NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

ANDRÉ ALEXANDRE PIRES DUARTE

Licenciado em Engenharia Eletrotécnica e de Computadores

Low-Power and Low-Noise Clock Generator for High-Speed ADCs

MESTRADO EM ENGENHARAIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa
Março, 2022

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES

Low-Power and Low-Noise Clock Generator for High-Speed ADCs

ANDRÉ ALEXANDRE PIRES DUARTE

Licenciado em Engenharia Eletrotécnica e de Computadores

Orientador: Luís Bica Oliveira,
Professor Associado, Universidade NOVA de Lisboa

Júri:

Presidente: Filipe de Carvalho Moutinho,
Professor(a) Catedrático(a), FCT-NOVA

Arguentes: João Carlos de Palma Goes,
Professor(a) Catedrático(a), FCT-NOVA

Orientador: Luís Bica Oliveira,
Professor Associado, FCT-NOVA

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa

Novembro, 2021

Acknowledgments

I will start by thanking Prof. Luis Olivera for all the motivation, support and reassurance he provided during the whole thesis. Even with the pandemic and in the most difficult times I went through he was always there and was always available to help.

I also want to thank my parents for all the motivation they gave me throughout the thesis and the entire course, because there were moments when I thought about giving up and they were the ones who gave me the necessary strength to finish it.

Finally, I want to thank a special friend who in my most difficult moments never abandoned me and who has always helped me throughout my life.

Resumo

O rápido desenvolvimento das tecnologias de comunicação de alto desempenho, reflete uma tendência clara na exigência dos requisitos impostos aos conversores analógico-digital (ADCs). Deste modo, verifica-se que estes requisitos implicam elevadas frequências não só sinal de entrada, como também frequências elevadas de amostragem o que se traduz numa maior sensibilidade do circuito ao ruído térmico e consequente aumento ruído de fase. Esta problemática, surge como propósito principal deste documento, no qual se procurará, como objetivo principal, o desenvolvimento de uma arquitetura que permita gerar múltiplos sinais de relógio a altas frequências de entrada e períodos de amostragem, com um baixo jitter e baixa energia consumida de forma a tornar mais eficiente e rápido o funcionamento de ADCs. Ruido térmico.

Esta dissertação propõe uma arquitetura composta por um amplificador de sinal de relógio que converte o duplo sinal de entrada num único sinal de saída, um amplificador digital que transforma uma onda sinusoidal numa onda quadrada e por fim um gerador de fase múltipla de sinais de relógio (MPCG), constituído por registos de deslocamento. Ambas as arquiteturas são implementadas em tecnologia CMOS de 130 nm.

A arquitetura é alimentada com um sinal LVDS de 200 mV de amplitude e com uma frequência de 1 GHz, de forma a obter à saída 8 sinais de relógio de onda quadrada com uma amplitude de 1,2 V e com 125 MHz de frequência. Os sinais obtidos à saída posteriormente alimentarão uma arquitetura de 8 canais com multiplexagem temporal.

A área total do circuito implementado é cerca de 8054,3 μm^2, para uma potência dissipada de 5,3 mW e para um valor de *jitte*r de 1,13 ps.

Esta nova arquitetura será direcionada para todo o tipo de entidades que trabalham com dispositivos que são constituídos por ADCs de alta velocidade de desempenho, de forma a poder melhorar o funcionamento desses mesmos dispositivos, tornando o processamento de sinal continuo para sinal discreto o mais eficiente possível.

Palavras-chave: amplificador de sinal de relógio, gerador de fase múltipla de sinais de relógio (MPCG), baixo jitter, baixa energia consumida, alta frequência, ADCs.

Abstract

The rapid development of high-performance communication technologies reflects a clear trend in demanding requirements imposed on analog-to-digital converters (ADCs). Thus, it appears that these requirements imply higher frequencies not only for the input signal but also higher sampling frequencies, which translates into a higher sensitivity of the circuit to thermal noise and consequent increase in phase-noise. This arises as to the main purpose of this document, which will seek, as its main objective, the development of an architecture that allows the generation of multiple clock signals at high input frequencies with low jitter and low power dissipation to make ADCs more efficient and faster.

This dissertation proposes an architecture implemented by a Clock Buffer that converts a differential input signal into a single-ended output signal, a Digital Buffer that transforms a sine wave into a square wave, and finally a Multi Clock Phase Generator (MPCG), consisting of Shift Registers. Both architectures are implemented in 130 nm CMOS technology.

The architecture is powered by a LVDS signal with an amplitude of 200 mV and a frequency of 1 GHz, in order to output 8 square wave clock signals with an amplitude of 1.2 V and with a frequency of 125 MHz. The signals obtained at the output later will feed an architecture of 8 Time-Interleaved ADCs.

The total area of the implemented circuit is about 8054.3 μm^2, for a dissipated power of 5.3 mW and a jitter value of 1.13 ps.

This new architecture will be aimed at all types of entities that work with devices that are made up of high-speed performance ADCs, to improve the operation of these same devices, making the processing from a continuous signal to a discrete signal as efficiently as possible.

Keywords: Clock Buffer, Multi-Phase Clock Generator, low jitter, low power, high frequency, ADCs.

Contents

Acknowledgments ..... **V**

Resumo ..... **VII**

Abstract ..... **VIII**

Contents ..... **IX**

List of Figures ..... **XIII**

List of Tables ..... **XVI**

Acronyms ..... **XVIII**

1. Introduction **1**

1.1. Context and Motivation 1

1.2. Problems and Objectives 2

1.3. Document Organization 3

2. Basic concepts of ADCs **4**

2.1. ADC Architectures 4

2.1.1. Parallel (Flash) ADC 5

2.1.2. Counter type ADC 6

2.1.3. Successive-Approximation Register (SAR) ADC 7

2.1.4. Dual-slope integration ADC 8

2.1.5. Pipeline ADC 9

2.2. Time-Interleaving ADCs 10

2.3. Performance Metrics 11

2.3.1. Aperture Jitter (jitter noise) 12

2.3.1.1. Cycle-to-Cycle Jitter 12

2.3.1.2. Period jitter 13
2.3.1.3. Long-term Jitter 13
2.3.1.4. Phase jitter 13
3. State of Art 15
3.1. Clock Buffer 15
3.1.1. Clock Recovery Circuitry 17
3.1.2. Existing topologies 18
3.2. Multi-Phase Clock Generator 23
3.2.1. Different Topologies 23
3.2.1.1. Shift Registers 24
3.2.1.2. Delay Locked Loop (DLL) 25
3.2.2. Output Jitter 25
3.2.2.1. SR Output Jitter 26
3.2.2.2. DLL Output Jitter 26
3.2.3. Jitter Comparison Between Topologies 28
3.2.3.1. Comparing Jitter Transferred from the Reference Clock 28
3.2.3.2. Comparing Jitter Generated Due to Thermal Noise 28
3.3. Discussion 28
3.3.1. Clock Buffer 29
3.3.2. Multi-Phase Clock Generator 30
3.4. Proposed Circuit 31
4. Circuit Implementation 33
4.1. Clock Buffer 33
4.1.1. Clock Buffer Sizing 34

4.1.2. Scaling W and L 36
4.1.3. DC Operating Point 42
4.2. Multi-Phase Clock Generator 42
4.2.1. Transmission Gates 43
4.2.2. Inverter 45
4.2.3. D-type Flip-Flops 45
4.2.3.1. Types of Memory 46
4.2.3.2. Different Topologies 47
4.2.3.3. Flip-Flop Sizing 49
4.2.4. Shift Registers 51
4.2.4.1. Ring-Counter 52
4.3. Digital Buffer 53
4.3.1. Digital Buffer Sizing 54
4.4. Clock Circuit 54
4.4.1. Rx 54
4.4.2. D-type Flip-flop 56
4.4.3. Shift Register 57
5. Electric Simulations 59
5.1. Description of the Electric Simulations 59
5.2. Time Response Simulations 59
5.3. Phase-Noise Analysis 61
5.3. Monte Carlo Simulation 66
5.4. Discussion 67
6. Layout 69

6.1. Rx Core 70

6.2. Shift Register 73

6.3. Analysis of the results 77

7. Conclusions 79

7.1. Future Work 80

8. References 81

Appendix A: DC Operating Point 85

Appendix B: Inverter operating mode 86

Appendix C: Jitter calculation function 88

List of Figures

Figure 2.1- N-bit flash ADC based in [4]. 5

Figure 2.2- Counter type ADC based in [4]. 6

Figure 2.3- A block diagram for a successive-approximation ADC based in [4]. 7

Figure 2.4- A basic block diagram for a dual-slope integration ADC based in [4]. 8

Figure 2.5- Pipeline ADC based in [4]. 9

Figure 2.6.- Time-interleaved ADC with N channels based in [5]. 10

Figure 2.7- Phase Noise plot based in [11]. 14

Figure 3.1- Simplified block diagram of a possible scheme for a clock recovery circuit. a) Current mode, b) Voltage mode c) Simple Buffer 16

Figure 3.2- Simplified block diagram of a possible scheme for ultra-low-jitter clock recovery circuit based in [15]. 17

Figure 3.3- Schematics of the selected low-jitter clock recovery circuits. a) SRL, b) SALPF, c) SRL+SALPF d) SAL+RCMF based in [15]. 21

Figure 3.4- More schematics of the selected low-jitter clock recovery circuits. a) QILA, b) HGRA, c) LNLGA based in [18]. 23

Figure 3.5- SR MPCG architecture based in [19]. 24

Figure 3.6- DLL MPCG architecture based in [19]. 25

Figure 3.7- Final schematic of the proposed circuit. 31

Figure 4.1- Current behavior of the SAL+RCMF circuit transistors as a function of their channel width. 37

Figure 4.2- Current/Power and Output voltage behavior of the SAL+RCMF circuit transistors M0 and MB as a function of their channel length. 37

Figure 4.3- Output voltage behavior of the SAL+RCMF circuit transistors as a function of their channel width. 38

Figure 4.4- Node voltage behavior of the SAL+RCMF circuit transistors a function of their channel width a) N2L, b) N1. 39

Figure 4.5- a) Power behavior of the SAL+RCMF circuit transistors as a function of their channel width. b) Current behavior of the SAL+RCMF circuit transistors as a function of their channel width. 41

Figure 4.6- CMOS Transmission gate and symbol representation presented in [22]. 43

Figure 4.7- CMOS Inverter and symbol representation..... 45

Figure 4.8- Single-threshold transmission gate flip-flop presented in [29]..... 48

Figure 4.9- Dynamic transmission gate flip-flop presented in [26]..... 49

Figure 4.10- Johnson Ring Counter..... 52

Figure 4.11- Chain of inverters with exponentially increasing size based in [40]..... 53

Figure 4.12- Rx core circuit..... 55

Figure 4.13- SR final circuit. 57

Figure 5.1- Simulation results of the Rx core,..... 60

Figure 5.2- Simulation results of the 8-phase IC. 60

Figure 5.3- Simulation results of the clock buffer output phase noise. 62

Figure 5.4- Simulation results of the Rx core output phase noise. 63

Figure 5.5- Simulation results of the IC output phase noise. 63

Figure 5.6- Simulation results of the CC and its elements output phase noise. a) 500 MHz, b) 1 GHz..... 64

Figure 5.7- Histogram of the simulated a) Phase Difference b) Phase (200 Monte Carlo runs). 66

Figure 6.1- Clock Buffer layout. 70

Figure 6.2- Digital Buffer layout 71

Figure 6.3- Rx core layout. 72

Figure 6.4- a) Inverter Layout, b) Transmission gate Layout..... 73

Figure 6.5- DTG flip-flop Layout. 74

Figure 6.6- SR Layout..... 74

Figure 6.7- Chip Layout..... 75

Figure B.1- Operations Regions of the CMOS Inverter based in [43]. 86

Figure C.1- Imporfile function code 88

Figure C.2- Code to calculate Jitter. 89

Figure C.3- Jitter function code. 89

List of Tables

Table 3.1- Advantages and disadvantages of the clock buffer architectures. 29

Table 4.1- Dimensions of the SAL+RCMF circuit transistors. 36

Table 4.2- New dimensions of the SAL+RCMF circuit transistors 41

Table 4.3- DC Operating Point – New voltages of the SAL+RCMF circuit transistors. . 42

Table 4.4- Truth Table of the Transmission Gate presented in [22]. 44

Table 4.5- Dimensions of CMOS Inverter transistors. 50

Table 4.6- Dimensions of CMOS transmission gate transistors 51

Table 4.7- Dimensions of the Digital Buffer transistors. 54

Table 4.8- Final dimensions of the SAL+RCMF circuit transistors. 55

Table 4.9- Power and Delay output result of the DTG and STTG flip-flop. 56

Table 5.1- Power consumption of the IC and its elements. 61

Table 5.2- RMS Jitter of the IC and its elements. 65

Table 5.3- RMS Jitter of the SR. 65

Table 5.4- Power consumption and jitter of the IC and its elements. 67

Table 5.5- Jitter of the SR. 67

Table 6.1- Layout areas of the elements of the Rx core. 72

Table 6.2- Layout areas of the elements of the SR. 75

Table 6.3- Layout areas of the elements of the IC. 76

Table 6.4- Comparison between the proposed Rx and other Rx core implementation @500 Mhz. 77

Table 6.5- Comparison between the proposed CC and other CC implementation @1 Ghz. 77

Table A.1- DC Operating Point - Voltages of the SAL+RCMF circuit transistors..... 85

Acronyms

ADC Analog-to-Digital Converter
CMOS Complementary metal-oxide-semiconductor
CP Charge Pump
D2S differential-to-single
DAC Digital-to-Analog Converter
DLL Delay-locked loop
DU Delay unit
ENOB Equivalent Number of Bits
FFD flip-flop D
FL Flip-Logic
HGRA High-Gain Regenerative Amplifier
CC Clock Circuit
LVDS Low-Voltage Differential Signal
LP Loop Filter
LSB Least Significant Bit
LNLGA Low-Noise Low-Gain Amplifier
MPCG Multi-Phase Clock Generator
MSB Most Significant Beat
NMOS N-channel metal-oxide-semiconductor
PD Phase Detector
PMOS P-channel metal–oxide-semiconductor
PNOISE Periodic Noise Analyses
PSS Periodic Steady State
QILA Quasi-Infinite Load Amplifier
RCMF continuous-time passive common-mode feedback circuit
SAL Differential Amplifier with active load
SALPF Differential Amplifier with active load and positive feedback
SFDR Spurious-Free Dynamic Range
SINAD Signal-to-Noise and Distortion Ratio
SNR Signal-to-Noise Ratio
SR Shift Registers
SRL Differential Amplifier with resistive load
VCDL Voltage Control Delay unit

1

1. Introduction

1.1. Context and Motivation

Nowadays, the technology of integrated circuits is in constant development and mutation, being, therefore, relevant to the appearance of new solutions in the most diverse areas of application [1]. In fact, in most of the new electronic systems, recently developed, there is a tendency to replace analog circuits by equivalent digital systems, which makes ADC systems with an increasing importance in all electronic systems, and they are subject to a constant need of evolution.

Data-converters have several applications, namely in communication systems, mobile devices, medical devices, among others. This shift to equivalent digital systems, it is directly reflected in an increase in the data storage process in the digital domain. The success of this modification will imply the implementation of high-resolution ADCs, with high sampling rate, to improve the efficiency and speed with which analog information can be converted into digital, that is, the rapid evolution of ADCs will allow signal conversion in continuous-time, for discrete-time signal, with great quality factor. However, there are several challenges in this type of implementation, since the resolution of the ADC decreases with the sampling period and with the input frequency, which results in a conflict between the accuracy and speed of the converter [2]. Thus, the main factor to consider when it comes to converter degradation is the clock jitter, it is essential to limit this factor to high input frequencies, since this causes uncertainties in the sampling instant, resulting in loss of ADC resolution. To counteract such consequences, it is important to develop an architecture that can balance both jitter minimization and signal sampling performance.

Another important factor to consider, not only in ADCs but in all electronic circuits, is the power dissipated by them, being necessary to find a balance between consumed power and jitter. Throughout this document, some architectures that meet the intended requirements, mentioned above, will be addressed.

1.2. Problems and Objectives

This dissertation aims to develop a multiphase clock generation circuit for high-speed ADCs with minimal phase-noise, jitter and the lowest power dissipation possible. However, during its design, several challenges will have to be considered, namely: sizing an architecture capable of generating a clock signal that will feed the converter, allowing to work with high input frequencies in a range between 500 MHz and 1 GHz and minimizing the jitter values, below 1 ps [3] resulting from the input frequency range; keep the power dissipation of the circuit at values within considerable ranges, below 10 mW.

The intended architecture will be divided into three main sub-architectures, depending on the desired requirements. In an initial phase, two sinusoidal signals of about 200 mV peak-to-peak will be generated by two ideal voltage sources, which will feed a clock buffer. In a second phase, the clock buffer will process these signals, which, in turn, will amplify them and transform the differential input signals into a single-ended output clock signal. This signal will later be processed in a digital buffer that will transform the sinusoidal wave into a new wave, a pulse with a square shape with 1.2 V peak-to-peak. Finally, this single-ended signal will feed a new architecture, which will generate an even number of signal clock phases. This architecture will be scaled with D flip-flops to obtain an 8-phase output signal with a 50 % duty cycle and a frequency range between 65 MHz and 125 MHz. The duty cycle can be changed according to the system's needs, for example, to 12.5 % using logical NAND gates.

This dissertation will focus essentially on the aforementioned architectures, which are the main points of interest. For the success of this work, it is intended to use existing architectures, size them with a 130 nm CMOS technology, and group them, so that they fulfill the intended simulation requirements.

1.3. Document Organization

This dissertation is organized by several themes, consisting of 7 Chapters and an Appendix. Note that each Chapter may include multiple sections.

Chapter 1 presents a first general approach to the content that will be presented, framing the importance that ADCs have gained in the technological world. In this context, the problem/focus of the dissertation is described, introducing the objectives.

In chapter 2, a brief review of concepts is made, namely what types of ADCs and metrics to consider when working with this type of devices.

Chapter 3 consists of a review of the available literature. This chapter addresses the work and research done in this field and what insights they can provide for the elaboration of the dissertation. In this way, the main operating characteristics of architectures capable of meeting what is intended are explored in detail, stating different implementation topologies. Subsequently, the main advantages and disadvantages of each topology are presented, and, through this information, it will be possible to develop a model that uses the best characteristics of the best topologies.

Chapter 4 presents the characteristics of the main topologies chosen for the implementation of the Clock Circuit (CC) as well as its sizing.

In Chapter 5, electrical simulations of the CC are elaborated to test its performance with the main interest in power and Phase jitter performance. At the end of the chapter, the electrical simulations results are duly discussed.

Chapter 6 presents the layout of the various CC elements and how they were implemented. In the final part of the chapter, the results obtained from the electrical simulations and from the layout are compared with other architectures previously studied and identical to those that were implemented.

Finally, in Chapter 7, the conclusions of the dissertation study are presented, proposing some future works that would be interesting to investigate to complement this dissertation.

2

2. Basic concepts of ADCs

2.1. ADC Architectures

The rapid evolution of digital integrated circuit technologies has led to the development of signal processing systems, increasing their level of sophistication. This development of signal processing systems brought a set of new advantages such as easy operation, high reliability and good stability. These integrated circuits operate on a wide variety of continuous-time signals, including telecommunications, consumer electronics, instrumentation, warfare systems, medical imaging, speech, among others [1]. The constant development of analog-to-digital converters (ADCs) is one of the main keys to the great success of these integrated circuits. The function of the ADC is to convert continuous analog signals to discrete digital signals, so to get a high performance of electronic systems, it is important to understand the important role that the digital-to-analog (DAC) converter plays. As a result, the number of signals to be digitized increased, which led to a diverse selection of data converters in terms of architecture, resolution and sampling rates [4].

Throughout this chapter, different types of ADC architectures will be analyzed. Each type of converter has its own set of features that satisfy different applications. There is, however, a set of ADCs that are most used in today's digital world, among them are: Flash ADC, Digital Ramp ADC, Dual-slope ADC, Successive Approximation Register ADC (SAR), Pipeline ADC. These types can vary on three main metrics: accuracy, interface, and speed.

2.1.1. Parallel (Flash) ADC

Parallel (Flash) ADC (Figure 2.1) is a kind of ADC with the highest converter rate and, therefore, is best suited for applications that require large bandwidth. However, this converter has some limitations. To be one of the fastest to convert the analog signal into a digital signal will need more chip area which translates into a higher cost, consumes more power and has a very limited resolution. These limitations will have a big impact on the ADC performance for high frequencies. An *N*-bit Flash ADC employs 2^N-1 cascading comparators and 2^N resistors. The input signal is applied to the ADC on all comparators. Then the comparators will compare the unknown input voltage with the voltage corresponding to each comparator. The voltage corresponding to each comparator depends on the periodicity level in the ADC, i.e., the resistance of the chain to which the comparator is associated. The result of the comparisons is a digital thermometer code that will later be decoded and translated into *N* bits. [4].

The resolution of the converter is directly dependent on the number of comparators and resistors, that is, for a higher resolution, more devices will have the ADC. If the goal is for the comparator to perform quickly, it must run at relatively high-power levels. Therefore due to the large number of high-speed comparators, the Flash ADC has some problems, these include high power dissipation which consequently leads to a rather large and expensive chip size (the bigger the chip the more expensive it is). In addition to these problems, it is important that the chain resistance remains relatively low to provide suitable bias current to the comparators.

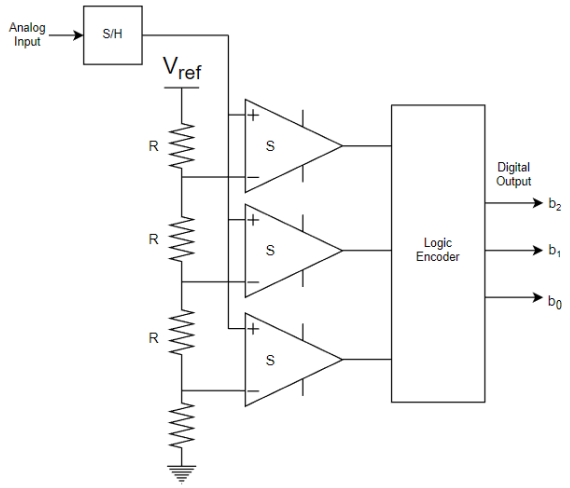


Figure 2.1- N-bit flash ADC based in [4].

2.1.2. Counter type ADC

Counter type ADC (Figure 2.2) is a converter that produces a sawtooth signal that rises or falls. When the conversion cycle begins, the timer starts counting until the voltage at the single oscillator output (ramp voltage) matches the ADC input signal. Then the comparator triggers and the value recorded by the timer is saved. Since the ramp voltage is normally generated by a single oscillator, the ramp time is more sensitive to temperature. To counteract this sensitivity, there is a solution that consists of using a counter with a clock to activate a DAC and using a comparator to preserve the time recorded by the counter. The main improvement of this new ramp implementation is that to store the voltage value obtained in the second signal comparison, the system only requires another register and another comparator [4].

There is a very simple way to implement a Counter type ADC with just a capacitor, a resistor and a microcontroller. When the analog input signal is connected to the comparator terminal, the binary counter is activated. The counter will later feed the DAC, to convert the digital signal to analog so that it can be compared to the input voltage. As the counter gradually increments the DAC output will increase. Until the DAC output voltage exceeds the unknown analog input voltage, this process will continue. This process will not only cause the counter to stop but will also affect the comparator output by changing its value. The corresponding time the counter takes to stop will represent the analog input voltage value.

