corresponding current. The DSOX91604A real-time oscillator from Agilent is used to measure the jitter; it outputs the N-period jitter as the standard deviation of 2×10^5 samples of the duration of 1×10^4 periods of the clock. All measurement results for a single individual (i.e., a single configuration of the PLL) are directly retrieved by the computer via a general-purpose interface bus (GPIB), which triggers the next measurement by sending the new configuration to the chip. Once the measurements of all the individuals in the generation are done, the algorithm does the selection of the fittest individuals and creates the next generation using the genetic operators. The algorithm stops once a given number of generations have passed, or a certain stopping criterion is achieved, e.g., the hypervolume metric [59].

3.3.3 Design of the Proposed Programmable PLL

Given the long simulation time for a single configuration of the PLL, the selection of the parameters has to be done with analytical expressions and simpler simulations because exploring the design space with the genetic algorithm is time-prohibitive. Therefore, in the following sections, we will detail how the programmable PLL was designed and how the ranges of the five degrees of freedom were selected.

3.3.3.1 Choice of Oscillator

The conventional architecture for VCOs based on a ring oscillator is the current-starved architecture. However, back-bias-controlled oscillators (BBCOs), a novel architecture enabled by the forward and reverse back-bias capabilities of the FD-SOI technology, offer some advantages over the conventional architecture. They can achieve more than 4× lower power consumption and slightly better performance in terms of the noise-power trade-off because of the shorter transistor stacks [51]. BBCOs also exhibit a more linear transfer function and thus a constant frequency gain over a wide PLL frequency range because of a more linear control of a transistor current through its back gate [51]. In the context of PLL design, this translates into phase noise reduction, and thus jitter, at the PLL output [70].

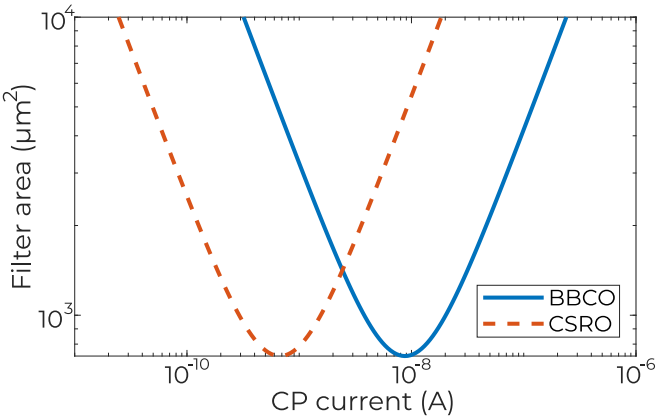

Fig. 3.18 Area of the conventional second-order passive low-pass filter in 22-nm technology for a PLL bandwidth of 3 kHz and a damping factor of $\sqrt{2}/2$. The considered capacitance and resistance densities are 9 fF/μm^2 and 125 kΩ/μm^2, respectively.

The silicon area of the LPF is an issue in charge-pump PLLs with low reference frequencies because of the large capacitors needed to obtain a low bandwidth, which is required for stability [71]. It thus needs to be minimized to avoid the need for off-chip components. As shown in Fig. 3.18, when a fixed bandwidth and damping factor of the PLL are considered, the minimum area occupied by the LPF is achieved at a higher CP current for the BBCO than for the current-starved ring oscillator (CSRO). This is because BBCOs have a frequency gain that is typically 10-20× lower than that of CSROs [51]. This increase in CP current remains acceptable in terms of power consumption because it does not actuate continuously, and its power remains low compared to the BBCO, which is the greatest contributor to the PLL power. The advantage of this current increase is that it decreases the phase noise due to the CP because its noise is scaled by the squared nominal value of the current [66].

Given the low frequency of operation, we use ultra-high-threshold-voltage (UHVT) transistors in the design of the BBCO to reduce its length and minimize its area. This transistor flavor has reverse back-bias (RBB) capabilities, i.e., in terms of absolute values, their threshold voltage can be increased when increasing their back-bias voltages. Therefore, these voltages can be used to control the oscillation frequency. However, negative voltages are needed to control the BBN in the case of RBB. Generating negative voltages for driving the BBCO can be done efficiently with switched-capacitor CPs for a few µW [UCP1]. However, this power overhead in the µW range is not negligible in this design. Therefore, to avoid the area and power overhead of generating a negative voltage, only the pMOS back-bias voltage (BBP) was kept as a tuning knob for the oscillator frequency. Simulations show that this degrades the duty cycle of the output clock signal by less than 2 %.

3.3.3.2 PLL Architecture

Fig. 3.19 shows the schematic of the proposed PLL. It is part of an ultra-low-power (ULP) MCU codenamed ICare, where the logic and SRAM memories are supplied at 0.65 V and the I/O voltage is 1.8 V. Three main modifications were introduced to the conventional architecture for controlling the BBCO. Firstly, although transistors in this technology can withstand up to 3 V on the back gate, this range was reduced to 1.8 V to comply with the I/O voltage of the MCU. A level shifter (LS) was added at the interface of the charge pump and the PFD, which works at 0.65 V as the rest of the blocks. Finally, the third modification was done to the LPF to limit the impact of

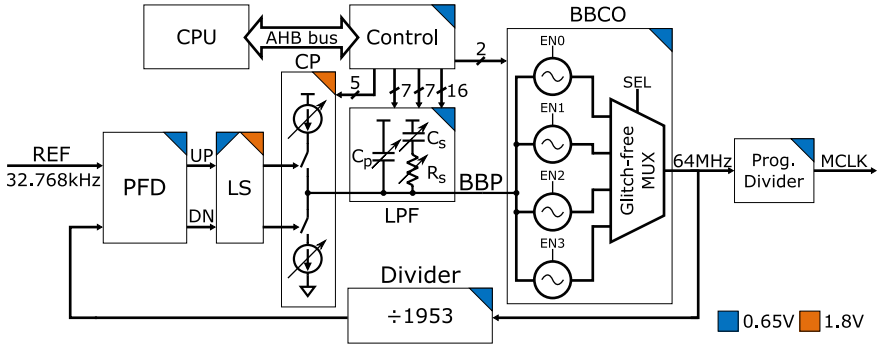

Fig. 3.19 Schematic of the proposed programmable PLL.

supply noise on the oscillator. An AC coupling between the BBP node and the 0.65-V BBCO supply was added by modifying the filter to have the capacitors connected to the supply instead of the ground.

3.3.3.3 BBCO Programmability

The inherent noise-power trade-off of the BBCO can be exploited to make the PLL programmable. At a fixed oscillation frequency, increasing the number of inverting stages in parallel in a ring oscillator is a good way to decrease the phase noise and the jitter, at the cost of an increased power consumption [51]. Nevertheless, the addition of switches at the input or output of the inverting stages has the risk of degrading the phase noise. Instead, we chose to design a bank of ring oscillators with different noise-power trade-offs. As can be seen in Fig. 3.19, four BBCOs were implemented in parallel and multiplexed through a glitch-free multiplexer. Each of them is disabled when it is not used to avoid power overheads. As UHVT transistors are used, the leakage overhead of implementing multiple BBCOs is negligible.

For selecting these four BBCOs, oscillators composed of different standard cells were simulated to select those with the best performance in terms of power, period jitter, and frequency gain. Inverter gates as well as higher fan-in inverting gates, i.e., NAND and NOR gates with short-circuited inputs, were considered. To guarantee functionality in the PVT corners, potential oscillators were sized to oscillate at the target frequency near half the input voltage range, i.e., 0.9 V. The results are shown in Fig. 3.20. The points are grouped according to their frequency gain (K_{VCO}). For the PLL design, the selected oscillators must have similar frequency gains to avoid

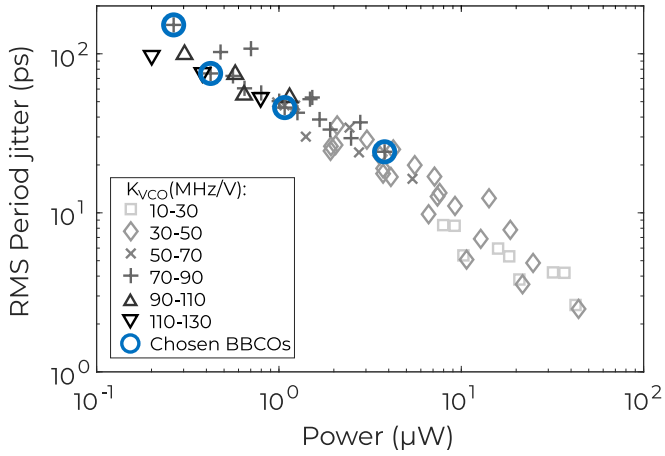

Fig. 3.20 Noise-power trade-off for different BBCOs grouped according to their frequency gain. A four-point Pareto front highlights the selected oscillators.

a large range for the LPF parameters needed to stabilize the system. Between the groups of oscillators with the highest number of points, the group whose gain is between 70 and 90 MHz/V was selected because a higher frequency gain is preferable to minimize the CP current and the LPF area. Inside this group, four optimal, evenly distributed points were selected in our implementation. They are highlighted in Fig. 3.20.

3.3.3.4 *Bandwidth Programmability*

Another way to add programmability to the PLL is through the bandwidth, which can be modified by changing the CP current (I_{CP}) or the values of the LPF passive components (C_S, C_p, and R_S). Changing the bandwidth contributes to the noise-power trade-off in the following way. A smaller bandwidth can be achieved by decreasing the CP current, thus lowering the peak power consumption. As depicted in Fig. 3.16(b), a smaller bandwidth implies a higher LT jitter because it accumulates for a longer time. On the contrary, a higher bandwidth makes the accumulation stop sooner but needs a higher CP current.

The bandwidth of the PLL can be obtained analytically with the expression:

$$f_{3\text{dB}} = \frac{\omega_n}{2\pi}\sqrt{1+(\omega_n R_S C_S)^2 - 2\zeta^2 + \sqrt{[2\zeta^2 - 1 - (\omega_n R_S C_S)^2]^2 + 1}}, \quad (3.3)$$

where the natural frequency ω_n and the damping factor ζ are parameters of the closed-loop transfer function of the PLL given by:

$$H(s) = \frac{N\omega_n^2(1 + sR_SC_S)}{s^2 + 2s\zeta\omega_n + \omega_n^2}, \tag{3.4}$$

with N being the division factor of the frequency divider, or the ratio between the output and reference frequencies.

The values of ω_n and ζ are linked to the circuit parameters through these expression [52]

$$\omega_n = \sqrt{\frac{I_{CP}K_{VCO}}{2\pi C_S N}}, \tag{3.5}$$

$$\zeta = \frac{R_S}{2}\sqrt{\frac{I_{CP}K_{VCO}C_S}{2\pi N}}. \tag{3.6}$$

A damping factor below one decreases the settling time of the PLL but increases the frequency ripple and overshoot, while a damping factor above one has no overshoot or ripple but increases the settling time.

For a given value of N, K_{VCO}, I_{CP}, bandwidth and damping factor, we can use (3.3), (3.5), and (3.6) to calculate the values of ω_n, C_S, and R_S. The value of the third passive component of the LPF, C_P, is usually taken five to ten times lower than C_S to reduce the impact of the pole it adds on the loop stability, while effectively removing the voltage ripples coming from the CP current flowing through R_S [52].

Table 3.3 summarizes the degrees of freedom and their corresponding range. A 5-bit current DAC controls the CP current. Two parallel 7-bit binary banks of capacitors were implemented for C_s and C_p. Finally, a series bank of resistors with logarithmically spaced values was implemented for R_s. The value was controlled through 16 bits with a one-hot code.

Table 3.3 Range of the degrees of freedom.

Parameter	Range	Number of bits
BBCO period jitter	24 to 152 ps	2
Charge pump current	5 to 160 nA	5
Series capacitance (C_s)	0.25 to 32 pF	7
Series resistance (R_s)	1 to 64 MΩ	4
Parallel capacitance (C_p)	0 to 32 pF	7

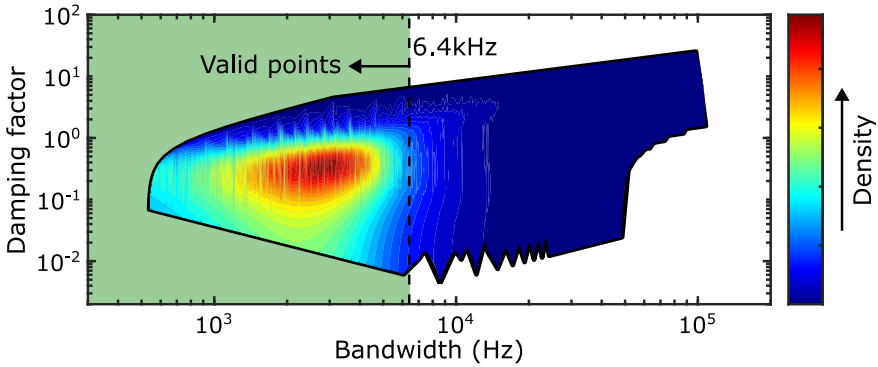

Fig. 3.21 Range of PLL bandwidth and damping factor resulting from the discrete choices of values for the degrees of freedom. The green zone shows the valid region in terms of bandwidth, corresponding to maximum 5 % of the reference frequency.

With these values, the resulting PLL bandwidth and damping factor are shown in Fig. 3.21. The color indicates the local density of points; a higher density means that more combinations of the degrees of freedom give the same values of bandwidth and damping factor. The green zone indicates the values that are considered valid in terms of bandwidth. A PLL with a bandwidth higher than 10 % of the reference frequency is not able to filter the signal at the output of the PFD correctly, thus resulting in non-optimal results. However, the values of the filter components are particularly subject to process variations, making the analytical approximation of the bandwidth not precise. Therefore, the constraint was relaxed by a factor of two. The parameter ranges were chosen such that the highest density of points falls in the valid region to maximize the chances of finding the optimal parameters after fabrication.

3.3.4 Measurement Results

The proposed circuit was prototyped in Global Foundries (GF) 22-nm FD-SOI technology. As shown in Fig. 3.22, it occupies an area of 0.0156 mm^2 on the 1.25×1.25 mm ICare Cortex-M4 MCU chip.

Fig. 3.23 shows the process of convergence across 50 generations from an initially random population. The only performance constraint in the algorithm concerns the bandwidth of the PLL. In the obtained Pareto front,

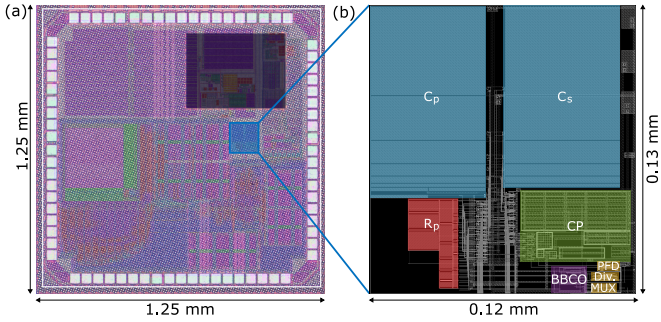

Fig. 3.22 (a) Chip microphotograph with overlaid layout and (b) layout of the proposed PLL.

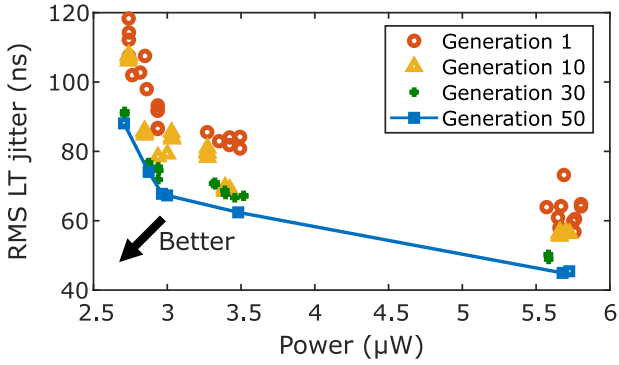

Fig. 3.23 Pareto front of the noise-power trade-off across 50 generations of the NSGA-II algorithm.

the point with the best power consumes 2.7 µW for an LT jitter of 88 ns, whereas the point with the best LT jitter is at 45 ns and consumes 5.7 µW.

The values of three of the five degrees of freedom for the optimal points in the Pareto front are shown in Fig. 3.24(a)-(c). The BBCO index corresponds to the digital code used to select one of the four BBCOs highlighted in Fig. 3.20, ordered in ascending order in terms of power consumption. The capacitances C_s and C_p are not shown because their selected values are shared by all the optimal points. C_s is fixed at 31 pF, whereas C_p relies only on the parasitic capacitances of the BBP node, estimated at 41 fF from post-layout parasitic extraction. Fig. 3.24(d) shows the PLL bandwidth resulting from these values, computed analytically. A linear dependence between power consumption and PLL bandwidth can be appreciated for the lower-

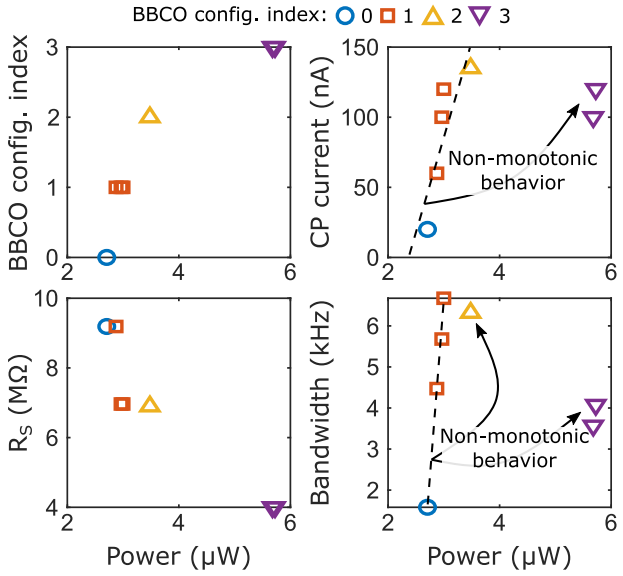

Fig. 3.24 Values of the degrees of freedom for the optimal points in the Pareto front: (a) digital code to select the BBCO ordered in ascending order in terms of power consumption, (b) charge pump current, and (c) series resistance of the low-pass filter. (d) Analytically approximated PLL bandwidth.

power points with a higher LT jitter. For fixed values of the capacitances, higher bandwidths are obtained by increasing the CP current. This results in a reduced jitter because the BBCO contributes less to the output noise, at the price of a higher power. Nevertheless, once the maximum bandwidth is reached, the relation becomes more complex and non-monotonic for the lower-jitter points. This shows the difficulty of predicting analytically the optimal design points in the noise-power trade-off of a PLL, motivating the need for post-silicon optimization.

It is also interesting to highlight that the BBCO dominates the noise-power trade-off of the PLL. The points that are near to others in the Pareto front share the same BBCO and only differ in the other filter parameters. To find other optimal points, a wider choice of oscillators must be included in the design. This is particularly true for points in the lower-jitter extreme of the Pareto front, as opposed to the lower-power extreme, where the power of the rest of the blocks imposes the lower limit achievable in terms of power consumption.

Table 3.4 presents a comparison to recent state-of-the-art MHz-range PLLs. After optimization, the proposed PLL exhibits a normalized power 5× lower than the design in [72], which generates the same output frequency. Furthermore, the FoM that allows comparing the power-jitter trade-off is 1.2 to 5.3 dB better, even though we use a reference frequency 15× lower, which tends to let the jitter accumulate more. Lower reference frequencies limit the PLL bandwidth and thus the filtering of the oscillator phase noise. The designs in [73, 74], which also use a 32-kHz reference but generate an output frequency 2.9 to 4.8× higher, exhibit a better phase noise. However, they rely on more complex active low-pass filter architectures, which have a power overhead resulting in a normalized power 29 to 142× higher.

Table 3.4 Comparison to state-of-the-art MHz-range phase-locked loops.

	This work	JSSC'19 [72]	ESSCIRC'16 [73]	TCAS-II'12 [74]
Technology	22 nm FD-SOI	65 nm Bulk	28 nm FD-SOI	45 nm Bulk
Supply voltage [V]	0.65/1.8	0.8	0.5/1.5	-
Frequency [MHz]	64	64	307.2	184
Ref. frequency [kHz]	32.768	500	32.768	32.768
N. power [nW/MHz]	42 to 89	466	1953	5978
Settling time [µs]	400*	-	2330	-
$\mathcal{L}$(10 kHz) [dBc/Hz]	−21 to −29	−57	−62	−48
$\mathcal{L}$(1 MHz) [dBc/Hz]	−83	−77	−110.6	−105.2
RMS period jitter [ps]	99.6 to 120.8	55.3	-	-
FoM [dB]†	−221.6 to −225.7	−220.4	-	-
Area [mm^2]	0.0156	0.016	0.0175	0.086

* After change of configuration

† $\text{FoM} = 20\log\left(\frac{\text{Period jitter}}{1\,\text{s}}\right) + 10\log\left(\frac{\text{Power}}{1\,\text{mW}}\right)$

3.4 Conclusions

In this chapter, the methodology used throughout this thesis is introduced. Based on a genetic algorithm, this multi-objective optimization extracts the Pareto front of the circuit by making a population of individuals evolve through several generations. The performance metrics of the circuit must be extracted for each individual at each generation. We propose two ways to perform this evaluation according to the complexity and time required to simulate the circuit.

