Design of a time-to-digital converter to be used as a phase detector in a PLL in 65 nm technology

Luis Guillermo Macias Rojas

September 15, 2025

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Acknowledgements

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Dedications

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

Systems that require precise timing and synchronization, such as high-speed communication systems, data converters, and clock generation circuits, often rely on phase-locked loops (PLLs) to maintain signal integrity and minimize jitter. For a long time the field of PLLs has been entirely dominated by analog designs, particularly the charge-pump PLL (CP-PLL) architecture, which has been the go-to solution for many applications. However, as technology scales down the limitations of analog PLLs (and analog circuits in general) become more pronounced because of the signal representation in the voltage domain [1]. This has led to a growing interest in digital PLLs (DPLLs), which exploit the several advantages that come with the technology scaling (like the inherit robustness of digital circuits against noise, coupling and PVT variations).

At the heart of most DPLLs is the time-to-digital converter (TDC), which serves as the phase detector (PD) by measuring the time difference between the reference clock and the feedback clock from the Digitally-controlled oscillator (DCO). In addition to the lower sensitivity to most disturbances that the TDC provides for the DPLL, it also eliminates the ever present problems

of charge pump mismatch and static phase error (dead-zone) of the analog PLL.

Between the several types of TDC developed in recent years, there hasn't been much interest in improving the simplest type of TDC, the delay line TDC (also called flash TDC because of its resemblance to flash ADCs). The majority of the research has been focused on improving the resolution and linearity of the TDC by using more complex architectures, such as local passive interpolation TDCs [2] or the TAC-ADCs [3]. While these architectures do provide better performance in terms of resolution and linearity, they also come with increased complexity, area and signal integrity issues. The delay line TDC, on the other hand is a simple, yet fast architecture that can provide sufficient resolution for many applications.

It is the motivation of this thesis to explore the design of a delay line TDC that can be used as a phase detector in a DPLL, with a focus on improving the dynamic range of a typical delay line TDC without compromising its conversion time or noise performance.

1.2 Objectives

1.2.1 General objective

Design a time-to-digital converter to be used as a phase detector in a phase-locked loop in 65 nm technology.

1.2.2 Specific objectives

- Review the state of the art of time-to-digital converters and phase-locked loops.
- Design a delay line TDC with a dynamic range of 20 ns and a resolution of 625 ps.

- Design a 5-bit digitally-controlled oscillator (DCO) with a tuning range of 1 GHz and a center frequency of 8 GHz.
- Simulate the designed TDC and DCO in a 65 nm CMOS technology using Cadence Virtuoso.
- Analyze the performance of the designed TDC in terms of resolution, linearity and power consumption
- Compare the performance of the designed TDC with other state-of-the-art TDCs.

1.3 Academic significance

The academic value of this research lies in the development of a new technique to extend the dynamic range of delay line TDCs and the establishment of a comprehensive design methodology for TDCs and DCOs in general. The findings of this research can be used as a reference for future research in the field of TDCs and DPLLs, and can also be applied to other areas of digital circuit design that require precise timing measurements. Furthermore, the design and simulation of the TDC and DCO in a 65 nm CMOS technology provides valuable insights into the challenges and considerations involved in designing high-speed mixed-signal circuits in advanced technology nodes.

1.4 Scope and limitations

This thesis focuses on the design and simulation of a delay line TDC to be used as a phase detector in a DPLL. The scope of the research is limited to the design of the TDC and DCO, and does not include the design of the complete DPLL. The performance of the TDC is evaluated in terms of resolution, linearity, noise and dynamic range, but does not include a detailed analysis of the impact of the TDC on the overall performance of the DPLL.

1.5 Thesis roadmap

This thesis is organized into seven chapters. Chapter 2 provides a theoretical framework for understanding the fundamental concepts and principles of PLLs and their building blocks. Chapter 3 reviews the state of the art of TDCs and DPLLs, highlighting the strengths and weaknesses of different architectures. Chapter 4 describes the design methodology used to develop the TDC and DCO, including the design choices and trade-offs made during the process. Chapter 5 presents the simulation results of the designed TDC and DCO, including a detailed analysis of their performance. Chapter 6 discusses the implications of the results and comments on future work. Finally, Chapter 7 states the conclusions.

References

[1] Stephan Henzler. *Time-to-Digital Converters*. Vol. 29. Springer Series in Advanced Microelectronics. Dordrecht: Springer, 2010.