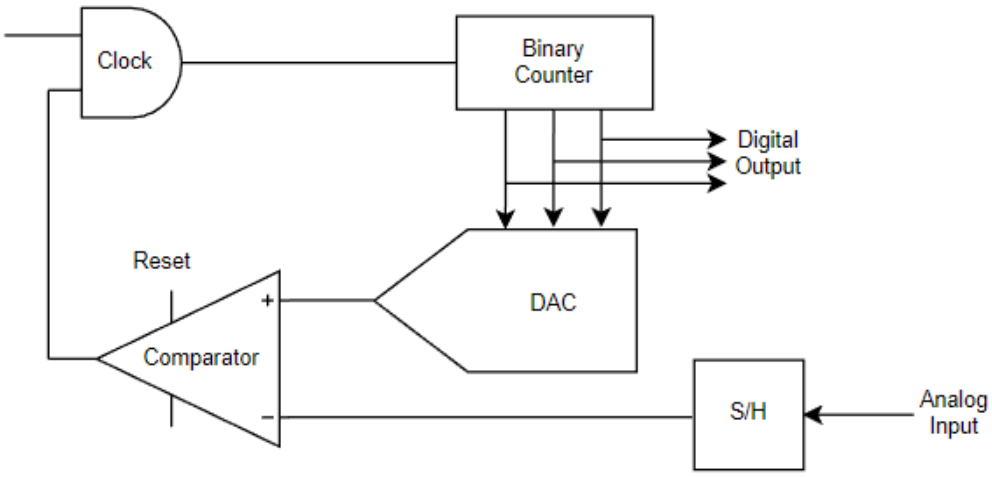


Figure 2.2- Counter type ADC based in [4].

2.1.3. Successive-Approximation Register (SAR) ADC

A successive-approximation Register ADC (Figure 2.3) works by using not only a comparator that allows finding the input voltage by performing a binary search, but also a digital-to-analog converter (DAC). Compared to digital ramp ADC it is much faster because the binary search is performed on all of the ADC quantization levels that it has at its disposal before converging to the final value. Explaining by other words the ADC converges to the value closest to the analog input voltage by using digital logic. Although it is more advantageous than the digital ramp ADC, it has a high resolution, which makes it slower, ceiling a maximum speed of about 5 Msps. Compared to other ADCs it has a simpler design which allows high resolution and high speed, a significant small chip area and a good balance between speed and cost. This features make this type of converter used in most modern IC ADCs [4].

As mentioned, there is a binary search to obtain the corresponding digital signal at ADC's output. The search begins with the most significant bit (MSB) and works on decoding the bits until it reaches the least significant bit (LSB). When the conversion starts the control logic sets the most significant bit of the successive approximation register (SAR) with a value of "1" and the other bits are initialized at "0". Afterwards the SAR information is fed into the digital-to-analog converter which produces an analog voltage (half of the reference voltage). This voltage is compared with the unknown voltage supplied to the input of the ADC. If the input voltage is smaller than the DAC output value, then the comparator induces the SAR control logic to reset the MSB back to "0" and set the next bit to "1" to repeat the process. If, on the other hand, the input voltage is higher than the voltage at the output of the DAC, this bit will remain at "1". After finding the value of the bit, the next significant bit will also be set to the value of "1" and the previous process is repeated. At the end of the search the result at the SAR output is a digital estimation of the analog signal sampled at the ADC input.

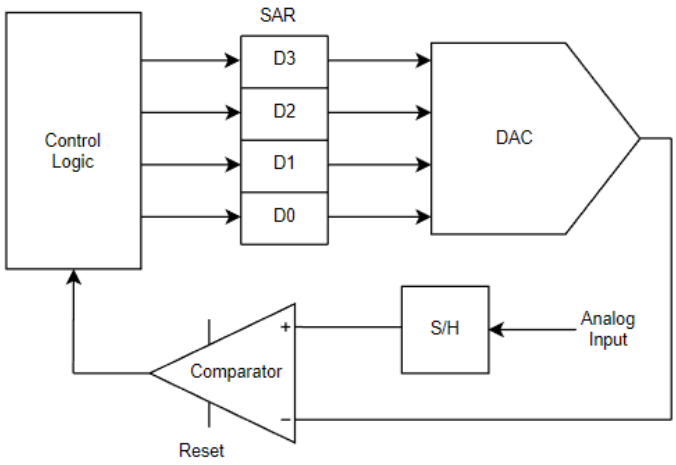


Figure 2.3- A block diagram for a successive-approximation ADC based in [4].

2.1.4. Dual-slope integration ADC

Dual-slope ADC (Figure 2.4) applies an unknown input voltage to the integrator and this voltage increases until a predetermined amount of time (T), at which time the binary counter is initialized. After that period, the reference voltage, but with opposite polarity is applied to the integrator. During this period the integration capacitor accumulates charge. Based on the integration of the charge accumulated by the input voltage until the reference voltage is injected into the integrator (T) and the time (t) at which the output of the integrator returns to zero a binary value is defined by the counter. This binary value will be a representation of the input voltage. The time measurement of (t) is done in the same unit as the converter clock, therefore longer integration times allows higher resolutions. However there is a trade-off between resolution and ADC speed, by sacrificing resolution the ADC can be faster [4].

Converters of this type are used for their linearity and flexibility. However, to further decrease power consumption, a single-slope ADC may be another option. Single-slope ADC can be a more attractive solution to minimize power consumption because the converter is implemented by digital circuits, which only consume current when there is a pulse transition. Dual-slope ADC applications are limited to low sampling frequency, once the ADC requires many conversion clocks for one sampling.

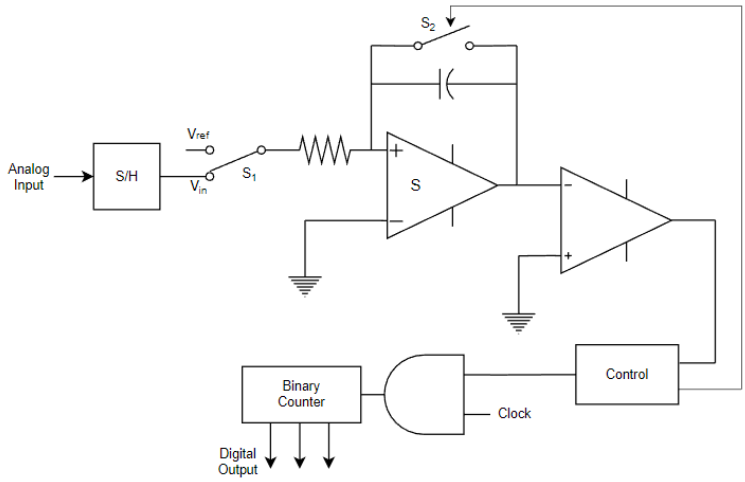


Figure 2.4- A basic block diagram for a dual-slope integration ADC based in [4].

2.1.5. Pipeline ADC

A pipeline ADC consists of several consecutive stages, each stage with 2 or 3 steps. In the first step, the input voltage is first sampled and held steady by a sample-and-hold while the ADC quantifies this voltage in n_2 bits. These bits will be fed to the DAC which generates an analog output where it will later be subtracted from the input signal. In the last step, the result obtained from the subtraction is amplified and fed to the next state. The ADC pipeline is an amplitude quantizer that has several stages in cascade where the digitization is performed. The number of stages depends on the resolution of the ADC, normally it matches the number of bits in the ADC. By using analog registers in the implementation of the ADC such as sample-and-hold amplifiers (SRAs) it is possible to obtain high conversion performance.[4].

Some modern pipelined ADCs employ a technique called "digital error correction" which greatly reduces the accuracy requirements of the ADCs used in their implementation. This implementation resorts to the use of individual comparators. The use of comparators allows the architecture to tolerate large displacements due to internal decision-level overlaps between successive stages. With this implementation, the conversion will depend exclusively on the precision, not only of the residual signals, but also on the conversion speed. However, there are still limitations as the gain is not corrected for errors in the individual DAC as well as in the gain amplifiers. Based on what has been mentioned, the pipeline is a very attractive architecture for applications that operate at high sampling rates.

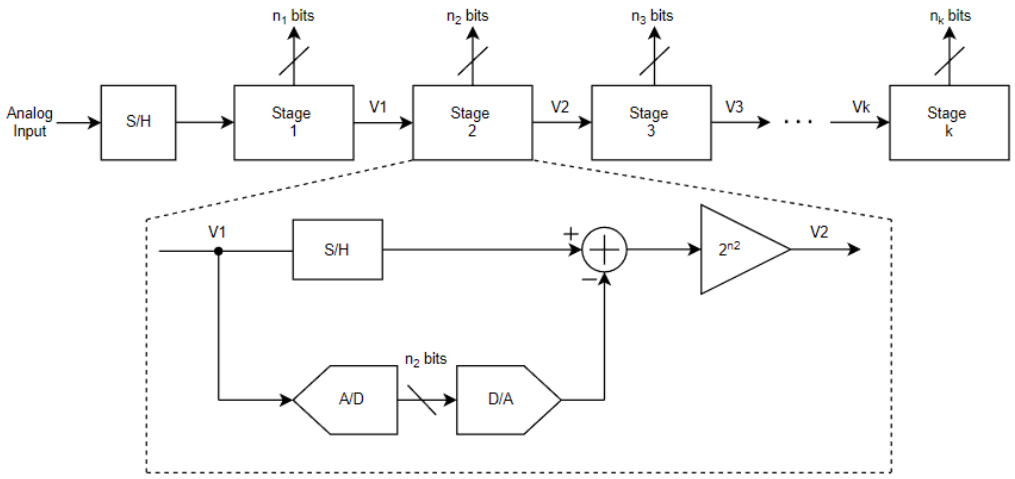


Figure 2.5- Pipeline ADC based in [4].

2.2. Time-Interleaving ADCs

A Time-Interleaved ADC (Figure 2.6) is made up of *N* parallel channels of any kind of ADC architecture that alternately each process a sample, where a channel's sampling frequency no need to match the Nyquist Criterion. However, when the analog samples are processed and converted to the digital domain, they form a sequence where the sampling frequency respects the Nyquist criterion. Thus, an ideal TI-ADC allows sampling with *N* channels equivalent to sampling with an ideal ADC with a sampling rate *N* times higher. The channels of a TI-ADC can have different implementations based on the conversion technology used and the intended objective, for example by selecting high-rate ADCs and then choosing which factor is more important to value, power or resolution. By increasing the number of channels, it is possible to increase the sampling rate of the TI-ADC. In an ideal and theoretical model, the dimensioning of the sampling rate would be according to the number of channels. Nevertheless, this process is not so linear, as there are mismatches between channels that directly influence and limit the performance of TI-ADCs. [5].

The fact that there is a tendency to downscaling IC technologies can complicate the matching of components that implement this type of architecture, but on the other hand, the increase in density makes it possible to add equivalent digital components with small costs. The existing method to mitigate the problems of analog conversion circuits is to add equivalent digital circuits. TI-ADCs are an excellent representative example of this type of technology where implementations will be increasingly worked on and improved.

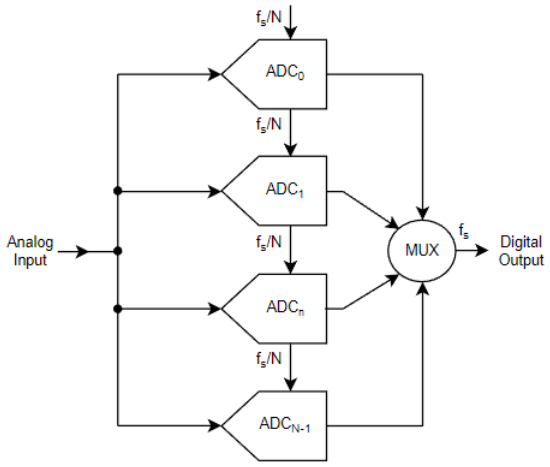


Figure 2.6.- Time-interleaved ADC with N channels based in [5].

As the main objective of this document is to obtain a system capable of processing high-resolution and high-speed sampling signals, it is necessary to consider an ADC architecture that complies with the new communication standards. Today's ADC technologies work very close to their limits and have limitations that prevent them from being properly pushed any further, as it is rather challenging to design ICs with sub-micron technologies. However, increasing the density of the components has a significant advantage since the circuit allows the dimensioning of a digital circuit with a smaller area. By reducing the chip area, it allows adding new components that will only have small additional costs. Parallelism is a method used to overcome the aforementioned performance limits. Splitting the analog information that is fed into the ADC input into multiple parallel channels, converting them independently, and finally recombining the samples into a digital output signal makes the signal conversion process much more efficient. Thus, the ADC architecture that will be considered will be the TI-ADC [6].

2.3. Performance Metrics

Before taking an approach to the clock signal that feeds the ADC, there is a need to adequately understand the main characteristics and parameters inherent to the ADC. Since there is a great variety of converters of this type, as has already been shown, it is essential to consider a large number of evaluation parameters, as there is still no universal parameter. However, we can consider a range of more relevant parameters, which will influence its normal functioning, they are: SNR; SFDR; THD; SINAD; and ENOB [7].

The SNR relates the effective power of the ADC input signal to the noise power. The higher this ratio, the lower the noise at the ADC input.

$$\mathrm{SNR} = 10\log\left(\frac{\mathrm{P_{signal}}}{\mathrm{P_{noise}}}\right) \tag{2.1}$$

The THD refers to the effective power of all harmonics present in the converter with the power of the input signal.

$$\mathrm{THD} = 10\log\left(\frac{\mathrm{P_{harmonics}}}{\mathrm{P_{signal}}}\right) \tag{2.2}$$

SINAD, like SNR, also relates the effective power of the ADC input signal, but also considers distortion rather than just noise power. SINAD as it considers the power of all

spectral components fully represents the overall dynamic performance of the ADCs. SINAD also differs from SNR, as it emphasizes harmonic distortion, more specifically the first five harmonics.

$$\mathrm{SINAD} = 10\log\left(\frac{\mathrm{P_{signal}}}{\mathrm{P_{noise}} + \mathrm{P_{distorcion}}}\right) \tag{2.3}$$

ENOB is a very important metric for ADC performance and directly reflects the equivalent number of ADC bits. To calculate this parameter, a method called the sine wave adjustment method is used, for a given sampling time.

$$\mathrm{ENOB} = \frac{\mathrm{SINAD} - 1.76}{6.02} \tag{2.4}$$

The SFDR has a formula that is very similar to THD. However, it refers to the effective power of the fundamental wave with the maximum harmonic power.

$$\mathrm{SFDR} = 10\log\left(\frac{\mathrm{P_{harmonic}}}{\mathrm{P_{signal}}}\right) \tag{2.5}$$

2.3.1. Aperture Jitter (jitter noise)

Jitter is characterized as the time difference of the final signal edge set, both rising and falling from the ideal signal values. Clock signals are subject to noise and other disturbances that are primarily responsible for causing oscillations. The main factors causing oscillations in clock signals include coupled interference from nearby circuits, power supply variations, device noise, load conditions and thermal noise.

One of the main noise meters is jitter and there are four primary measures for clock jitter. Based on the performance requirements that the clock signal needs to achieve depending on the application for which it is applied, the measure for the jitter calculation is chosen. The following are the major types of jitter [10]: Long Term Jitter, Cycle to Cycle Jitter, Phase Jitter and Period Jitter.

2.3.1.1. Cycle-to-Cycle Jitter

Cycle-to-Cycle jitter is the maximum deviation between adjacent consecutive clock periods. These periods are defined for a fixed number of cycles, which can vary but are

normally within 1,000 cycles or 10,000 cycles. In other words, Cycle-to-Cycle jitter takes into account the rising edge of two adjacent clock periods and calculates the maximum deviation between them, so it can be reported as an RMS value in ps. If there is a need to limit a sudden jump in frequency, cycle-to-cycle jitter is used as a monitoring metric. This means that the characteristics that this type of jitter has are normally used to demonstrate the instability of systems that work with spread-spectrum clock signals, since they have less sensitivity to the spread-frequency feature [10].

2.3.1.2. Period jitter

Period jitter is the difference between two clock periods for a given random number of cycles. The selection of periods, however, is not random and the highest clock period and the lowest clock period are selected. There are several publications that take a different approach and define this type of jitter as the difference between the ideal period and the measured clock period. By having access to several individual clock periods, they can be measured which allows calculating not only the average clock period, but the peak-to-peak value and the standard deviation, as well [10].

2.3.1.3. Long-term Jitter

Long-term jitter measures the clock signal deviation at the clock signal output from the ideal position over numerous cycles. To choose the most suitable number of cycles to obtain the jitter value, it is necessary to consider the application of the signal. This type of jitter has different characteristics from both Cycle-to-Cycle and Period Jitter because for a designated continuous flow of clock cycles over a long time interval, it considers the cumulative jitter value, that is, the jitter increases as the number of cycles increases. It is for this reason that long-term jitter is sometimes called accumulated jitter [10].

2.3.1.4. Phase jitter

Before talking about Phase Jitter it makes sense to explain what Phase-Noise is. Phase-noise can be described in two ways, either as a graphical representation of continuous noise for a given range of frequencies or a set of noise values at different frequency shifts. Let's focus on the first definition, considering a curve that represents the continuous noise for a given frequency spectrum, it is possible to integrate this curve in which the result of this integration will represent the Phase Jitter value expressed in seconds. When the signal, where the jitter is to be analyzed is a square wave, it is necessary to consider the carrier frequency. It makes sense to make this consideration because it is in the carrier frequency that are located most of the

noise energies [10]. Nevertheless, there is always a trace of signal energy that is "leaked" in a range of frequencies that can go to either side of the carrier. Phase jitter will be the jitter on which this document will focus the most and as such, it is necessary to find the method to find its value. To calculate this value, it is necessary to consider two offset frequencies and then relate the amount of phase noise energy contained between them relative to the carrier [11] (Figure 2.7).

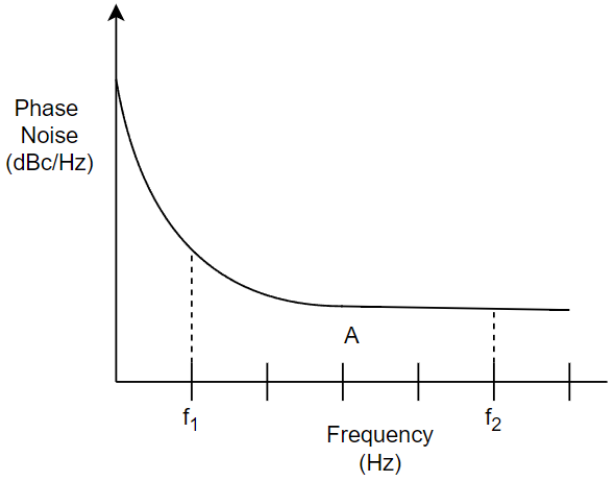


Figure 2.7- Phase Noise plot based in [11].

The RMS Phase Jitter between f_1 and f_2 is defined by equation (2.6):

$$\text{RMS Phase Jitter} = \frac{\sqrt{2\int_{f_1}^{f_2} 10^{A/10}}}{2\pi f_c} \tag{2.6}$$

3

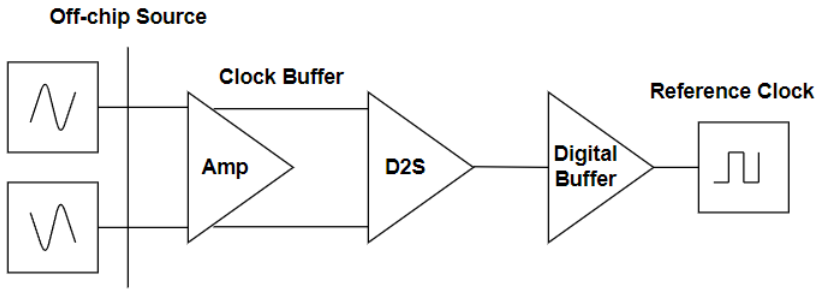
3. State of Art

Nowadays, one of the most important challenges for ADC architectures is to achieve the combination between a high value of SNR and a low value of THD, being, therefore, necessary to obtain a low jitter clock signal [12]. This problem becomes more difficult to mitigate when we are dealing with converters that require clock signals with high input frequencies. At the initial instant of sampling, there is a phenomenon called jitter, mentioned above, which directly influences the performance and noise of the system. Thus, digitizing high-speed signals requires adequate low-jitter reconstruction of the clock signal to fulfill the targeted sampling performance. For this purpose, a circuit called clock buffer was created, which aims to reconstruct the initial clock signal transforming it into a low jitter signal.

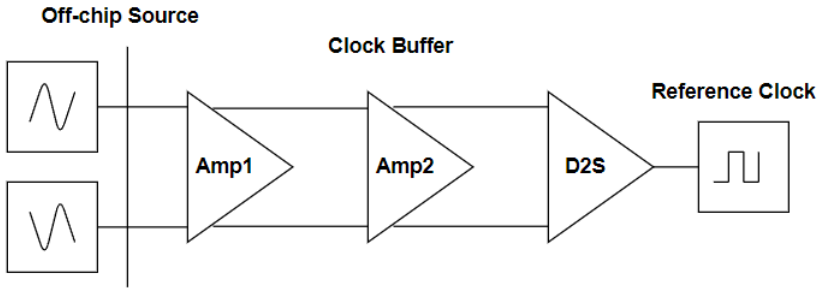
3.1. Clock Buffer

The clock buffer's function is to pre-amplify the input signal and transform the sinusoidal input wave into an output wave, with characteristics identical to the square wave. To obtain a wave with as perpendicular a slope as possible, the buffer needs to have a large mesh gain and, in turn, produce a low intrinsic noise. The first stage of the buffer is quite simple and consists of a differential-input and differential-output structure. One aspect to consider, to minimize noise, is to use resistors as load sources, instead of the active loads, if this possibility is valid. The second floor of the buffer is built by a differential input, but by a single output, unlike what was sized on the first stage. Additionally, still, regarding the second stage, it should be noted that the structure was selected, considering that one of the purposes of the buffer is to transform a sinusoidal differential signal into a single output signal (D2S circuit) [1], [13]. The buffer will require another architecture to connect to manually, a digital buffer that aims to flatten the output signal curve, both in the ascending and descending phases, accelerating the rise and fall time of the clock signal, which will be used to power the MPCG. The transistors and resistors are independent of each other, for the noise existing in the first stage, this must be amplified by the second stages of the buffer, which will have a significant impact on the total output noise. For this reason, the first floor is sized with active

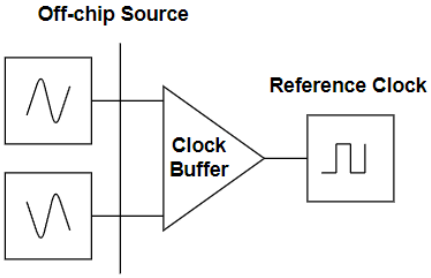
loads. Figure 3.1 shows schematics of base clock buffer architectures, where all types of architectures will be analyzed.



(a)



(b)



(c)

Figure 3.1- Simplified block diagram of a possible scheme for a clock recovery circuit. a) Current mode, b) Voltage mode c) Simple Buffer

3.1.1. Clock Recovery Circuitry

The on-chip clock recovery circuit (Rx) (Figure 3.2) considers several types of architectures, with several stages, which feature in its constitution a front-end differential amplifier, an optional differential to single-ended converter (D2S), which can do so much. operate in voltage mode as in current mode [14]. Afterwards, this converter is followed by a buffer with a back-end output. If the architecture does not have the D2S amplifier in the circuit, its role is played by the digital output buffer. However, this approach will not be the most suitable, which may imply some consequences, namely in the significant degradation in the performance of the ADC, for a given value of consumed power. As the purpose is to obtain a signal with low jitter and power, the use of the D2S stage becomes mandatory [15].