At first, the design of a balun noise-canceling LNA is used as an example of a simple analog block. Since evaluating its performance needs less than 2 s using simulations, the genetic algorithm based on simulations can be used to design the amplifier from the design-space exploration until the final selection of circuit parameters that give the optimal combinations.

Complex mixed-signal circuits cannot be designed this way because of the long time it takes to simulate them. This is why in the second part of the chapter, we design a highly programmable 64-MHz PLL based on a back-bias-controlled ring oscillator and optimize it with a post-silicon "hardware-in-the-loop" optimization scheme based on the genetic algorithm used for the LNA. The optimization scheme was run on an automatic setup for the measurement of the power consumption and the jitter of the PLL. The evolution of the five degrees of freedom across the Pareto front shows a non-monotonic behavior for the low-jitter points, highlighting the difficulty of predicting them analytically and the advantages of post-silicon optimization. Furthermore, the optimization time was decreased with the proposed scheme to only 17 h, in contrast to the simulation-based alternative that would require more than a year.

Even if the sizing of the circuits is done with the help of an optimization algorithm, the experience and knowledge of the designer are still crucial to obtain an optimal design. When simulations are in the loop, like for the LNA designed in this chapter, the designer must choose an appropriate range for the design parameters to speed up the optimization process. Giving at the beginning a random and arbitrarily large range of parameters to the genetic algorithm could result in a long optimization time, as the time it could take the algorithm to find an initial population that respects all constraints is unknown. There is also a risk that this algorithm finds sub-optimal results if the initial population is far from the optimal solution.

Having the hardware in the loop makes even more crucial the experience and knowledge of the designer, as the optimal solution must be included in the final implementation. The advantage, in this case, of the optimization algorithm is that it finds the optimal sizing of the circuit taking into consideration unknown variables at design time, like the PVT corner or transistor mismatch. Performing this optimization on several chips, at different temperatures, and with slight supply voltage variations could give the designer interesting insights for future designs if trends appear in the resulting optimal parameters. This, unfortunately, is not possible with the PLL designed in this chapter, as the optimization algorithm was run on a single chip. Fortunately, the extensive existing literature on PLLs and their optimization was sufficient to obtain a satisfactory result of the genetic optimization algorithm with the hardware in the loop.

4

Programmable Narrowband RF Front End

The first of two prototypes of a programmable receiver is presented in this chapter. We begin with the design of only the RF front end to estimate the potential power savings that are possible by tuning the noise-power trade-off of the blocks in the receiver. This RF front end includes the LNA, quadrature mixers, and an analog PLL. Since the LNA is the first block in the receiver chain, it usually dominates the input-referred noise. It is thus interesting to explore the benefits of exploiting its inherent trade-offs. The analog PLL is also designed to be programmable in order to validate the specifications obtained in Chapter 2. The mixers are simply optimized to minimize both their power consumption and noise, while the contribution of the other blocks in the receiver chain is simply estimated at this stage and will be refined in Chapter 5.

4.1 Architecture

The schematic of the proposed programmable direct-conversion RF front end is shown in Fig. 4.1. It is implemented in Global Foundries 22-nm FD-SOI techonology and supplied at 0.8 V. It features a programmable

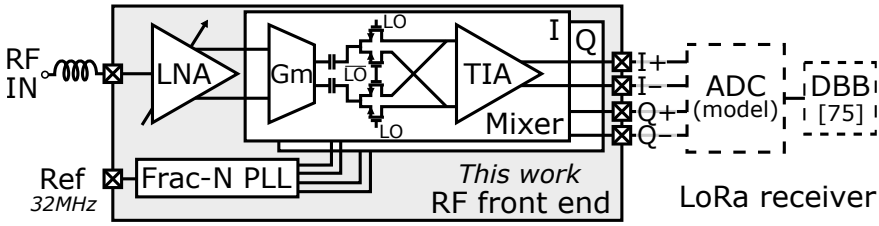

Fig. 4.1 Schematic of the proposed programmable RF front end. Models for the power consumption of the ADCs and the DBB processor from [75] are used to approximate the power consumption of a complete LoRa receiver.

balun noise-canceling LNA, whose noise-power trade-off can be configured dynamically. It also includes a fractional-N PLL with a 32-MHz clock reference consuming 370 µW, and two passive mixers for the in-phase (I) and quadrature (Q) paths. A direct-conversion receiver architecture was chosen because of its simplicity. Band-pass filters are replaced by simpler low-pass filters, and there is no need for complex image-rejection filters due to the absence of image frequencies. On the downside, the flicker noise dominates in the band of interest. To mitigate this, a passive current-mode mixer was chosen, and, by AC-coupling the switches, their contribution to the flicker noise of the system is negligible [76]. Moreover, spread-spectrum techniques, as used in LoRa, perform a despreading operation at the receiver, after which the noise energy is spread over the entire bandwidth [77], thereby limiting the effect of flicker noise on low frequencies. A transconductance (Gm) stage was added to interface the current-mode mixer with the voltage-domain output of the LNA. Finally, a transimpedance amplifier (TIA) converts back the signal into the voltage domain and provides first-order low-pass filtering at 500 kHz, the maximum bandwidth allowed in LoRa [21]. Simulations of the mixer alone give a power consumption of 281 µW, and an integrated double-sideband NF of 24.8 dB.

4.2 Design of the Programmable LNA

In this section, we provide more details on the architecture of the LNA whose design was presented in Chapter 3 as well as the post-layout simulation results.

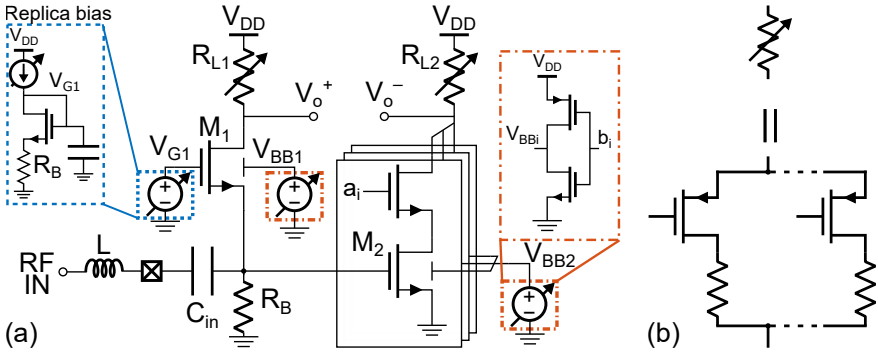

Fig. 4.2 (a) Schematic of the proposed programmable balun noise-canceling LNA. The M_1 gate voltage is generated by a diode-connected replica, whereas both back-bias voltages are generated by inverters. (b) Implementation of the variable resistance as a parallel bank of resistors.

4.2.1 Architecture

Fig. 4.2(a) shows the schematic of the proposed programmable LNA, based on the noise-canceling balun LNA architecture [68]. It is composed of a common-gate branch on the left, which provides the gain on the positive path and the 50-Ω matching to the antenna, and a common-source branch on the right, which provides the gain on the negative path and is designed to cancel out the noise from the common-gate transistor M_1. M_1 and M_2 are implemented with RF LVT transistors. By combining the LNA and balun functions, we can take full advantage of the suppression of all common-mode signals coming from the use of fully differential blocks, without the insertion losses of a standalone balun needed to interface with single-ended antennas.

A high-Q off-chip inductor offers narrowband input matching by resonating at 868 MHz with the 2.17-pF input MOM capacitor. This results in a higher interference immunity, which is important when working on license-free frequency bands, as with LoRa. Another advantage of the high-Q inductor is that a relatively low power consumption is needed for input matching and low NF [78].

The programmability of the LNA is implemented through four tuning knobs: the load resistors (R_{L1} and R_{L2}), the gate voltage of transistor M_1, V_{G1}, the width of transistor M_2, and the back-bias voltage of both transistors, a tuning knob enabled by the forward back-biasing of the FD-SOI

Table 4.1 Range of the tuning knobs.

Parameter	Range	Number of bits
R_{L1}	518 to 619 Ω	2
R_{L2}	159 to 562 Ω	3
DAC current	7 to 15 µA	6
M_2 width	120 to 616 µm	3
M_1 back-bias voltage	0 or 0.8 V	1
M_2 back-bias voltage	0 or 0.8 V	1

technology.

A couple of banks of resistors allow tuning the load resistances, as shown in Fig. 4.2(b). The resistors are implemented with unsilicided N+ diffusion resistors in the thin active SOI film, and the pMOS switches with RF HVT transistors. The gate voltage V_{G1} is controlled by modifying the current flowing through a diode-connected replica of transistor M_1 with a simple current digital-to-analog converter. For tuning the width of transistor M_2, we used several parallel branches composed of common-source transistors of varying widths and switches. The switches are controlled by the digital signals a_i and are implemented with RF SLVT transistors. Finally, the back-bias voltage of both transistors was restricted to only two values for simplicity of implementation. An inverter connects the back gate either to the supply voltage or to ground. Table 4.1 shows the range and resolution of each of these tuning knobs. The genetic optimization algorithm described in Chapter 3 was used with the post-layout netlist to choose the optimal combinations of the chosen values for these parameters, summarized in Table 4.1.

4.2.2 Post-layout Simulation Results

Fig. 4.3(a) shows the Pareto front for the proposed LNA resulting from the genetic optimization based on post-layout simulations. The implemented programmability allows changing the configuration dynamically to any of the points in the Pareto front with a limited area overhead. These results are compared to sub-GHz narrowband and sub-12-GHz wideband LNAs in Table 4.2. The FoM used to compare the LNA objectively to the state of the art is the following:

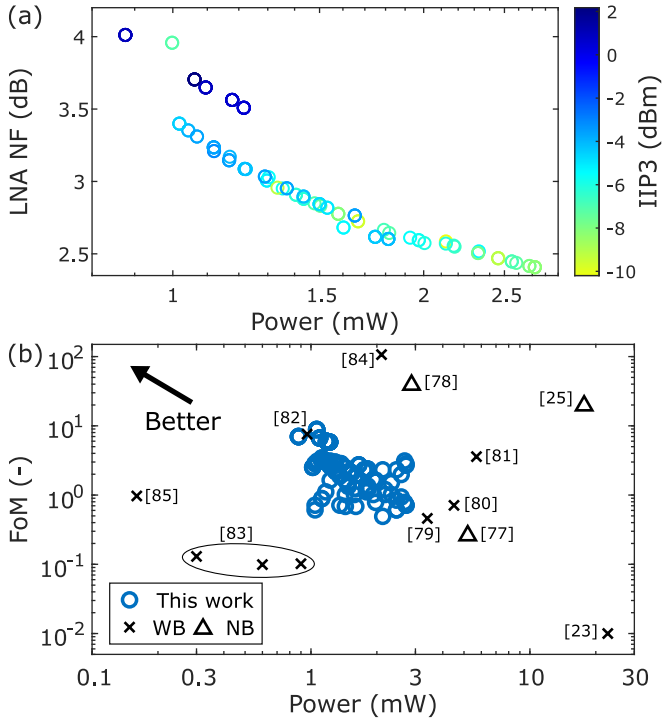

Fig. 4.3 (a) LNA Pareto front resulting from a genetic optimization based on post-layout simulations. (b) Comparison to state-of-the-art narrowband (NB) and wideband (WB) LNAs in terms of the FoM and power consumption.

$$\text{FoM} = \frac{\text{Gain}\left[\frac{\text{V}}{\text{V}}\right] \times \text{IIP3}\,[\text{mW}]}{(F-1) \times \text{Power}\,[\text{mW}]}, \tag{4.1}$$

where F is the noise factor, i.e., the NF in linear scale.

As shown in Fig. 4.3(b), the proposed LNA is comparable to other state-of-the-art designs and is among the best in terms of the trade-off between the FoM and power consumption. Those achieving a better FoM use architectures focused on improving the IIP3, resulting in either an increased power consumption [25] or in complex feedback structures that make the programmability difficult [79, 85]. Compared to other programmable designs [23, 25, 84], the proposed LNA offers, at the same time, the low power consumption needed for LPWAN, a large tuning range of the main perfor-

Table 4.2 Comparison to state-of-the-art sub-GHz narrowband LNAs and sub-12-GHz wideband LNAs.

	Narrowband				Wideband								
Publication	**This work**	[78]	[25]	[79]	[80]	[81]	[82]	[83]	[84]			[85]	[86]	[23]
	ISCAS	VLSI	JSSC	TMTT	ISSCC	JSSC	TCAS-I	TCAS-I	ISLPED			S3S	JSSC	TCAS-I
Year	2024	2021	2015	2015	2022	2021	2020	2018	2017			2017	2016	2012
Technology	22 nm FD-SOI	0.13 µm bulk	65 nm bulk	0.13 µm bulk	28 nm bulk	28 nm bulk	65 nm bulk	65 nm bulk	28 nm FD-SOI			28 nm FD-SOI	0.13 µm bulk	0.18 µm bulk
Programmable	✓	✗	✓	✗	✗	✗	✗	✗	✓			✗	✗	✓
Supply [V]	0.8	1.2	1.6/1	1.2	1	1	1	1.2	0.6	0.8	1	1	0.4	1.8
Power [mW]	0.88 to 2.72	5.2	15.8 to 20.2	2.88	3.4	4.5	5.7	0.96	0.3	0.6	0.9	2.1	0.16	18 to 39.6
f_c / BW [GHz]‡	0.868	0.868	0.2 to 1.6	0.9	2.8	4.5	2.1	2.5	5.6	7.5	9.5	11.1	2.5	1.4
Gain [dB]	17.4 to 19.7	14.21	14 to 24	18	19.6	15.2	27.5	21.2	16.8	18.8	21.5	19.2	13	5 to 12.8
NF [dB]	2.4 to 4	0.92	≥ 3.6	1.94	2.7	2.09	2.3	3	7.3	6.7	6.3	2.2	4.5	1.88†
IIP3 [dBm]	−10.2 to 2.2	−12	14.5	9	−8.5	4.62	−2.2	−2		−16		12.1	−12	−28.9 to −10
FoM	0.49 to 8.84	0.26	19.56*	38.9	0.46	0.71	3.59	7.58	0.13	0.10	0.10	106.78	0.97	0.01†
Area [mm²]	0.0126	0.047	0.2	0.099	0.0078	0.03	0.046	0.05		0.0015		0.00091	0.39	1
I/O	Balun	SE	Balun	FD	SE	SE	Balun	FD		SE		Balun	SE	SE

SE: Single-Ended FD: Fully Differential * Estimated from Table III † Performance of non-programmable LNA
‡ Narrowband LNAs are characterized by their central frequency (f_c) and wideband LNAs by their bandwidth (BW)

mance metrics, and a FoM on par with the rest of non-programmable LNAs.

4.3 Current-Domain Passive Mixer

4.3.1 Architecture

The schematic of the mixer is shown in Fig. 4.4(a). First, a Gm block interfaces the current-domain mixer with the voltage-domain LNA. It is composed of a self-biased resistive-feedback inverter implemented with RF LVT transistors. The resistor obliges both the input and output of the inverter to the same DC voltage, making it work in its amplification range. The resistor is implemented with an unsilicided N+ diffusion resistor in the thin active SOI film. Then, the switches perform the actual mixing operation with the four clock phases coming from the fractional-N PLL. The AC coupling makes their impact on the overall flicker noise negligible because no DC current passes through them [76]. MOM capacitors are used for the AC coupling. Finally, a transimpedance amplifier (TIA) transforms the signal back to the voltage domain while performing first-order filtering at 500 kHz, the highest LoRa bandwidth.

The fully differential amplifier used in the TIA is a Miller-compensated two-stage amplifier, as shown in Fig. 4.4(b). A local resistive feedback stabilizes the common mode of the first stage, while the second stage is stabilized by the auxiliary amplifier shown in Fig. 4.4(c). This additional common-mode feedback (CMFB) amplifier fixes the main amplifier's output DC voltage to 335 mV, the value of the voltage V_{ref} which is fixed by a resistor ladder. Identical resistors and capacitors are connected between the outputs of the main amplifier to generate a mean voltage to feed to the other input of the CMFB amplifier. Both amplifiers are implemented with LVT transistors. For the passive components, MOM capacitors are used, and the resistors are implemented with the denser unsilicided narrow high-resistance N+ polysilicon resistors. We do not use the same type of resistor as in the Gm block because of the several orders of magnitude of difference in resistance. It occupies less area to implement these resistors in the MΩ or the hundreds of kΩ with a denser resistor, while the 50-Ω resistors in the Gm occupy less area with the less dense resistor because a lower number of resistors must be connected in parallel to achieve such a small value.

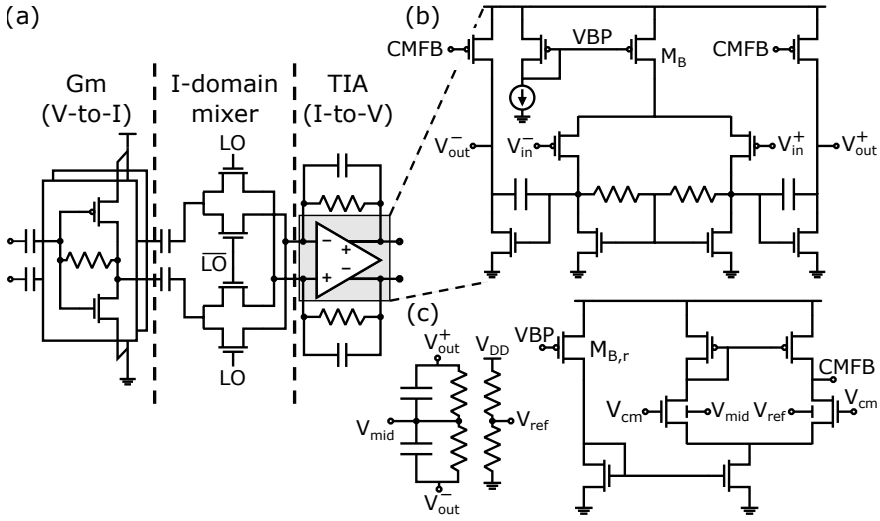

Fig. 4.4 Schematics of (a) the current-domain passive mixer, (b) the fully differential amplifier used in the transimpedance amplifier of the mixer, and (c) the common-mode feedback (CMFB) auxiliary amplifier for stabilizing the common mode of the fully differential amplifier.

The auxiliary amplifier must have an nMOS input stage so that its output can correctly bias the pMOS current sources of the main amplifier. However, designing this input stage when its input voltage must be relatively low is challenging. Therefore, the forward back-bias (FBB) capabilities of the FD-SOI technology are exploited to create a robust back-bias-driven amplifier. The voltage V_{CM} that biases the gates of the CMFB amplifier is also generated through a resistor ladder. Finally, the bias current for this auxiliary amplifier is generated from the same current source biasing the principal amplifier, but transistor $M_{B,r}$ is much smaller than M_B to minimize the power overhead of this auxiliary amplifier.

4.3.2 Simulation Results

Fig. 4.5 shows the pre-layout results of the mixer. Fig. 4.5(a) shows the NF as a function of frequency, resulting in an integrated NF of 24.8 dB. The integrated noise power breakdown on the different blocks of the mixer is shown in Fig. 4.5(e). The integrated noise power is dominated by the thermal noise due to the considered bandwidth. The Gm block that largely dominates

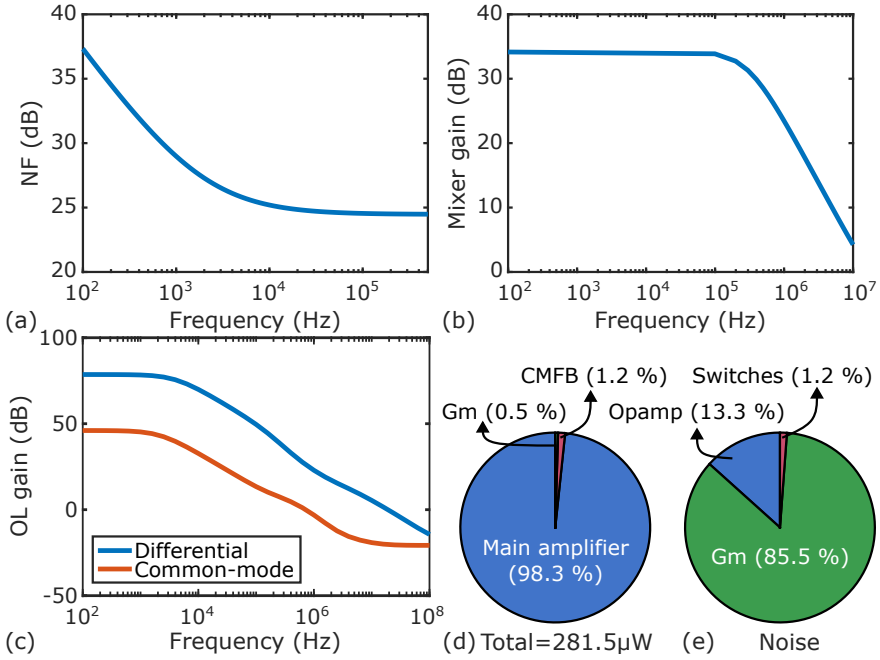

Fig. 4.5 Pre-layout simulation results of the mixer. (a) Noise figure (NF) versus frequency. (b) Mixer conversion gain. (c) Open-loop (OL) gain of the TIA's main amplifier in differential and common modes. (d) Power breakdown. (e) Integrated noise breakdown.

the noise, contributing more than 85 %. Given the chosen architecture, the transistors work in strong inversion, where the thermal noise is higher compared to the weak inversion region. This is a limitation of the design proposed in this chapter. It will be revised in Chapter 5 to drastically reduce its noise contribution at a low power overhead by biasing the transistors in weak inversion.