[2] Antti Mantyniemi, Timo Rahkonen, and Juha Kostamovaara. "A CMOS Time-to-Digital Converter (TDC) Based On a Cyclic Time Domain Successive Approximation Interpolation Method". In: *IEEE Journal of Solid-State Circuits* 44.11 (2009), pp. 3067–3078. DOI: 10.1109/JSSC.2009.2032260.

[3] V. Kratyuk et al. "A Digital PLL with a Stochastic Time-to-Digital Converter". In: *2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.* 2006, pp. 31–32. DOI: 10.1109/VLSIC.2006.1705297.

Chapter 2

Theoretical framework

2.1 Phase-locked loop fundamentals

2.1.1 Basic concept

The PLL is a feedback control system that is used to synchronize the phase and frequency of an output signal to a reference signal. The reference signal is a very stable clock signal locally generated usually by a crystal oscillator. The PLL compares the phase of this reference signal with the phase of the output signal and generates an error signal that is used to adjust the frequency of the output signal.

There's two types of applications for PLLs broadly speaking: clock generation and clock data recovery. In the first case, the PLL is used to generate a clock signal that is synchronized to the reference signal. In the second case, the PLL is used to recover the clock signal from a data stream.

The PLL consists of three main components: a phase detector (PD), a loop filter (LF) and a voltage-controlled oscillator (VCO). The PD compares the phase of the reference signal with the phase of the output signal and generates an error signal that is proportional to the phase difference between

the two signals. The LF smooths out the ripple riddled signal generated by the PD, reduces the high frequency noise of the loop and provides a stable control voltage to the VCO. The VCO generates an output signal whose frequency is proportional to the control voltage. Lastly, the output signal is fed back to the PD, thereby creating a closed-loop system.

If the PLL also has a frequency divider in the feedback path, the system is also capable of generating a frequency that is a multiple or fraction of the reference frequency. This is useful in applications where a higher frequency is needed, such as in RF synthesizers.

2.1.2 Key PLL parameters

2.1.2.1 Phase noise / jitter

Jitter is defined as the time deviation (Δ_t) of a signal's transition edges from their ideal positions in time [1]. It is a metric of the outmost importance in the design of PLLs as it is a direct measure of the quality of the clock signal generated. Phase noise describes the same phenomenon in the frequency domain (the phase noise of a signal is the Fourier transform of the jitter) and is usually expressed in dBc/Hz. It is often used in most figures of merit (FOM) to benchmark PLLs [2].

2.1.2.2 Output frequency

It is defined as the range of frequencies that the PLL is capable of generating and can be determined by the VCO output range and the division ratio of the feedback frequency divider. This is a key metric in establishing the application of the PLL (e.g., clock generation or RF synthesizer) and it bears significant importance in the design process due to the tradeoff it has with the phase noise performance of the PLL.

2.1.2.3 Loop bandwidth

The closed-loop bandwidth of a PLL is the frequency range over which the PLL can track the phase/frequency variations of the input signal (from DC to either -3 dB from the open-loop gain or to the unity-gain frequency at 0 dB). It affects the acquisition time and phase noise performance of the PLL.

2.1.2.4 Noise bandwidth

It is the PLL's noise due to the filter's loop bandwidth, the higher frequencies of the voltage-controlled oscillator (VCO) are the major contributor to this effect [3], therefore the need to set a low enough bandwidth in order to suppress the most out of it. Even if reducing the loop bandwidth would be desirable to suppress in-band phase noise, it's also necessary to take into consideration that there is a tradeoff between the lock-in time and in-band phase noise.

2.1.2.5 Beat-note period

Defined as the time interval between successive "beats" (oscillations) observed at the phase detector output, in other words, is the inverse of the frequency difference's magnitude between the reference and feedback signals as shown in figure 2.2. Beat-note period is closely related to the acquisition time of the PLL and can be estimated by equation (2.1). This metric is a direct measure of the frequency mismatch between the reference and the feedback signals.

$$T_{beat} = \frac{1}{|f_{ref} - f_{feed}|} \tag{2.1}$$

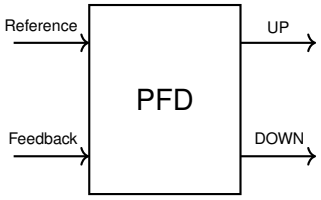

Figure 2.1: a) PFD block.