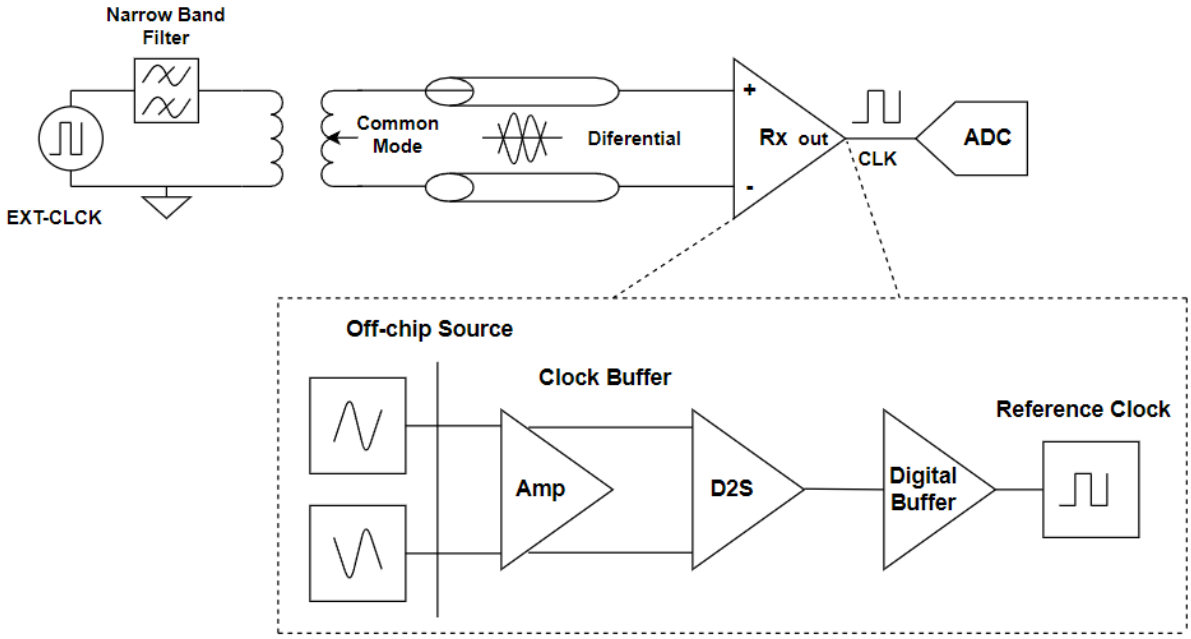


Figure 3.2- Simplified block diagram of a possible scheme for ultra-low-jitter clock recovery circuit based in [15].

In fact, as the aim is to minimize jitter, the D2S stage gains significant importance, as it plays a fundamental role in the clock signal reconstruction process [16]. The functioning of the D2S has a great impact on the slope of the clock signal curve, when it has a positive value, but it also impacts the gain that is obtained in the various stages of the circuit. As already mentioned, the objective is to reach a topology that allows to obtain performances with very low jitter values. To achieve this performance, the clock signal is applied as a differential

waveform, with characteristics very similar to the sine wave. This signal is obtained with the aid of a narrowband filter, together with a low jitter oscillator. The purpose of the circuit is to recover the internal clock signal, without clock jitter degradation, inside the chip. Once the input signal enters the chip, the front-end clock recovery (Rx) circuit will transform the differential signal into a low jitter single clock (CLK) signal that will later feed the ADC. With this approach, the clock signal is retrieved at the intended requirements.

The main objective in a clock signal recovery circuit is the signal reconstruction through a filtered wave very identical to the sinusoidal one, which is fed to the input. From a more functional point of view, the circuit must detect the zero point of voltage between the positive and negative input of the circuit. The clock buffer consists of a core (Rx), which basically consists of a comparator between the zero point of voltage crossing, in each period, and a digital output buffer, which provides the necessary conduction capacity for the load of output and consists of two cascaded CMOS inverters [15].

3.1.2. Existing topologies

As already discussed, the clock buffer can either operate in both voltage or in current mode so that, for both modes, different architectures will be presented. According to Juan Núñez et al.[15], there are several types of architectures, which can be compared to each other. On the other hand, two important characteristics were detected in the sizing of the low jitter clock signal recovery circuit, the D2S operation and the achievable gain.

Differential stage with resistive load (SRL)

The first architecture (Figure 3.3 a)) is based on a differential amplifier with resistive load (SRL) and represents a simplified scheme of the Rx core. The main disadvantage observed in this approach is that the conversion performed on the D2S circuit inputs is done in voltage mode by the digital output buffer (BUF). An additional, dummy buffer will have to be added to compensate the load of the transistor's nodes N1 and N2. As a result of this approach, the circuit will show relatively small gains and inefficient power consumption as a result half of the RX core energy is wasted. Furthermore, the size of this architecture can be quite complicated for low jitter and power consumption [15].

Differential stage with active load and positive feedback (SALPF)

The SRL architecture (Figure 3.3 b)) has many limitations, so, to alleviate the problems, it faces, a new topology was suggested, which uses a front-end differential amplifier, with active load together with a differential to single-ended (D2S) circuit in current mode, formed by transistors. The transistors M13 and M14 are in diode configuration and they represent the

active load of the circuit, they are also part of two current mirrors, while the transistors which are responsible to increase the output gain and are optionally used as positive feedback (PF), are the M15 and M16. The amount of feedback desired in the circuit can be controlled by the ratio of the original diode effect transistors to the M15-M16 cross pair.

This architecture can be called SALPF, as it presents a solution with positive feedback and at the same time can reach a considerable gain factor. The use of diodes in the circuit makes it possible to fix the first stage's operation point, at the low impedance output node. In extreme cases, where there is no positive-feedback, which is abbreviated to SAL, transistors M15 and M16 are no longer part of the circuit and the gain of the first stage is lower, because of the diode connection [15].

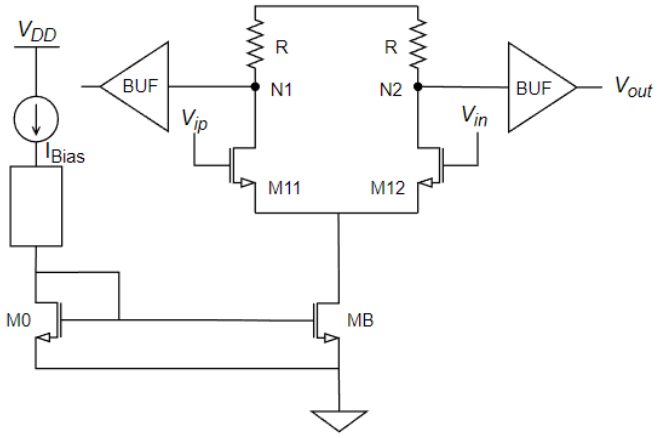
Two-stage architecture based on a SRL and a SALPF (SRL + SALPF)

In the third structure, a new approach was explored, in which two amplifiers with different functions were grouped together. This approach considered the possibility of using a dedicated D2S circuit, fully differential, to integrate the second stage of the circuit, to improve the gain. The resulting circuit, identified as SRL + SALPF, can be represented as a cascade circuit of receiver cores from both the SRL architecture and the SALPF architecture, in which the differential to single-ended signal conversion is decoupled from the first stage. As in the SALPF architecture (Figure 3.3 c)), transistors M13 and M14 connected to the diode fix the operating point, while transistors M15 and M16, responsible for the partial feedback of the second stage and increase circuit's voltage gain. Finally, transistors M21 to M24 perform de D2S operation. [15].

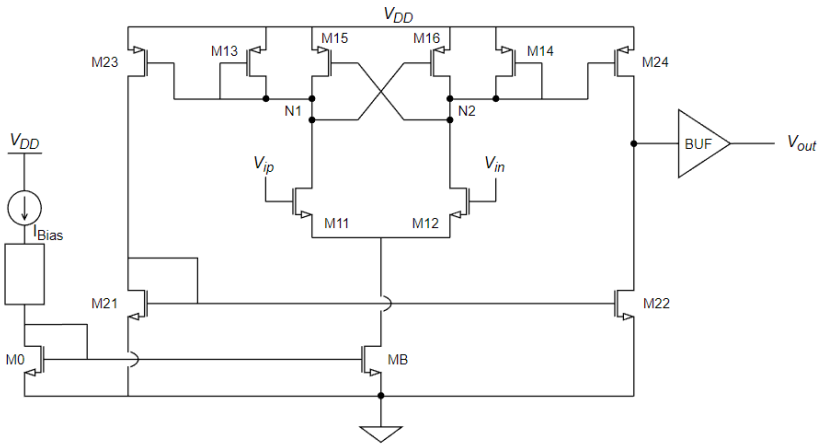
Differential stage with active load and passive common mode feedback (SAL + RCMF)

Taking these considerations into account, an alternative architecture (SAL + RCMF) has been decvided. This architecture (Figure 3.3 d)) consists of a first stage, completely differential, with an active load (SAL) and incorporates the D2S circuit in current mode (transistors M21 to M24). To establish the DC operating point in the drain of the PMOS transistors (N1 and N2), a new circuit was considered. This new approach consists of a continuous-time, with common-mode feedback circuit (RCMF), that embodies two resistors. This type of approach will not only allow an increase in the SAL circuit gain, which translates into an increase in the first stage gain, but, at the same time, it manages to define an appropriate input voltage for the D2S circuit (transistors M23 and M24) as well. In addition to what was mentioned, this architecture, demonstrated by the simulation carried out by the name of the author of the article, will allow a significant reduction in power consumption for a given jitter value. It is a very attractive approach, but it is important to emphasize that he main upper hand of this topology does not

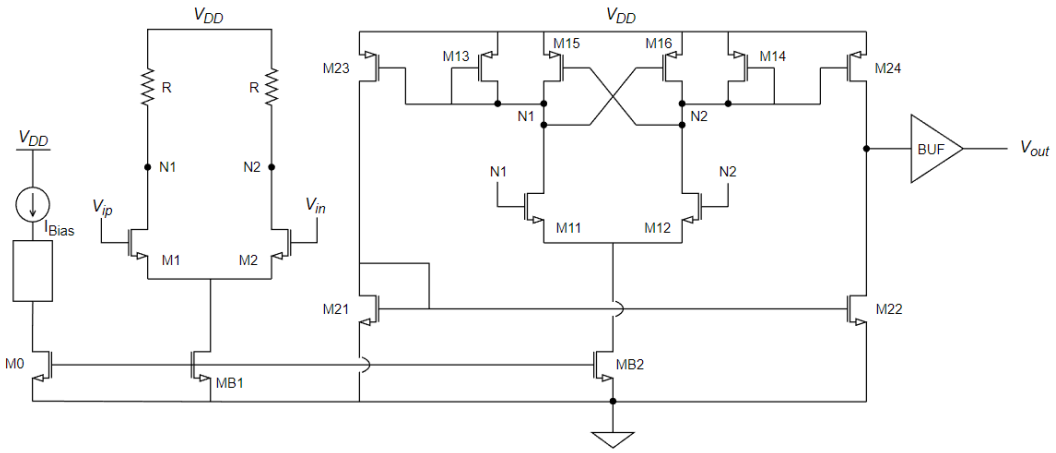
depend exclusively on the differential pair of the first stage, but also on the relationship between the two stages, that is, on the combination with the D2S circuit, since without its presence the jitter could be considerably higher [15].



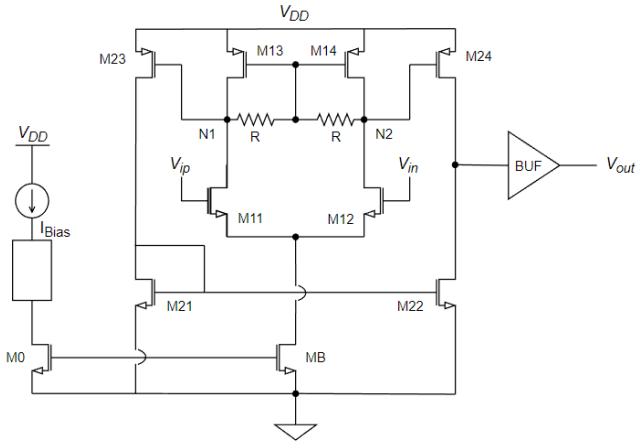
(a)



(b)



(c)



(d)

Figure 3.3- Schematics of the selected low-jitter clock recovery circuits. a) SRL, b) SALPF, c) SRL+SALPF d) SAL+RCMF based in [15].

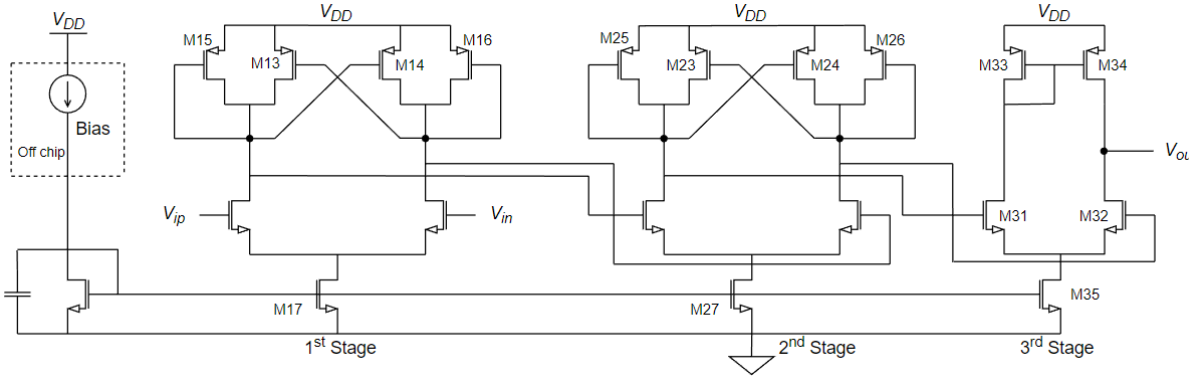
Quasi-Infinite Load Amplifier (QILA)

According to Cheng, Long et al.[18], a clock recovery circuit is presented, with characteristics quite identical to those already mentioned, however with some alterations. This topology is composed of two SALPF circuits and a D2S circuit which, unlike the previous topologies, operates in voltage mode and not in current mode, as in the other architectures. Another apparent difference in this topology is the fact that the SALPF and D2S circuits are independent of each other and, as such, this topology is called Quasi-Infinite Load Amplifier (QILA).

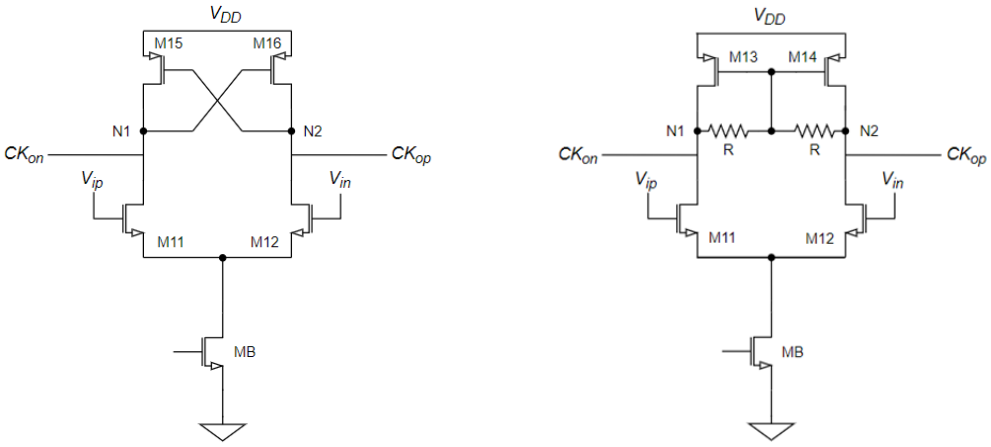
As can be seen from Figure 3.4 a), it becomes quite evident that the active load transistors, M13 and M14, work in diode configuration, in the first SALPF circuit, as well as transistors M23 and M24 in the second SALPF circuit. On the other hand, the set of transistors M15 and M16 in the first SALPF circuit and the set M25 and M26 in the second SALFP circuit are used as positive feedback to increase the amplifier gain. Finally, it should be mentioned that the topology will be composed of the D2S circuit that works in voltage mode.

Low-noise low-gain amplifier (LNLGA) and High-gain regenerative amplifier (HGRA)

In the same article a comparison of this amplifier was made to two others, widely used in clock recovery circuits, a Low-noise low-gain amplifier (LNLGA) and High-gain regenerative amplifier (HGRA) [16]. From what was presented in the article, the Quasi-Infinite Load Amplifier has superior characteristics when compared to the LNLGA (Figure 3.4 c)) and HGRA (Figure 3.4 b)), since, through the circuit simulation, the LNLGA presented a substantially lower gain and the output signal does not show a peak, making it more vulnerable to noise. The slope of the HGRA output signal has a more defined peak, but it is still subject to a cancellation factor, which happens when the amplitude of the input signal is small. Thus, the amplitude of the output signal does not change [18]. A possible solution to this problem would be to increase the value of the current at the input, which would result in an increase in power consumption.



(a)



(b) **(c)**

Figure 3.4- More schematics of the selected low-jitter clock recovery circuits. a) QILA, b) HGRA, c) LNLGA based in [18].

3.2. Multi-Phase Clock Generator

The next step focuses on the implementation of an architecture, which will make it possible to convert the signal obtained at the end of the clock buffer and reproduce it into eight signals with identical characteristics, but with different phases. These signals will later feed into the TI-ADC architecture. Note that, in this case, eight clock signals were considered, but the scaling can be done for any even number of outputs. As mentioned above, jitter plays an important role in clock signal processing and taking this into account, it is necessary to find the type of architecture that best minimizes this factor.

3.2.1. Different Topologies

To implement the intended architecture, Delay-locked loops (DLLs) and Shift Registers (SRs) are the options to consider. Therefore, it is necessary to carry out a study that allows us to infer which of the two has better characteristics, in order to more adequately satisfy the objectives that must be fulfill.

3.2.1.1. Shift Registers

The architecture of an SR MPCG (Figure 3.5) consists of a chain of D flip-flops (DFF) with N (where N represents the number of output signals) identical DFF's. A reference clock signal, with a frequency $N \cdot f$, feeds the chain of flip-flops. To generate the input clock signal, considering that we are working at high frequencies, we chose to use a sine wave signal, to avoid timing and reflection errors due to the input impedance adaptation problems. input. Next, a clock buffer is designed to amplify the input clock signal by modifying the input waveform to a square-wave clock signal. The possible architectures for the clock buffer structure have already been analyzed in previous chapters.

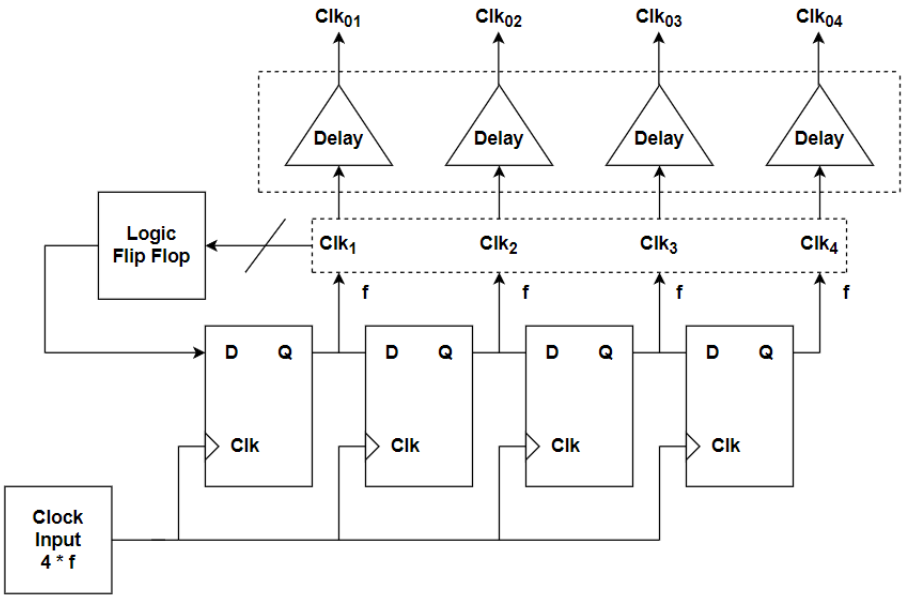


Figure 3.5- SR MPCG architecture based in [19].

The signal obtained at the output of the clock buffer will be the reference clock signal, which will feed the chain of D flip-flops (DFF). In other words, the outputs of the D flip-flops chain work with a given input frequency and the MPCG, constituted by Shift Registers, works as a frequency divider, depending on the number of outputs. Since an DFF is sensitive to rising peaks, the Q output of each DFF is delayed, relative to the output of the previous DFF, by a reference clock period, which is equivalent to a 2π / N phase delay [19].

8

3.2.1.2. Delay Locked Loop (DLL)

A Delay-Locked Loop (Figure 3.6) (DLL) consists of four main components: a phase detector (PD), a charge pump (CP), a closed loop filter (LF) and a voltage-controlled delay line (VCDL). The system has N delay units (DU) per each N DLL outputs.

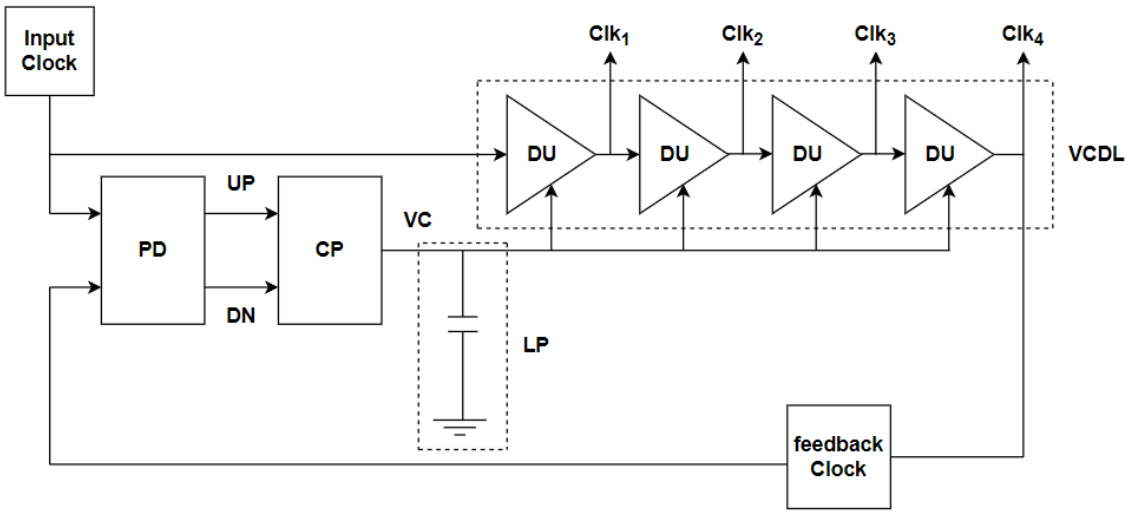


Figure 3.6- DLL MPCG architecture based in [19].

The DLL operation starts with the phase detector PD which compares the phase difference between the reference clock signal and the delayed clock signal, i.e., the signal from one of the VCDL outputs. On the other hand, the charge pump converts the phase detector output into a control voltage (VC), by charging or discharging the filter capacitor. The VC changes the delay of each delay cell in the VCDL, this means that when the phase of the feedback clock signal is more advanced than that of the reference clock signal, VC increases the delay of VCDL and vice versa. The negative feedback in the closed loop will have the function of adjusting VC, to reduce the phase error, until reaching the moment when the rise time of the reference clock signals, and the feedback signals are aligned. Since each delay cell provides the same delay, with the same control voltage value, the DLL will produce a well-proportioned multiphase clock signal if the loop is closed [19].

3.2.2. Output Jitter

Both implementations have advantages and disadvantages, which essentially depend on the type of use that is given to them, so there is a need to understand the reasons that can lead to opting out of one of them, depending on the other. Thus, it should be noted, once again,

that one of the main points influencing the performance of the ADC is noise, which is associated with jitter. Next, an attempt will be made to understand the impact of jitter on the two architectures under analysis and, thus, choose the one that best responds to the intended problem.

3.2.2.1. SR Output Jitter

The output jitter of the SR MPCG can be divided into two parts that will be duly analyzed and studied: jitter transferred from the reference clock signal and jitter generated by the chain of flip-flops D. The logic inversion circuit only has the function of changing the logic value of the entry of the first FFD and will have no contribution to jitter. Regarding the jitter transferred from the reference clock signal, the SR MPCG does not show improvement when compared to the VCDL. It is important to note that any timing error in the reference clock signal will be transferred to the outputs of the FFD chain., The FFD chain generates two types of jitter: mismatch jitter and noise jitter. However, the FFD chain does not accumulate jitter in the transition from one flip-flop to the other since each FFD output acts only as an input to the next FFD. An FFD can be designed in many ways and one of the approaches is the master/slave class methodology. For a design aimed at optimal characteristics, the only latch that contributes to jitter is the slave one, as the master functions as an "enabler" [19].