The open-loop gain of the main amplifier of the TIA is shown in Fig. 4.5(c), both in differential and common mode. In differential mode, the amplifier has a DC gain of 78.6 dB, a transition frequency of 20 MHz, and a phase margin of 60.9°. In common mode, the DC gain is 46 dB, the transition frequency is 750.1 kHz, and the phase margin is 87.6°. Ideally, the transition frequency in both modes should be similar for more robustness against common-mode perturbations. However, to save power, some of this robustness was sacrificed so the contribution of the auxiliary amplifier to the total

Table 4.3 Pre- and post-layout simulation results comparison for the mixer.

	Pre-layout	Post-layout
Power	281.48 µW	100 µW
Integrated NF	24.8 dB	30 dB
DC gain	34.3 dB	32.3 dB
Bandwidth	296.8 kHz	74.07 kHz

power consumption was negligible, as shown in the power breakdown in Fig. 4.5(d).

Fig. 4.5(b) shows the Bode diagram of the complete mixer with a DC gain of 34.3 dB and a bandwidth of 296.8 kHz. This is below the bandwidth of the TIA but still higher than the most common LoRa bandwidth of 250 kHz. This drop in bandwidth is probably due to the dominant pole of the main amplifier. A decrease in the gain of the amplifier affects the conversion gain of the switches doing the frequency down-conversion, which supposes a perfect virtual ground at the input of the amplifier.

In post-layout simulations, all performance metrics of the mixer are significantly degraded. Table 4.3 shows the comparison between pre- and post-layout simulations. The significant drop in power consumption suggests that the performance of the main amplifier must have significantly changed, directly impacting all other performance metrics of the mixer. Indeed, the DC point at the amplifier's pMOS input stage increased from 325 mV to 616 mV. Layout proximity effects are probably the cause for this change in DC point. This directly affects the current source biasing the amplifier, explaining the significant reduction of power consumption, which in turn impacts the gain of the amplifier and its noise. This explains why the other performance metrics of the mixer are also affected. Unfortunately, this issue was found after the chip went to fabrication. In the second prototype, presented in Chapter 5, the architecture of this CMFB auxiliary amplifier is modified for more robustness to avoid repeating this issue.

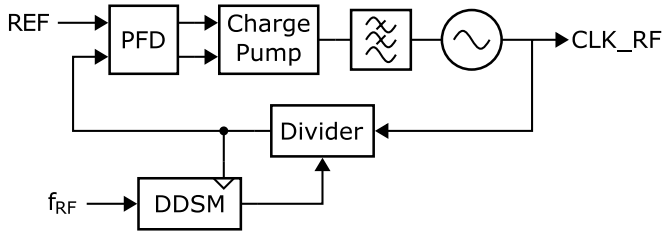

Fig. 4.6 Fractional-N phase-locked loop.

4.4 Programmable Frequency Synthesis

Commercial LoRa transceivers synthesize the carrier frequency from a 32-MHz crystal oscillator. Generating 868 MHz from this reference frequency, and with channels separated by 200 kHz [41], requires a fractional-N PLL instead of an integer-N PLL as the one designed in Section 3.3.3.

Frequency dividers can work with integer division factors only. Implementing a fractional one relies thus on the dithering between integer factors with the right proportion, which will give, on average, the fractional division factor of interest. For example, the frequency synthesizer for a LoRa transceiver can achieve a division factor of 27.125 by dividing the input clock by 28 once every 8 output clock cycles (12.5 % of the time) and by 27 the other clock cycles.

The main problem with this approach is that, although it works well on average, it creates spurious tones at the output of the PLL due to its instantaneous behavior. While dividing the output clock by an integer factor instead of the exact fractional one, the phase error at the output of the PFD accumulates, returning to zero after increasing the integer factor by one and creating a periodic signal [87]. Its period depends thus on the fractional division factor, creating the often called fractional spur at the output. In the above example, the fractional spur would appear at an eighth of the reference frequency, i.e., 4 MHz, because every 8 clock cycles the pattern repeats itself.

A better alternative, and the mainstream choice for fractional-N PLLs in the literature, is to exploit the noise-shaping capabilities of digital delta-sigma modulators (DDSMs) to push the quantization noise to high frequencies so that it can be canceled by the low-pass filter of the loop. Fig. 4.6 shows the simplified schematic of a fractional-N PLL, where the input of the

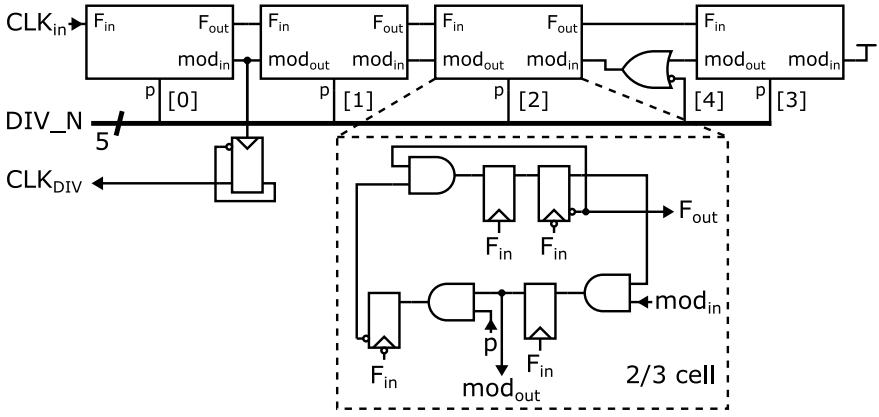

Fig. 4.7 Schematic of the multi-modulus divider.

DDSM, f_{RF}, is highlighted. This digital word controls the fractional division factor and is used to select the communication channel. In the next sections, each of the new or modified blocks with respect to an integer-N PLL will be discussed in detail.

4.4.1 Multi-Modulus Divider

A simple counter is sufficient for integer-N PLLs that need a fixed division factor. For fractional-N PLLs, where the division factor changes continuously, a more optimized architecture is needed instead. The alternative studied here is the modular design presented in [88].

Fig. 4.7 shows the schematic of the implemented multi-modulus divider (MMD). It is composed of cascaded 2/3 divider cells that divide the frequency of their input clock (F_{in}) by 2 or 3, depending on the *mod* and *p* signals. They divide by 3 only when both signals are high. The mod signal is generated by the rightmost cell and propagates leftward, being re-clocked at each cell. The p signal controls the division factor of the divider and is thus driven by the DDSM.

In the design proposed in [88], the output divided clock is taken as the mod signal input to the leftmost 2/3 cell. Even though the frequency of this mod signal is indeed the one of interest, its duty cycle is far from 50 % and decreases with the division factor, as shown in Fig. 4.8(b). This could be problematic for digital circuits using this signal as a clock, as is the case with the DDSM in the PLL. To guarantee a 50-% duty cycle, the mod signal

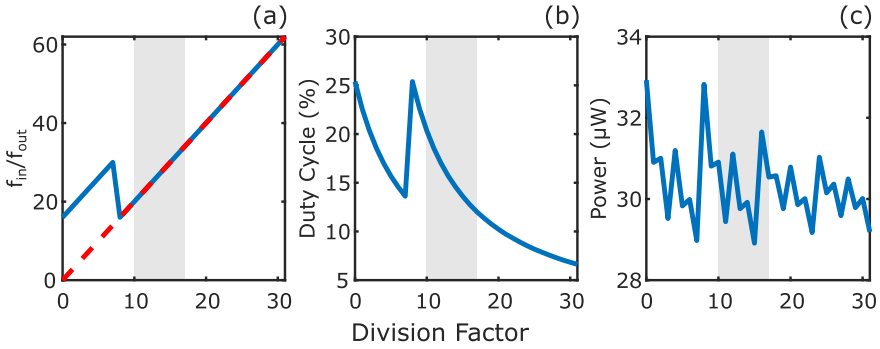

Fig. 4.8 Post-layout simulation results of the multi-modulus divider as a function of the input division factor. The shaded area represents the division factors needed in the PLL. (a) Ratio between input and output frequency. The red dashed line represents the ideal ratio. (b) Duty cycle of the input mod signal of the leftmost 2/3 cell. (c) Power consumption.

is first divided by 2 before outputting the divided clock.

As the 2/3 cells are only able to divide by 2 or 3, cascading n cells allows for a division factor between 2^n and $2^{n+1} - 1$ [88]. This range does not completely cover all the division factors needed in the PLL for any n. To remedy this, an OR gate can be used to disable the rightmost 2/3 cell for some division factors, as shown in Fig. 4.7. The divider can thus be used as a cascade of either 3 or 4 cells, extending sufficiently the range to cover the division factors of interest. As shown in Fig. 4.8(a), the range extends now from 8 to 31 but stops working below 8 as expected.

The divider is implemented with LVT standard digital cells and consumes, on average, 30.34 µW in post-layout simulations. The power consumption as a function of the division factor is shown in Fig. 4.8(c).

4.4.2 Digital Delta-Sigma Modulator

Digital delta-sigma modulators are a type of modulator that reproduces the signal at their input with a reduced number of bits while shaping the quantization noise.

The output signal y[n] can be written in the z domain as a function of the input signal x[n] and the quantization noise e_q[n] [89]

$$Y(z) = \mathrm{STF}(z)X(z) + \mathrm{NTF}(z)E_q(z). \tag{4.2}$$

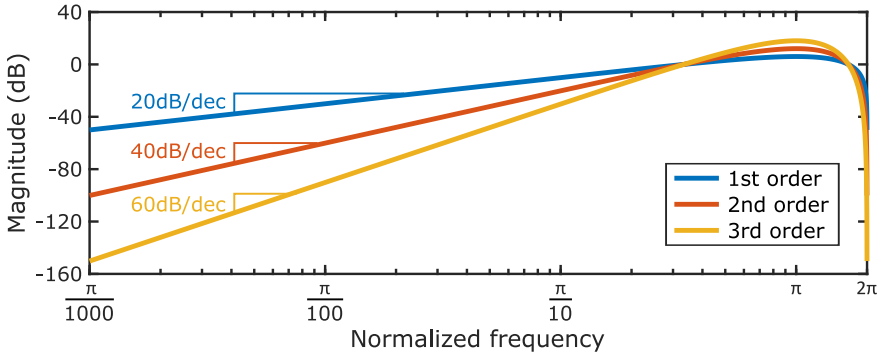

Fig. 4.9 Z-domain noise transfer function of the DDSM for three different orders.

The signal transfer function (STF) is typically equal to 1/M, representing the fact that the output signal has a reduced number of bits compared to the input signal. The noise transfer function (NTF) has a more complex expression due to the noise shaping effect of the DDSM:

$$(1 - z^{-1})^L, \tag{4.3}$$

where L represents the order of the DDSM. This transfer function describes a high-pass filter in the z domain whose slope is $+20L$ dB/dec, as shown in Fig. 4.9. It implies that the white quantization noise is reduced inside the signal bandwidth and pushed to high frequencies, where it can be filtered without affecting the signal of interest.

The error feedback modulator shown in Fig. 4.10(a) is an example of a first-order DDSM. This modulator owes its name to the fact that the quantization error is the quantity that is fed back to the input instead of the output [89]. It has the advantage of having a simple digital implementation, needing only a full adder with overflow flag and a register, as shown in Fig. 4.10(c) [90, 91]. The overflow flag represents the quantized single-bit output, while the sum without overflow calculates the quantization error directly. From its linearized model depicted in Fig. 4.10(b) [89], we can write in the z domain

$$V(z) = X(z) + E(z) \tag{4.4}$$

$$Y(z) = \frac{1}{M} V(z) + E_q(z) \tag{4.5}$$

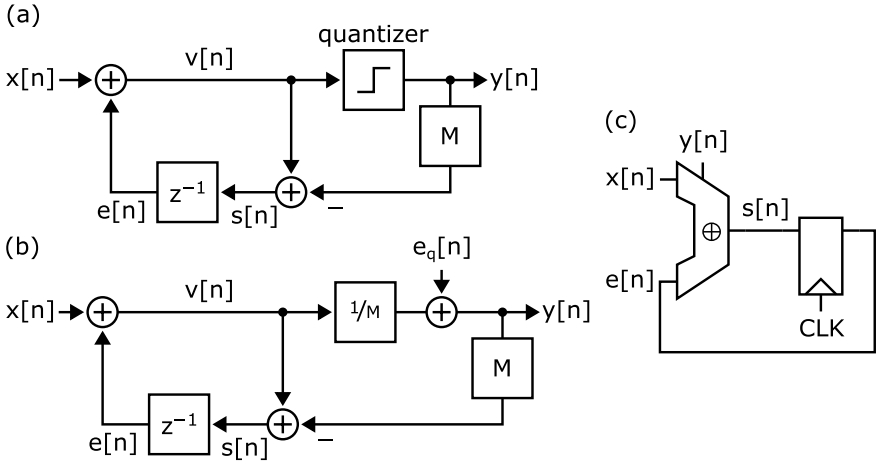

Fig. 4.10 First-order error feedback modulator (a) block diagram, (b) linearized model, and (c) digital implementation.

$$E(z) = z^{-1}\left(V(z) - MY(z)\right). \tag{4.6}$$

Inserting (4.5) in (4.6), a useful relation between the quantization error and the feedback error signal can be deduced:

$$E(z) = -Mz^{-1}E_q(z). \tag{4.7}$$

Finally, replacing (4.4) and (4.7) in (4.5), we obtain

$$Y(z) = \frac{1}{M}X(z) + (1 - z^{-1})E_q(z), \tag{4.8}$$

where we recognize using (4.2) the STF and NTF defined before for the first-order DDSM.

4.4.2.1 MASH Architecture

Several techniques exist in the literature to increase the order of DDSMs, including multi-bit quantizers, multi-loop architectures, and hybrid approaches [90]. However, the architecture with the best trade-off between complexity and noise shaping is the multistage noise shaping (MASH) architecture [90]. It consists of a cascade of error feedback modulators as the one in Fig. 4.10. For this reason, it is the simplest architecture as it only requires adders and registers to be implemented. Furthermore, it is inherently

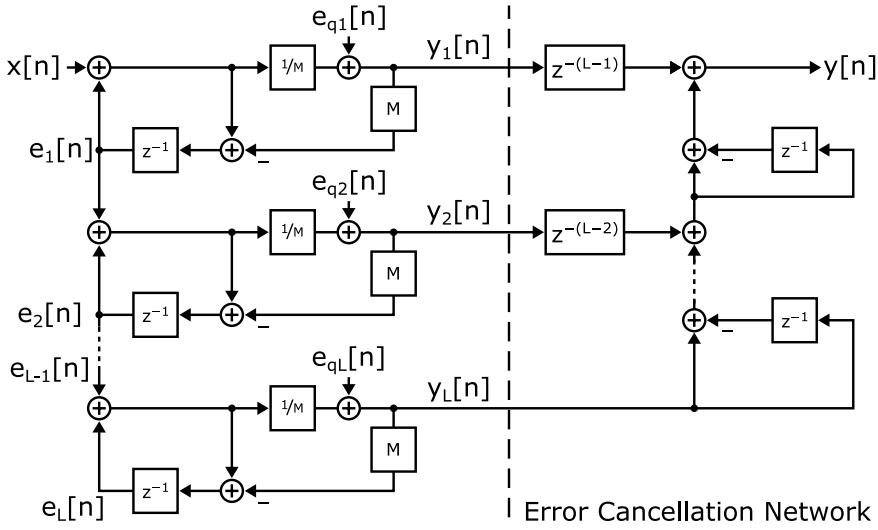

Fig. 4.11 Block diagram of an Lth-order MASH DDSM.

stable, its NTF has the same high-frequency behavior as other multi-loop architectures, and its STF does not affect the input, which is important in fractional-N PLLs [90]. The only disadvantage is its high probability of producing spurious tones at the output due to its periodic behavior. This issue, along with possible solutions, will be discussed later.

The block diagram of an Lth-order MASH DDSM is shown in Fig. 4.11. The core of the modulator is on the left, composed of a cascade of L first-order error feedback modulators. The input signal $x[n]$ is fed only to the input of the first stage, while the other stages receive at their input the $e[n]$ signal coming from the previous stage. On the right of the figure, the error cancellation network perfectly cancels the quantization error of all quantizers except that of the last stage, which is filtered with the NTF in (4.3).

We can do a similar analysis in the z domain as for the first-order case to obtain the global transfer function. As each stage is composed of the same error feedback modulator but with the feedback error signal of the previous stage at its input, we can rewrite (4.8) for the ith stage of a MASH DDSM as:

$$Y_i(z) = \frac{1}{M} E_{i-1}(z) + (1 - z^{-1}) E_{q,i}(z), \tag{4.9}$$

which gives the following expression after inserting (4.7):

$$Y_i(z) = -z^{-1}E_{q,i-1}(z) + (1 - z^{-1})E_{q,i}(z). \tag{4.10}$$

After the error cancellation network, the output of the second-order MASH DDSM, usually named MASH 1-1 due to the cascade of two first-order modulators, is given respectively by:

$$Y(z) = z^{-1}Y_1(z) + (1 - z^{-1})Y_2(z), \tag{4.11}$$

while the output of the third-order MASH DDSM, or MASH 1-1-1, can be expressed as:

$$Y(z) = z^{-2}Y_1(z) + z^{-1}(1 - z^{-1})Y_2(z) + (1 - z^{-1})^2Y_3(z). \tag{4.12}$$

Using (4.10), (4.11) and (4.12) are simplified to obtain the complete transfer function of the second- and third-order MASH DDSMs:

$$Y(z) = \frac{1}{M}z^{-1}X(z) + (1 - z^{-1})^2E_{q,2}(z), \tag{4.13}$$

$$Y(z) = \frac{1}{M}z^{-2}X(z) + (1 - z^{-1})^3E_{q,3}(z). \tag{4.14}$$

Again, we recognize the STF and NTF we were looking for, with the exponent in the NTF increasing as expected with the order of the DDSM. Furthermore, the error cancellation indeed perfectly cancels the quantization error of all quantizers but the last one, which is the only one appearing in the transfer function. Notice in the STF that the input is delayed. This is not problematic as the input to the DDSM in a fractional-N PLL is mostly constant; it only changes to switch to another frequency channel.

4.4.2.2 Integration in the Fractional-N PLL

In fractional-N PLLs, the ratio between the output frequency f_{out} and the reference frequency f_{ref} is a rational number. Its integer part N_{int} is achieved through the division range of the multi-modulus divider, while its decimal part is achieved through the DDSM. Depending on the input signal of the DDSM (f_{RF}), we can calculate the output frequency with the following expression [89]

$$f_{\text{out}} = \left(N_{\text{int}} + \frac{f_{\text{RF}}}{2^n}\right) f_{\text{ref}}, \tag{4.15}$$

where n is the number of bits of the DDSM. To calculate the needed input signal, it suffices to isolate this variable from the previous equation:

$$f_{\mathrm{RF}} = 2^n \left(\frac{f_{\mathrm{out}}}{f_{\mathrm{ref}}} - N_{\mathrm{int}} \right). \tag{4.16}$$

It must be highlighted that given the chosen architecture of multi-modulus divider in Fig. 4.7, where a divide-by-2 cell is added at the output to ensure a 50-% duty cycle, half the frequency is actually generated. To correct this, we use twice the reference frequency in (4.16) to obtain the correct input signal for the DDSM, increasing it to 64 MHz.