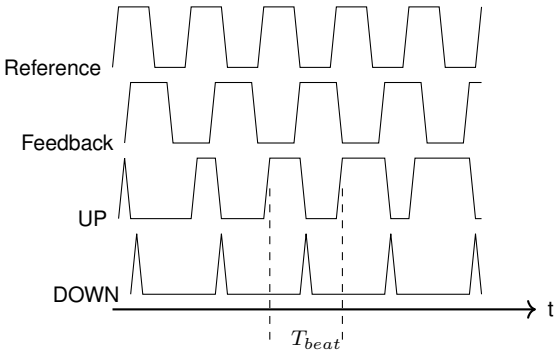

Figure 2.2: b) PFD output.

2.1.2.6 Lock-in time

Lock-in time (also called acquisition or settling time) is defined as the time for the PLL to lock on to the input reference phase and frequency within one beat-note period (T_{beat}) after a frequency change or startup of the system. This parameter measures the time required for the PLL to achieve the final phase lock once the VCO frequency is within the lock-in range. The lock-in time can be estimated with equation (2.2).

$$T_{lock-in} = \frac{ln(\epsilon)}{\zeta\omega_n} \tag{2.2}$$

where:

$\epsilon =$ acceptable phase error

$\zeta =$ damping factor

$\omega_n =$ natural frequency

2.1.2.7 Pull-in time

In contrast with the lock-in time, the pull-in time is the required amount of time for the PLL to initially acquire lock on to the reference signal from an

arbitrary starting frequency (when the VCO's initial frequency is far from the target).

2.1.2.8 Lock-in range

The frequency range within which the PLL locks to the reference frequency in one T_{beat}. Once the feedback signal is in this range, the PLL will enter lock state within the next T_{beat}.

2.1.2.9 Pull-in range

Range of frequency within which the PLL can acquire lock on to the reference signal once the VCO has the correct frequency and several beat-note periods have passed.

2.1.2.10 Pull-out range

The maximum allowed frequency or phase abrupt change applied to the reference signal beyond which the PLL unlocks.

2.1.2.11 Hold range

This parameter establishes the maximum theoretical frequency range for an input reference beyond which the PLL never locks [4]. Hold range is bigger than both lock-in and pull-in ranges.

2.1.2.12 Power consumption

An important parameter in measuring the performance of integrated circuits. Often used in most FOMs.

2.1.2.13 Spurs

Spurious tones are unwanted periodic signals that appear at the output spectrum of the PLL. Spurs are completely deterministic, thus they are distinct from phase noise which is random and spreads across the spectrum of

the signal.

2.1.3 Analog phase-locked loop

The principles of operation of the digital PLL stem from its analog counterpart, therefore this subsection describes the fundamental building blocks of the analog PLL as well as a simplified model of the system which allows for the circuit designer to analyze its behavior.

2.1.3.1 Linear phase model of the PLL

The PLL is a non-linear system, yet it can be approximated quite accurately into the linear system of figure 2.1 if certain assumptions are taken in to consideration [5]. The first of this assumptions is that the phase error (ϕ_e) must be small enough, meaning the PLL is near lock state. Another one is that the characteristic of the phase detector must be approximately linear and given by (2.3). Lastly, the VCO's tuning range must be within its linear region of operation (it is generally desirable to limit the variation of K_{VCO} to no more than 20%) like the frequency range between ω_1 and ω_2 in figure 2.4, only in this conditions the VCO can be considered a linear system and thus be described by (2.4).

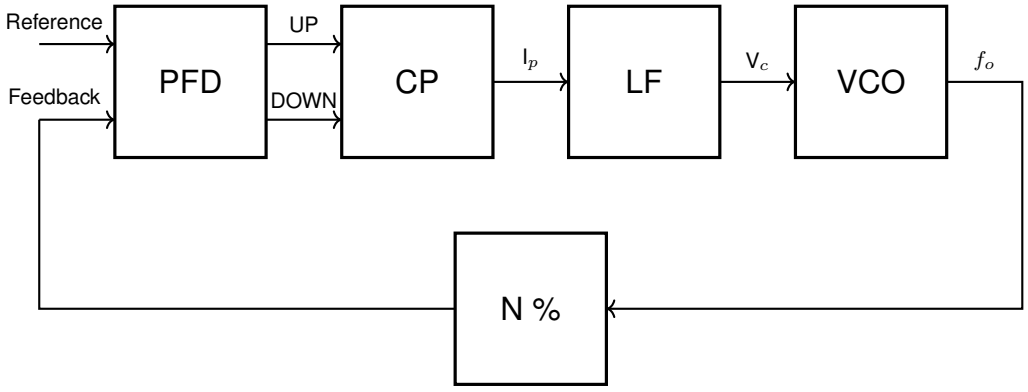

Figure 2.3: Block diagram of the linear phase PLL.