If the rms noise and mismatch noise of an FFD (latch) unit are defined as $\sigma^2_{t,latch,noise}$ and $\sigma^2_{t,latch,mis}$, respectively, the average variance in the jitter value, for the set of clock signals with N phases generated, through the Shift Registers, will be calculated as follows:

$$\left(\sigma^2_{t,SR,noise}\right)_{medN} = \frac{1}{N} \cdot \sum_{n=1}^{N} n \cdot \sigma^2_{latch,noise} = \sigma^2_{latch,noise} \tag{3.1}$$

$$\left(\sigma^2_{t,SR,mis}\right)_{medN} = \frac{1}{N} \cdot \sum_{n=1}^{N} n \cdot \sigma^2_{latch,mis} = \sigma^2_{latch,mis} \tag{3.2}$$

3.2.2.2. DLL Output Jitter

The DLL MPCG output jitter can be distinguished into three main divisions, but the focus will be more on the first two: The jitter transferred by the reference clock signal to the DLL and the jitter generated by the VCDL. When the reference clock signal is transferred to the DLL outputs, the jitter has peak values of this transfer. The DLL, however, has some limitations as it not able to decrease the jitter of the reference clock signal. On the other hand, by choosing a low closed loop bandwidth for the DLL, peak jitter values can be reduced.. For an ideal DLL architecture, the closed loop jitter contribution is negligible and can therefore be

ignored. Given the previous information, the main problem and concern is essentially with the jitter resulting from the VCDL. In an MPCG DLL, the architecture used in its implementation, VCDL can generate two types of jitter: random noise jitter, in which the main responsible for this type of jitter is phase noise, and mismatch jitter, which, as the name implies, is caused by the mismatch of the delay units. For the first type of jitter, that is, the noise jitter produced by VCDL, the DLL circuit does not show significant improvements. Again, the VCDL noise jitter is lower for low values of the closed loop bandwidth, a situation that would be quite identical to a free-running VCDL execution. Jitter, as it is transmitted, from delay unit to delay unit, will accumulate. The DLL's closed loop can significantly improve deterministic mismatch jitter, contrary to the effect it has on phase noise jitter,. The fact that the reference clock signal is properly aligned on both the input and the output of the VCDL causes the deterministic time error to be equal to zero, in contrast to the middle of the VCDL where the mismatch jitter is maximum [19].

If the jitter noise variation of a delay unit is represented by $\sigma^2_{t,DU,noise}$ and, assuming the presence of uncorrelated white noise, the jitter noise variation at the output of the n^{th} delay unit will be n times greater. To quantify the jitter value of a set of N-phase clock signals, the average jitter variance of the N signals has a significant amount. The average jitter variance of noise generated by DLL can be calculated by:

$$\left(\sigma^2_{t,DU,noise}\right)_{medN} = \frac{1}{N} \cdot \sum_{n=1}^{N} n \cdot \sigma^2_{t,DU,noise} = \frac{N+1}{2} \cdot \sigma^2_{t,DU,noise} \tag{3.3}$$

By defining the mismatch jitter variance of each delay unit as $\sigma^2_{t,DU,mis}$, the jitter variation at the output of the n^{th} delay unit can be calculated by:

$$\sigma^2_{t,DU,mis} = \frac{n(N+n)}{2} \cdot \sigma^2_{t,DU,mis} \tag{3.4}$$

To obtain the average value, the following expression is considered:

$$\sigma^2_{t,DLL,mis} = \frac{N^2-1}{6N} \cdot \sigma^2_{t,DU,mis} \tag{3.5}$$

3.2.3. Jitter Comparison Between Topologies

3.2.3.1. Comparing Jitter Transferred from the Reference Clock

From what was analyzed in the previous point, it is possible to notice that both the DLL and the SR MPCGs do not present improvements in the jitter of the reference clock signal. Taking this into account, it is known that the SR MPCG needs a reference clock signal with a frequency *N* times greater than the DLL, where *N* represents the number of clock signals intended for the output. If both signals are generated by the same source, the one that generates the signal for the SR must operate with a frequency *N* times greater than that of the DLL. However, higher operating frequencies will influence another important factor, power consumption [19].

Although the SR MPCG works with a frequency *N* times higher, the jitter produced is the same, in both cases, considering that both have the same power and the same design quality. On the other hand, it is important to be aware of the limitations that exist when increasing the frequency, and in addition to the reference signal source, the power consumption of the clock buffer can also become an issue.

3.2.3.2. Comparing Jitter Generated Due to Thermal Noise

In most architectures designed to generate clock signals, jitter and power are, in fact, two very important parameters, as has been mentioned throughout the document. By manipulating the admittance, both noise and mismatch jitter can always be reduced, as long as they balance with an increase in power consumption. DLL is made up of *N* delay units, which contribute to jitter and power consumption. In turn, an SR has *N* branches contributing to the jitter and the same number of branches contributing to the power dissipation. The result of the comparison depends on the amount of delay units, that is, on the number of signals you want to obtain. In an MPCG DLL, the input source sets the working frequency and the VCDL sets the delay between the *N* output clock signals. Both the reference signal generator source and the delay units need to be adjusted so that the entire circuit runs at the same frequency. In contrast, the SR MPCG allows for greater flexibility. For different frequency values, only the source generating the reference signal needs to be adjusted, since the frequency and delay between the N output clock signals are defined by the clock period of the input source [19]

3.3. Discussion

In this chapter, a final balance will be made between the different implementations, listing the advantages and disadvantages of the various possibilities, making it necessary to choose not only the best architecture for the clock buffer, but also for the MPCG.

3.3.1. Clock Buffer

Table 3.1 shows the advantages and disadvantages of the different architectures presented in the state of the art.

Table 3.1- Advantages and disadvantages of the clock buffer architectures.

Clock Buffer	Advantages	Disadvantages
SRL	-Very simplified design.	- It is necessary to add a dummy buffer. - Low gain and inefficient power consumption. - High jitter value. -The D2S circuit operates in voltage mode.
SALPF	- Superior features to SRL. - Relatively higher gain.	-Limitations on power dissipation.
SRL + SALPF	- Superior features to SALPF. - Joins two types that architectures. - High power consumption, for a given jitter value. - The D2S circuit operates in current mode.	- Higher Power/Jitter trade-off value than SALPF.
SAL + RCMF	- Buffer with better Power/jitter trade-off - The lowest power dissipation for a given jitter value.	- It requires accurate transistor electrical models at high frequency.
LNLGA	- Low jitter and power dissipation.	- Higher noise instability - Low gain.
HGRA	- Higher gain and robustness against noise than LIGA.	- Higher jitter values and power dissipation than LGNA
QILA	- Higher gain and has a sharper clock signal peak than LNLGA. - Less noise and lower power dissipation than HGRA.	- The D2S circuit operates in voltage mode.

Based on what is described in the table the two best architectures for the buffer implementation are SAL+RCMF and QILA. However, the SAL+RCMF architecture, as it operates in current mode, has a significant advantage over QILA. This will be the circuit to implement and size as far as the clock buffer is concerned.

3.3.2. Multi-Phase Clock Generator

Based on what has already been mentioned and on the comparisons made between the two possibilities of MPCG, SR has a lower noise instability than the DLL counterpart, for a given power consumption.

The analysis made before shows that for mismatched jitter values DLL MPCG may have a small advantage in some high frequency cases.

Regarding the implementation of both circuits, the SR MPCG is easier, since it does not require so many components, namely a phase detector, a charge pump, a closed-loop filter. On the other hand, it can be more difficult to implement in applications that have a high number of outputs (N) and a high operating frequency (f) as requirements, as the SR works based on the following denotation: $2N*f$. On the other hand, it is also important to consider, in this type of approach, the loading of the input source to be more severe in the SR MPCG, since it requires adding $2N$ DFFs. However, this problem can be solved by manipulating the value of admittances, reducing them considerably, which would result in a decrease in the size of the DFFs. In this way, less jitter would be generated, compared to the delay units, saving energy and chip area [19].

To finalize the comparison between the two circuits, MPCG consisting of Shift Registers will be the solution to implement since it is a digital circuit that has a large range of operating frequencies. In addition to what has been mentioned, an SR can change its output frequency and has the flexibility to generate clock signals with different duty cycles [20].

3.4. Proposed Circuit

Based on the conclusions obtained in the state of the art, it is possible to visualize the model of the architecture to be implemented (Figure 3.7).

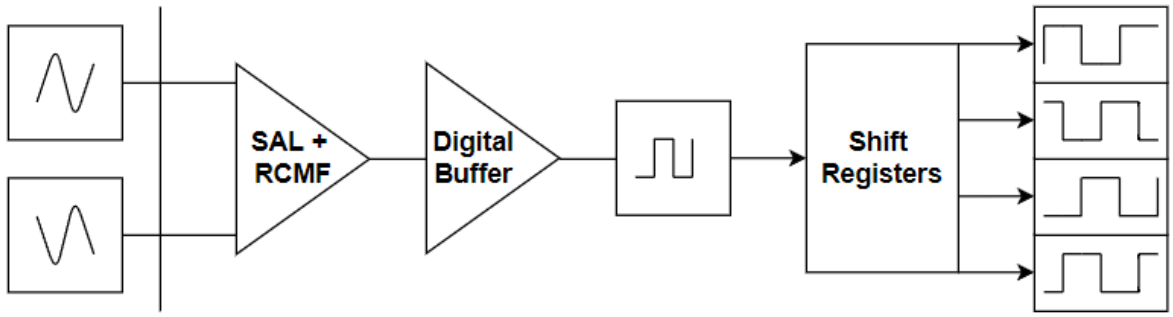


Figure 3.7- Final schematic of the proposed circuit.

4

4. Circuit Implementation

4.1. Clock Buffer

In the state of the art, it was concluded that the clock buffer circuit would be SAL+RCMF (Figure 3.3 d)), and in this chapter the sizing of the transistors of this circuit will be discussed. The circuit geometry is described through the length (L) and width (W) of each transistor and a careful selection of these values for each transistor is necessary, in order to make the circuit as efficient as possible.

The starting point of the clock buffer sizing methodology is to select for all transistors in the circuit their nominal polarization condition, where an input signal with a voltage close to $V_{DD}/2$ will be considered, which is the maximum recommended value. To understand bias conditions, it makes sense to define overdrive voltages for a given bias current.

$$(|v_{OD}| = |V_{GS}| - |V_{TH}|) \tag{4.1}$$

Since the current that feeds the differential pair is half of the Bias source current and the relationship between current mirrors and transistors are a concern of the first stage, it is possible to significantly reduce the independent variables with the proposed parameterization.

In the sizing methodology is not a critical factor the specific selection of the initial polarization, as it will have an optimization later. At the beginning of the DC operating point, for a complete analysis, the main criteria are as follows:

- Bias all transistors in the saturation region and strong inversion.

$$V_{DS} > V_{GS} - V_{TH} \tag{4.2}$$

$$V_{GS} > V_{TH} \tag{4.3}$$

$$I_D = I_D\,(V_{DS}) \cong I_{Dsat} \equiv \frac{\mu_n C_{ox}}{2}\frac{W}{L}(V_{GS} - V_{TH})^2 \tag{4.4}$$

- Ensure that the DC voltages at the critical nodes (N1, N2, N3) of the circuit are between 80% and 120% of $V_{DD}/2$, to provide wide operating margins.

- In the zone of differential pairs, consider an overdrive voltage, around 150 mV.

- Define the lengths of the differential pair at significantly low values, thus guaranteeing the circuit's performance speed and, simultaneously, avoiding being harmed by possible capacitances.

- As a last point, for the differential pair NMOS current sources, use relatively large channel lengths (above 2 μm), because these transistors have the most significant contribution to the jitter value [21].

4.1.1. Clock Buffer Sizing

To effectively start the transistor size, it is essential to know the operating current and voltage. A nominal current of 100 μA was injected by the Bias current source and the circuit is fed by a differential input voltage of $V_{DD}/2$ and in this case a V_{DD} of 1.2 V was considered.

The first pair of transistors to size will be the NMOS transistors that make up the differential pair, M11 and M12. Since we are dealing with a differential pair, the drain current of each transistor in the pair will be half of the nominal current. Admitting a V_{sat} of about 150 mV and assigning a value of 180 nm to the L of the transistor it is possible to calculate the value of W through the equations.

Next, the second pair of transistors to be dimensioned will be those that have the current mirror function, the M0 and MB transistors. These have the function of mirroring the nominal current for the first stage of the circuit and as such this will be their drain current. Apart from this aspect it was mentioned that these have a big impact on the jitter value, so the width of the transistors was considered equal to 2 μm. Regarding the value of V_{sat}, it is necessary to consider the DC voltage at the critical nodes (N1 and N2). Considering a fictitious node, V_x between the source of the differential pair and the drain of the current mirror MB, it is possible to obtain a condition for the V_{sat}. Assuming the following equalities:

$$\begin{gathered} N1, N2, N3 = [480\,;720]\ \text{mV} \\ V_{dsat11,12} = V_{GS11,12} - V_{TH} = 150\ \text{mV} \\ N1 = V_{DS11,12} + V_{DS,MB} \end{gathered} \tag{4.5}$$

Using the equation (4.5) that states whether the transistor of the differential pair is in the saturation region, the following conditions are met:

$$N1 - V_x \geq V_{GS11,12} - V_{TH} \tag{4.6}$$

$$V_x \leq N1 - V_{dsat11,12}$$

The conditions that determine the MB and M0 transistors saturation region are:

$$V_x \geq V_{GS,MB} - V_{TH} \qquad (4.7)$$
$$V_x \geq V_{dsat,MB,M0}$$

If the node N1 has the lowest voltage value, that is, 480 mV, the condition that allows knowing the minimum saturation voltage for the MB and M0 transistors are obtained:

$$V_{dsatM} < N1 - V_{dsat,M11,12} \qquad (4.8)$$
$$V_{dsat,MB,M0} < 330\ \text{mV}$$

To determine the W and L of transients M13,14 and M23.24, the conditions and equalities that were considered for this purpose are quite identical:

$$V_{DD} - N1 = V_{DS,M13,14} \qquad (4.9)$$
$$V_{DD} - N3 = V_{DS,M23,24}$$

$$V_{DD} - N1 \geq V_{GS,M13,14} - V_{TH} \qquad (4.10)$$
$$V_{DD} - N3 \geq V_{GS,M23,24} - V_{TH}$$

Now if the node N1 and N3 has the highest voltage value, that is, 720 mV, the condition that allows knowing the minimum saturation voltage for the M13,14 and M24,24 transistors are met:

$$V_{DD} - N1 \geq V_{dsat,M13,14} \qquad (4.11)$$
$$V_{DD} - N3 \geq V_{dsat,M23,24}$$
$$V_{dsat,M13,14}\ , V_{dsat,M13,14} \leq 480\ \text{mV}$$

Finally, the last pair of transistors that need to be dimensioned is the M21.22. Regarding the condition that allows us to know the saturation voltage, the equation is quite elementary:

$$N3 = V_{DS,M21,22} \qquad (4.13)$$

$$N3 \geq V_{GS,M21,22} - V_{TH} \qquad (4.14)$$
$$N3 \geq V_{dsat,M21,22}$$
$$V_{dsat,M21,22} \leq 480\ \text{mV}$$

Considering the previous conditions and equations that allowed sizing the transistors, a table represents the values of W and L assigned to each pair of transistors (Table 4.1). Although the buffer has high frequency applications, which would lead to the selection of a minimum L for the transistor dimensions, low values of L cause high phase noise. Therefore, it is preferable for transistors to be more susceptible to capacitances but with low Phase Noise than vice versa. So, the recommended minimum value is about 1.5 times the minimum L, so the L selected for the transistors was 180 nm. This sizing criterion was maintained for the remaining circuits used in the IC implementation.

Table 4.1- Dimensions of the SAL+RCMF circuit transistors.

1	Current	V_{dsat} (mV)	L (µm)	W (µm)	nf
M0	I_B	250	2	13	2
MB	I_B	250	2	13	2
$M_{11,12}$	$\frac{I_B}{2}$	150	0.18	3	1
$M_{13,14}$	$\frac{I_B}{2}$	300	0.18	2	1
$M_{21,22}$	$\frac{I_B}{2}$	250	0.5	1.6	1
$M_{23,24}$	$\frac{I_B}{2}$	300	0.18	2	1

Once the sizing of the clock buffer is done, a DC analysis will be carried out because it is necessary to infer whether the transistors are in the saturation zone (Appendix A:)

Analyzing the voltages values it is possible to verify that all transistors are in the strong inversion saturation region, except for the MB.

Since, through the initial conditions, it was not possible to have all the transistors in the saturation region (Table A.1), it is necessary to understand which metrics to manipulate and study the impact that increasing or decreasing the size of the transistors would have on the buffer performance.

4.1.2. Scaling W and L

To better understand the impact that the values of W and L have on the voltage of nodes N1, N2 and N3, on the current and on the power consumed by the circuit, a study was carried out. For the elaboration of the study, the values of L of the MB transistor and the W of all transistors were varied during a considerable interval, and then the consequences that this variation had on the metrics under study were verified.

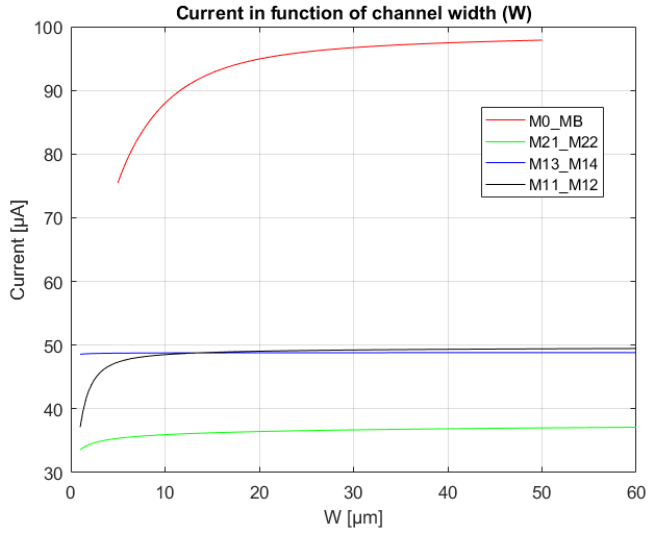


Figure 4.1- Current behavior of the SAL+RCMF circuit transistors as a function of their channel width.

Based on the analysis of the Figure 4.1, it should be noted that to obtain the current value close to the nominal value circulating in the drain of the transistor MB, it is necessary to increase the value of W. On the other hand, it also makes sense to increase the value of L to minimize the jitter value, as this is one of the transistors that contributes the most to this variable. However, it is necessary to pay attention to what this increase can influence the other metrics of the circuit.

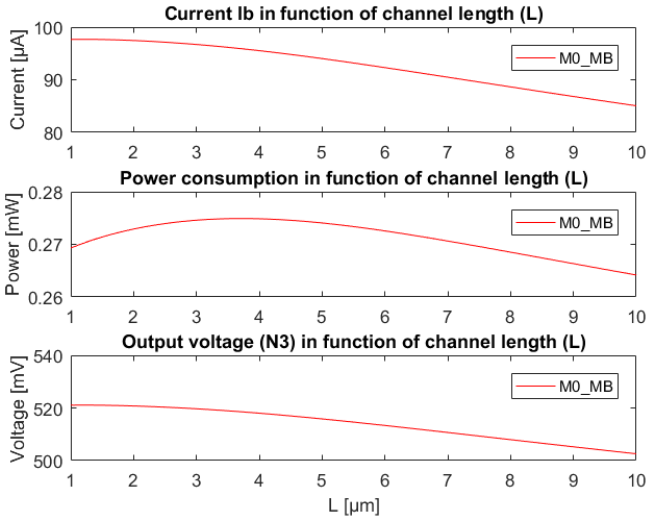


Figure 4.2- Current/Power and Output voltage behavior of the SAL+RCMF circuit transistors M0 and MB as a function of their channel length.

By looking at the Figure 4.2 it is possible to see that increasing the L of the MB and M0 transistors decrease the consumed power of the circuit, but on the other hand, the current also decreases significantly. It is therefore important to find a value of L that best satisfies these two metrics. Regarding the output voltage, the voltage decreases with increasing L, but this decrease will not be significant for the circuit operation as it is within the expected ranges

The next analysis to consider is the influence that increasing transistor W has on the output voltage (node N3) and what values will be sensible to consider for this parameter so that the voltage is within the range of 80 % to 120 % of $V_{DD}/2$.

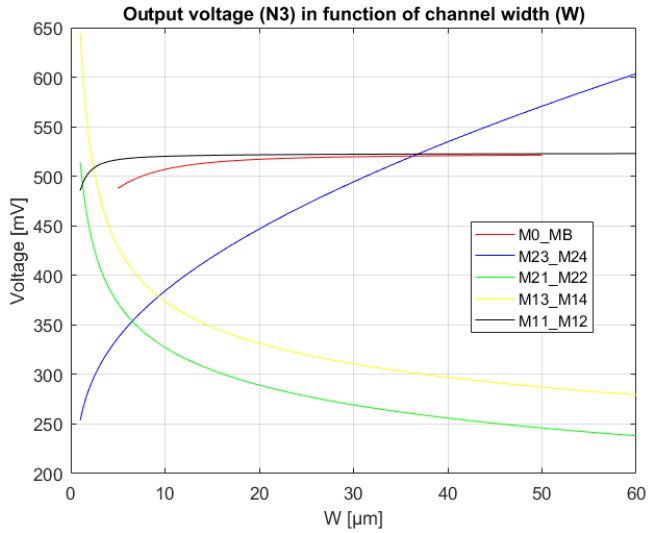
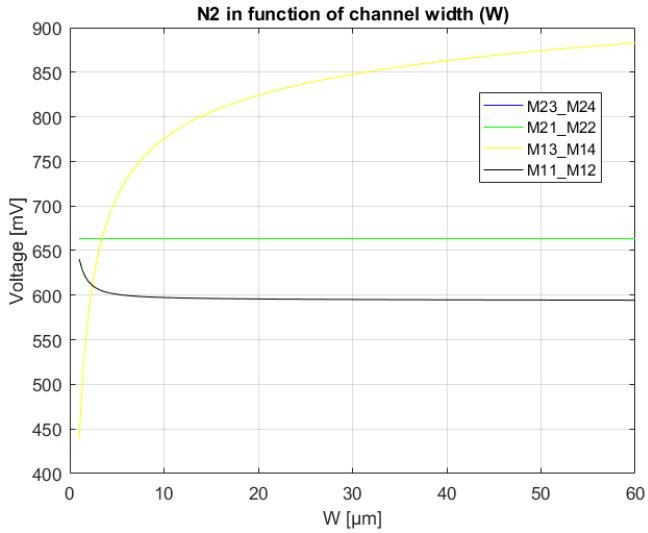


Figure 4.3- Output voltage behavior of the SAL+RCMF circuit transistors as a function of their channel width.

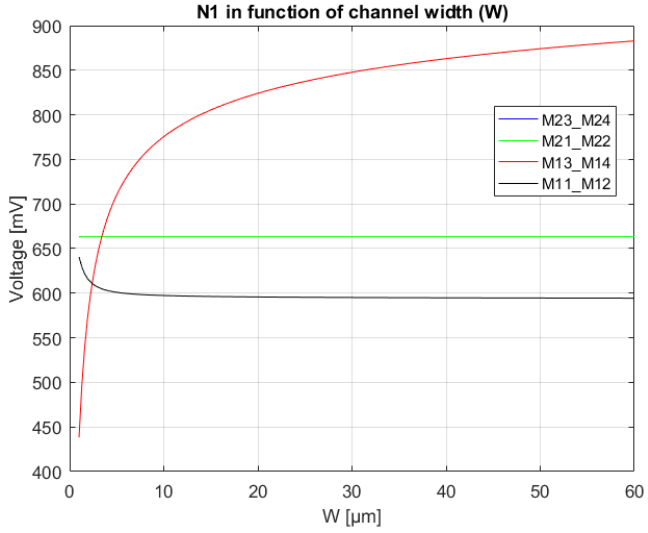
The Figure 4.3 tells us that for pair M23,24 the increase in W causes an increase in the output voltage and, on the contrary, it tells us that for pairs M21,22 and M13,14 increasing this parameter will reduce the output voltage. The idea would be to increase the value of W for the pair of transistors M23,24 and regarding the other two pairs since they are much more limited in this increase, and it is best not to touch them. For the other pairs that were not mentioned, the impact of the increase in W is not significant since the voltage remains almost stable during the interval. However, it will be important to understand what this increase will translate into the other metrics.