The number of bits used in the DDSM determines its resolution, as can be seen in (4.15). Indeed, when increasing f_{RF} by one, the output frequency increases by

$$\Delta f = \frac{f_{\mathrm{ref}}}{2^n}. \tag{4.17}$$

From which we can deduce the formula for the number of bits needed for a certain resolution:

$$n = \log_2 \frac{f_{\mathrm{ref}}}{\Delta f}. \tag{4.18}$$

The specification for the frequency resolution can be deduced from the maximum CFO tolerated in a LoRa receiver. There is no sensitivity degradation for a CFO up to 25 % of the signal bandwidth [21]. Taking the smallest bandwidth of 125 kHz, the needed resolution is 31.25 kHz. This results in a DDSM of 11 bits when using (4.18), but we increase this to 12 as a precaution.

4.4.2.3 Spur Tones in the MASH Architecture

It is well-known in the literature that DC inputs are the worst-case inputs for DDSMs [91, 92, 93]. With constant input signals, they behave as finite-state machines and thus produce periodic outputs. This is caused by the fact that the modulation is done in the digital domain with a finite word length [89, 92, 93, 94]. However, depending on the modulation order, the initial conditions, and the actual input value, the length of the periodic sequence varies. As the sequence gets longer, the quantization error energy spreads over a higher number of tones and approaches a white spectrum [94]. To verify this, the quantization error must fulfill three conditions: have a zero mean, an average power of $\mathrm{LSB}^2/12$, and an autocorrelation function which is zero for all non-zero lag indexes [91]. The quantization error needs to be

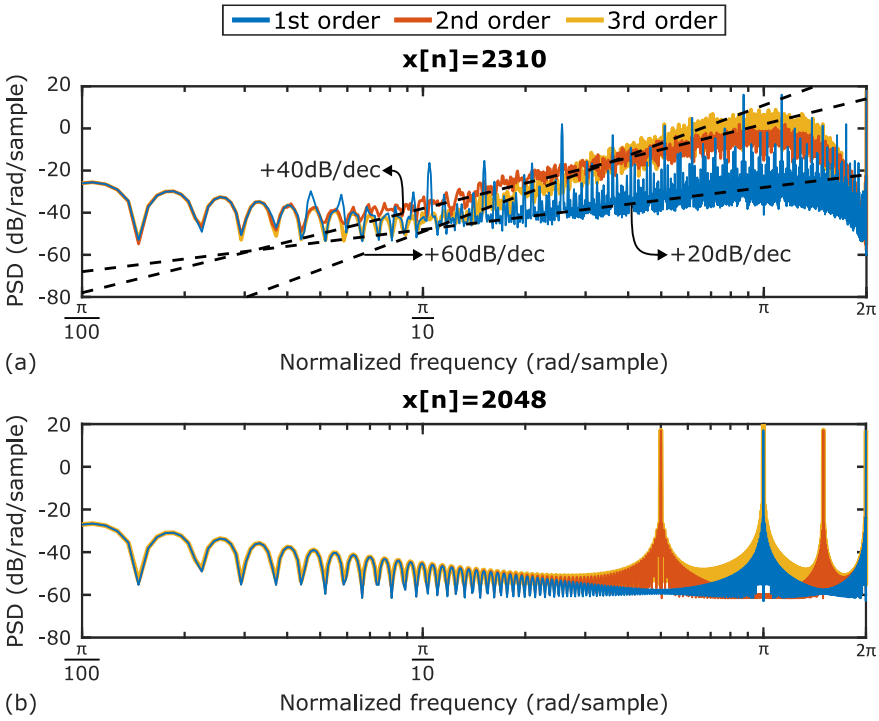

Fig. 4.12 Power spectral density (PSD) of the output of a 12-bit MASH DDSM of three different orders for a constant input signal equal to (a) 2310, and (b) 2048, and without initial conditions.

white so that it is shaped as expected by the DDSM.

The impact of the input value, the modulation order, and the initial conditions on the power spectral density (PSD) of the DDSM output can be observed through simulation. For this, three 12-bit MASH DDSMs of varying order were simulated for two different input signals and two different initial conditions.

Starting with the case without initial conditions, where all registers are reset at the beginning of the simulation, the impact of the input signal can be highlighted. Fig. 4.12(a) shows the PSD for an input signal equal to 2310. This input signal corresponds to the control word needed to generate 868 MHz in a fractional-N PLL with a 32-MHz reference clock and the multi-modulus divider of Fig. 4.7. As shown with the help of the dashed lines, the quantization noise is shaped with the expected slope for the three

modulation orders. However, the spectrum of the first-order modulator contains significant spurs at several frequencies. This can be explained through the PSD of the quantization noise shown in Fig. 4.13(a)-(c) along with its mean and variance. The quantization noise was normalized to have a unitary LSB. Even though the mean and variance correspond to those of a uniformly distributed noise, the PSD of the first-order modulator has significant spurs, unlike that of the other two orders. This is because the condition on the autocorrelation is not respected, as shown in Fig. 4.14(a). Its autocorrelation function has non-zero values for non-zero lag indexes. This behavior of first-order modulators is exhaustively studied in [95].

Considering now an input signal equal to 2048, we can see strong spurs on the output spectrum in Fig. 4.12(b) without any noise shaping. This is considered the worst-case input to a DDSM as only the most-significant bit (MSB) is one. The noise spectrum in Fig. 4.13(a)-(c) and its autocorrelation function in Fig. 4.14(c) show that in this case the quantization noise is far from being white, and consequently, neither of the three conditions is respected. Moreover, signals in this case have a very short period, with their sequence length being below four. The noise energy is thus shared among a few tones and results in strong spurs.

Dithering and specific initial conditions are the main solutions found in the literature to avoid these strong spurs in MASH DDSMs [89, 90, 91, 94]. We chose to work with the initial conditions, as it is a simpler solution for the same results as dithering without the extra hardware. Extensive simulations done in [94] show that having an odd initial condition in the first register of a MASH DDSM maximizes the sequence length as it guarantees a sequence length close to or equal to the maximum possible given the number of bits in the modulator. However, the mathematical approach used in [91] proves that this only guarantees a white quantization noise for modulators of order three or more. We can verify this in the case of a 12-bit DDSM.

Considering that all registers in the modulator are reset at the beginning of the simulation, except the first set to one, we obtain the output spectra shown in Fig. 4.15 for the two input signals studied before. For an input equal to 2310, no visible difference can be seen with the initial conditions. However, for the worst-case input, we see that the third-order modulator achieves the expected noise shaping without spurious tones. A slight noise shaping can be seen for the second-order one, but it still has strong spurious tones. The first-order modulator remains unchanged with the initial conditions. Comparing the PSD of the quantization noise with and without initial conditions in Fig. 4.13, we confirm the previous observations.

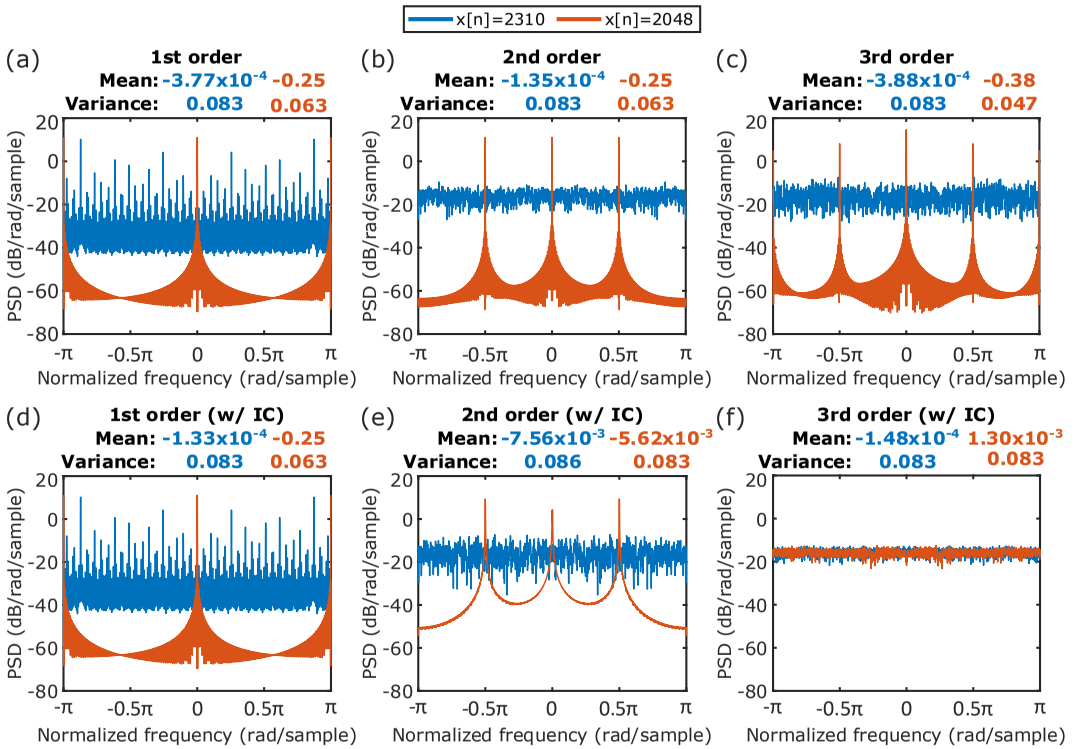

Fig. 4.13 Power spectral density (PSD) of the quantization error in a 12-bit MASH DDSM for two different constant input signals. The modulator is of (a,d) first order, (b,e) second order, and (c,f) third order. (a)-(c) do not have initial conditions, whereas (d)-(f) have an initial condition in the first modulator equal to one.

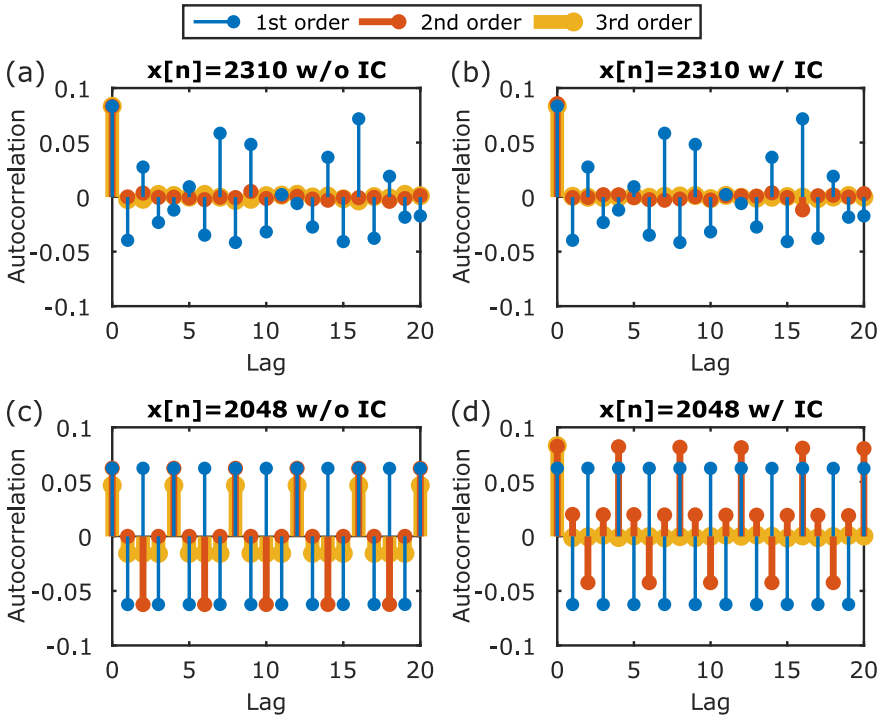

Fig. 4.14 Autocorrelation function of the quantization error in a 12-bit MASH DDSM of three different orders for a constant input signal equal to (a) 2310, without initial conditions; (b) 2310, with an initial condition in the first modulator equal to one; (c) 2048, without initial conditions; and (d) 2048, with an initial condition in the first modulator equal to one.

The third-order modulator has a white spectrum for both input signals, whereas in the other two cases, there are strong spurious tones for an input signal equal to 2048. In particular, for the second-order modulator, we can see that the initial conditions made the quantization error have the expected mean and variance, but its autocorrelation function in Fig. 4.14 still has non-zero values for non-zero lag indexes. Only the third-order modulator respects this property for the worst-case input signal.

Even though the third-order modulator has the best noise-shaping performance of the three orders considered, it may not be the best choice for a PLL due to the slope of the quantization noise. The PLL low-pass filter must be able to sufficiently filter the quantization noise pushed to higher

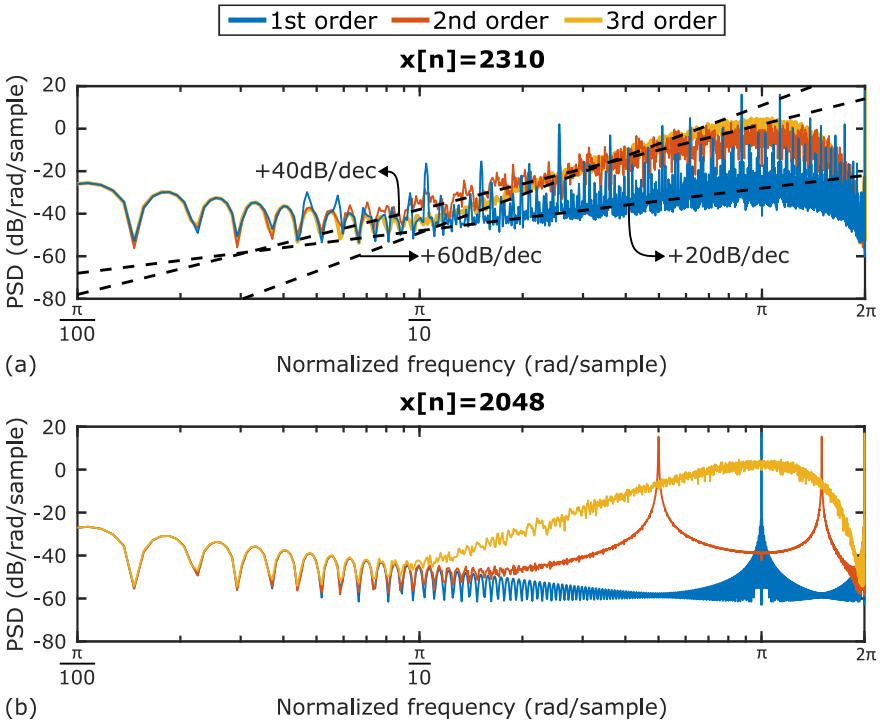

Fig. 4.15 Power spectral density (PSD) of the output of a 12-bit MASH DDSM of three different orders for a constant input signal equal to (a) 2310, and (b) 2048, and for an initial condition in the first modulator equal to one.

frequencies to avoid having the total phase noise at the output dominated by the shaped quantization noise. As will be discussed later, this may introduce instability risks in the PLL and an overall more complex sizing of the filter. There is thus a trade-off in the design of the DDSM involving power consumption and area at the modulator level, and also stability at the PLL level, along with other less straightforward trade-offs due to the interaction with the other blocks in the loop. It is thus interesting to develop a programmable modulator so the order can be chosen at run-time to exploit these trade-offs.

4.4.2.4 *Implementation and Results*

Fig. 4.16 shows the block diagram of the implemented programmable DDSM based on the MASH 1-1-1 architecture. It was implemented in

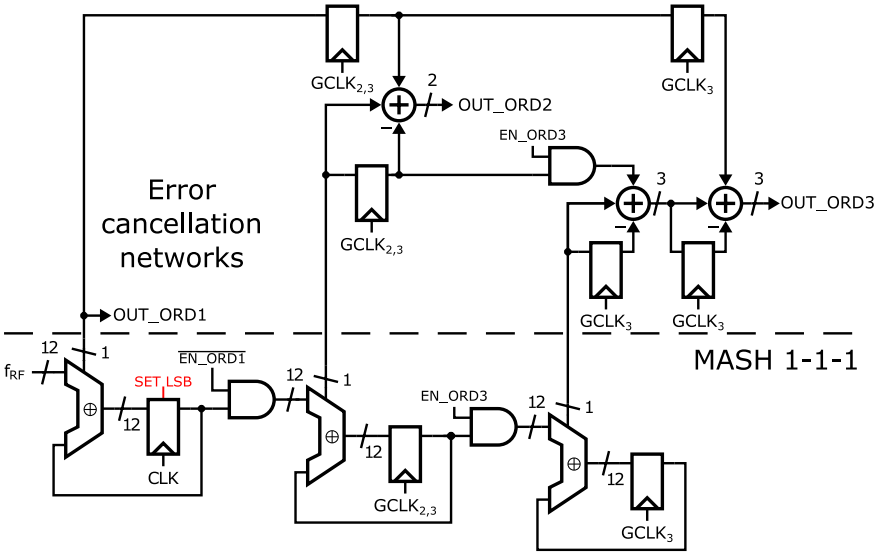

Fig. 4.16 Schematic of the programmable DDSM based on the MASH 1-1-1 architecture.

GlobalFoundries 22-nm FD-SOI technology using LVT transistors. In red is highlighted the initial condition necessary for a correct operation of the third-order modulator. Notice that both clock and data gating are used to avoid unnecessary toggling in blocks that are not used for a certain modulation order.

We can see the effect of these power-saving strategies in Table 4.4 as power savings of 43.56 % and 41.88 % are obtained when passing from a third-order to a second-order modulator and from a second-order to a first-order modulator, respectively. The leakage power remains negligible even though LVT transistors were used.

The output of the DDSM is an unsigned integer with a varying number of bits depending on the modulation order, as shown in Fig. 4.16. A decoder is thus needed between this output and the multi-modulus divider. For a fractional-N PLL with a 32-MHz reference clock and the multi-modulus divider of Fig. 4.7, which includes notably a divide-by-2 cell for duty-cycle concerns, the decoder has the logic shown in Table 4.5.

Table 4.4 Post-synthesis power results of the programmable DDSM.

	Order		
	1	2	3
Leakage (µW)	0.03	0.03	0.03
Internal (µW)	0.77	1.95	3.94
Switching (µW)	1.67	2.27	3.56
Total (µW)	2.47	4.25	7.53
Power savings	–41.88 %	–43.56 %	-

Table 4.5 Logic for division factor generation.

Output	0	1	2	3	4	5	6	7
1st order	13	14						
2nd order	13	14	15	12				
3rd order	13	14	15	16	17	10	11	12

4.4.3 Loop Filter

The additional noise source from the DDSM forces us to review the design of the loop filter in the PLL to make sure the shaped quantization noise is correctly filtered. We saw before that an Lth-order DDSM produced a shaped quantization noise at its output with a slope of +20L dB/dec. However, due to the integration effect of the multi-modulus divider, this slope is reduced to +20$(L-1)$ dB/dec at the divider output [66].

The contribution of the DDSM to the output phase noise is obtained by the following expression [66]:

$$S_{\Phi_{\text{out}}}(\omega) = T|G(j\omega)|^2(2\pi)^2(2\sin(\omega T/2))^{2(L-1)}\frac{1}{12}, \tag{4.19}$$

where T is the period of the reference clock and G(jω) is a low-pass filter defined in the Laplace domain as:

$$G(s) = \frac{\frac{I_{\text{CP}}K_{\text{VCO}}H_{\text{filt}}(s)}{2\pi Ns}}{1+\frac{I_{\text{CP}}K_{\text{VCO}}H_{\text{filt}}(s)}{2\pi Ns}}. \tag{4.20}$$

Therefore, the loop filter (H_{filt}) must introduce one less pole than the order of the DDSM. For example, a second-order PLL with the loop filter shown

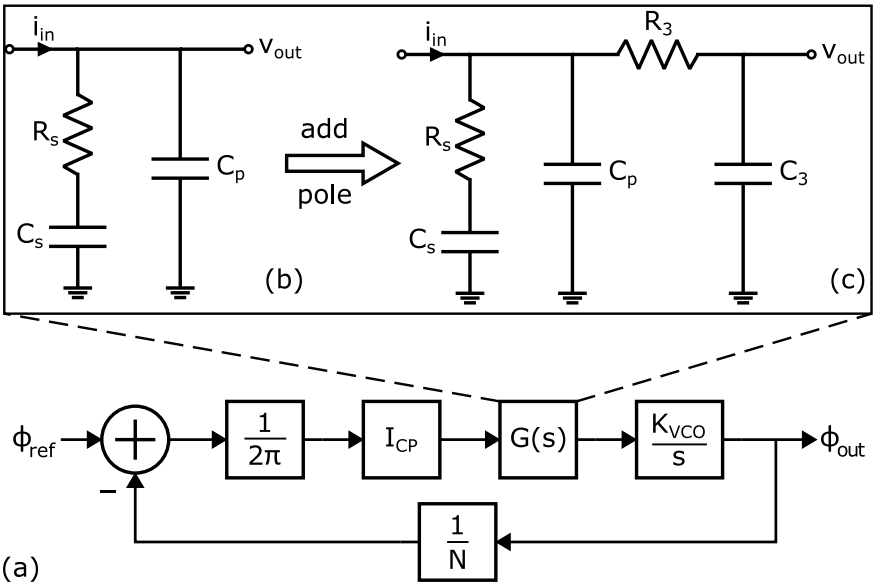

Fig. 4.17 (a) Linearized model of the PLL. Schematic of (b) the second-order loop filter and (c) the third-order loop filter.

in Fig. 4.17(b) can filter correctly the quantization noise of a DDSM of up to order 2. The quantization noise of a third-order DDSM would produce a flat phase noise at high frequencies.