The increase of W in pair M23,24 and M21,22 will have no influence on the value of voltages N1 and N2 (Figure 4.4) since this value is constant for the considered interval. Regarding the consumed power, the increase in W in the pair M23,24 will cause a considerable increase in power consumption (Figure 4.5 a)), so this factor must be considered. The W value

of this pair of transistors will also influence the current of the second stage of the amplifier, both values are directly proportional, one increase and as such the other also increases.



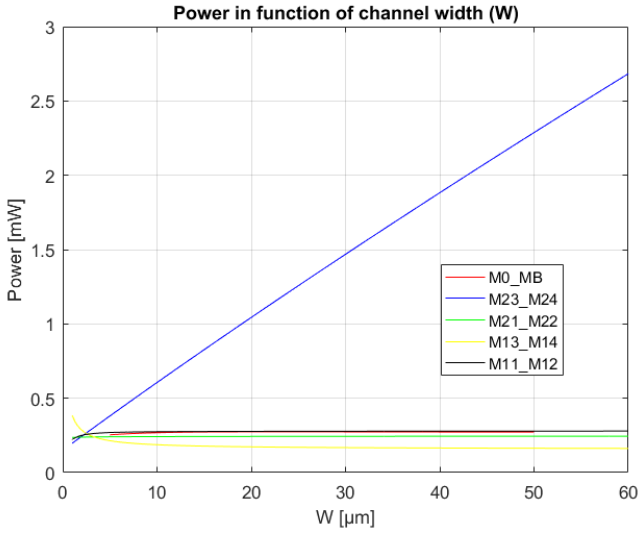
(a)



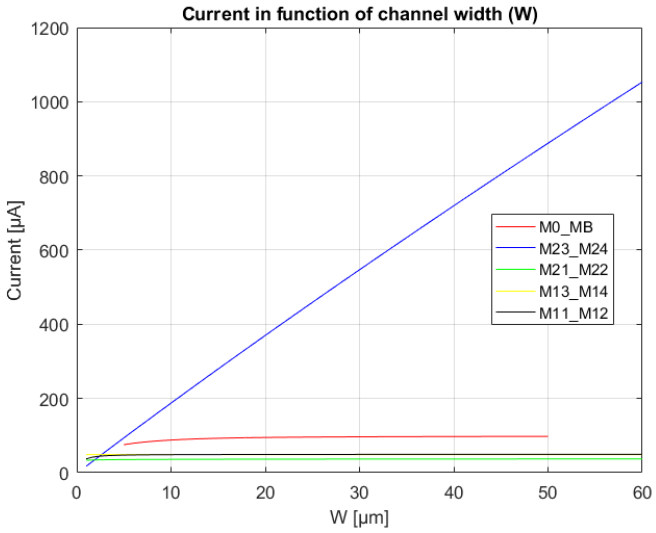
(b)

Figure 4.4- Node voltage behavior of the SAL+RCMF circuit transistors a function of their channel width a) N2L, b) N1.

When analyzing the graph of Figure 4.5, the increase in the value of W of the pair M13,14 will have a great influence on the voltage of nodes N1 and N2. Based on this information and on the information in the graphs of the current Ib and the output voltage, there is a range of values for W where it is possible to satisfy all the desired characteristics. The power drawn from the circuit, such as the drain current of this pair of transistors, is not greatly influenced by the increase in the value of W



(a)



(b)

Figure 4.5- a) Power behavior of the SAL+RCMF circuit transistors as a function of their channel width. b) Current behavior of the SAL+RCMF circuit transistors as a function of their channel width.

Using the information provided by the graphics, new L and W parameters were assigned to the Buffer transistors to place all transistors in the saturation region and to optimize the operation point of the circuit (Table 4.2).

Table 4.2- New dimensions of the SAL+RCMF circuit transistors

Transistor	L (µm)	W (µm)	nf
M0	3	40	5
MB	3	40	5
$M_{11,12}$	0.18	10	4
$M_{13,14}$	0.18	3	1
$M_{21,22}$	0.5	2	1
$M_{23,24}$	0.18	5.5	2

4.1.3. DC Operating Point

Based on the data provided by the Table 4.3, it is verified that all transistors are and in the saturation region in moderate/strong inversion since they fulfill all the conditions for this purpose.

Table 4.3- DC Operating Point – New voltages of the SAL+RCMF circuit transistors.

Transistor	I_d (µA)	V_{GS} (mV)	V_{DS} (mV)	V_{TH} (mV)	V_{dsat} (mV)
M0	100	359.9	359.9	227.5	152.1
MB	97.86	359.9	227.4	227.5	152.1
$M_{11,12}$	48.93	372.6	454	359.7	96.08
$M_{13,14}$	48.97	537.5	537.5	287.5	242.3
$M_{21,22}$	94.34	579.9	579.9	269.4	277.3
$M_{23,24}$	94.34	537.5	620.1	283.7	246

4.2. Multi-Phase Clock Generator

The next chapter will address the architecture responsible for transforming the single clock signal into a signal with 8 different phases. The architecture responsible for this phenomenon will be the Shift Registers, these can have different designs depending on the characteristics and needs of the circuit. Before implementing Shift Registers, it is important to understand in more detail the different architectures available and which one best fits the intended characteristics of this work.

Shift Registers are constituted of chains of D flip-flops. However, as mentioned, there are different CMOS circuits for the implementation of flip-flops and each one of them has its advantages and disadvantages for the basic circuit operation.

4.2.1. Transmission Gates

Before approaching each topology, it makes sense to define and explain how the transmission gates work, since both topologies have this component in their constitution.

Instead of just talking about Transmission gates (Figure 4.6), it is important to mention the analog switch, analog switch is a solid-state semiconductor switch that works like a relay and has the function of controlling the transmission path of analog signals. Switches have both closed and open operations and are generally controlled by digital logic gates. There are several switch designs available with different configurations according to your usage.

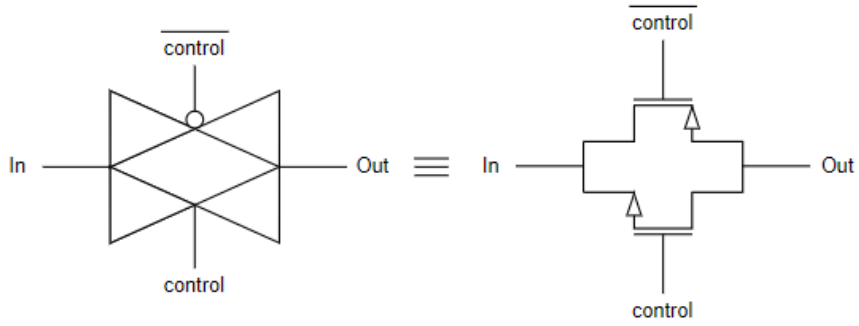


Figure 4.6- CMOS Transmission gate and symbol representation presented in [22].

Mechanical relays can switch and route digital and analog signals that can be triggered either by voltage or current. Although relays manage to have these characteristics, their operating process is significantly slow and expensive. One option to minimize the cost and speed up the process is using devices that use metallic oxide semiconductors, such as solid-state electronic switches. These devices have fast-acting digital analog ports that allow signal currents to shift from their input to their output. This type of technology, MOS (metal-oxide-semiconductor), uses two types of transistors, NMOS and PMOS. These devices used individually or together can control the operation of logic switches.

For cases where transistors are implemented in the same gate circuit, they have two working modes: they allow to pass (closed condition) or block (open condition) an analog or digital signal. This process is dependent on the digital logic level.[23], [24].

An example of a bilateral switch that uses the type of technology mentioned above is the Transmission Gate. To create a Transmission Gate, only two CMOS transistors connected in parallel are required, one PMOS and one NMOS. The circuit needs control voltages, which as the name implies will control the output value based on the input value. For a transmission gate to be properly implemented, there is a need to use an inverter. The inverter will be responsible for supplying the complementary control voltage to the gate of the circuit transistors. Transmission gates, despite having a similar function to a conventional logic gate

have some significant differences. Unlike a basic logic gate, a transmission gate has a symmetrical feature, or in other words is bilateral, which makes its input and output interchangeable. The operating mode of this device is quite simple to understand when the control voltage (Control) that feeds the gates of the transistors is low, both transistors are cut-off, i.e., the switch is open. When the value of the control voltage is high, the transistors are biased into conduction and the switch that was previously open is now closed. [22].

Thus, the transmission gate works as a "closed" switch when Control =1. When Control =0 the gate works as an "open" switch when $V_C = 0$

Table 4.4 shows the truth table of the transmission gate operation.

Table 4.4- Truth Table of the Transmission Gate presented in [22].

Control	Input	Output
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1
Out=In.Control		

It is possible to see from the truth table above, that the output (Out) of the transmission gate depends only on the logic level of the control (Control) and on the logic level of the input (In). Therefore, both In and Control defines the logic level value of Out and the respective Boolean expression for a transmission gate is formed: Out=In.Control

As mentioned earlier there is an easier way to implement a CMOS switch with characteristics like a transmission gate which is through a single transistor either PMOS or NMOS. However, it is more advantageous to combine the two transistors in parallel. To better understand this statement, it is necessary to consider that the transistor channels are resistive, so it is more profitable to connect the ON-resistors in parallel. As the ON -resistance of a FETs is a function of the voltage between the gate and the source V_{GS}, there is a tradeoff between the conduction of the transistors. When the NMOS gate is turned on this transistor becomes less conductive which consequently will make the PMOS take over and become more conductive. Thus, the combined value of the two ON-resistors remains almost unchanged and constant. If the switch was implemented with only one transistor, it would not have this feature [22].

4.2.2. Inverter

Another element that, in addition to transmission gates, is also part of both topologies is the CMOS inverter (Figure 4.7). The inverter is composed of two MOSFETs. The top is a PMOS type device while the bottom is an NMOS type. Both gates are connected to the input line and the output line connects to the drains of both FETs [25].

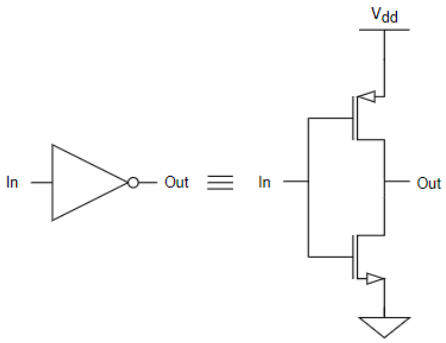


Figure 4.7- CMOS Inverter and symbol representation.

Its operation is promptly understood with the help of the simple switch model MOS of the MOS transistor. In Appendix B: the operation of the CMOS inverter is properly explained.

One aspect to bear in mind about the CMOS inverter is the power consumption of the circuit [26]. In an inverted CMOS it is mainly due to its dynamic power consumption, which is described by:

$$P_{dyn} = {V_{DD}}^2 \cdot C_L \cdot f_{in} \tag{4.15}$$

Where C_L represents the capacitance at the output of the circuit and f_{in} is the frequency with which the circuit is supplied. Therefore, the higher the frequency, the greater the power consumed by the inverter.

4.2.3. D-type Flip-Flops

Combinational logic circuits, like the CMOS inverter and the transmission gate are characterized by the output of the logic block only depending on the current input values, assuming that enough time for the logic gates to settle down has already elapsed. However, there is a need for all systems in another property, the storage of state information. These systems need, requires the use of another class of circuits called sequential logic circuits. In these circuits, the output, unlike in combinational logic circuits, depends not only on the

current values of the inputs, but also on the previous input values. In other words, the circuit must have memory to be able to store the previous input value.

An example of this type of circuit, a register, can be a flip-flop where the outputs work depending on the input information and the current state. This means that next state is reached based on current input value and current state. However, to carry out this process, it will be necessary to add a clock signal that signals the register that it needs to process the input state. When the clock signal is on the rising edge, the next state bits are copied to the register output and a new cycle begins. This process is not immediate as there is some propagation delay. Until there is a new rising edge in the clock signal the register ignores all changes in the input signals [24]. Registers can have two modes of operation, either positive edge-triggered or negative edge-triggered (where the date is copied on either the positive or negative edge of the clock sign).

4.2.3.1. Types of Memory

The registers can have two types of memories that can be dynamic or static. Static memories are able to preserve the current state while the circuit is powered and can be built either through the use of positive feedback or through regeneration. The circuit topology that allows this type of memory is implemented through a combinational circuit where intentional connections between the output and the input of the circuit are designed.

Static and Dynamic Memories

Dynamic memories can only store a state for a short period of time, no more than a few milliseconds. These compared to static memory present superior characteristics when the registry does not need to be updated for long periods of time. Dynamic memories have simpler designs, which translates into less power dissipation and significantly higher performance. This type of memory is mostly used in data path circuits that are periodically clocked and require high levels of performance [24]. Two flip-flop implementations will be analyzed in which one presents the information statically and the other dynamically, they are respectively, Single-Threshold Transmission Gate flip-flop [27]–[29] and Dynamic Transmission Gate flip-flop [26], [30], [31].

4.2.3.2. Different Topologies

The inverter and transmission gate CMOS circuits are part of the two flip-flop D architectures that will be discussed.

STTG Flip-Flop

The first flip-flop D topology to be analyzed is the Single-Threshold Transmission Gate flip-flop (Figure 4.8) due to its greater degree of complexity.

The D flip-flop was implemented using eight previously studied CMOS architectures, four inverters and four transmission gates. This architecture has a master/slave mode of operation and it is through the clock signal that it is possible to trigger the different stages of the flip-flop. The clock signal drives the first stage of the flip-flop (master), however the second stage of the flip-flop (slave) is driven by the inverted clock signal. In other words, the master stage has a higher sensitivity to the positive level, while the slave stage has a higher sensitivity to the negative level. The master-slave flip-flop design is simple to use and allows for good noise margins. This type of implementation works for high-speed applications as it offers more clock leeway and, with the right design, can handle clock skew.

The operation of this circuit can be explained as follows: The input D of the first stage is sampled, when a clock signal with a voltage corresponding to a logic value of "1" is injected into the transmission gates (example: VDD). On the other hand, the slave stage is storing the previous value. When there is a transition in the clock signal, and its voltage drops drastically, i.e., it goes from high to low (0 V), the master stage stops following the input and the D value when this transition occurs is stored. While this operation is happening, the slave latch has no influence on the process, since its only function is to pass the value stored by the master stage to its output, i.e., the Q output of the flip-flop. In this architecture the flip-flop output is not affected by the input value, because the input D is disconnected from the master stage. When there is a new change in the clock signal from low to high, the slave latch stores the output value of the master latch, which allows the master latch to sample a new input signal.[27].

This model is not only extensively used in sequential systems but has lower power consumption and is one of the fastest flip-flop designs as well.

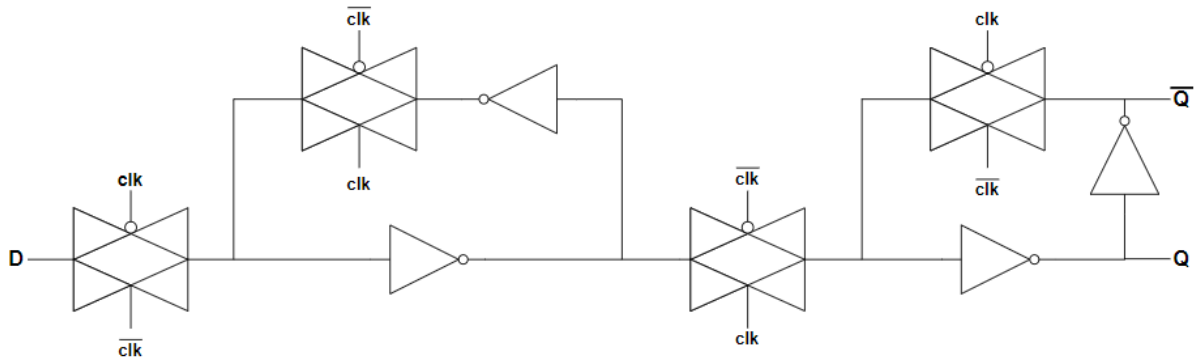


Figure 4.8- Single-threshold transmission gate flip-flop presented in [29].

DTG Flip-Flop

The second flip-flop D topology to be analyzed is the Dynamic Transmission Gate flip-flop (Figure 4.9) due to its smaller degree of complexity.

A fully dynamic positive edge driven register is also based on the master-slave concept. When the clock signal has the logic value "0", the input data is sampled in storage node X. This node has an equivalent capacitance consisting of the capacitance of the inverter at the input of the flip-flop, the capacitance junction of the transmission gate, and the transmission gate's overlapping gate capacitance. At this instant the node Y is in a high-impedance state, while the slave state is in standby mode. The transmission gate that is directly connected to the circuit's output, when the clock edge rises, turns on, the value that just before the edge of the rising clock signal was sampled at the node X propagates to the output Q. Node Y now stores the inverted logic value of node X. It is noticeable that node X manages to show stability during the high clock phase since the first transmission gate is turned off [32].

An edge-triggered register can be implemented rather efficiently as it only requires 8 transistors (if the flip flop does not have the complement output). The fact that this flip-flop is implemented with a reduced number of transistors makes it attractive for high-performance, low-power systems.

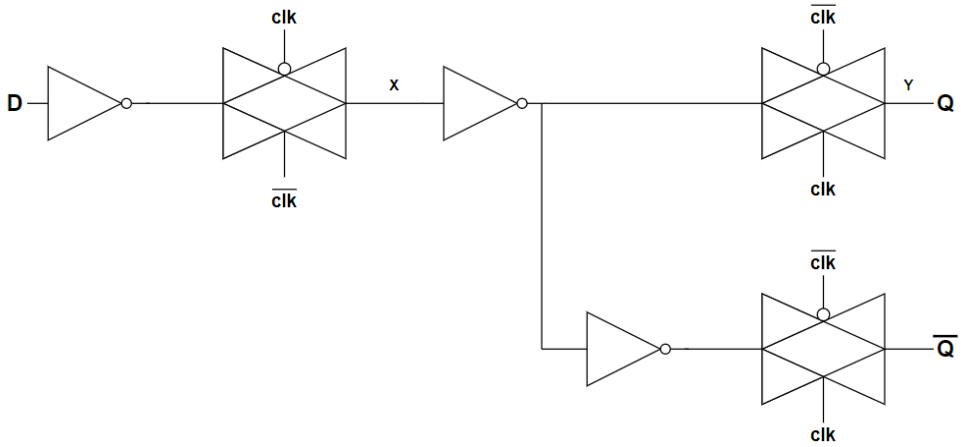


Figure 4.9- Dynamic transmission gate flip-flop presented in [26].

4.2.3.3. Flip-Flop Sizing

The next chapter aims to size and test the circuits mentioned in the previous chapter, the CMOS inverter, the transmission gate, and both are part of the design of two flip-flop topologies.

4.2.3.3.1. Inverter

The first circuit that we are going to investigate is the CMOS inverter. Its design consists of a PMOS transistor and a NMOS transistor connected in cascade. This means that the PMOS drain connects to the NMOS drain and the gates of both transistors are also linked together. The inverter threshold voltage V_{th} is one of the most important parameters that characterize the steady-state behaviour of the CMOS inverter circuit. A proper design of W/L ratio is very important to find the current driving capability of gate in both directions [33].

As seen in the circuit analysis in the previous chapter, when the inverter is in the operation zone and both transistors are in the saturation region, there is a point that proves the following equality $V_{in} = V_{out}$. In an ideal case $V_{in} = V_{out} = V_{th}$. Through this equality, it can be deduced that the drain current of both transistors is equal, so it is possible to obtain the following expression [34]:

$$\frac{K_n}{2}(V_{th} - V_{Tn})^2 = \frac{K_p}{2}(V_{DD} - V_{th} - |V_{Tp}|)^2 \tag{4.16}$$

By isolating the terms K_n and K_p from the expression (4.16) it is possible to create a new equality, but this time in function of these terms:

$$\sqrt{\frac{K_n}{K_p}} = \frac{V_{DD} - V_{th} - |V_{Tp}|}{V_{th} - V_{Tn}} \tag{4.17}$$

To get a symmetrical voltage transfer curve, V_{th} is set to $V_{DD}/2$:

$$\sqrt{\frac{K_n}{K_p}} = \frac{\frac{1}{2}V_{DD} - |V_{Tp}|}{\frac{1}{2}V_{DD} - V_{Tn}} \tag{4.18}$$

If in a process this condition is met $|V_{Tp}| = V_{Tn}$, the device aspect ratios for a symmetrical inverter are related by:

$$\frac{K_n}{K_p} = \frac{\mu_p (W/L)_p}{\mu_n (W/L)_n} \tag{4.19}$$

If the gate oxide thickness and hence, the gate oxide capacitance have the same value for both NMOS and PMOS transistors [33]. The ratio condition for the ideal symmetric invert is:

$$\frac{(W/L)_p}{(W/L)_n} = \frac{\mu_n}{\mu_p} \tag{4.20}$$

Since $\mu_n = 580\ CM^2V - S$ and $\mu_p = 230\ CM^2V - S$:

$$(W/L)_p \approx 2.5\ (W/L)_n \tag{4.21}$$

By setting the L value in both transistors to 180 nm by the features was already mentioned in the previous chapter and considering the above equality (equation 4.20), the dimension for the CMOS inverter is finally reached.

Table 4.5- Dimensions of CMOS Inverter transistors.

Transistor	L (μm)	W (μm)	nf
PMOS	0.180	18	2
NMOS	0.180	6	1

The circuit has a large W/L ratio in both NMOS and PMOS because by having a large ratio the current I_d is relatively large enough to increase the circuit drive strength. Therefore, the transconductance also increases and so does the capacitance of both transistors [35].

4.2.3.3.2. Transmission Gate

Regarding the transmission gate sizing, the ratio between the PMOS and NMOS transistors was the same as that used for the CMOS inverter. Relating to the W/L ratio of each transistor, this value was slightly increased to decrease the channel resistance of both transistors, increasing the switch drive strength. This phenomenon allows the current to drive the next state, which is the CMOS inverter, and so on with greater speed [35].

Table 4.6- Dimensions of CMOS transmission gate transistors

Transistor	L (µm)	W (µm)	nf
PMOS	0.180	30	3
NMOS	0.180	10	1

After sizing the circuits that integrate the two D flip-flop architectures, it is possible to obtain the final circuits for both architectures and later test them.

4.2.4. Shift Registers

Flip-flop architectures can only store a single bit of data. However, there are other architectures that allow storing multiple data bits. By connecting more than one set of D flip-flops in series this phenomenon becomes possible. This device that allows storing such information is called Register. The information stored within these Registers can be transferred with the help of Shift Registers [36].

A Shift Register basically consists of a group of flip-flops used to store multiple bits of data, one flip-flop for each data bit. An n-bit shift register can be formed by connecting n flip-flops together in a serial type of daisy-chain arrangement where the bits stored can be made to move from the output of one data latch to the input of the next one or just in and out of the data latch by applying a clock signal [37].

Generally, shift registers operate in one of four different modes, these are:

Serial-in to Parallel-out (SIPO) - The register is loaded serially with bits, one at a time, and the stored data is available at the output of the register in parallel form.

Serial-in to Serial-out (SISO) - The register is loaded with serial data that is shifted serially out of the register, one bit at a time in either a left or right direction under clock control.

Parallel-in to Serial-out (PISO) - Parallel data is loaded into the register simultaneously, i.e. in parallel and is shifted out of the register serially one bit at a time under clock control.

Parallel-in to Parallel-out (PIPO) - The parallel data is loaded in parallel to the register and the result is transferred to the register outputs through the same clock pulse.

4.2.4.1. Ring-Counter

A predefined number of clock cycles is required for the serial data to move through the register, thus allowing the SISO to function as a time delay circuit when processing the original input data signal.

It is important to enhance that for the intended implementation there will be no additional source that directly feeds the register input, the only signal provided to the register is the clock signal. Considering this statement, a new approach is devised that allows the implementation of the logger without having to add an extra source. An example of this approach is the Ring-Counter. A Ring-Counter is a closed loop circuit where the output of the register is directly connected to the input of the register, allowing data to flow through the flip-flops while they are fed with clock pulses. [38]. As mentioned, as long as clock pulses are applied, there will be data pattern circulating inside the shift register. The Ring -Counter can have two types of implementations, one of them is implemented as mentioned above and the other the only difference is that the output signal of the flip-flop chain is inverted before being connected to the input of the chain. This second implementation is called Johnson Ring-Counter (Figure 4.10) and will be used in this work. This type of approach has significant advantages over other standard Ring Counters. The Johnson Ring-Counter manages to get the same number of output states, but only needs half the number of flip-flops in its implementation. The circuit consists of *N* DFFs that are connected and instead of producing a sequence of n states like a standard ring counter, it produces *2N* different states ("mod-*2N* counter") [39].

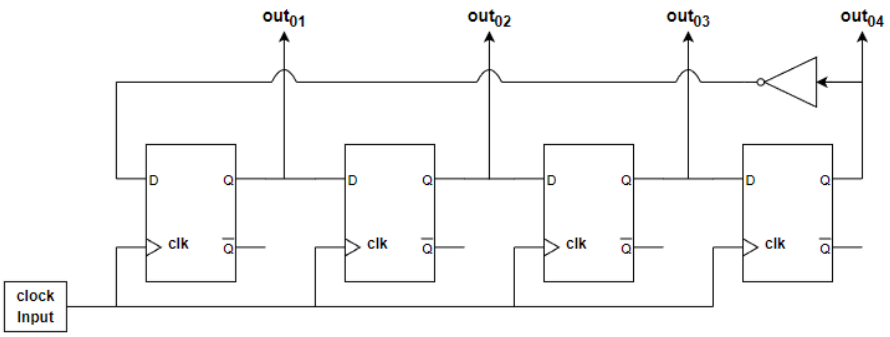


Figure 4.10- Johnson Ring Counter.

.

4.3. Digital Buffer

This chapter only appears at the end and not right after the clock buffer sizing, because this buffer will serve as a transition point between the clock buffer and the shift registers, despite being an element of the Rx core, so it will be necessary to take these factors into account in its sizing. It is important to remember that the output waveform of the clock buffer does not have a square characteristic as intended. In these two articles, an architecture for the digital buffer is suggested, which consists of a chain of inverters, specifically two inverters in cascade.

Although the two cascade inverters achieve the intended operation, as the frequency of the circuit's input signal (LVDS) increases, this phenomenon gains some weaknesses. In both works carried out in the previous articles, the frequency range used was less than 500 MHz. For input frequency values higher than 500 MHz, which is the focus of this work, the clock signal at the buffer output has a significant increase in the slope and gets a rounder form. To counteract this increase in signal slope, another architecture is considered [40].

The design of a buffer consisting of a chain of CMOS inverters with exponentially increasing gate widths is called "exponential horn [41]" (Figure 4.11) is meant to drive a large capacitive load with minimal propagation delay. This circuit will have the functionality to convert the high frequency sinusoidal output signal from the clock buffer to a square signal of the same frequency but with an amplitude of 1.2 V. However, the total bandwidth of a cascade of two or more amplifier stages is narrower than that of a constituent stage. An exponential horn, having a very high gain, therefore has a rather limited bandwidth and gives rise to increased jitter [40]. Thus, the focus will be redoubled when it comes to the sizing of transistors to not only counteract these previous points and the problem regarding high-speed buffering as well.

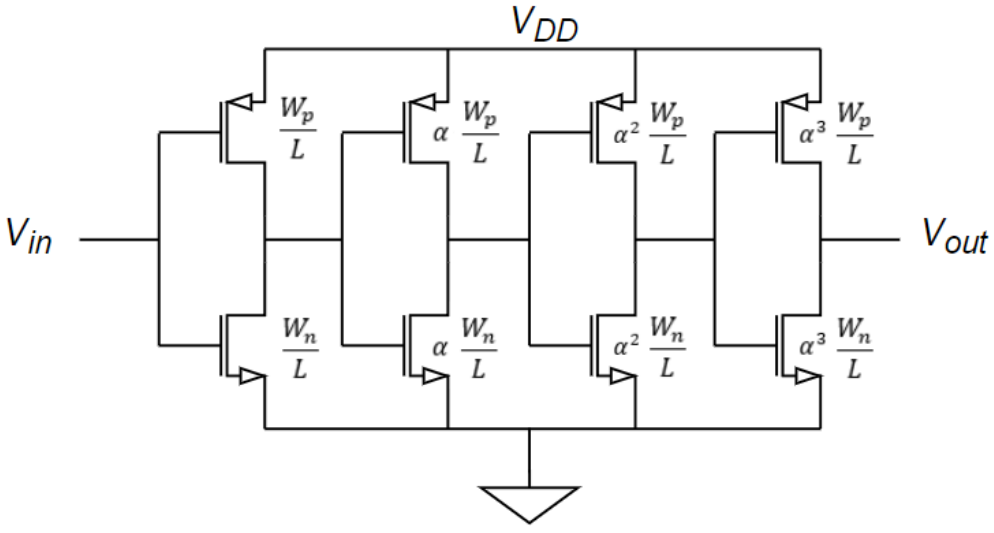


Figure 4.11- Chain of inverters with exponentially increasing size based in [40].

4.3.1. Digital Buffer Sizing

Regarding the sizing of inverters of the buffer, the ratio between the PMOS and NMOS transistors was the same as that used for the CMOS inverter in the previous chapter. Regarding W/L ratio of each transistor, will depend on the position that the transistor has in the chain.

The starting point for the chain sizing was the last inverter in the chain. For the drive strength to be ensured in the conduction of the current to the next state, from the amplifier to the shift register, it was defined that the dimensions of the last transistor in the chain were equal to those of the transmission gate. The alpha scaling factor value was determined based on the last inverter in the chain. Since $W_n = 10$ µm for Inverter 4 and the Inverter 1 of the chain makes the direct connection with the clock buffer then it is essential to ensure that the NMOS transient of this inverter respects the following condition $W_n \geq 1$ µm. Parameter α has been chose to be equal to 2.15. Based on the conditions mentioned above, the following table is obtained:

Table 4.7- Dimensions of the Digital Buffer transistors.

Inverter	$\frac{W_n}{L}$	L (µm)	W_n (µm)	nf	W_p (µm)	nf
1	$\frac{W_n}{L}$	0.180	1.00	1	3.00	1
2	$\alpha\frac{W_n}{L}$	0.180	3.00	2	9.00	2
3	$\alpha^2\frac{W_n}{L}$	0.180	5.00	2	15.00	3
4	$\alpha^3\frac{W_n}{L}$	0.180	10.00	4	30.00	5

4.4. Clock Circuit

Before proceeding with the electrical simulations of the Clock Circuit and its elements, it is essential to make final adjustments to the different architectures of the CC to optimize their performance, and that will subsequently improve the quality of the results.

4.4.1. Rx

The clock buffer sizing was done for frequencies in the order of 100 MHz and as discussed throughout the document, the working frequency is around 1 GHz. With that in mind, it will be necessary to make some adjustments to the clock buffer sizing to meet the

intended requirements. The main factor to consider with increasing the frequency at the input of the buffer will be the drastic gain reduction. The best way to counteract this decrease in gain is to increase the dimensions of the transistors in the output stage of the D2S circuit. Thus, increasing its current and transconductance will represent a significant increase in amplifier gain. As mentioned in [15] the increase in frequency causes the number of harmonics that contribute to the jitter to decrease, which causes the jitter value at the clock buffer output to decrease.

Table 4.8shows the new W and L values of the clock buffer transistors.

Table 4.8- Final dimensions of the SAL+RCMF circuit transistors.

Transistor	L (µm)	W (µm)	nf
M0	5	40	4
MB	5	40	4
$M_{11,12}$	0.18	15	4
$M_{13,14}$	0.18	3	1
$M_{21,22}$	0.5	27	4
$M_{23,24}$	0.18	60	15

The circuit formed by the Clock Buffer whose output is connected to the Digital buffer is called Rx. The Figure 4.12 shows the Rx architecture.

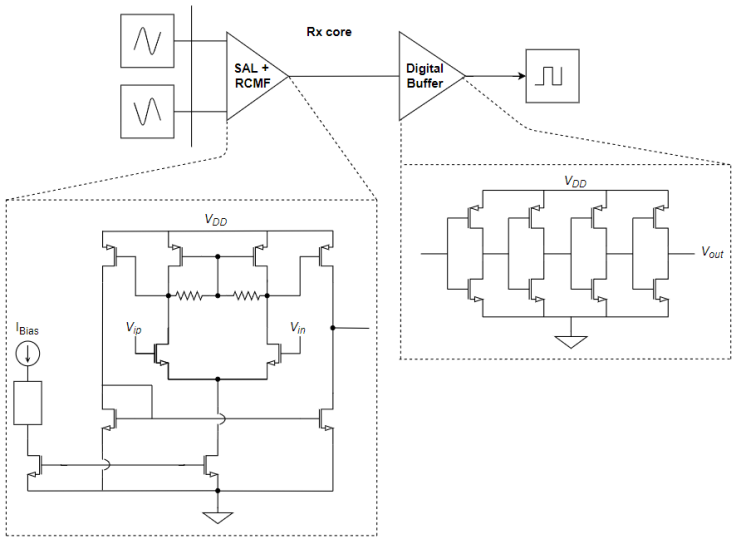


Figure 4.12- Rx core circuit.

4.4.2. D-type Flip-flop

Previously, two different architectures were analyzed for the implementation of the D flip-flop, the STTG flip-flop and the DTG flip-flop depending on the type of memory they use. To decide which one made the most sense to implement, a simulation of these circuits was performed to obtain the power consumed and the delay of the signal at the output of the flip-flop. Both circuits were supplied with two ideal sources one for the D input of the flip-flop and another for the clock signal that powers the flip-flop. Input D was supplied with a square pulse of 1.2 V with a period of 8 ns, the clock signal supplying the flip-flop was also a square pulse of 1.2 V but with a period of 4ns, i.e., half the period of the input D.

Table 4.9 shows the results of the simulation of the two architectures.

Table 4.9- Power and Delay output result of the DTG and STTG flip-flop.

Flip-flop D	Delay (ps)	Power (mW)
STTG	519.8	0.79
DTG	339.6	0.72

Based on the values in Table 4.9 for both the consumed power and the delay obtained in the simulation, it was inferred that the DTG flip-flop would be the best implementation.

4.4.3. Shift Register

Regarding the architecture of the Shift Registers, as mentioned in the state of the art, it was necessary to add an inverter to the output of the flip-flops. The signal at the output of the shift registers has a very noisy zero value. Passing this signal through an inverter this can be solved by raising the threshold of the inverter, which results in an improvement in the signal response [33].

Since the Shift Registers model used is Johnson Counter, from 4 D flip -flops it is possible to obtain 8 phases (Figure 4.13) at the IC output all with the same frequency. Reducing the number of flip-flops to obtain the same number of phases significantly reduces the power consumed by the circuit. The 8 phases will have a duty cycle of 50%. The signals at the output of the Shift Register all have a duty cycle of 50% however this value can be changed to 12.5% by recombining the outputs with AND logic gates.

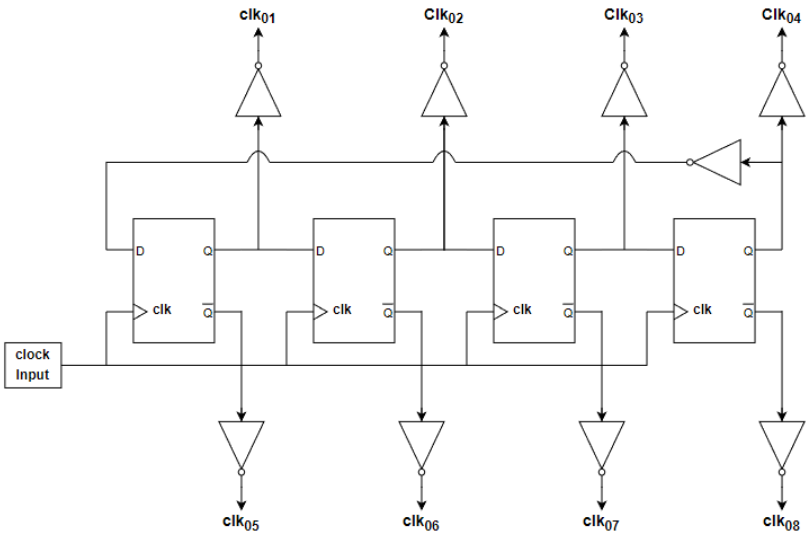


Figure 4.13- SR final circuit.

5

5. Electric Simulations

5.1. Description of the Electric Simulations

On this chapter the results of the electrical simulations of the complete circuit will be shown. The simulations are divided in three parts:

• The first one is the time-response analysis of the two main architectures of the circuit which are the Rx and the CC itself.

• The second is the Noise Phase Analysis of the same two main architectures but also of the Clock Buffer and the SR. This simulation will be done not only for a working frequency of 500 MHz but for 1 GHz, as well.

• The third is a Monte Carlo simulation with 200 runs for both the phase difference between two adjacent Clock Circuit outputs and the phase of one of the clock circuit outputs.

5.2. Time Response Simulations

To recap what was established in the previous chapters, the Clock Circuit (Rx+SR) has been excited with a LVDS signal with 0.6 V common mode, with an amplitude of 200 mV, with a frequency of 1 GHz. The circuit's input sources are ideal as opposed to what was mentioned earlier to facilitate the simulation process. The initial objective would be to use a non-ideal source as mentioned before, but it was defined that this element would not be part of the CC, so the simulation was done with ideal sources.

Figure 5.1 shows the simulation result. It is possible to see that the two input signals (LVDS) were transformed into a single pulse with the same frequency and with an amplitude of 1.2 V. In addition to this characteristic, there is a small delay in the output response, that is, the output signal takes around one clock period to settle. Another visible aspect of the Rx circuit response is that the output signal has small voltage spikes.

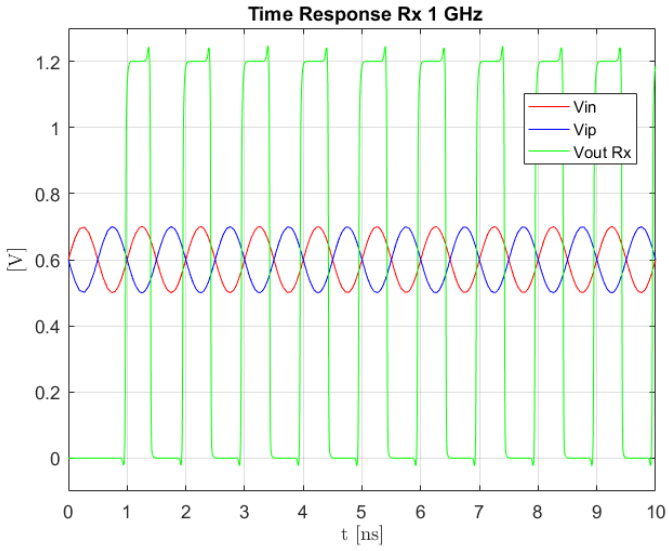


Figure 5.1- Simulation results of the Rx core,

The square pulse obtained at the output of the Rx circuit is subsequently fed to the Shift Register to obtain the 8-phase signal, both with a frequency of 125 MHz.

Figure 5.2 shows the simulation of the IC outputs. Since the input signal already had an initial delay, the Shift Register outputs also presented this initial delay, which will also be demonstrated in the Shift Register outputs. The delay value is now changed from 1ns to 2ns. Regarding the duty cycle, the value of both phases is near the same.

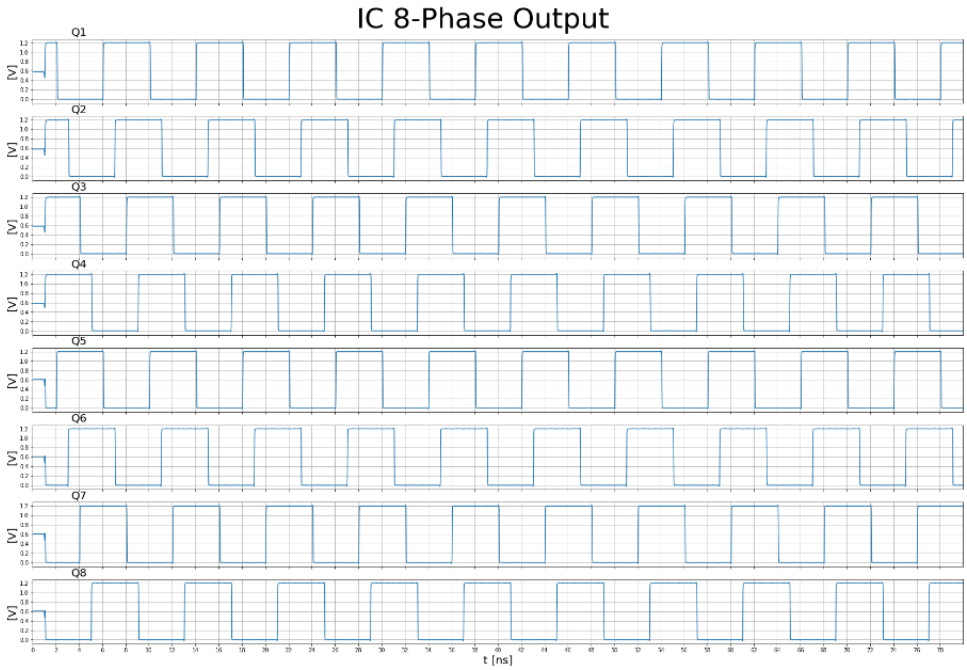


Figure 5.2- Simulation results of the 8-phase IC.

By simulating the time responses of the circuits, it is possible to calculate the total power consumed by the IC's constituent architectures. The power calculation was done by multiplying the value of V_{DD} with the output stage current. Since the current is not constant throughout the simulation and varies depending on the time instant, it is necessary to find its average value. Using the average value of current and multiplying the value of V_{DD} it is possible to obtain the power consumed by the various elements of the IC.

Table 5.1 shows the results of the power obtained at the output of the various components of the IC. The power values are obtained at the output of each Clock Circuit element, first of the clock buffer, after the RX Core (Clock Buffer + Digital Buffer) and finally of the IC (Rx Core + SR).

Table 5.1- Power consumption of the IC and its elements.

Frequency (GHz)	Circuit	Power (mW)
0.5	Clock Buffer	2.751
	Rx	3.014
	IC	4.104
1.0	Clock Buffer	2.753
	Rx	3.239
	IC	5.304

5.3. Phase-Noise Analysis

The objective of this work focuses on two essential factors low jitter and low power. Previously it was seen that the power is calculated through the response in time, the jitter is calculated through the Phase Noise output. However, before proceeding with the jitter calculation, it is worth observing the Phase Noise output of the various CC elements for frequencies of 500 MHz and 1 GHz off the differential input sources. The Phase Noise output simulation is performed in two phases. First, a periodic steady-state analysis (PSS) with a realistic input stimulus is performed to determine the quiescent situation at large signal regime. From there, a periodic noise analysis (PNOISE) is performed at the first harmonic to evaluate the sampled noise density function, in other words output Phase Noise. This analysis was performed for a spectrum of frequencies between 10 KHz and 100 MHz.

Figure 5.3 shows the simulation of the phase-noise output of the Clock Buffer for a frequency of 500 MHz and 1 GHz. The graph shows that both frequencies present a very identical characteristic curve with lower values of Phase Noise for the frequency of 1 GHz.

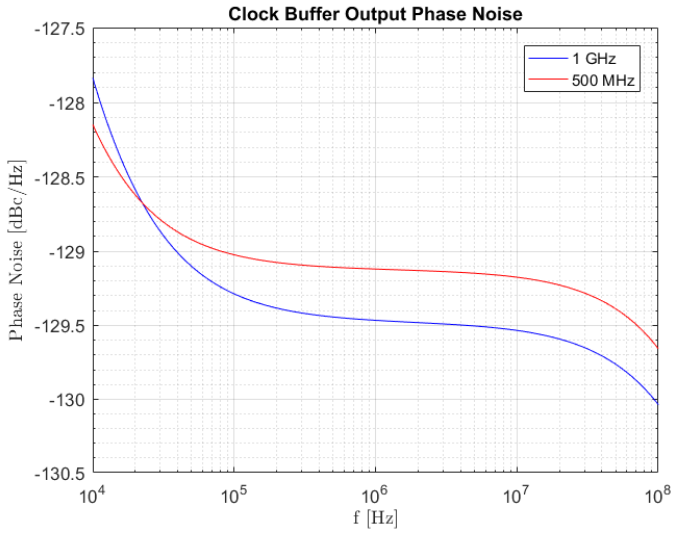


Figure 5.3- Simulation results of the clock buffer output phase noise.

Figure 5.4 shows the simulation of the phase-noise output of the Rx architecture for a frequency of 500 MHz and 1 GHz. The graph shows that both frequencies have, as in the previous simulation, a very identical characteristic curve. However, contrary to the previous simulation, the Phase Noise values for the 1 GHz frequency are higher than for the 500 MHz frequency.

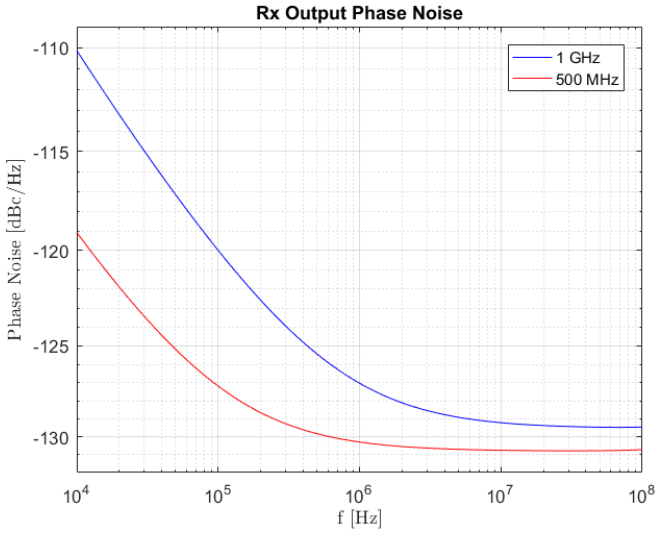


Figure 5.4- Simulation results of the Rx core output phase noise.

Finally, the same simulation is done but for the Clock Circuit. Figure 5.5 shows the Phase Noise simulation for the final circuit, that is, at the output of the CC. As in the first and second simulations, the characteristic curves obtained are quite identical. Relative to the obtained Phase Noise values, there is a decrease of about -6 dB when the frequency of the input signals is reduced by half.

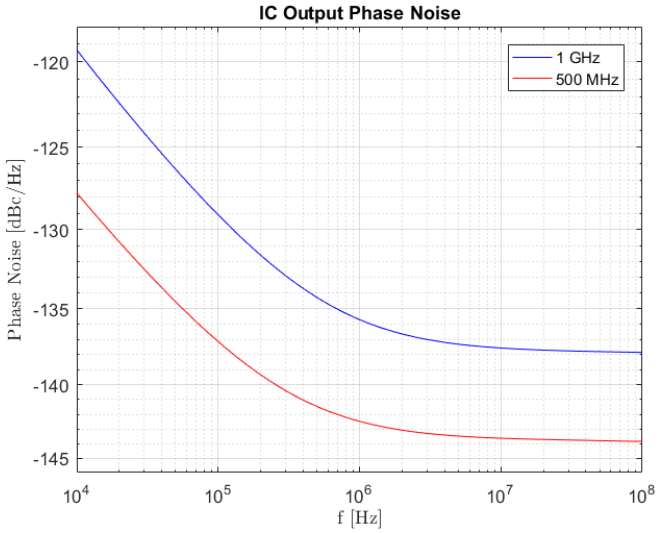
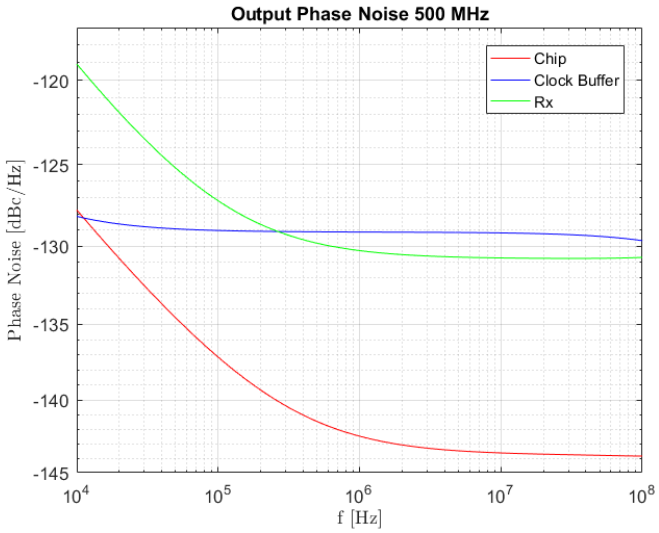


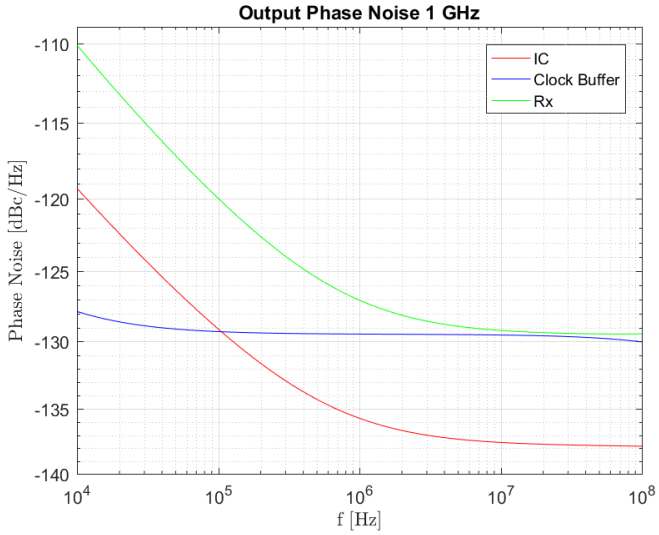
Figure 5.5- Simulation results of the IC output phase noise.