A solution for this is to add an additional pole to the loop filter, as shown in Fig. 4.17(c). A careful design of such a filter requires careful sizing to avoid stability issues in the PLL. This methodology is extracted from [96].

4.4.3.1 *Sizing of a third-order filter*

The transfer function of the third-order loop filter is given by:

$$H_{\text{filt}}(s) = \frac{1 + s\tau_2}{sA_0(1 + s\tau_1)(1 + s\tau_3)}, \tag{4.21}$$

with

$$A_0 = C_\text{s} + C_\text{p} + C_3, \tag{4.22}$$

$$A_1 = A_0(\tau_1 + \tau_3) = R_\text{s}C_\text{s}(C_\text{p} + C_3) + R_3C_3(C_\text{s} + C_\text{p}), \tag{4.23}$$

$$A_2 = A_0\tau_1\tau_3 = R_\text{s}R_3C_\text{s}C_\text{p}C_3, \tag{4.24}$$

and

$$\tau_2 = R_s C_s \tag{4.25}$$

The open-loop transfer function of the linearized model of the PLL shown in Fig. 4.17(a) is therefore given by:

$$H(s) = \frac{I_{CP} K_{VCO}}{2\pi N s} \frac{1 + s\tau_2}{s A_0 (1 + s\tau_1)(1 + \tau_3)}. \tag{4.26}$$

Defining the unity-gain frequency as ω_c, the magnitude of the transfer function in (4.26) must be unity at this frequency, and we can write

$$A_0 = \frac{I_{CP} K_{VCO}}{2\pi N \omega_c^2} \sqrt{\frac{1 + (\omega_c \tau_2)^2}{(1 + (\omega_c \tau_1)^2)(1 + (\omega_c \tau_3)^2)}}. \tag{4.27}$$

This gives us the fourth equation in a system with five unknowns. The fifth equation needed to have a unique solution for the system is obtained by maximizing C_3. This capacitor is connected in parallel with the input capacitance of the VCO; therefore, maximizing C_3 minimizes the impact of this parasitic capacitance on the system stability. The maximization of C_3 is done with respect to C_p. Using (4.22), (4.23), and (4.24), we can write C_3 as a function of C_p and the other system parameters as:

$$C_3 = \frac{-\tau_2^2 C_p^2 + \tau_2 A_1 C_p - A_2 A_0}{\tau_2^2 C_p - A_2}. \tag{4.28}$$

Setting the derivative of this function to zero, we obtain the value of C_p that maximizes C_3:

$$C_p = \frac{A_2}{\tau_2^2} \left(1 + \sqrt{1 + \frac{\tau_2}{A_2} (\tau_2 A_0 - A_1)} \right). \tag{4.29}$$

Now that we are able to calculate the component values from the filter coefficients, we need to link these coefficients to the stability of the PLL. From (4.26), we can deduce that the phase margin is given by:

$$\phi_M = \tan^{-1}(\omega_c \tau_2) - \tan^{-1}(\omega_c \tau_1) - \tan^{-1}(\omega_c \tau_3). \tag{4.30}$$

If we want to maximize the phase margin at the unity-gain frequency, then

the following condition must be met

$$\frac{\tau_2}{1+(\omega_c\tau_2)^2} - \frac{\tau_1}{1+(\omega_c\tau_1)^2} - \frac{\tau_3}{1+(\omega_c\tau_3)^2} = 0, \tag{4.31}$$

which can be approximated to [96]

$$\tau_2 = \frac{1}{\omega_c^2(\tau_1+\tau_3)}. \tag{4.32}$$

However, maximizing the phase margin at ω_c is not always the best choice for lock time and spur attenuation [96], which is why we introduce the parameter γ in the following way:

$$\tau_2 = \frac{\gamma}{\omega_c^2(\tau_1+\tau_3)}. \tag{4.33}$$

Another useful parameter to define is the ratio between the two poles of the system:

$$T_{31} = \frac{\tau_3}{\tau_1}. \tag{4.34}$$

Finally, substituting (4.33) and (4.34) in (4.30) yields the following

$$\phi_M = \tan^{-1}\left(\frac{\gamma}{\omega_c\tau_1(1+T_{31})}\right) - \tan^{-1}(\omega_c\tau_1) - \tan^{-1}(\omega_c T_{31}\tau_1). \tag{4.35}$$

We can solve this equation numerically to obtain the value of τ_1, and then use (4.33) and (4.34) to obtain τ_2 and τ_3, which allows us to obtain the values of the five filter components.

4.4.4 Voltage-Controlled Oscillator

One common way of generating the four precise clock phases needed for the I and Q mixers requires a ring oscillator with an even number of stages [97, 98]. As shown in Fig. 4.18(b), fully differential delay elements are needed to ensure oscillations with a 4-stage ring oscillator. The multiplexers between the stages serve the same purpose of ensuring oscillations. Even though only two are required for this, two dummy multiplexers were included to have the same load for all delay elements. Unlike [97, 98] where BBCOs were used to exploit their capabilities of lower power consumption and better noise-power trade-off [51], we chose the VCO architecture shown

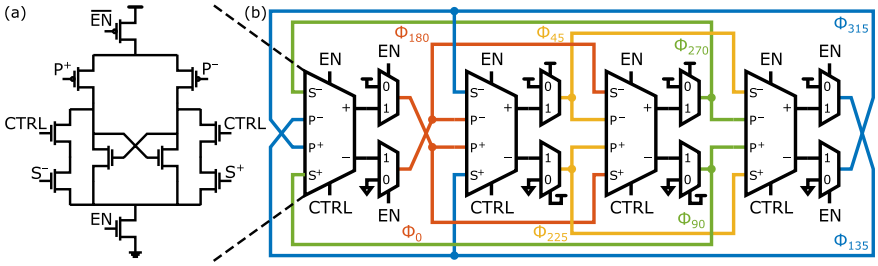

Fig. 4.18 Schematic of (a) the fully differential delay element, and (b) the voltage-controlled oscillator. RVT transistors are used for the delay element and LVT transistors for the multiplexers.

in Fig. 4.18(a), inspired by [99] for its low-noise capabilities. This choice avoids two issues that come with BBCOs. First, no negative voltage needs to be generated for driving the back-bias of pMOS transistors with FBB capabilities, which avoids area and power overheads. Second, possible reduced duty-cycle issues are avoided in case only the back-bias of nMOS transistors was used to control the BBCO.

Two changes were implemented in the VCO compared to the architecture proposed in [99]. The circuit was flipped to have the control voltage at the gate of nMOS transistors. This makes it slightly easier to design the PLL as the frequency increases when the control voltage increases, and no modification to the conventional PLL architecture needs to be done. The second change was to include header and footer transistors for disabling the oscillator, which is useful for how the programmability is implemented in this block.

The VCO dominates the power of the PLL and contributes greatly to the phase noise. This makes it the perfect candidate for exploiting the noise-power trade-off of PLLs. As shown in Fig. 4.19(a), increasing the number of parallel delay elements decreases the phase noise of the ring oscillator without changing its frequency, but at the cost of an increased power consumption [51]. The noise-power trade-off in oscillators can be quantified with the oscillator FoM given by

$$\mathrm{FoM}[\mathrm{dBc/Hz}] = \mathcal{L}(\Delta f) + 20\log\left(\frac{\Delta f}{f_0}\right) + 10\log\left(\frac{P}{1\,\mathrm{mW}}\right), \qquad (4.36)$$

where $\mathcal{L}(\Delta f)$ is the phase noise at a Δf offset, f_0 is the oscillation frequency, and P is the power consumption. The FoM remains approximately un-

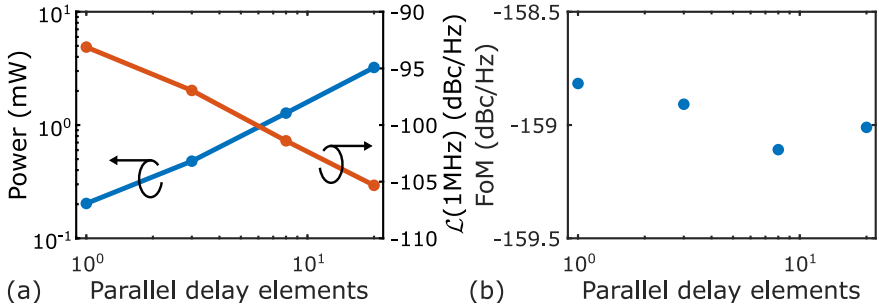

Fig. 4.19 Impact of the number of parallel delay elements on (a) the power and phase noise of the VCO, and (b) the oscillator FoM, from post-layout simulations.

changed, as depicted in Fig. 4.19(b), meaning that the noise-power trade-off of the oscillator is not degraded by the increase in the number of parallel delay elements. The FoM is close to the theoretical limit for ring oscillators of −165 dBc/Hz [51], showcasing the good noise-power trade-off in the VCO.

The programmable VCO is implemented as a bank of four parallel VCOs with 1, 3, 8, and 20 parallel delay elements per stage. They were not integrated into a single VCO to avoid noise degradation due to the switches. Instead, four different enable signals are generated in the digital controller of the PLL to disable the unused ring oscillators. A glitch-free multiplexer is then used at their outputs to select which oscillator is connected to the rest of the loop. Regular-threshold-voltage (RVT) transistors were used to implement the delay element to reduce the leakage when each VCO is disabled.

Another important characteristic of VCOs for the design of PLLs is their frequency gain. Fig. 4.20(a) shows the transfer function of the four parallel VCOs, and Fig. 4.20(c) their frequency gain as a function of the control voltage. Both were obtained from post-layout simulations. It is important to mention that two and four multiplexers had to be put in parallel for the VCOs with 8 and 20 parallel delay elements, respectively, to obtain an acceptable frequency range due to parasitics in the layout. The four VCOs have approximately the same frequency gain within 25 %. This allows the use of the same parameters in the rest of the blocks of the PLL without changing its dynamic behavior or its stability significantly.

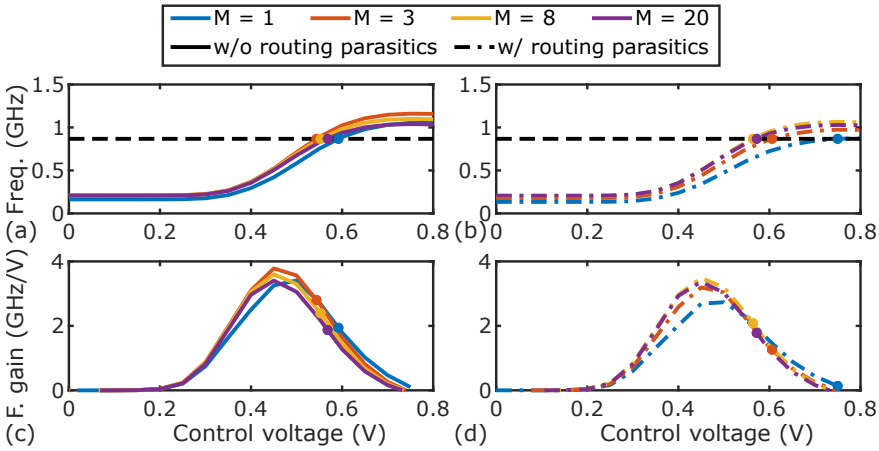

Fig. 4.20 (a)-(b) Transfer function of the four parallel VCOs, and (c)-(d) their corresponding frequency gain from post-layout simulations. The routing parasitics correspond to the routing of the four clock phases from the VCO to the glitch-free multiplexer inside the PLL.

4.4.5 Implementation and Post-Layout Results

Fig. 4.21 shows the complete schematic of the programmable fractional-N PLL. In the LPF, pass-gates are used to select the filter order. Table 4.6 summarizes the degrees of freedom and their ranges. A 6-bit current DAC controls the CP current. One 8-bit and two 5-bit binary banks of capacitors were implemented for C_s, C_p, and C_3, respectively. Finally, two series banks of resistors with logarithmically spaced values were implemented for R_s and R_3. The 4-bit configuration value for the resistances was converted to 16 bits with a one-hot code before arriving at the switches controlling the series banks. All capacitors were implemented with MOM capacitors and all resistors with unsilicided narrow high-resistance N+ polysilicon resistors.

With these values, the resulting PLL bandwidth, phase margin, γ, and T_{31} are shown in Fig. 4.22. The color indicates the local density of points; a higher density means that more combinations of the degrees of freedom give the same values of the bandwidth, the phase margin, and the two design parameters. The density was estimated from 1×10^6 random combinations of the degrees of freedom because using all the 3.2×10^9 combinations poses memory and computation time problems. The green zone indicates the

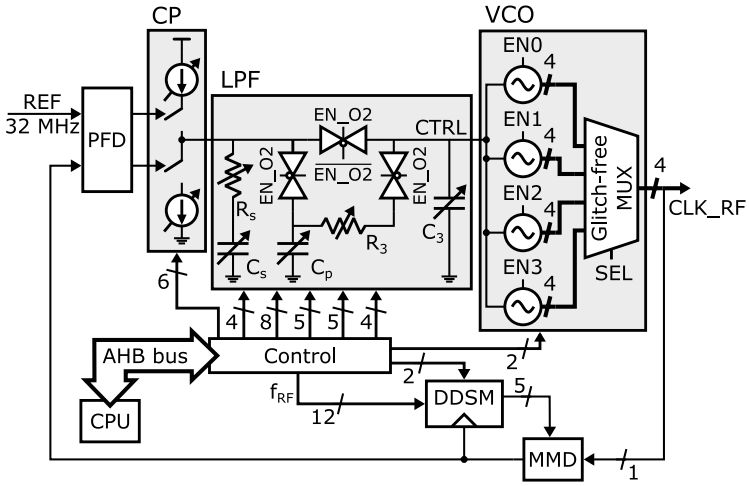

Fig. 4.21 Schematic of the programmable PLL.

values that are considered valid in terms of bandwidth and phase margin. As was the case in Chapter 3, PLLs with a bandwidth higher than $f_{ref}/10$ are discarded because they are not able to filter the control signal correctly. For the phase margin, values above 30° are taken to ensure the stability of the PLL. A limit of 45° would be better for the stability of the PLL, but it is better to relax the constraints since passive components are particularly subject to process variations and can impact the analytic predictions of the PLL performance.

Characterizing the phase noise of a PLL cannot be done as for VCOs where noise contributions are calculated around a steady state, because PLLs are not periodic, especially fractional ones. Instead, we must rely on noise transient simulations and estimate the phase noise from the output clock signal. This can be done, as detailed in [49], by estimating the PSD of the extra phase of the clock. For a vector t containing the times when the clock completes a period, e.g., times when the clock crosses half the supply on the positive edges, we can calculate the extra phase by comparing this vector to an ideal one t_{id}:

$$\phi_e = 2\pi f_0(t - t_{id}), \tag{4.37}$$

where f_0 is the frequency of oscillation. Estimating the PSD of this extra phase signal (S_ϕ), we can obtain the phase noise of the clock signal. Note that

Table 4.6 Range of degrees of freedom.

Parameter	Range	Number of bits
DDSM order	1, 2, or 3	2
Parallel delay elements in the VCO	1, 3, 8, or 20	2
Charge pump current	15 to 960 nA	6
Series capacitance (C_s)	250 fF to 64 pF	8
Series resistance (R_s)	5 kΩ to 20 MΩ	4
Parallel capacitance (C_p)	250 fF to 8 pF	5
Third-order-filter resistance (R_3)	1 to 200 kΩ	4
Third-order-filter capacitance (C_3)	0 to 7.75 pF	5

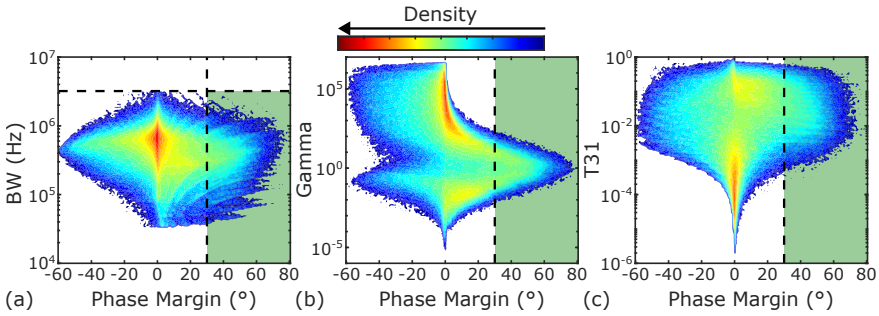

Fig. 4.22 Range of (a) bandwidth, (b) γ, and (c) T_{31} as a function of the phase margin. The green zone shows the valid regions for a phase margin above 30° and a bandwidth below 10 % of the 32-MHz reference frequency.

the units of this phase noise are simply dB/Hz because it is not normalized to the carrier power. However, this is equivalent to the PSD of the voltage clock signal normalized by the carrier power, in dBc/Hz.

Using this methodology, Fig. 4.23 shows the post-layout simulation results of the phase noise for the four parallel VCOs. The rest of the PLL parameters are constant to give a bandwidth of 480 kHz and a damping factor of 0.45 on average, using a second-order filter and DDSM. The average values are given due to the small variations in the VCO gain when the number of parallel delay elements changes. The first notable thing in the figure is that the smallest VCO has less phase noise than the rest, which is the opposite of what was expected. The clock routing inside the PLL changed the transfer function of this small VCO significantly because the PLL locked at a control voltage close to the supply voltage, as shown in

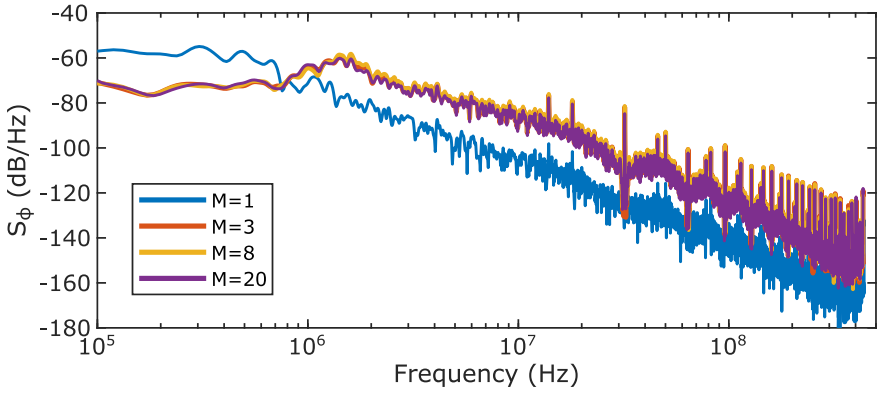

Fig. 4.23 Post-layout simulation results of the PLL phase noise. The PLL uses the second-order filter and DDSM with a bandwidth of 480 kHz and a damping factor of 0.45. The different curves represent the number of parallel delay elements (M) inside the VCO.

Fig. 4.20(b). The other three were not as impacted, probably because the parasitic capacitance of the clock routing becomes negligible compared to that of the delay elements as more of these are put in parallel. The VCO should be designed at the same time as the buffers involved in its routing to the rest of the PLL to avoid this issue.

The other thing we can see in Fig. 4.23 is that the phase noise is constant for the other three VCOs. This means that the VCO is not the dominant noise source. We can see in Fig. 4.24(c) that the contribution of the quantization noise of the DDSM to the output phase noise obtained using (4.19) is the dominant noise source at high offset frequencies. This is unfortunate as we lose an important lever in the noise-power trade-off of the PLL when the VCO does not dominate the phase noise.

The spurs in the phase noise do not come from the quantization noise; instead, they come from noise at the control voltage node. In Fig. 4.24(a), we can see the control voltage trace when the PLL is locked. There are significant periodic components that explain the presence of spurs in the output phase noise. The FFT of this trace, shown in Fig. 4.24(b), reveals peaks at 32 MHz and its harmonics. These are probably caused by clock injection in the charge pump as well as the loop actuation at the reference frequency. The other significant peak of the FFT appears at the frequency of oscillation of the VCO. This is due to capacitance coupling in the transistor whose gate is driven by the control voltage in Fig. 4.18(a). A higher capacitance

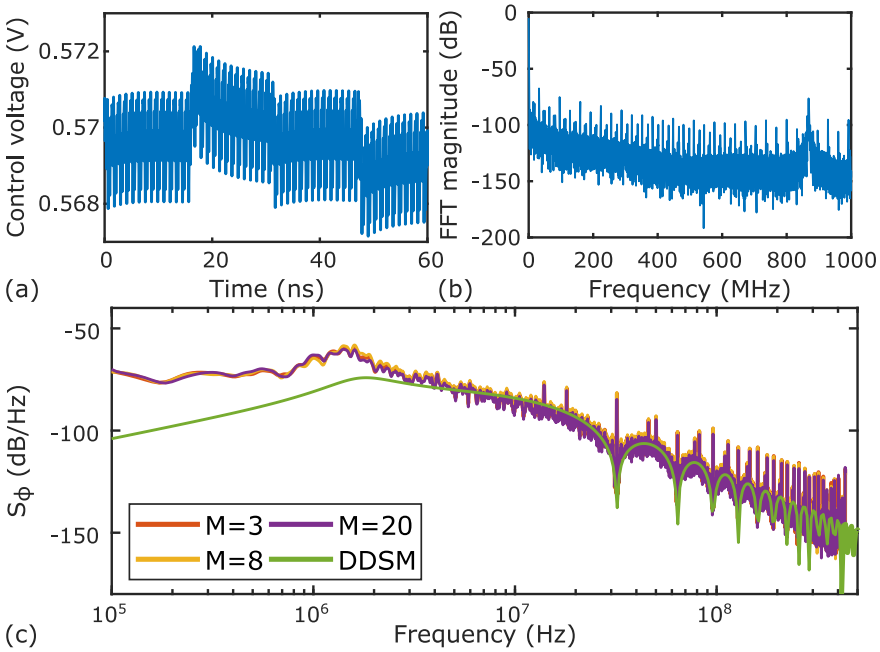

Fig. 4.24 (a) Control voltage of the VCO in lock. (b) Magnitude of the FFT of the control voltage. (c) Post-layout phase noise results, including the filtered quantization noise from the DDSM.

should be connected to the control node to reduce this effect, while carefully resizing the rest of the filter to avoid stability issues.

4.5 Measurement Results

The proposed programmable direct-conversion RF front end was prototyped in Global Foundries (GF) 22-nm FD-SOI technology. As shown in the microphotograph in Fig. 4.25(a), it occupies an area of 0.127 mm^2 on the $1.04 \times 1.04\,mm^2$ Cortex-M4 MCU chip codenamed CERBERUS. Fig. 4.25(b) shows the layout the RF front end.

The supply voltage had to be increased by 3.125 % to 0.825 V because the VCO was unable to reach the target frequency at a lower voltage. This is most probably explained by a slow process corner during fabrication. Nevertheless, the LNA has a good impedance matching with the 50-Ω

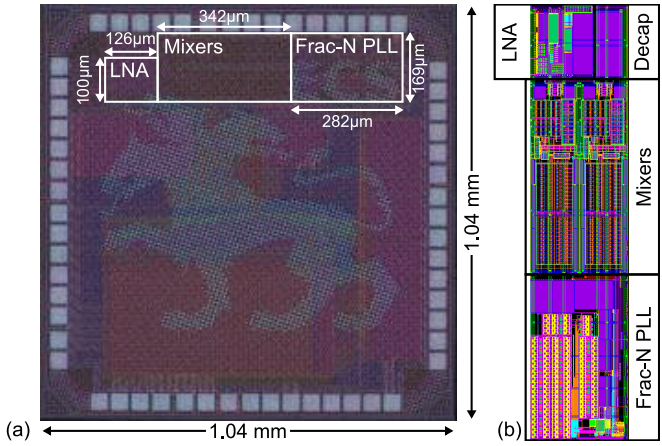

Fig. 4.25 (a) Chip microphotograph with overlaid layout. (b) Layout of the RF front end rotated 90° clockwise.

antenna, Fig. 4.26(b) shows that the magnitude of the S_{11} parameter is below −10 dB for all configurations. Fig. 4.26(c) shows the complete S_{11} curve as a function of frequency for configuration 16 of the LNA. Even though the peak is located 765 MHz, the matching is done over a sufficiently broad frequency range to stay below −10 dB in the frequency band used by LoRa.

The total power consumption of the RF front end is shown in Fig. 4.26(a) for the different LNA configurations. It varies between 1.80 and 2.64 mW. The configuration of the PLL was fixed, with notably the VCO that has three delay elements in parallel. From post-layout simulations, the RF front end in these conditions consumes between 1.49 and 3.33 mW. Fig. 4.27 shows a comparison between the measured and the simulated power consumption for a subset of configurations. The highest mismatch is at the extreme configurations with approximately ±21 %. The fact that configurations consuming more than 2.2 mW had their power consumption overestimated with post-layout simulations, when the opposite happens with lower-power configurations, suggests that the IR drops in the supply voltage are significant. We did not observe them in post-layout simulations because the parasitic extraction was performed at the level of independent circuit blocks and therefore did not capture the drops in the global power rail on the chip, in the wirebonds, and in the printed circuit board (PCB) traces. The different blocks in the RF front end were simulated separately post-layout, and

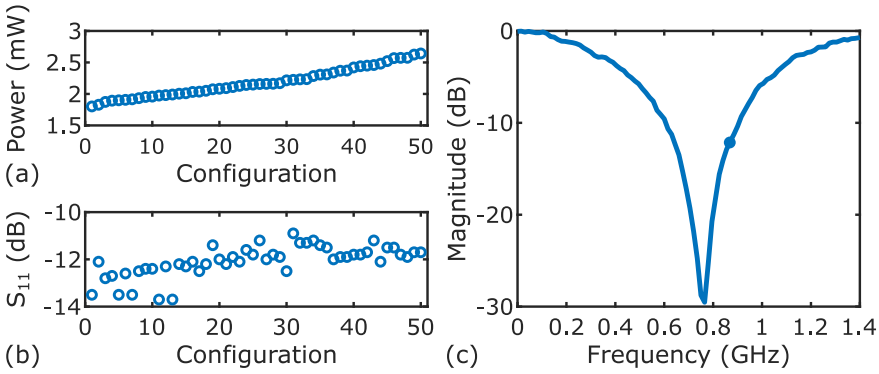

Fig. 4.26 Measurement results. (a) Total power consumption of the RF front end as a function of the LNA configuration. (b) Magnitude of the S_{11} parameter for the different LNA configurations. (c) Magnitude of the S_{11} parameter at different frequencies for configuration 16 of the LNA.

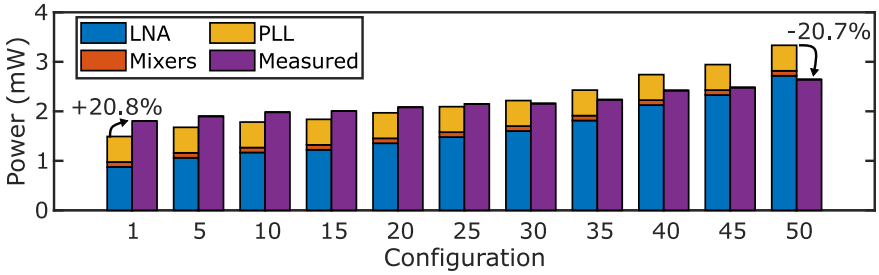

Fig. 4.27 Comparison between the power consumption of the RF front end in measurements and post-layout simulations.

the power supply distribution was thus not included in these simulations, which could explain the mismatch with the measurements.

The phase noise of the PLL was measured for a configuration using the second-order DDSM, and the second-order filter for a bandwidth of 110 kHz and a damping factor of 1.6. It is shown in Fig. 4.28. The measured bandwidth is approximately 180 kHz, corresponding to an increase of 63.6 %. This could be due to variations in the values of the passive components of the filter, but the VCO gain could also have changed significantly due to the slow process corner estimated in measurement and the increased supply voltage. The spectrum analyzer computes the phase noise as an average of five measurements, which could explain the unexpected behavior around

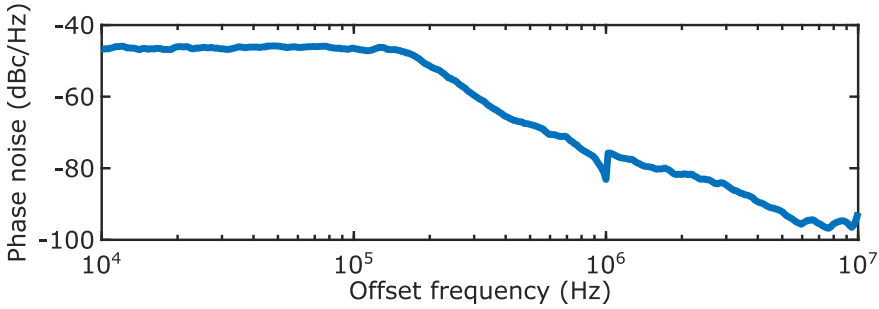

Fig. 4.28 Measured phase noise for a PLL running at 868.09 MHz, using a second-order filter and DDSM, and with 110 kHz of bandwidth and a damping factor of 1.6. The curve is an average of five measurements and was corrected by a factor of $20\log(64) = 36$ dB as the phase noise was measured on a clock divided by 64.

1 MHz. It is important to mention that the 3.2-mA digital output pads of the chip were not capable of outputting the clock at 868 MHz, so it was divided by 64 by an on-chip clock divider to respect the maximum allowed frequency of the pads. The phase noise of a divided clock is lower than the original by a factor of $20\log(N_{\text{div}})$ [49], where N_{div} is the division factor. Therefore, Fig. 4.28 includes a correction factor of 36 dB to show the correct phase noise of the clock at 868 MHz.

Unfortunately, the gain and noise of the RF front end could not be measured. Only noise can be seen at the output of the mixers, no matter the power of the signal put at the input of the LNA. The probable cause of this is a bad clock routing between the PLL and the mixers. The clock is generated at the right edge of the PLL where the glitch-free multiplexer and the output buffers are located in the layout, while the mixer switches are located near the LNA at the left edge of the mixer, to avoid a long connection of the RF output of the LNA. As can be seen in Fig. 4.25, this represents at least 624 µm of metal length. Middle-level metal of minimal width was used for this routing, representing a total resistance of 4 kΩ and a total capacitance of 10 pF. Even though they are distributed along all the connections, a single RC circuit was used to simulate the impact of these parasitic elements on the clock signal. This should give a worst-case scenario. Fig. 4.29(a) shows the mixer output with and without the routing parasitic elements, and for two different input power levels. This is a post-layout simulation of the complete RF front end, where the PLL was replaced by an ideal clock

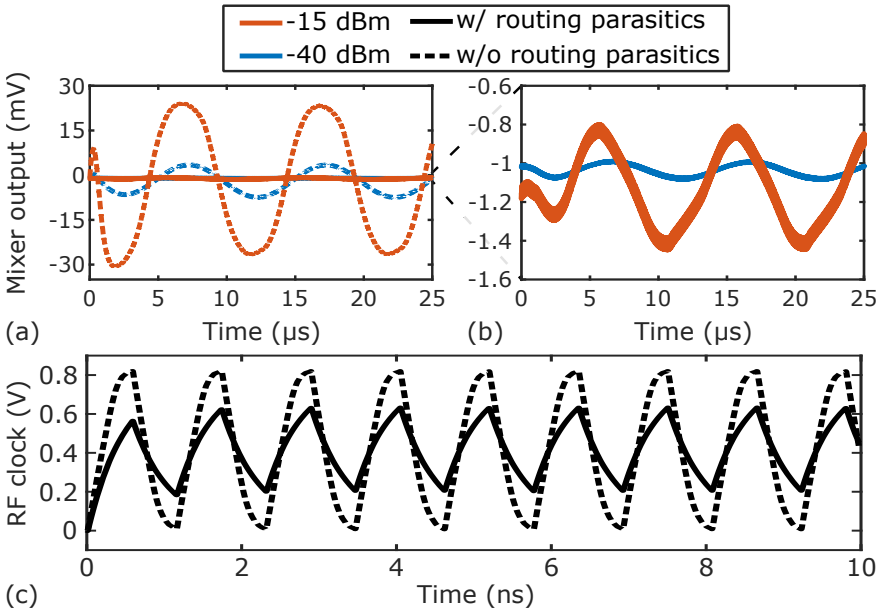

Fig. 4.29 (a) Post-layout simulation of the RF front end without PLL showing the output signal for an input signal at −15 and −40 dBm. The output signal is shown with and without the parasitic elements linked to the routing of the clock. (b) Zoom in on the y-axis to show the output signal when the parasitic elements are considered. (c) High-frequency clock with and without the parasitic elements linked to its routing.

source and the output buffers that drive the RC circuit representing the long metal connections until the mixer. The simulation was performed in the slow nMOS and slow pMOS process corner (SS) corner at 0.825 V to emulate the conditions present in the measurements. The gain of the mixer is severely degraded by the presence of the parasitic elements. It can be seen in a zoom of the y-axis in Fig. 4.29(b) that the amplitude of the output signal indeed increases with the input power, but it remains significantly lower than expected. Fig. 4.29(c) shows the clock signal with and without the routing parasitic elements at the mixer switches. The clock transitions, even without the parasitic elements, are not sharp enough due to the SS corner. A stronger buffer was needed to drive the large switches of the mixer. Moreover, the clock amplitude is severely degraded when the RC circuit is included, explaining the loss of gain in the mixer. In future designs, a more careful design of the clock buffering is necessary, both on the number

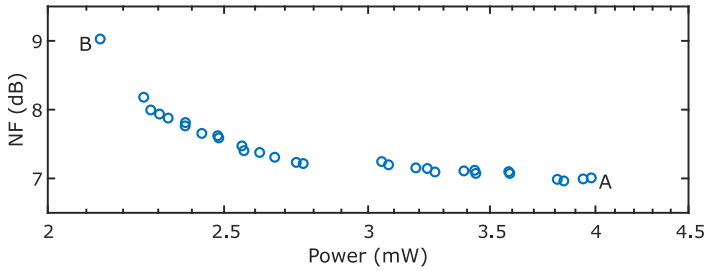

Fig. 4.30 Pareto front of the noise-power trade-off of the complete LoRa receiver. The power consumption of the missing blocks was estimated.

of buffers to use as well as their size. A better floorplanning is desired as well as it also helps reduce the power consumption in the clock distribution.

4.6 System Adaptation

Given that we were unable to complete the measurements of the RF front end, we will continue the system study with simulation results. Overall, the NF of the LNA can be tuned from 2.4 to 4 dB, resulting in a power consumption between 0.88 and 2.72 mW. To this power consumption, we must add the power of two mixers consuming 281 µW each and the power of the fractional-N PLL with a fixed configuration consuming 371 µW. Then, we can extrapolate this result to estimate the power consumption of a complete LoRa receiver. For the DBB signal processing, we use the power from [75] as an upper bound because it implements a LoRa software-defined radio with an SF of 7 and a bandwidth of 125 kHz. It uses 12-bit I and Q samples acquired at 2 MHz, i.e., with an oversampling factor of 16, and consumes 316 µW. Regarding the ADC, we use the lower bound of 0.67 fJ/conv−step from [100] obtained empirically, resulting in a power consumption of 10.97 µW for the two required ADCs. Finally, we neglect the power consumption of the analog LPFs since they operate in baseband and are not typically a major contributor to the overall power consumption. With these estimations, the Pareto front of the noise-power trade-off of the complete LoRa receiver is shown in Fig. 4.30. The power breakdown of the extreme points of this front, called configurations A and B to refer easily to them, is shown in Fig. 4.31(b). The entire receiver would consume

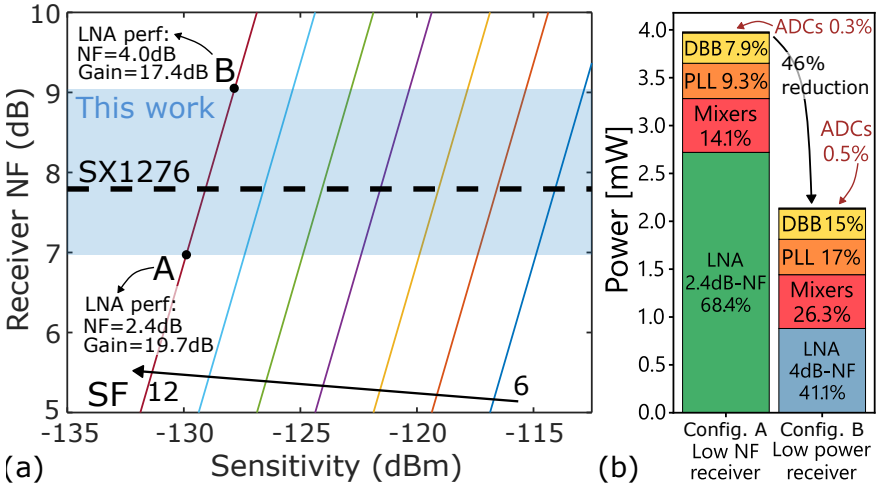

Fig. 4.31 (a) NF required at the receiver to attain the sensitivity of LoRa as a function of the SF. The dotted line represents the mean NF of the SX1276 chip from Semtech. Two LNA configurations, A and B, are highlighted, exhibiting respectively the best and worst NF. (b) Extrapolated power consumption of a LoRa receiver for configurations A and B of the LNA.

between 2.14 and 3.98 mW depending on the configuration, resulting in a 46-% power reduction at the expense of a 2-dB sensitivity degradation.

Fig. 4.31(a) shows the NF required to reach the sensitivity of LoRa at the maximum allowed bandwidth of 500 kHz. The dotted line indicates the mean NF for all SFs of the SX1276 chip [21], one of the commercial LoRa transceivers developed by Semtech, the company that owns the LoRa physical layer IP. The NF of the proposed receiver can be changed dynamically in a range going from point A to point B, saving up to 2× in power consumption. Point A in Fig. 4.31(a) indicates the design point usually considered for receivers, where the best performance is reached, and the signal is correctly demodulated even in the worst-case channel conditions. However, if the SNR of the received signal is, for instance, 2 dB higher (point B), the NF could be degraded by the same amount, thereby maintaining the initial PER and reducing the power consumption at the same time. A non-programmable receiver operating at point A therefore incurs a power overhead for its lower but unnecessary NF, while the SF cannot be decreased yet to benefit from a shorter packet to reduce the transmitter (TX) packet energy.

4.7 Conclusions

In this chapter, we present an RF front end for narrowband LPWAN technologies that is programmable so it can be adapted to the conditions of the wireless channel with the goal of limiting the power overhead due to the sizing for the worst-case channel conditions. The programmability stems from the LNA, which achieves an NF between 2.4 and 4 dB for a corresponding power consumption between 2.72 and 0.88 mW. The IIP3 can also be tuned between −10.2 and 2.2 dBm. The complete RF front end consumes between 1.81 and 3.65 mW in post-layout simulations with an integrated double-sideband NF between 7 and 9 dB. By extrapolation with state-of-the-art ADCs and digital demodulator, power consumption savings of 46 % can be achieved at the receiver level at the expense of a 2-dB sensitivity degradation. These findings corroborate the ideas developed in Chapter 2 about the potential power savings of adaptive receivers. The design of the rest of the receiver chain must be completed to confirm these findings at the full system level.

The choice of receiver architecture allows having an NF compatible with LoRaWAN even with the presence of flicker noise in the band. Nevertheless, the contribution of the mixer to the global NF is significant and limits its tunability range. As two instances of the mixer are required for signal demodulation, its contribution to the overall power consumption is also significant, making its noise-power trade-off interesting to exploit in the next prototype introduced in Chapter 5. However, some problems encountered in the design presented in this chapter must be addressed before. First, the architecture of the Gm block must be reviewed to reduce its noise contribution, which seems feasible with limited power penalty given the fact that in the prototype presented in this chapter its power is almost negligible. Second, as the TIA's amplifier dominates the power consumption, this is where tunability should be added. However, a more robust design in terms of common-mode feedback is required to make the tuning easier to design. Finally, but key for the mixer functionality, is the design of clock buffering near the mixer switches to avoid losing gain due to the routing parasitics that increase the transition times and decrease the amplitude of the clock signal.

The second objective of the thesis of designing a long-range wake-up receiver will oblige us to revise the design of the LNA and the PLL. The

current design of LNA consumes too much power for a wake-up receiver; even the configuration with the lowest power consumption is too close to the 1-mW upper limit for its power consumption, and does not leave enough room for the design of the rest of the chain with an acceptable power budget. The next chapter will thus explore the use of a matching network to remove the constraint on the input impedance that sets a minimum bound to the power consumed by the LNA. Concerning the PLL, an all-digital architecture will be explored for two reasons. The ability to duty-cycle the PLL without frequency drifts, and sharper digital filters with an acceptable power and area to reduce the quantization noise from the DDSM, which dominated the phase noise of the PLL designed in this chapter.

Given the extensive changes required in all blocks, a change of technology can be considered. For practical reasons linked to a more relaxed tape-out schedule and a more relaxed layout process, we decided to implement the second prototype presented in Chapter 5 in Taiwan Semiconductor Manufacturing Company (TSMC) 65-nm bulk technology. Using this older technology node will result in an increased power consumption of digital blocks and the loss of the back-bias as a tuning knob, but should have a limited impact on the performance of analog blocks. The end of Chapter 5 will be focused on the comparison of both prototypes and will highlight the differences caused by the change in technology.