Figure 5.6 (a) and b)) show the comparation between the Phase noise simulations between the various elements of the CC and the output of the CC itself for frequencies of 500 MHz and 1 GHz of the input signals. For both frequencies the results are quite identical.

However, there are some differences between graphs. There is a more accentuated reduction in the Phase Noise values of the CC compared to the other architectures when the circuit is simulated with 500 MHz.



(a)



(b)

Figure 5.6- Simulation results of the CC and its elements output phase noise. a) 500 MHz, b) 1 GHz.

By simulating the Phase noise, it is possible to calculate the Phase Jitter. The first step in this process involves calculating the integrated phase noise power over the frequency range of interest phase, i.e., the area of the curve. The integration interval defined was between 12KHz and 20 MHz [9]. After calculating the power of the Phase noise for the selected integration interval, it is necessary to divide it by the power value of the carrier frequency, thus obtaining the Phase Jitter in rms (Appendix C).

Table 5.2 shows the jitter results obtained at the output of the various components of the Clock Circuit. RMS Jitter values are obtained through the Phase Noise at the output of each Clock Circuit element, first of the clock buffer, after the RX Core (Clock Buffer + Digital Buffer) and at the end of the CC (RX Core + SR).

Table 5.2- RMS Jitter of the IC and its elements.

Frequency (GHz)	Circuit	RMS Jitter (ps)
0.5	Clock Buffer	0.691
	Rx	0.587
	CC	1.09
1.0	Clock Buffer	0.332
	Rx	0.378
	CC	1.13

Subsequently, a simulation was made only with SR, injecting a square pulse of 1.2 V with a frequency of 1 GHz and 500 MHz. The remaining elements of the circuit were not considered in this simulation. This simulation aims to understand the contribution that this circuit has in the jitter value obtained in the CC output.

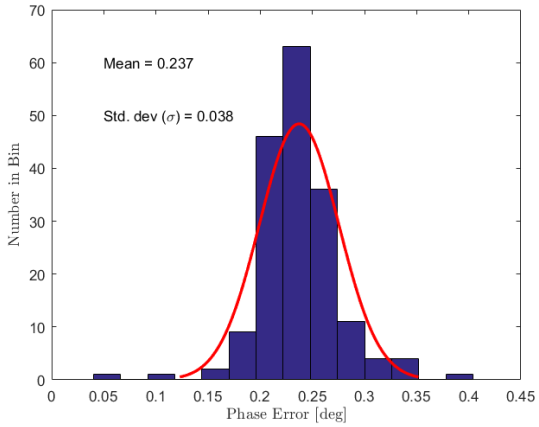
Table 5.3- RMS Jitter of the SR.

Frequency (GHz)	Circuit	RMS Jitter (ps)
0.5	SR	0.037
1.0	SR	0.019

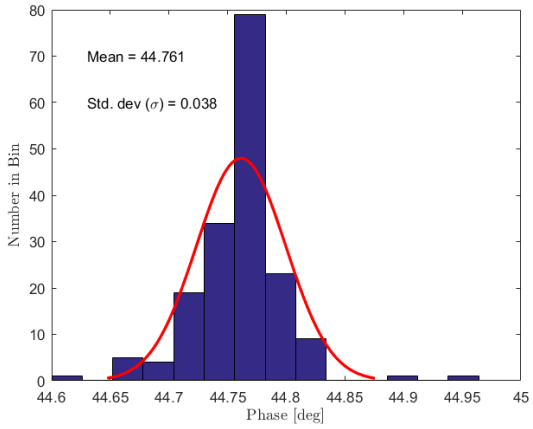
Table 5.3 shows the jitter values obtained in the SR output. The jitter values obtained for the SR were calculated through the simulation of the Phase Noise Output.

5.3. Monte Carlo Simulation

The Clock Circuit was subjected to a Monte Carlo simulation with 200 runs both to visualize the phase difference values between two adjacent circuit outputs and for the phase of one of the circuit's outputs. Figure 5.7 a) and b) show the respective simulations with a confidence level of 89%. Based on the information provided by histograms, it is inferred that the standard deviation value is identical in both simulations. Another visible feature is that adding the average value obtained in the two simulations gives 44.998 degrees, which is a very close value to the phase that each circuit output should ideally have, 45 degrees.



(a)



(b)

Figure 5.7- Histogram of the simulated a) Phase Difference b) Phase (200 Monte Carlo runs).

5.4. Discussion

In the beginning of this chapter, the results of the electrical simulations are shown. As mentioned, the simulations were divided into two parts:

- The first one is the time-response analysis of the two main architectures of the circuit, obtaining the power consumed by each of them
- The second is the Noise Phase Analysis of the same two main architectures but also of the Clock Buffer and the SR, obtaining the corresponding Phase Jitter value.

In this chapter, the results obtained will be duly discussed.

Table 5.4 and Table 5.5 show the calculated power consumption and phase jitter values for the various circuit elements.

Table 5.4- Power consumption and jitter of the IC and its elements.

Frequency (GHz)	Circuit	RMS Jitter (ps)	Power (mW)
0.5	Clock Buffer	0.691	2.751
	Rx	0.587	3.014
	CC	1.09	4.104
1.0	Clock Buffer	0.332	2.753
	Rx	0.378	3.239
	CC	1.13	5.304

Table 5.5- Jitter of the SR.

Frequency (GHz)	Circuit	RMS Jitter (ps)
0.5	SR	0.037
1.0	SR	0.019

Regarding the power consumed, there is a constant increase as more elements of the CC are considered for its calculation. Increasing the input frequency to double has little impact on the increase in power consumed by the Clock Buffer. In electrical signals, there is no relationship between frequency and power. Two signals with the same amplitude but different

frequencies will have the same power. For this reason, the difference in power consumed does not vary with the increase in frequency [15].

However, when analyzing the power consumed by the clock buffer connected to the digital Buffer (Rx) this statement is no longer verified. This phenomenon happens because the digital buffer is made up exclusively of CMOS inverters. In the CMOS inverter, the power consumption is mainly due to its dynamic power consumption, which depends on the frequency of the power supply [26]. This phenomenon will also be verified for the Shift Registers that are also constituted of CMOS Inverters.

The next thing to be analyzed are the phase jitter values, it is inferred that the Shift Registers are the ones that contribute most to the high final value obtained, when the clock signal that supplies the Register already has Phase Noise. On the other hand, when the SR is supplied with an ideal source with a frequency of 1 GHz and 500 MHz, the jitter value obtained is much lower when compared to the values obtained when the supply signal comes from the Rx core. This phenomenon can be easily explained since the source is ideal and the only contribution to the jitter value is given by the flip-flop chain and the reference clock signal no longer contributes to the jitter value.

When analyzing the jitter values obtained both for the clock buffer and later for the Rx architecture, it is noted that when the input source frequency increases, the jitter value decreases. Jitter decreases with clock frequency, an intuitive way to analyze this is to assume that the clock signal is, in general, a linear combination of its harmonics and that the receiver has a bandwidth. Therefore, the higher the clock frequency, the fewer harmonics contribute to noise and the lower the jitter [15]. However, there is a non-linearity in the jitter values obtained since the values at the CC output are higher for 1 GHz in relation to the values obtained for 500 MHz. The jitter at the Rx output is lower for 1 GHz so you would expect it to be lower than the CC output as well. This discrepancy in the values may have some explanations, among them: the fact that the Phase Noise simulation is relative to the first harmonic and does not consider all harmonics that may contribute to jitter; another possible explanation is the fact that in the jitter calculation the value is divided by the carrier power, which makes the obtained values very close, however there is a discrepancy of -6 dB in the Phase Noise curve.

6

6. Layout

After analyzing the results of the electrical simulations, the circuit layout must be made to produce it. The layout is intended to give an estimated idea of the area that the CC will occupy. In this chapter, the layout of the Clock Circuit will be done.

When making the circuit layout, it is necessary to keep in mind that the price of its assembly is strongly related to the area it occupies, that is, the larger the area, the more expensive it is. So, the main objective when making layout models is to keep the area as small as possible.

In addition to minimizing the area, other aspects were considered during the layout process. One of the main aspects in the layout design rules is the minimum distance between the different layers and according to [35] this process was respected. In addition to this aspect, the symmetry style was adopted for the layout design whenever possible. Symmetry eliminates the gradient effect along the substrate, minimizing mismatch of devices that occupy a large area. In the clock buffer differential pair, the common-centroid layout was used to minimize mismatch effects [42].

6.1. Rx Core

The Rx architecture is formed by the Clock Buffer and the Digital Buffer. The clock buffer was implemented with 9 transistors, 4 PMOS and 5 NMOS. The transistors that occupy the largest area of the circuit are those that have the current mirror function due to the high value of L and the PMOS transistors of the output stage of the buffer with high value of W. The number of fingers makes it possible to better distribute the area of the transistors so as not to concentrate all the dimensions in height or length. In the clock buffer layout (Figure 6.1), two types of metals are used, metal 1 and metal 2, this differentiation only serves to overlap connections so that it does not appear that they are the same connection and have the same label.

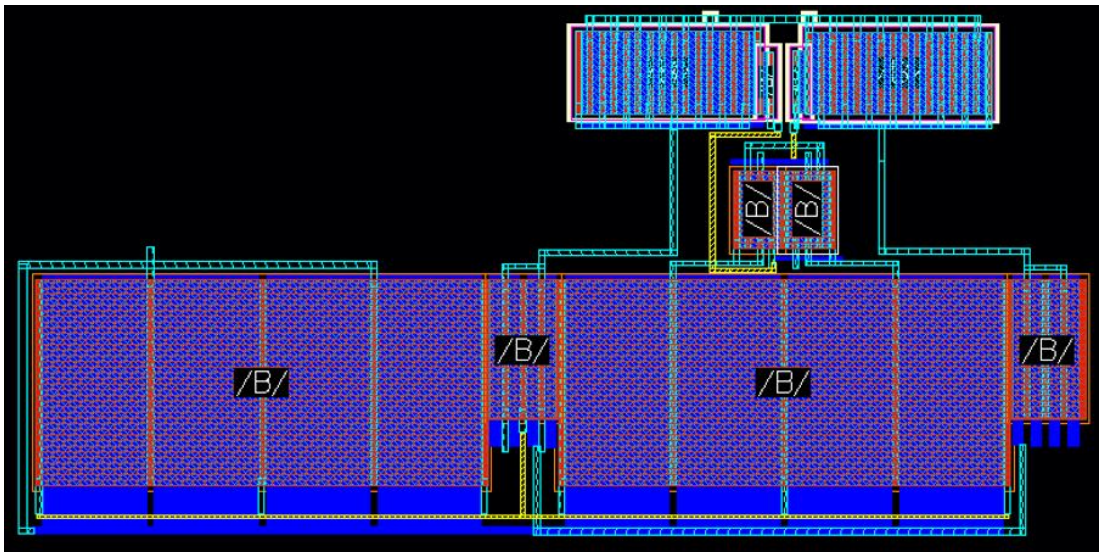


Figure 6.1- Clock Buffer layout.

The Digital buffer is implemented with 4 CMOS inverters, that is, by 4 PMOS transistors and 4 NMOS where the PMOS has 3 times greater W than the NMOS. The size of the transistors gradually increases, with the last ones presenting the largest dimensions and consequently it will be the last PMOS transistor in the chain that will contribute the most to the area occupied by the circuit. Figure 6.2 shows the layout of the Digital Buffer.

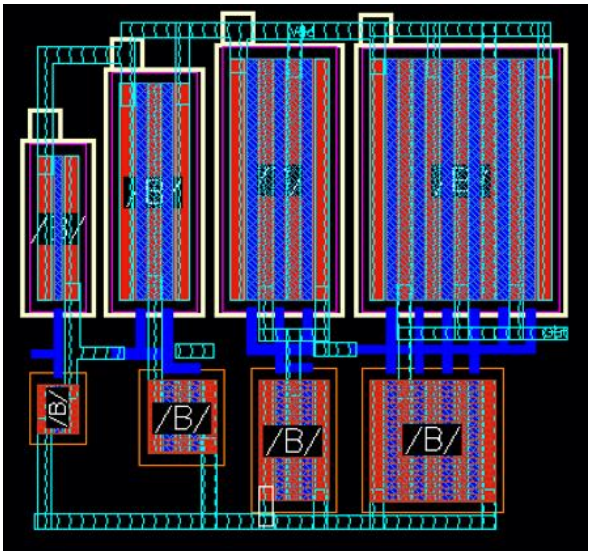


Figure 6.2- Digital Buffer layout

The Rx core layout (Figure 6.3) is implemented by joining the clock buffer output to the digital buffer input.

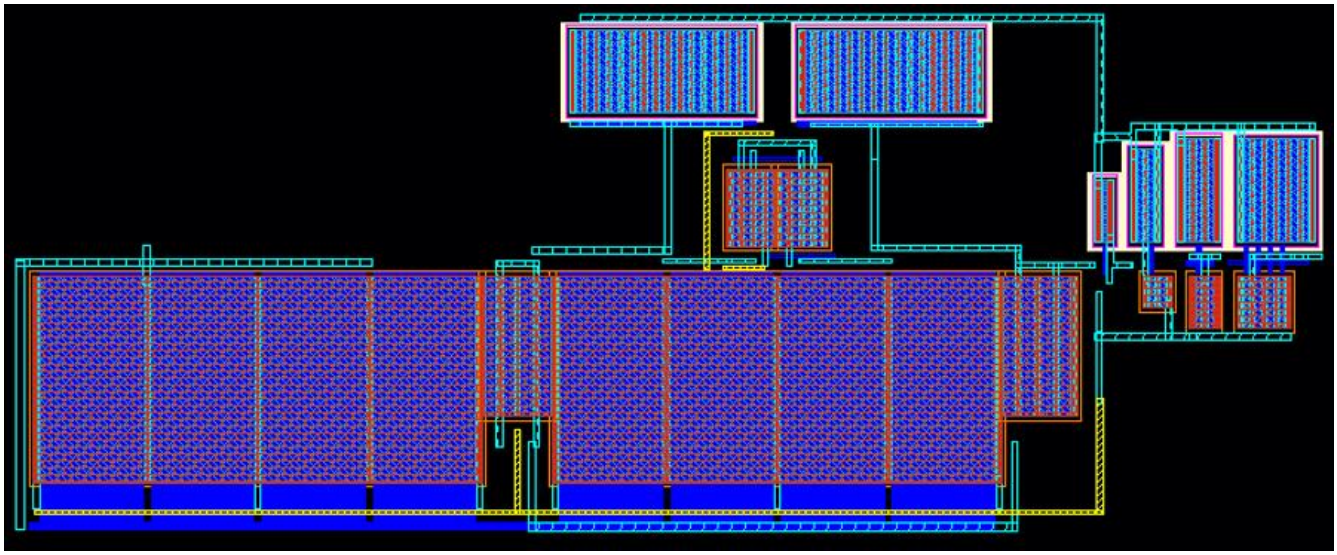


Figure 6.3- Rx core layout.

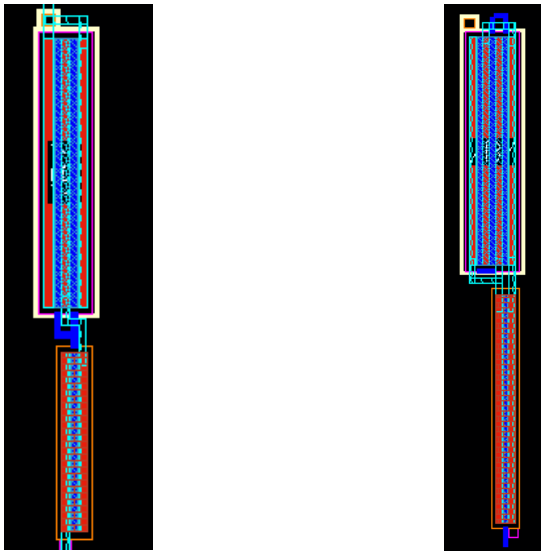
The areas occupied by the circuits that are part of the Rx core are represented in Table 6.1.

Table 6.1- Layout areas of the elements of the Rx core.

Circuit	Area (μm^2)
Clock Buffer	1154.1
Digital Buffer	121.1
Rx	1415.3

6.2. Shift Register

Shift Registers are implemented with flip-flops, each flip-flop is implemented through transmission gates and inverters. Not only an inverter but also a transmission gate are quite simple to implement since they only need two transistors, an NMOS and a PMOS, with the PMOS being the main responsible for the area occupied in the two circuits. The layout of the inverter (Figure 6.4 a)) and the transmission gate (Figure 6.4 b)) will also be quite simple to design.



(a) **(b)**

Figure 6.4- a) Inverter Layout, b) Transmission gate Layout.

Using the layout circuits of the inverter and the transmission gate, it is possible to obtain the layout of the flip-flop D (Figure 6.5). The flip-flop D is designed with 3 transmission gates and 3 inverters. For the layout design of this circuit, metal 2 was again used to clarify the overlapping of the circuit wires.

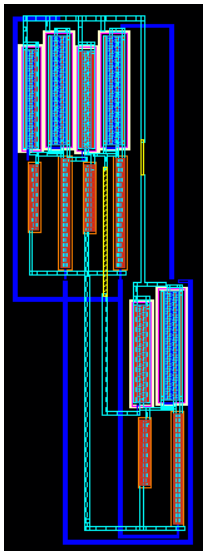


Figure 6.5- DTG flip-flop Layout.

The Shift Register consists of connecting 4 D flip-flops in parallel, but as mentioned before, it is necessary to add an inverter to the output of each flip-flop. Figure 6.6 shows the layout of the Shift Register taking this feature into account. Due to the irregular shape of the flip-flops and inverters, the inverters have been glued horizontally instead of vertically so that they take up as little space as possible. However, the spacing between the elements and between the wires is in accordance with the rules.

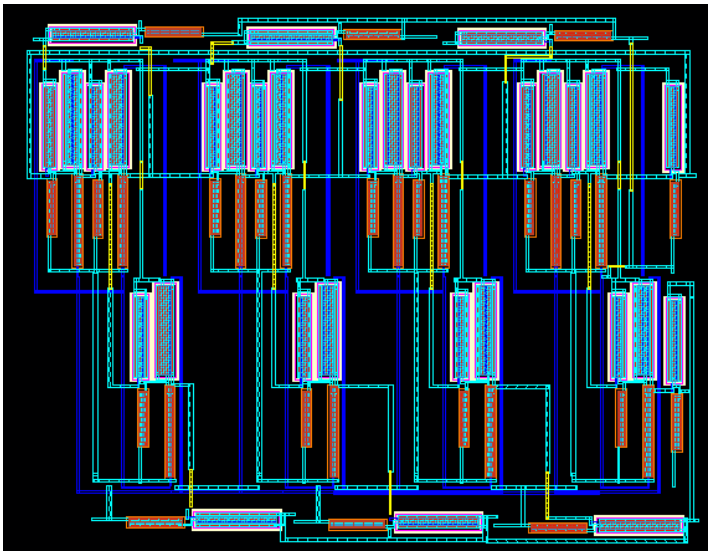


Figure 6.6- SR Layout.

Table 6.2 shows the areas of elements that implement the Shift Register and of the Shift Register itself. As expected, the area occupied by the Shift Register is larger than the area of its elements.

Table 6.2- Layout areas of the elements of the SR.

Circuit	$Area\ (\mu m^2)$
Inverter	36.93
Transmission Gate	62.02
Flip-flop D	789.69
Shift Register	4338.7

The CC is implemented by connecting the output of the Rx core to the input of the Shift Register. The CC layout (Figure 6.7) is also designed using the same principle however there are other connections that are also connected between them namely the ground of both circuits as well as the V_{DD}. The two circuits have very different sizes and there are some wires that are neither part of one circuit nor the other, they work only as connecting elements, but contribute to the chip area. It is also because of this disproportion of circuit sizes that the clock buffer, being the smallest, is in a central position in relation to the Shift Register. In this implementation, along with others, there is also the presence of a metal 2 for the reasons mentioned above.

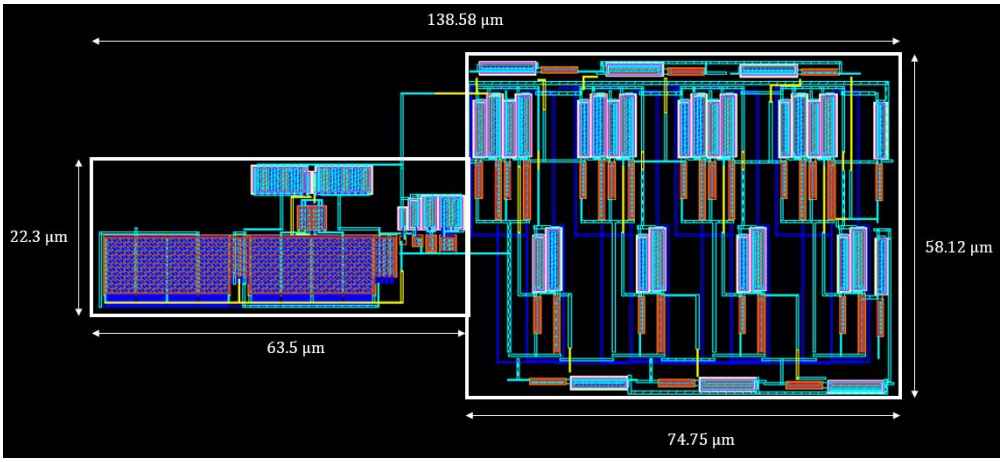


Figure 6.7- Chip Layout.

Table 6.3 shows the different areas of the CC and by the circuits on the CC. The chip (IC) has an area of 8054. and the main element that contributes to this value is the Shift Register.

Table 6.3- Layout areas of the elements of the IC.

Circuit	$Area\ (\mu m^2)$
Rx	1415.3
Shift Register	4338.7
CC	8054.3

6.3. Analysis of the results

Finally, based not only on the results obtained in the electrical simulations chapter, but also on the calculated chip area, it is possible to compare with other studies carried out for the same circuits (Rx [15] and CC [6]). Table 6.4 and Table 6.5 show the difference in the values obtained for different simulations.

Table 6.4- Comparison between the proposed Rx and other Rx core implementation @500 Mhz.

Rx	[15]	This work
Technology	130 nm	130 nm
Frequency (GHz)	0.5	0.5
Area (μm^2)	78400	2117
RMS Jitter (fs)	73.05	1090
Power (mW)	18.22	3.014

Table 6.5- Comparison between the proposed CC and other CC implementation @1 Ghz.