7

Conclusions

THE research presented in this thesis focuses on minimizing the power consumption of RF receivers in IoT nodes. The goal is to enable the use of mesh networks in LPWANs, whose predominant topology is the star network, by reducing the power overhead associated with the increased use of their receivers. Unlike star networks, mesh networks provide a larger communication range, which is particularly helpful in environmental monitoring applications in remote locations. Wake-up receivers are also introduced as an alternative to network synchronization. Ultra-low-power receivers that continuously scan the wireless channel for incoming packets can further reduce the power overhead of mesh networks by eliminating the need to periodically enable the more power-intensive main receiver just for synchronization.

To this end, Chapter 2 explores how a programmable receiver can exploit the intrinsic variability of wireless channel conditions to save power when channel attenuation is lower, leveraging its noise-power trade-off. This chapter also presents specifications for the frequency synthesizer as well as the proposed scheme for a wake-up receiver based on the LoRa modulation. Chapter 3 then describes the methodology used throughout the thesis to design programmable analog and mixed-signal circuits based on a genetic optimization algorithm. Two examples show how to use this algorithm to optimize simple analog circuits with simulations and complex mixed-signal circuits using measurements. The potential power savings

obtained from programmable receivers are estimated using two different prototypes detailed in Chapter 4 and 5. Finally, Chapter 6 shows how the latter prototype was extended to include a wake-up receiver.

7.1 Contributions

This section summarizes the main findings of this thesis and their limitations.

Interpolating Between Spreading Factors

The idea of RF transceivers adapting to the conditions of the wireless channel is well known in telecommunications, mainly for adapting the transmission power. In LoRa, for example, transmitters benefit from the trade-off associated with the choice of spreading factor. When channel attenuation is lower, reducing the SF increases the data rate and decreases the energy required for packet transmission. In contrast, when channel attenuation is higher, e.g., when a longer communication range is needed, the SF is increased, which reduces the data rate and raises transmission energy in order to achieve the required sensitivity. However, the SF can only be adjusted when the SNR changes in steps of 2.5 dB.

The fact that this adaptation brings limited energy savings for receivers motivates the first research question, which we initially simplify as follows:

Can an IoT receiver reduce its power consumption by adapting its noise-power trade-off according to the wireless channel conditions while preserving communication quality?

In Chapter 2, we tackle this question by calculating the NF that is strictly needed at the receiver to attain a certain sensitivity at a fixed PER of 1 %. The same 2.5-dB difference between SFs is also found in the linear relation between NF and sensitivity. This suggests that, for SNR changes within 2.5 dB, a conventional receiver with a fixed NF has an unnecessarily low NF. Given the trade-off between power consumption and NF in receivers, an unnecessarily low NF directly translates into a power overhead that we can eliminate by giving the receiver the ability to dynamically adapt its noise-power trade-off according to the SNR of the received signal. This can be seen as a sort of interpolation between two SFs, as we propose new operating points for receivers within the 2.5-dB steps provided by the SF.

As the LoRaWAN standard does not allow SFs below 7, the energy savings on the transmitter typically stop when reaching an SNR of –7.5 dB. Therefore, the range of possible NF values does not need to be limited to 2.5 dB. Higher values could potentially become useful in case of exceptionally good SNRs.

Design of Programmable Receivers

Having answered positively to the previous question, we can now tackle the complete research question:

How can we design highly configurable circuits for IoT receivers that can dynamically tune their noise-power trade-off to adapt to the wireless channel conditions, thereby minimizing power consumption while preserving communication quality?

Several contributions of this thesis contribute toward addressing this question, as detailed in the following sections.

Methodology

An initial step toward answering this question involves the methodology used in the design of programmable analog and mixed-signal circuits in an RF receiver. The general method taken to obtain the optimal Pareto front of the complete receiver consists in extracting the Pareto fronts of the sub-blocks that have a significant impact on both the NF and the power consumption. Sub-blocks that do not respect this condition are simply optimized so their contribution to both metrics is as small as possible.

A genetic optimization algorithm was chosen as the method to extract the Pareto fronts of the sub-blocks. For this, we used an existing optimization framework that had already shown good performance in analog blocks [59]. Chapter 3 details how the algorithm can be used for the complete design of an LNA, from design-space exploration to choosing the optimal combination of discrete configuration values.

Complex mixed-signal circuits like PLLs cannot be optimized with the same method as an LNA because of the long time it takes to complete the simulation needed to extract the required performance metrics. This motivated the novel optimization methodology presented in Chapter 3, based on the post-silicon genetic optimization of a highly programmable prototype. The optimization time was decreased with the proposed scheme to only 17 h, in contrast to the simulation-based alternative that would require more than a year. The values to implement for each tuning knob

must be carefully chosen to ensure that the optimum can be achieved once the circuit is fabricated. Analytic expressions are particularly useful for this purpose, but they are not sufficient, as complex interactions between the tuning knobs and circuit non-idealities make it difficult to predict the optimal combinations. The PLL designed using this methodology exhibited a competitive performance when compared to the state of the art.

Even though this optimization methodology can be adapted to optimize other metrics of a PLL or even to optimize different complex circuits, the optimization setup is consequent. The automation of the optimization process is thus difficult and needs further research before it can be used in a commercial product. The way current and voltage references, or temperature sensors, are calibrated post-silicon could be a source of inspiration for future research in this direction.

Phase Noise Specification for Frequency Synthesizers in a LoRa Receiver

As LoRa is a proprietary physical layer, there is no official specification for the phase noise required in the generated carrier frequency. Commercial LoRa chips by Semtech use LC oscillators exhibiting excellent phase noise performance. However, integrating the LC tank is problematic in the frequency band used by LoRaWAN due to the size of the integrated inductors. In Chapter 2 we study the possibility of using ring oscillators for their simplicity, compactness, and lower power compared to LC oscillators. After adapting the MATLAB testbench presented in [40], it was found that an in-band phase noise of −80 dBc/Hz is enough to avoid a significant impact on the PER. This level of phase noise can be achieved by ring oscillators with an acceptable power consumption.

Further research in this area is necessary to obtain a more precise specification in terms of the SF, packet length, and phase-noise profile used. In this preliminary study, we did not find a significant impact of the SF. However, only small packets were simulated, and with a simplified PLL phase-noise profile without flicker noise. The influence of the PLL bandwidth merits a more careful study as it already showed significant impacts on the results. Finally, experimental validation of the resulting specification is necessary.

Programmable RF Front End

Chapter 4 presents a first prototype included in the MCU codenamed CERBERUS, which was designed to have a first estimation of the potential power savings of a programmable receiver. This prototype includes a programmable LNA able to dynamically change its NF between 2.4 and 4 dB

for a corresponding power consumption between 2.72 and 0.88 mW. The prototype also includes a mixer and an analog fractional PLL based on a ring oscillator, both with fixed performance. By extrapolating the power consumption of these three blocks with state-of-the-art ADC and digital demodulator, power savings of 46 % were estimated at the expense of a 2-dB sensitivity degradation. This demonstrates the power savings that may be obtained when limited to interpolating between SFs.

The PLL included in this prototype was initially designed to have a programmable trade-off between power consumption and phase noise. However, it was found after the tape-out that the quantization noise of the DDSM dominates the phase noise at high offset frequencies instead of the VCO. As the DDSM has a negligible power consumption compared to the rest of the PLL, there is no noise-power trade-off to exploit for this particular design, unlike the design optimized in Chapter 3.

In addition, a problem with the clock routing in the chip did not allow the measurement of the NF due to a significantly reduced gain. Therefore, the influence of the phase noise on the NF could not be validated with measurements to confirm that ring oscillators can be used in LoRaWAN nodes.

Programmable LoRa Receiver

A complete LoRa receiver included in the MCU codenamed BREWER is detailed in Chapter 5. Programmable LNA and mixers control the noise-power trade-off of this second prototype. As the mixer is a large contributor to both metrics, exploiting its inherent trade-off enlarges the range of achievable NF and potential power savings. In this design, power consumption savings of 62.6 % are possible at the cost of an 8.3-dB sensitivity degradation. When limited to interpolating between SFs, a similar reduction in power consumption is obtained as the previous prototype, with a 49.8-% reduction.

With this prototype, we not only show the potential behind programmable receivers, but we also propose a receiver that consumes between 3.5 and 8× less than commercial LoRa chips from Semtech for the same sensitivity.

The sensitivity had to be extracted from post-layout simulations. Even though the functionality of the receiver was validated on-chip, an offset voltage issue resulted in a significant increase of the NF. Unfortunately, this problem prevented experimental verification of the impact of phase noise on the NF and the sensitivity again.

Long-Range Wake-Up Receiver

Mesh networks, unlike the star topology typically used in LPWANs, offer significantly greater communication range in remote locations but at the cost of a power consumption overhead. Wake-up receivers offer a solution to this as they eliminate the need for synchronizing the network. To this end, the second research question aims at extending the idea of programmable receivers to include a wake-up mode:
Can we exploit the extremes of the noise-power trade-off of RF receivers to enable wake-up receivers able to detect signals with a degraded noise performance but an ultra-low power consumption?
Chapter 6 shows how the second prototype was modified to include this wake-up mode. The three blocks dominating the power consumption of the receiver: LNA, mixers, and PLL were modified to enable this mode of operation.

In an effort to reduce the power consumption of the PLL, we explored the possibility of duty-cycling it for using a free-running oscillator most of the time and enabling the complete control loop just for correcting possible frequency drifts. We show that an analog PLL cannot be duty-cycled because the leakage currents cause a considerable frequency drift during the time of a LoRa symbol. An all-digital PLL was thus designed instead based on a ring oscillator. A novel architecture for the precise control of the frequency of the current-starved DCO is proposed. To the best of our knowledge, this is the first fractional ADPLL based on a ring oscillator. After experimental validation, we conclude that the high phase noise of the free-running DCO does not allow a correct detection of a LoRa packet and that the PLL must remain enabled to ensure functionality.

Nevertheless, the changes brought to the LNA and mixer reduced the power consumption in wake-up mode by more than $3\times$, compared to the main receiver at its best sensitivity. However, its power consumption is still too high to consider an always-on operation in an IoT node, for which sub-mW consumption is needed for extending its lifetime.

7.2 Perspectives

Having listed the main contributions of the thesis and their limitations, let us conclude the thesis by giving some perspectives on future work.

The thesis was centered around the design of programmable circuits. An approach that can result in significant power savings, even though it implies an additional non-negligible design effort to ensure functionality of a greater number of designs by limiting their impact on each other. This is particularly the case of the LNA designed in this thesis due to the switches being inserted in the design and to the parasitics added by the inactive configurations implemented in parallel. The sizing done with the help of the genetic optimization algorithm managed to optimize almost all performance metrics of the LNA despite these difficulties, even if some configurations had to be dropped to ensure the functionality of the others, particularly the low-power ones. However, we showed that the non-linearity measured through the IIP3 was indeed impacted by the extra transistors in the stack. Slightly more complex designs like [85] could be considered as a way to improve the non-linearity of the LNA by including distortion cancellation. The other programmable circuits are less impacted by the addition of tunability, thanks to a lower operating frequency than the LNA or a less invasive implementation of the tunability. The mixer performance was configured through the bias current, which does not imply any modification to the circuit architecture, as for the charge pump of the PLL. The PLL filter, thanks to a significantly lower frequency of operation, was not significantly impacted by the inserted switches. The oscillator, the other programmable high-frequency circuit, handled the multiple configurations in the more robust digital domain with a multiplexer for choosing the clock among the parallel oscillators.

Coming back to the non-linearity, we decided to leave aside the IIP3 metric to simplify the design of the LoRa receiver in the second prototype because of the lack of interference in the target application. Nodes deployed in remote areas for environmental monitoring applications should not indeed suffer from strong interference. However, this is not the case for many other applications since LoRaWAN operates in unlicensed frequency bands, which are subject to more interference. Not optimizing the IIP3 is probably why the proposed receiver outperforms some state-of-the-art designs, particularly commercial LoRa receivers, but this is not reflected in the used FoMs because they do not include this metric. Future works on programmable receivers should focus on optimizing the IIP3 to add robustness to interference and therefore be suited for more applications. Exploring the tunability of the IIP3 is also an interesting research path because the RF receiver could adapt to the dynamic presence of interference, or it could give versatility to the IoT node to adapt its IIP3 performance to the

target application, e.g., a low IIP3 for environmental monitoring with low interference and high IIP3 for applications in urban environments where the risk of interference is higher.

The first research question of this thesis was not completely answered, as we only showed the potential power consumption savings that could result from dynamically tuning the noise-power trade-off of LoRa receivers with the two proposed prototypes. Research efforts are still needed at different levels. Finding the best way to estimate the channel attenuation on-chip should be a priority, as this information is key for real-time adaptation to the wireless channel conditions. Machine learning could be useful to develop a strategy for choosing the best configuration in real time. Finally, at the network level, this information could be exploited to search for a global minimum in power consumption.

Concerning the wake-up receiver, as we propose the first wake-up receiver employing the LoRa modulation, there remain many opportunities for optimization. Its power consumption must first be reduced below 1 mW before it can be used in IoT applications. To achieve this, the power consumed by the PLL must be the first priority, as it largely dominates the power in the wake-up receiver. Switching to a more advanced technology node and decreasing the supply voltage could help thanks to the all-digital architecture of the PLL. Another option could be to design an analog PLL with a relaxed phase-noise requirement to limit its power consumption. However, exploring simplified architectures, like injection locking or frequency-locked loops, for limiting the phase noise of the free-running oscillator without a significant power overhead could offer better advantages. The possibility of using a free-running LC oscillator could also be studied, but this requires a thorough analysis of the impact of phase noise in the LoRa modulation to avoid having to wait for the chip measurements before knowing if it fulfills the modulation requirements. Finally, duty-cycling the receiver chain could also be a possible option, as this is a common practice in state-of-the-art wake-up receivers to trade off latency for lower power consumption.

Bibliography

[1] K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, Jun. 2009.

[2] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241–261, Jan. 2019.

[3] Cisco. (2020) Cisco annual internet report - Cisco annual internet report (2018–2023) white paper. Accessed: Aug., 2025. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[4] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[5] M. R. M. Kassim, "IoT applications in smart agriculture: Issues and challenges," in *2020 IEEE Conference on Open Systems (ICOS)*. Kota Kinabalu, Malaysia: IEEE, Nov. 2020, pp. 19–24.

[6] G. Mois, S. Folea, and T. Sanislav, "Analysis of three IoT-based wireless sensors for environmental monitoring," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2056–2064, Aug. 2017.

[7] P. Asha, L. Natrayan, B. Geetha, J. R. Beulah, R. Sumathy, G. Varalakshmi, and S. Neelakandan, "IoT enabled environmental toxicology for air pollution monitoring using AI techniques," *Environmental Research*, vol. 205, p. 112574, Apr. 2022.

[8] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Inf.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[9] Statista. (2025) Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034. Accessed: Aug. 2025. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[10] T. Pirson and D. Bol, "Assessing the embodied carbon footprint of IoT edge devices with a bottom-up life-cycle approach," *J Cleaner Prod*, vol. 322, Nov. 2021.

[11] A. Plepys, "The grey side of ICT," *Environmental Impact Assessment Review*, vol. 22, no. 5, pp. 509–523, Oct. 2002.

[12] L.-P. P.-V. Clément, Q. E. Jacquemotte, and L. M. Hilty, "Sources of variation in life cycle assessments of smartphones and tablet computers," *Environmental Impact Assessment Review*, vol. 84, p. 106416, Sep. 2020.

[13] A. Müller, L. Friedrich, C. Reichel, S. Herceg, M. Mittag, and D. H. Neuhaus, "A comparative life cycle assessment of silicon PV modules: Impact of module design, manufacturing location and inventory," *Solar Energy Materials and Solar Cells*, vol. 230, p. 111277, Sep. 2021.

[14] G. Callebaut, G. Leenders, J. Van Mulders, G. Ottoy, L. De Strycker, and L. Van der Perre, "The art of designing remote IoT devices—technologies and strategies for a long battery life," *Sensors*, vol. 21, no. 3, p. 913, Jan. 2021.

[15] LoRaWAN Alliance. (2017) LoRaWAN specification v1.1. Accessed: Aug., 2025. [Online]. Available: https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/

[16] Semtech, "An in-depth look at LoRaWAN class A devices," 2019, [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lorawan-class-a-devices/.

[17] J. Bauwens, P. Ruckebusch, S. Giannoulis, I. Moerman, and E. D. Poorter, "Over-the-air software updates in the Internet of Things: An

overview of key principles," *IEEE Commun. Mag.*, vol. 58, no. 2, pp. 35–41, Feb. 2020.

[18] The Things Network. What are LoRa and LoRaWAN. Accessed: Aug. 2025. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

[19] *Attenuation in vegetation*, ITU-R P.833-10. International Telecommunication Union, Sep. 2021.

[20] A. Goldsmith and S.-G. Chua, "Adaptive coded modulation for fading channels," *IEEE Trans. Commun.*, vol. 46, no. 5, pp. 595–602, May 1998.

[21] Semtech, "137 MHz to 1020 MHz low power long range transceiver," SX1276/77/78/79 Datasheet, 2020.

[22] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT nodes: Opportunities and challenges," *IEEE Des. Test*, vol. 36, no. 2, pp. 7–40, Apr. 2019.

[23] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Trans. Circuits and Syst. I*, vol. 59, no. 5, pp. 946–957, May 2012.

[24] M. Mukherjee, A. Tchamkerten, and C. Jabbour, "Reconfigurable adaptive channel sensing," *IEEE Trans. on Green Commun. Netw.*, pp. 1–1, 2023.

[25] J. Zhu, H. Krishnaswamy, and P. R. Kinget, "Field-programmable LNAs with interferer-reflecting loop for input linearity enhancement," *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 556–572, Feb. 2015.

[26] M. Shafiee and S. Ozev, "An in-field programmable adaptive CMOS LNA for intelligent IoT sensor node applications," *IEEE Trans. Comput.-Aided Des. Integr. Circuits and Syst.*, vol. 41, no. 2, pp. 201–210, Feb. 2022.

[27] J. Zaini, F. Hameau, T. Taris, D. Morche, L. Q. V. Tran, and P. Audebert, "Channel aware receiver front end for low power 2.4 GHz wireless sensor network: A system level analysis," in *2016 14th IEEE Int. New Circuits and Syst. Conf. (NEWCAS)*, pp. 1–4.

[28] D. Das, M. Liu, M. Abdi, F. Restuccia, and M. Onabajo, "Digitally tunable CMOS mixer design for adaptive RF front-ends," in *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*. London, United Kingdom: IEEE, May 2025, pp. 1–5.

[29] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Int. Conf. on Parallel Problem Solving from Nature*, 2000, pp. 849–858.

[30] R. Dekimpe and D. Bol, "Cross-domain optimization of low-power mixed-signal sensor systems under classification accuracy constraints," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 43, no. 8, pp. 2507–2517, Aug. 2024.

[31] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019.

[32] J. Peña Queralta, T. Gia, Z. Zou, H. Tenhunen, and T. Westerlund, "Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: Beyond LoRa and LoRaWAN," *Procedia Computer Science*, vol. 155, pp. 343–350, 2019.

[33] M. Perez, F. E. Sierra-Sanchez, F. Chaparro, D. M. Chaves, C.-I. Paez-Rueda, G. P. Galindo, and A. Fajardo, "Coverage and energy-efficiency experimental test performance for a comparative evaluation of unlicensed LPWAN: LoRaWAN and SigFox," *IEEE Access*, vol. 10, pp. 97 183–97 196, 2022.

[34] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *2016 IEEE Wireless Communications and Networking Conference*. Doha, Qatar: IEEE, Apr. 2016, pp. 1–5.

[35] O. Seller and N. Sornin, "Low power long range transmitter," Patent US 14/170,170, Aug., 2014.

[36] ——, "Low complexity, low power and long range radio receiver," Patent US 15/620,364, May, 2019.

[37] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.

[38] M. Chiani and A. Elzanaty, "On the LoRa modulation for IoT: Waveform properties and spectral analysis," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8463–8470, Oct. 2019.

[39] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "Lora digital receiver analysis and implementation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 1498–1502.

[40] M. Xhonneux, O. Afisiadis, D. Bol, and J. Louveaux, "A low-complexity lora synchronization algorithm robust to sampling time offsets," *IEEE Internet Things J.*, 2021.

[41] The Things Network. Frequency plans. Accessed: Aug. 2025. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/frequency-plans/

[42] O. Afisiadis, M. Cotting, A. Burg, and A. Balatsoukas-Stimming, "On the error rate of the LoRa modulation with interference," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1292–1304, Feb. 2020.

[43] T. Mahjoub, M. B. Said, and H. Boujemaa, "On the performances of packet error rate for LoRa networks," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*. Dubrovnik, Croatia: IEEE, May 2022, pp. 372–377.

[44] The Things Network. EU863-870 MHz band. Accessed: Aug. 2025. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/regional-parameters/eu868/

[45] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2117–2157, 2017.

[46] P. P. Mercier, B. H. Calhoun, P.-H. P. Wang, A. Dissanayake, L. Zhang, D. A. Hall, and S. M. Bowers, "Low-power RF wake-up receivers: Analysis, tradeoffs, and design," *IEEE Open J. Solid-State Circuits Soc.*, vol. 2, pp. 144–164, 2022.

[47] K.-M. Kim, H.-G. Seok, O.-Y. Jung, K.-S. Choi, B. Yun, S. Kim, W. Oh, E.-R. Jeong, J. Ko, and S.-G. Lee, "A –123-dBm sensitivity split-channel BFSK reconfigurable data/wake-up receiver for low-power wide-area networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2656–2667, Sep. 2021.

[48] K.-M. Kim, K.-S. Choi, H. Jung, B. Yun, J. Xu, J. Ko, and S.-G. Lee, "A -124-dBm sensitivity interference-resilient direct-conversion duty-cycled wake-up receiver achieving 0.114 mW at 1.966-s wake-up latency," *IEEE J. Solid-State Circuits*, vol. 58, no. 6, pp. 1667–1680, Jun. 2023.

[49] N. Da Dalt and A. Sheikholeslami, *Understanding Jitter and Phase Noise: A Circuits and Systems Perspective*, 1st ed. Cambridge University Press, 2018.

[50] X. Chen, H.-S. Kim, and D. D. Wentzloff, "An analysis of phase noise requirements for ultra-low-power FSK radios," in *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. IEEE, 2017, pp. 37–40.

[51] M. Schramme, L. Van Brandt, D. Flandre, and D. Bol, "Comprehensive analytical comparison of ring oscillators in FDSOI technology: Current starving versus back-bias control," *IEEE Trans. Circuits Syst. I*, 2022.

[52] B. Razavi, "Phase-locked loops," in *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.

[53] G. Gielen, T. McConaghy, and T. Eeckelaert, "Performance space modeling for hierarchical synthesis of analog integrated circuits," in *Des. Automat. Conf.*, 2005, pp. 881–886.

[54] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," *Proc. IEEE*, vol. 95, no. 3, pp. 640–669, Mar. 2007.

[55] B. G. Arsintescu, E. Charbon, E. Malavasi, U. Choudhury, and W. H. Kao, "General AC constraint transformation for analog ICs," in *Des. Automat. Conf.*, 1998, pp. 38–43.

[56] Y. Fellah, N. Abouchi, T. Tixier, A. Aubert, and L. Labrak, "Top-down, constraint driven design: Application to sigma-delta modulator," in *2004 IEEE Int. Conf. on Ind. Technol.* IEEE, 2004, pp. 132–136.

[57] S. Dam and P. Mandal, "Modeling and design of CMOS analog circuits through hierarchical abstraction," *Integration*, vol. 46, no. 4, pp. 449–462, Sep. 2013.

[58] R. Phelps, M. J. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums, "A case study of synthesis for industrial-scale analog IP: Redesign of the equalizer/filter frontend for an ADSL CODEC," in *Des. Automat. Conf.*, 2000, pp. 1–6.

[59] R. Dekimpe and D. Bol, "A configurable ULP instrumentation amplifier with pareto-optimal power-noise trade-off achieving 1.93 NEF in 65nm CMOS," *IEEE Trans. on Circuits and Syst. II: Express Briefs*, vol. 68, no. 7, pp. 2272–2276, 2021.

[60] F. Crupi, R. D. Rose, M. Paliy, M. Lanuzza, M. Perna, and G. Iannaccone, "A portable class of 3-transistor current references with low-power sub-0.5 V operation," *Int. J. of Circuit Theory and Appl.*, vol. 46, no. 4, pp. 779–795, 2017.

[61] Q. Huang, C. Zhan, L. Wang, Z. Li, and Q. Pan, "A -40 °C to 120 °C, 169 ppm/°C nano-ampere CMOS current reference," *IEEE Trans. Circuits Syst. II*, vol. 67, no. 9, pp. 1494–1498, Sep. 2020.

[62] M. Lefebvre and D. Bol, "A family of current references based on 2T voltage references: Demonstration in 0.18-µm with 0.1-nA PTAT and 1.1-µA CWT 38-ppm/°C designs," *IEEE Trans. Circuits Syst. I*, vol. 69, no. 8, pp. 3237–3250, Aug. 2022.

[63] H. Jiang, P.-H. P. Wang, P. P. Mercier, and D. A. Hall, "A 0.4-V 0.93-nW/kHz relaxation oscillator exploiting comparator temperature-dependent delay to achieve 94-ppm/°C stability," *IEEE J. of Solid-State Circuits*, vol. 53, no. 10, pp. 3004–3011, 2018.

[64] J. Mikulic, G. Schatzberger, and A. Baric, "Post-manufacturing process and temperature calibration of a 2-MHz on-chip relaxation oscillator," *IEEE Trans. on Circuits and Syst. I: Regular Papers*, vol. 68, no. 10, pp. 4076–4089, 2021.

[65] G. Yu and P. Li, "Hierarchical analog/mixed-signal circuit optimization under process variations and tuning," *IEEE Trans. on Computer-Aided Des. of Integr. Circuits and Syst.*, vol. 30, no. 2, pp. 313–317, 2011.

[66] M. Perrott, M. Trott, and C. Sodini, "A modeling approach for Σ-Δ fractional-N frequency synthesizers allowing straightforward noise analysis," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1028–1038, Aug. 2002.

[67] B. Razavi, "Jitter-power trade-offs in PLLs," *IEEE Trans. Circuits Syst. I*, vol. 68, no. 4, pp. 1381–1387, Apr. 2021.

[68] S. C. Blaakmeer, E. A. M. Klumperink, D. M. W. Leenaerts, and B. Nauta, "Wideband balun-LNA with simultaneous output balancing, noise-canceling and distortion-canceling," *IEEE J. Solid-State Circuits*, vol. 43, no. 6, pp. 1341–1350, Jun. 2008.

[69] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann, "A CPPLL hierarchical optimization methodology considering jitter, power and locking time," in *Des. Automat. Conf.*, 2006, pp. 19–24.

[70] E. Ayranci, K. Christensen, and P. Andreani, "Enhancement of VCO linearity and phase noise by implementing frequency locked loop," in *The Int. Conf. on "Computer as a Tool"*. IEEE, 2007, pp. 2593–2599.

[71] A. Lahiri, N. Gupta, A. Kumar, and P. Dhadda, "A 600µA 32 kHz input 960 MHz output CP-PLL with 530ps integrated jitter in 28nm FD-SOI process," in *40th Eur. Solid-State Circuits Conf. (ESSCIRC)*. IEEE, Sep. 2014, pp. 87–90.

[72] J. Zhu, W.-S. Choi, and P. K. Hanumolu, "A 0.016 mm^2 0.26-µW/MHz 60–240-MHz digital PLL with delay-modulating clock buffer in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 8, pp. 2186–2194, Aug. 2019.

[73] A. Lahiri and N. Gupta, "A 0.0175mm^2 600µW 32kHz input 307MHz output PLL with 190psrms jitter in 28nm FD-SOI," in *42nd Eur. Solid-State Circuits Conf.* IEEE, Sep. 2016, pp. 339–342.

[74] Xiao Pu, A. Kumar, and K. Nagaraj, "Area-efficient low-noise low-spur architecture for an analog PLL working from a low frequency reference," *IEEE Trans. Circuits Syst. II*, vol. 59, no. 6, pp. 331–335, Jun. 2012.

[75] M. Xhonneux, J. Louveaux, and D. Bol, "A sub-mW Cortex-M4 microcontroller design for IoT software-defined radios," *IEEE Open J. Circuits and Syst.*, vol. 4, pp. 165–175, 2023.

[76] S. Chehrazi, A. Mirzaei, and A. Abidi, "Noise in current-commutating passive FET mixers," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 2, pp. 332–344, Feb. 2010.

[77] D. Torrieri, "Chapter 2 direct-sequence systems," in *Principles of Spread-Spectrum Communication Systems*. Cham: Springer International Publishing, 2015, pp. 79–145.

[78] A. Thakur and S. Chatterjee, "A 4.4-mA ESD-safe 900-MHz LNA with 0.9-dB noise figure," *IEEE Trans. VLSI Syst.*, vol. 29, no. 2, pp. 297–306, Feb. 2021.

[79] H. G. Han, D. H. Jung, and T. W. Kim, "A 2.88 mW +9.06 dBm IIP3 common-gate LNA with dual cross-coupled capacitive feedback," *IEEE Trans. Microwave Theory Techn.*, vol. 63, no. 3, pp. 1019–1025, Mar. 2015.

[80] Z. Liu, C. C. Boon, C. Li, K. Yang, Y. Dong, and T. Guo, "A 0.0078mm^2 3.4mW wideband positive-feedback-based noise-cancelling LNA in 28nm CMOS exploiting g_m boosting," in *2022 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 142–144.

[81] A. Bozorg and R. B. Staszewski, "A 0.02–4.5-GHz LNTA in 28-nm CMOS for 5G exploiting noise reduction and current reuse," *IEEE J. Solid-State Circuits*, vol. 56, no. 2, pp. 404–415, Feb. 2021.

[82] S. Kim and K. Kwon, "Broadband balun-LNA employing local feedback gm-boosting technique and balanced loads for low-power low-voltage applications," *IEEE Trans. Circuits and Syst. I*, vol. 67, no. 12, pp. 4631–4640, Dec. 2020.

[83] Z. Pan, C. Qin, Z. Ye, Y. Wang, and Z. Yu, "Wideband inductorless low-power LNAs with gm enhancement and noise-cancellation," *IEEE Trans. Circuits and Syst. I*, vol. 65, no. 1, pp. 26–38, Jan. 2018.

[84] J. Zaini, F. Hameau, T. Taris, D. Morche, P. Audebert, and E. Mercier, "A tunable ultra low power inductorless low noise amplifier exploiting body biasing of 28 nm FDSOI technology," in *2017 IEEE/ACM Int. Symp. on Low Power Electronics and Des. (ISLPED)*, pp. 1–6.

[85] C. Gimeno, F. Stas, G. de Streel, D. Bol, and D. Flandre, "Improving noise and linearity of CMOS wideband inductorless balun LNAs

for 10-GHz software-defined radios in 28nm FDSOI," in *2017 IEEE SOI-3D-Subthreshold Microelectronics Technol. Unified Conf. (S3S)*, pp. 1–2.

[86] M. Parvizi, K. Allidina, and M. N. El-Gamal, "Short channel output conductance enhancement through forward body biasing to realize a 0.5 V 250 µW 0.6–4.2 GHz current-reuse CMOS LNA," *IEEE J. Solid-State Circuits*, vol. 51, no. 3, pp. 574–586, Mar. 2016.

[87] B. Miller and R. Conley, "A multiple modulator fractional divider," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 3, pp. 578–583, Jun. 1991.

[88] C. Vaucher, I. Ferencic, M. Locher, S. Sedvallson, U. Voegeli, and Z. Wang, "A family of low-power truly modular programmable dividers in standard 0.35-µm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1039–1045, Jul. 2000.

[89] K. Hosseini and M. P. Kennedy, *Minimizing Spurious Tones in Digital Delta-Sigma Modulators*. New York, NY: Springer New York, 2011.

[90] V. R. Gonzalez-Diaz, M. A. Garcia-Andrade, G. E. Flores-Verdad, and F. Maloberti, "Efficient dithering in MASH sigma-delta modulators for fractional frequency synthesizers," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 9, pp. 2394–2403, Sep. 2010.

[91] M. Kozak and I. Kale, "Rigorous analysis of delta-sigma modulators for fractional-N PLL frequency synthesis," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 6, pp. 1148–1162, Jun. 2004.

[92] H. Mo and M. P. Kennedy, "Masked dithering of MASH digital delta-sigma modulators with constant inputs using linear feedback shift registers," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 8, pp. 1131–1141, Aug. 2016.

[93] ——, "Influence of initial conditions on the fundamental periods of LFSR-dithered MASH digital delta-sigma modulators with constant inputs," *IEEE Trans. Circuits Syst. II*, vol. 64, no. 4, pp. 372–376, Apr. 2017.

[94] M. Borkowski, T. Riley, J. Hakkinen, and J. Kostamovaara, "A practical Δ-Σ modulator design method based on periodical behavior analysis," *IEEE Trans. Circuits Syst. II*, vol. 52, no. 10, pp. 626–630, Oct. 2005.

[95] R. Gray, "Spectral analysis of quantization noise in a single-loop sigma-delta modulator with DC input," *IEEE Trans. Commun.*, vol. 37, no. 6, pp. 588–599, Jun. 1989.

[96] D. Banerjee, *PLL Performance, Simulation, and Design*, 5th ed. Dog Ear Publishing, LLC, 2017.

[97] G. De Streel, F. Stas, T. Gurne, F. Durant, C. Frenkel, A. Cathelin, and D. Bol, "SleepTalker: A ULV 802.15.4a IR-UWB transmitter SoC in 28-nm FDSOI achieving 14 pJ/b at 27 Mb/s with channel selection based on adaptive FBB and digitally programmable pulse shaping," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 1163–1177, Apr. 2017.

[98] M. Schramme, C. Gimeno, A. Cathelin, D. Flandre, and D. Bol, "A 2.5-GHz clock recovery circuit based on a back-bias-controlled oscillator in 28-nm FDSOI," *IEEE Solid-State Circuits Lett.*, vol. 3, pp. 478–481, 2020.

[99] B.-K. Kim, D. Im, J. Choi, and K. Lee, "A low noise amplifier simultaneously achieving input impedance and minimum noise matching," in *2011 IEEE Radio Freq. Integr. Circuits Symp.*, pp. 1–4.

[100] B. Murmann, "ADC Performance Survey 1997-2023," [Online]. Available: https://github.com/bmurmann/ADC-survey.

[101] P. Harpe, H. Gao, R. V. Dommele, E. Cantatore, and A. H. M. Van Roermund, "A 0.20 mm^2 3 nW signal acquisition IC for miniature sensor nodes in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 240–248, Jan. 2016.

[102] S.-W. M. Chen and R. W. Brodersen, "A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13-µm CMOS," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2669–2680, Dec. 2006.

[103] M. Xhonneux, J. Louveaux, and D. Bol, "Implementing a LoRa software-defined radio on a general-purpose ULP microcontroller," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. Coimbra, Portugal: IEEE, Oct. 2021, pp. 105–110.

[104] T. T. Nguyen, H. H. Nguyen, R. Barton, and P. Grossetete, "Efficient design of chirp spread spectrum modulation for low-power wide-area

networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9503–9515, Dec. 2019.

[105] Z. Xu, S. Tong, P. Xie, and J. Wang, "From demodulation to decoding: Toward complete LoRa PHY understanding and implementation," *ACM Trans. Sen. Netw.*, vol. 18, no. 4, pp. 1–27, Nov. 2022.

[106] B. Razavi, "The design of a biquadratic filter [The analog mind]," *IEEE Solid-State Circuits Magazine*, vol. 16, no. 1, pp. 6–13, 2024.

[107] Semtech, "SX1272/73 - 860 MHz to 1020 MHz low power long range transceiver," SX1272 datasheet, 2019.

[108] STMicroelectronics, "Multiprotocol LPWAN dual core 32-bit Arm Cortex-M4/M0+ LoRa, (G)FSK, (G)MSK, BPSK, up to 256KB flash, 64KB SRAM," STM32WL55CC datasheet, 2022.

[109] B. Razavi, "The biquadratic filter [A circuit for all seasons]," *IEEE Solid-State Circuits Magazine*, vol. 10, no. 2, pp. 11–109, 2018.

[110] J. Zhuang and R. B. Staszewski, "All-digital RF phase-locked loops exploiting phase prediction," *Information and Media Technologies*, vol. 7, no. 1, pp. 1–14, 2014.

[111] L. Zhang, D. Duvvuri, S. Bhattacharya, A. Dissanayake, X. Liu, H. L. Bishop, Y. Zhang, T. N. Blalock, B. H. Calhoun, and S. M. Bowers, "A -102-dBm sensitivity multichannel heterodyne wake-up receiver with integrated ADPLL," *IEEE Open J. Solid-State Circuits Soc.*, vol. 4, pp. 43–56, 2024.

[112] F. Andersson and A. Johnson, "Modeling and characterization of all-digital phase-locked loop," Master's thesis, Linköpings universitet, Norrköping, Sweden, Feb. 2010.

[113] K. Kwon and D. D. Wentzloff, "Automated design flow for synthesizable ADPLL: From specification to GDS," *AEU - International Journal of Electronics and Communications*, vol. 177, p. 155204, Apr. 2024.

[114] T. Okazawa and I. Akita, "A robust and low-power synchronization technique of coarse-and-fine conversion parts in ring-oscillator-based time-to-digital converters," in *2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. Batumi: IEEE, Dec. 2017, pp. 239–242.

[115] V. Kratyuk, P. K. Hanumolu, U.-K. Moon, and K. Mayaram, "A design procedure for all-digital phase-locked loops based on a charge-pump phase-locked-loop analogy," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 3, pp. 247–251, Mar. 2007.

[116] B. Jiang and T. Xia, "ADPLL design parameters determinations through noise modeling," *Integration*, vol. 48, pp. 138–145, Jan. 2015.

[117] Semtech, "Ultra-low power integrated UHF transceiver," SX1211 datasheet, 2013.

[118] K.-M. Kim, K.-S. Choi, H. Jung, B. Yun, S. Kim, W. Oh, E.-S. Lee, S. Park, E.-R. Jeong, J. Ko, and S.-G. Lee, "An LPWAN radio with a reconfigurable data/duty-cycled-wake-up receiver," in *2022 IEEE Int. Solid-State Circuits Conf. (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2022, pp. 404–406.

[119] M. K. Raja, Z. Bin, Y. D. Lei, and C. J. Leo, "A -121dBm sensitivity, 2.8µJ/bit Rx, 47.5% efficient Tx, narrowband IoT transceiver," in *2020 IEEE Asia-Pacific Microw. Conf. (APMC)*. Hong Kong, Hong Kong: IEEE, Dec. 2020, pp. 457–459.

[120] D. Lachartre, F. Dehmas, C. Bernier, C. Fourtet, L. Ouvry, F. Lepin, E. Mercier, S. Hamard, L. Zirphile, S. Thuries, and F. Chaix, "A TCXO-less 100Hz-minimum-bandwidth transceiver for ultra-narrow-band sub-GHz IoT cellular networks," in *2017 IEEE Int. Solid-State Circuits Conf. (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2017, pp. 134–135.

[121] N. Scolari, F. X. Pengg, K. Manetakis, C. A. Salazar, A. Vouilloz, E. P. Serna, A. Dissanayake, P. Persechini, V. Kopta, E. L. Roux, F. Chicco, S. Cillo, N. Gerber, C. Barbelenet, F. Epifano, P. A. D. Fabbro, and N. Raemy, "A 1mm^2 software-defined dual-mode bluetooth transceiver with 10dBm maximum TX power and -98.2dBm sensitivity 2.96mW RX power at 1Mb/s," in *2024 IEEE Int. Solid-State Circuits Conf. (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2024, pp. 402–404.

[122] A. H. Masnadi Shirazi, H. M. Lavasani, M. Sharifzadeh, Y. Rajavi, S. Mirabbasi, and M. Taghivand, "An ultra-low-power current-reused direct-conversion bluetooth-low-energy receiver front-end in 40-nm CMOS," *IEEE Trans. Microwave Theory Techn.*, vol. 69, no. 5, pp. 2697–2711, May 2021.

[123] P. Xu, D. Flandre, and D. Bol, "Analysis, modeling, and design of a 2.45-GHz RF energy harvester for SWIPT IoT smart sensors," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2717–2729, Oct. 2019.

[124] T. J. Odelberg, J. Im, and D. D. Wentzloff, "A 2.1mW -109dBm NB-IoT wake-up receiver," in *2021 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. Atlanta, GA, USA: IEEE, Jun. 2021, pp. 235–238.

[125] M. Moosavifar, J. Im, and D. D. Wentzloff, "An interference-resilient bit-level duty-cycled ULP receiver leveraging a dual-chirp modulation," *IEEE J. Solid-State Circuits*, vol. 59, no. 2, pp. 337–348, Feb. 2024.

[126] A. Dissanayake, H. L. Bishop, J. Moody, H. Muhlbauer, B. H. Calhoun, and S. M. Bowers, "A multichannel, MEMS-less -99dBm 260nW bit-level duty cycled wakeup receiver," in *2020 IEEE Symposium on VLSI Circuits*. Honolulu, HI, USA: IEEE, Jun. 2020, pp. 1–2.

[127] A. Dissanayake, H. L. Bishop, S. M. Bowers, and B. H. Calhoun, "A 2.4 GHz -91.5 dBm sensitivity within-packet duty-cycled wake-up receiver," *IEEE J. Solid-State Circuits*, vol. 57, no. 3, pp. 917–931, Mar. 2022.