CC	[6]	This work
Technology	130 nm	130 nm
Frequency (GHz)	1.0	1.0
Area (μm^2)	18000	8054
RMS Jitter (fs)	93	1130
Power (mW)	3.88	5.4

Analyzing the results of the table it is verified that at the power level both are quite congruent and identical. However, there is still a small discrepancy regarding the value of the power of the Rx, and the one obtained in this work was lower than in the study [6]. Regarding the area occupied by the chip, the one implemented in this work is significantly lower than in the other works. Regarding the jitter values there is a large discrepancy, this phenomenon can be explained by the method used to calculate jitter. The fact that the Phase Noise is related to the first harmonic and only the first 10 harmonics are considered to generate the Phase Noise curve may be the main factor.

7

7. Conclusions

This project aimed to implement a low-power and low-noise clock generator for high-speed ADCs in 130 nm CMOS technology. The generator is implemented on a chip using state-of-the-art circuitry. This generator can provide clock signals with a power not exceeding 5.4 mW and a phase jitter around 1 ps of the output signal for an input LVDS signal with 200 mV for a frequency of 1GHz, being implemented to work with non-ideal sources of supply.

The Generator can power an 8 channel TI-ADC at 1 GHz. However, this generator can supply lower frequency ADCs, and the ideal working frequency is between 500 MHz and 1 GHz.

In Chapter 2, a brief overview of the various ADCs used in today's technological market was given. Based on the CC architecture chosen to implement the clock generator, the TI-ADC was the one that presented the most attractive characteristics for this purpose. In addition to the various ADCs, a set of metrics were presented that influence the functioning of the ADC, how to calculate them to better understand how to mitigate them.

In chapter 3, a review of the architectures already implemented, and their main characteristics is made. In this chapter, a balance is made of the positive and negative points of these architectures and what features they can offer in the implementation of the CC.

Based on what was mentioned, it was concluded that the circuit would be implemented with two main architectures, an Rx core and an MPCG. The Rx core would in turn be implemented by two circuits, a clock buffer and a digital buffer, each with its own purpose for the circuit. In this chapter it was also closed that the Clock Buffer would be implemented by a SAL + RCMF circuit and that the MPCG would be implemented by Shift Registers. SARL + RCMF was the topology of choice for the clock buffer due to the fact that not only the D2S circuit operates in current mode, but also has the best trade-off between jitter and power. Regarding the choice of Shift Register to implement the MPCG instead of, for example, using A DLL since it can change its output frequency and has the flexibility to generate clock signals with different duty cycles.

In chapter 4, the circuits and sub-circuits that implement the CC are presented in greater detail. In this chapter, the dimensioning of these components is done and how this dimension will affect the operation of the circuit.

In chapter 5 we show the results of the electrical simulations. The CC is simulated in the time domain and in the frequency domain and is powered with a source with a voltage of 200 mv peak to peak with 0.6 V of offset and with a frequency that can be 500 MHz or 1 GHz depending on the simulation. The temporal response is a simulation that allows observing the characteristic of the signal at the output of the CC, in this case the output signal presents 8 different phases, all with the same frequency. Through this simulation it is possible to calculate the power consumed by the CC and its components. The frequency domain response allows obtaining the Phase Noise at the output of the CC and its components and it is through this simulation that the jitter value can be calculated. After this process, at the end of this chapter, the results of the electrical simulations are discussed. Based on the results obtained, it is concluded that the power value reached is in accordance with what was previously established. Regarding the jitter value reached, it is concluded that there were some limitations for the calculation of this value, however, despite the limitations, the values are very close to the intended limit.

Chapter 6 presents the CMOS circuit layouts used to implement the CC. Through the layout it is possible to calculate the chip area, with the final value obtained being 8054.3 μm^2. It was concluded that the SR is the one that contributes the most to the area value, due to the large dimensions of the inverters and transmission gates that constitute it. At the end of the chapter, using the results of the previous chapter and the calculated layout areas, a comparison is made with the results of identical architectures studied in other works. Based on this analysis it is essentially concluded that the circuit area is much smaller in this work, however the jitter is much higher. The implementation made in this work, despite being able to lose in terms of the Phase Noise generated, gains in relation to the cost of the chip area.

Finally, the values obtained are in accordance with the expected, however, using a lower CMOS technology, it would be possible to reduce the jitter value and power obtained or increase the CC supply frequency to values greater than 1 GHz.

7.1. Future Work

The Clock Circuit that was designed and developed in this research thesis reached power and jitter values within the expected ranges. However, there are some points that can be investigated to improve the results obtained. Since the process technology used in the implementation of the CC was 130 um and nowadays the available technologies are already quite inferior. Using a lower technology, for example 65 nm, it would be interesting to analyze the results obtained and compare the differences between them

8. References

[1] Y. H. Gao, Y. L. Wang, and Z. P. Zhang, 'A multi-phase clock design for super high-speed time interleaved analog-to-digital converter', *Proc. Int. Conf. Anti-Counterfeiting, Secur. Identification, ASID*, no. 24, pp. 17–20, 2012, doi: 10.1109/ICASID.2012.6325291.

[2] J. Nunez, A. J. Gines, E. J. Peralias, and A. Rueda, 'Low-jitter differential clock driver circuits for high-performance high-resolution ADCs', *2015 Conf. Des. Circuits Integr. Syst. DCIS 2015*, pp. 15–18, 2016, doi: 10.1109/DCIS.2015.7388558.

[3] D. Redmayne and A. Steer, 'Understanding the effect of clock jitter on high-speed ADCs', *Electron. Eng. Times*, no. 1542, pp. 45–46, 2008.

[4] M. B. Labhane and P. Palsodkar, 'Various architectures of analog to digital converter', *2015 Int. Conf. Commun. Signal Process. ICCSP 2015*, pp. 1199–1203, 2015, doi: 10.1109/ICCSP.2015.7322696.

[5] C. Vogel, 'Time-Interleaved Analog-To-Digital Converters : Status and Future Directions', *2006 IEEE Int. Symp. Circuits Syst.*, pp. 3386–3389, 2006, doi: 10.1109/ISCAS.2006.1693352.

[6] S. A. Zahrai, N. Le Dortz, and M. Onabajo, 'Design of clock generation circuitry for high-speed subranging time-interleaved ADCs', *Proc. - IEEE Int. Symp. Circuits Syst.*, pp. 31–34, 2017, doi: 10.1109/ISCAS.2017.8050895.

[7] M. Zhang, Y. Han, F. Geng, C. Zhang, and B. Duan, 'Research on high-speed ADC test and evaluation system', *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, no. 3, pp. 4927–4931, 2017, doi: 10.1109/CAC.2017.8243652.

[8] 'High-speed ADC Input Clock Issues High-speed ADC Input Clock Issues', *Teledyne e2v*, pp. 1–21, 2007.

[9] B. Brannon, 'Sampled systems and the effects of clock phase noise and jitter', *Appl. Note AN-756*, no. 1, p. 12, 2004, [Online]. Available: http://qtwork.tudelft.nl/~schouten/linkload/jitter-appnote-andev.pdf.

[10] 'Clock Jitter Definitions and Measurement Methods', *SiTime*, pp. 1–17, 2014.

[11] W. Kester, 'Converting Oscillator Phase Noise to Time Jitter', *Analog Integr. Circuits Signal Process. Devices*, pp. 1–10, 2009.

[12] C. R. Parkey, W. B. Mikhael, D. B. Chester, and M. T. Hunter, 'Modeling of jitter and its effects on time interleaved ADC conversion', *AUTOTESTCON (Proceedings)*, pp. 367–372, 2011, doi: 10.1109/AUTEST.2011.6058747.

[13] Y. Chen, C. Fan, and J. Zhou, 'Low jitter clock driver for high-performance pipeline ADC', *Proc. Int. Conf. ASIC*, no. 1, pp. 1–4, 2013, doi: 10.1109/ASICON.2013.6812047.

[14] H. Barthelemy, 'Current Mode and Voltage Mode: basic considerations', *2003 46th Midwest Symp. Circuits Syst.*, pp. 161–163, 2003, doi: 10.1109/mwscas.2003.1562243.

[15] J. Núñez, A. J. Ginés, E. J. Peralías, and A. Rueda, 'Design methodology for low-jitter differential clock recovery circuits in high performance ADCs', *Analog Integr. Circuits Signal Process.*, vol. 89, no. 3, pp. 593–609, 2016, doi: 10.1007/s10470-016-0870-6.

[16] A. Zanchi and F. (Ching-Y. Tsay, 'A 16-bit 65-MS/s 3.3-V Pipeline ADC Core in SiGe BiCMOS With 78-dB SNR and 180-fs Jitter', *IEEE J. Solid-State Circuits*, vol. 40, no. 6, pp. 1225–1237, 2005.

[17] F. Yuan, *CMOS Current-Mode Circuits for Data Communications*. 2007.

[18] L. Cheng, H. Yang, L. Luo, and J. Ren, 'Analysis and design of low-jitter clock driver for wideband ADC', *ICSICT-2010 - 2010 10th IEEE Int. Conf. Solid-State Integr. Circuit Technol. Proc.*, pp. 515–517, 2010, doi: 10.1109/ICSICT.2010.5667355.

[19] E. A. M. Klumperink, X. Gao, and B. Nauta, *Polyphase Multipath Circuits for Cognitive Radio and Flexible Multi-phase Clock Generation*. 2009.

[20] X. Gao, E. A. M. Klumperink, and B. Nauta, 'Advantages of Shift Registers Over DLLs for Flexible Low Jitter Multiphase Clock Generation', *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 55, no. 3, pp. 244–248, 2008, doi: 10.1109/TCSII.2008.918972.

[21] J. Núñez, A. J. Ginés, E. J. Peralías, and A. Rueda, 'An approach to the design of low-jitter differential clock recovery circuits for high performance ADCs', *2015 IEEE 6th Lat. Am. Symp. Circuits Syst. LASCAS 2015 - Conf. Proc.*, pp. 2–5, 2015, doi: 10.1109/LASCAS.2015.7250431.

[22] 'Transmission Gate', *Electronics Tutorials*. https://www.electronics-tutorials.ws/combination/transmission-gate.html (accessed Oct. 30, 2021).

[23] W. Xu and E. G. Friedman, 'Clock feedthrough in CMOS analog transmission gate switches', *Proc. Annu. IEEE Int. ASIC Conf. Exhib.*, pp. 181–185, 2002, doi: 10.1109/ASIC.2002.1158052.

[24] 'Designing Combinational Logic Gates in Cmos', *Berkeley*, pp. 1–65, 2000, [Online]. Available: http://bwrcs.eecs.berkeley.edu/Classes/icdesign/ee141_f01/Notes/chapter6.pdf.

[25] B. Shelley, 'The CMOS Inverter Explained', *Calpoly*. https://courseware.ee.calpoly.edu/~dbraun/courses/ee307/F02/02_Shelley/Section2_BasilShelley.htm (accessed Feb. 01, 2022).

[26] M. D. Fernandes, L. B. Oliveira, J. Goes, and J. P. Oliveira, 'Design of a low phase error multiphase clock generator for modern wideband receivers', *2016 12th Conf. Ph.D. Res. Microelectron. Electron. PRIME 2016*, pp. 1–4, 2016, doi: 10.1109/PRIME.2016.7519504.

[27] K. Mehta, N. Arora, and B. P. Singh, 'Low Power Efficient D Flip Flop Circuit', *Proc. Publ. by Int. J. Comput. Appl.*, pp. 16–19, 2011.

[28] H. Kumar, A. Kumar, and A. Islam, 'Comparative analysis of D flip-flops in terms of delay and its variability', *2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015*, pp. 1–6, 2015, doi: 10.1109/ICRITO.2015.7359339.

[29] L. Li and J. Hu, 'A transmission gate flip-flop based on dual-threshold CMOS techniques', *Midwest Symp. Circuits Syst.*, pp. 539–542, 2009, doi: 10.1109/MWSCAS.2009.5236037.

[30] E. Klumperink, R. Dutta, Z. Ru, B. Nauta, and X. Gao, 'Jitter-Power minimization of digital frequency synthesis architectures', *Proc. - IEEE Int. Symp. Circuits Syst.*, no. May, pp. 165–168, 2011, doi: 10.1109/ISCAS.2011.5937527.

[31] R. Dutta, E. Klumperink, X. Gao, Z. Ru, R. Van Der Zee, and B. Nauta, 'Flip - Flops for Accurate Multi - Phase Clocking : Transmission Gate versus Current Mode Logic', *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 60, pp. 422–426, 2013, doi: 10.1109/TCSII.2013.2261173.

[32] J. Rabaey, 'Designing sequential logic circuits', *Berkeley*, pp. 270–315, 2000.

[33] 'The Cmos Inverter', *Berkeley*, pp. 144–192, 1999, [Online]. Available: http://bwrcs.eecs.berkeley.edu/Classes/icdesign/ee141_f01/Notes/chapter5.pdf.

[34] 'CMOS Inverter', *StudyLib*. https://studylib.net/doc/18339482/5.-cmos-inverter.

[35] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation, 3rd Edition (IEEE Press Series on Microelectronic Systems)*, vol. 3rd. 2010.

[36] H. Pandey and A. Kai, 'Shift Registers in Digital Logic', *GeeksforGeeks*, 2019. https://www.geeksforgeeks.org/?p=168099 (accessed Feb. 12, 2022).

[37] 'The Shift Register', *Electronics Tutorials*. https://www.electronics-tutorials.ws/sequential/seq_5.html (accessed Nov. 01, 2021).

[38] G. C. Naguboina and K. Anusudha, 'Realization and Synthesis of Ring Counter and Twisted Ring Counter Using Reversible Logical Computation with Minimum Quantum Cost', *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, pp. 926–931, 2018, doi: 10.1109/ICIRCA.2018.8597345.

[39] 'Johnson Ring Counter', *Electronics Tutorials*. https://www.electronics-tutorials.ws/sequential/seq_6.html.

[40] T. Maekawa, S. Amakawa, N. Ishihara, and K. Masu, 'Design of CMOS inverter-based output buffers adapting the Cherry-Hooper broadbanding technique', *ECCTD 2009 - Eur. Conf. Circuit Theory Des. Conf. Progr.*, no. September, pp. 511–514, 2009, doi: 10.1109/ECCTD.2009.5275025.

[41] W. J. Dally and J. W. Poulton, *Digital systems engineering*. Cambridge University Press, 1998.

[42] S. Youssef, D. Dupuis, R. Iskander, and M. M. Louërat, 'Automatic stress effects computation based on a layout generation tool for analog IC', *Proc. IEEE Int. Work. Behav. Model. Simulation, BMAS*, no. March 2015, pp. 7–12, 2010, doi: 10.1109/BMAS.2010.6156590.

[43] L. Valavala, K. Munot, and K. B. R. Teja, 'Design of CMOS inverter and chain of inverters using neural networks', *Proc. - 2018 IEEE 4th Int. Symp. Smart Electron. Syst.*

iSES 2018, pp. 269–274, 2018, doi: 10.1109/iSES.2018.00065.

[44] A. Buck, 'Phase Noise to jitter', *MATLAB Central File Exchange*, 2022. https://www.mathworks.com/matlabcentral/fileexchange/22038-phase-noise-to-jitter (accessed Feb. 22, 2022).

Appendix A: DC Operating Point

This appendix contains all information regarding the DC operating point of the clock buffer.

Once the sizing of the clock buffer is done, a DC analysis will be carried out because it is necessary to infer whether the transistors are in the saturation zone. A table was created to facilitate the visualization of the results obtained by the DC simulation of the clock buffer.

Table A.1 shows not only has the voltage values of V_{GS}, V_{DS}, V_{TH} and V_{dsat} for each transistor, but also the current that flows through it.

Table A.1- DC Operating Point - Voltages of the SAL+RCMF circuit transistors.

Transistor	I_d (µA)	V_{GS} (mV)	V_{DS} (mV)	V_{TH} (mV)	V_{dsat} (mV)
M0	100	486.5	486.5	229.3	253.3
MB	82.6	486.5	170.5	229.3	253.5
$M_{11,12}$	41.31	429.6	454	359.7	96.08
$M_{13,14}$	41.34	575.6	575.6	288	272.5
$M_{21,22}$	43.24	499.1	499.1	258	230.2
$M_{23,24}$	43.2	575.6	700.9	284.6	275.3

Analyzing the values in the table all transistors are in the strong inversion saturation region, except for the MB.

This transistor has a current significantly less than the nominal value. It is known that the W/L ratio is directly proportional to the current flowing through the transistor, so if the W value increases, the drain current of the transistor will increase as well.

The V_{dsat} values are close to the theoretical values idealized in the design, there is only a small discrepancy in this value when compared to the theoretical value. This discrepancy may be due to the body effect that transistors are subjected to.

Appendix B: Inverter operating mode

This appendix contains all information regarding the Inverter operating mode.

This leads to the following interpretation of the inverter. When Vin is high and equal to V_{DD}, the NMOS transistor is on, while the PMOS is off, because exists a direct path between V_{out} and the ground node, resulting in a steady-state value of 0 V. On the other hand, when the input voltage is low (0 V), NMOS transistor is off and PMOS transistor is on because exists a path between V_{DD} and V_{out}, yielding a high output voltage [25].

Figure B.1 shows the various regions of operation of the CMOS inverter:

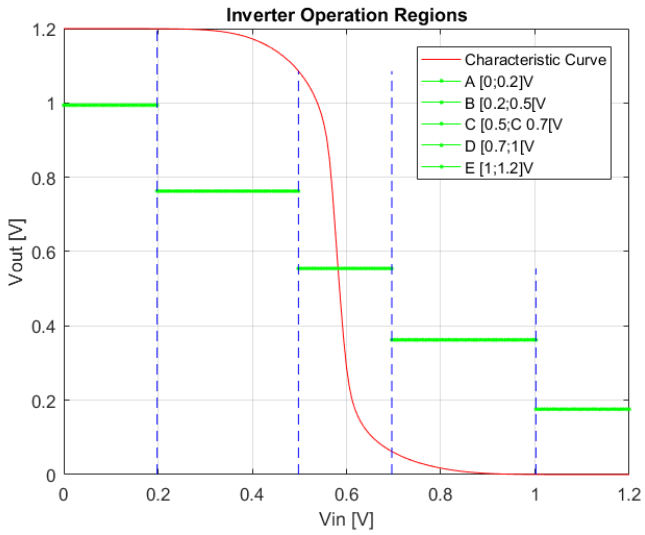


Figure B.1- Operations Regions of the CMOS Inverter based in [43].

Region A

In this region the input voltage is between 0 and V_{TC}. the PMOS device is forward biased and therefore on in the linear region. The NMOS device is cut off since the input voltage is below V_{TC}The power dissipation is zero.

Region B

Here the input voltage is above V_{TC}. The PMOS device is in the linear region and the NMOS device is in the saturation region. The current now flows through both devices and because of that the power dissipation is no longer zero.

The maximum allowable input voltage at the low logic state (V_{IL}) occurs in this region. V_{IL} is the value of V_{in} at the point where the slope of the V_{TC} is -1. In other words, V_{IL} occurs when the condition in (A.3) occurs.

$$\frac{dV_{out}}{dV_{in}} = -1 \tag{A.3}$$

Region C

In this region there exists a point in the middle where $V_{in} = V_{out}$. Not only is the PMOS device is in the saturation region, but the NMOS device is in the saturation region, as well. Regarding power dissipation, this value reaches a peak in this region, namely at where $V_{in} = V_{out} = V_M$

Region D

Region D occurs between an input voltage slightly higher than V_M but lower than $Vdd - V_{TP}$. The PMOS device is in the saturation region and the NMOS device is forward biased and therefore on, in the linear region.

The minimum allowable input voltage at the logic high state (V_{IH}) occurs in this region. V_{IH} occurs at the point where the slope of the V_{TC} is –1 (A.3).

Region E

The PMOS device is cut off when the input is near V_{DD}. On the other hand, the NMOS device is forward biased and therefore on in the linear region. Just like in the region A the total power dissipation is zero.

Appendix C: Jitter calculation function

This appendix contains all information regarding the method used to calculate jitter.

The first step to calculate the jitter was to transfer the Phase Noise output values in dB/Hz and the respective frequency value obtained through the cadence simulations to MATLAB. The data obtained in cadence were exported to an Excel file that were then imported into MATLAB. After uploading the Excel file, a function is created through Matlab that automatically saves the Excel values in variables. Below is the code that allows saving the imported file.

```
[freq_buffer,Lf_buffer]=importfile_ClockBuffer('Phase Noise.xlsx','Phase Noise Clock Bufferf'); %Import Clock Buffer Phase Noise values

%-----------------------------------------------------------------------------------------------------------------------------------

%Fucntion importfile_Clock Buffer

function [f,Lf] = importfile(workbookFile,sheetName,startRow,endRow)

%% Input handling

% If no sheet is specified, read first sheet
if nargin == 1 || isempty(sheetName)
    sheetName = 1;
end

% If row start and end points are not specified, define defaults
if nargin <= 3
    startRow = 2;
    endRow = 402;
end

%% Import the data
data = xlsread(workbookFile, sheetName, sprintf('A%d:B%d',startRow(1),endRow(1)));
for block=2:length(startRow)
    tmpDataBlock = xlsread(workbookFile, sheetName, sprintf('A%d:B%d',startRow(block),endRow(block)));
    data = [data;tmpDataBlock]; %#ok<AGROW>
end

%% Allocate imported array to column variable names
f = data(:,1);
Lf = data(:,2);
```

Figure C.1- Imporfile function code

These values were stored in an array. Then it was necessary to verify which vector position corresponded to the integration frequency for the respective Phase Noise value, because the simulation was performed for frequencies between 10 KHz and 100 MHz, but the integration frequency was between 12 KHz and 20 MHz. Once this position is found, a new vector is created not only for the frequency but also for the phase noise, which only contains the values that correspond to the integration interval. The code below represents this process.

```
fb2=freq_buffer500(9:330); %fb2 only have frequency values bteween 12k and 20M
Lf_b2=Lf_buffer500(9:330); %Lf_b2 only have Phase Noise values between 12K and 20M
Jiterr_ClockBuffer500=Pn2Jitter(fb2,Lf_b2,0.5e9) %Jitter calculation for the Clock Buffer with fc=500 MHz
```

Figure C.2- Code to calculate Jitter.

These new vectors will be the input variables in the jitter calculation function [44], Lf being the Phase Noise values, f the frequency values corresponding to Lf and fc the carrier frequency. The following code allows the calculation of the RMS jitter.

```
function Jitter = Pn2Jitter(f, Lf, fc)
%
% Summary: Jitter (RMS) calculation from phase noise vs. frequency data.
%
% Calculates RMS jitter by integrating phase noise power data.
%  Phase noise data can be derived from graphical information or an
%  actual measurement data file.
%
% Usage:
%   Jitter = Pn2Jitter(f, Lf, fc)
% Inputs:
%   f:  Frequency vector (phase noise break points), in Hz, row or column.
%   Lf: Phase noise vector, in dBc/Hz, same dimensions, size(), as f.
%   fc: Carrier frequency, in Hz, a scalar.
% Output:
%   Jitter: RMS jitter, in seconds.

L = length(Lf);
if L == length(f)
% Fix ill-conditioned data.
I=find(diff(Lf) == -10); Lf(I) = Lf(I) + I/10^6; % Diddle adjacent Lf with
                                                 % a diff=-10.00dB, avoid ai:/0
% Just say "No" to For loops.
lp = L - 1; Lfm = Lf(1:lp); LfM = Lf(2:L); % m~car+, M=cdr
fm  = f(1:lp); fM = f(2:L);  ai = (LfM-Lfm) ./ (log10(fM) - log10(fm));
% Cull out problematic fine-sieve data from the PN9000.
Iinf = find( (fm.^(-ai/10) == inf) | fm.^(-ai/10)<10^(-300)); % Find Inf

fm(Iinf) = []; fM(Iinf-1) = []; Lfm(Iinf) = []; LfM(Iinf-1) = [];
ai(Iinf) = []; f(Iinf) = []; Lf(Iinf) = [];

% Where's the beef?
Jitter = ...
    1/(2*pi*fc)*sqrt(2*sum( 10.^(Lfm/10) .* (fm.^(-ai/10)) ./ (ai/10+1)...
    .* (fM.^(ai/10+1) - fm.^(ai/10+1)) ));

else
    disp('> > Oops!');
    disp('> > > The f&Lf vector lengths are unequal.  Where''s the data?')
    Jitter = sqrt(sqrt(-12446784));
end % if L
toc
```

Figure C.3- Jitter function code.