$$v_d = K_{PD}\,\phi e + \underbrace{V_{do}}_{\text{free running voltage}} \tag{2.3}$$

$$\Delta_{\omega_o} = K_{VCO}\,(V_c - V_{co}) \tag{2.4}$$

Each of the blocks in 2.3 has a transfer function associated with it, which can be expressed as follows: The PFD/CP combination is modeled as a current source with a gain of $I_p/2\pi$. The VCO is modeled as an ideal integrator with gain K_{VCO} and its transfer function is given by K_{VCO}/S. The feedback divider is modeled as a simple gain of $1/N$. Finally, the loop filter (LF) depends on the order of the filter and the type of components used.

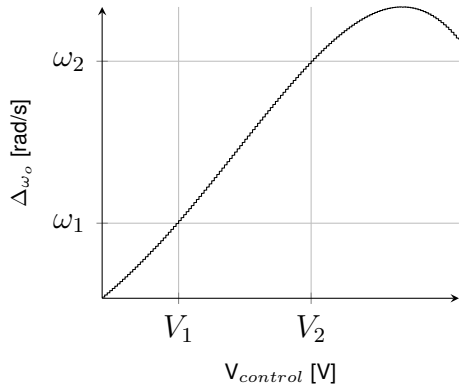

Figure 2.4: VCO characteristic example.

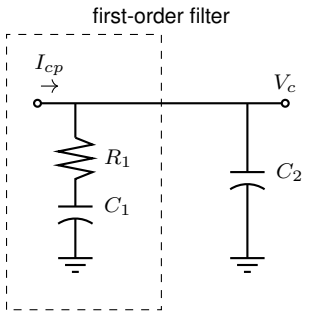

Figure 2.5: Second-order passive loop filter.

The most common implementation is a passive RC filter, which can be modeled either as a first-order or a second-order system. For the sake of simplicity in this analysis, the LF is modeled as a first-order circuit like the one enclosed by dashed lines in figure 2.5. The transfer function of the LF is then given by $(R_1SC_1 + 1)/(SC_1)$.

Knowing the transfer functions of each block, one can represent the PLL system as in figure 2.6. Then with the help of Mason's rule (2.5), the overall closed-loop transfer function (2.6) of the PLL can be obtained.

$$H(s) = \frac{\Sigma_{F.P}(1 - \Sigma_{loops\,not\,touching})}{1 - \Sigma_{loops}} \tag{2.5}$$

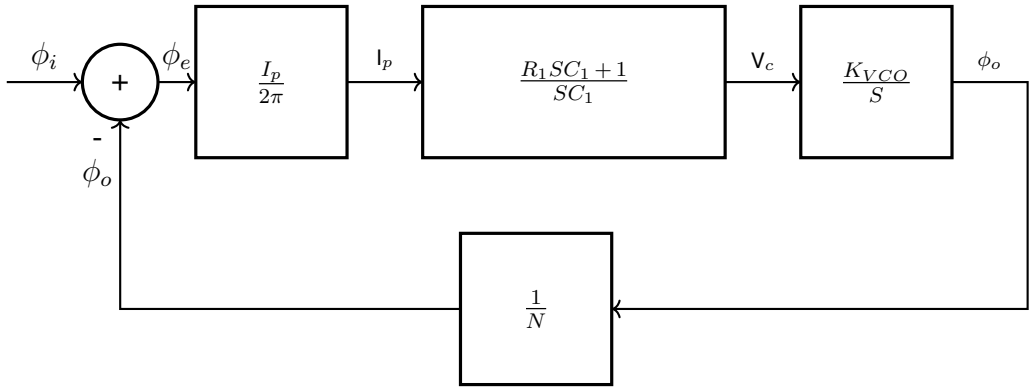

Figure 2.6: Block diagram of the linear phase PLL with each block transfer function.

$$A_{CL} = \frac{\frac{I_p K_{VCO}}{2\pi C_1}(R_1 S C_1 + 1)}{S^2 + S\frac{I_p}{2\pi}K_{VCO}R_1 + \frac{I_p K_{VCO}}{2\pi C_1}} \tag{2.6}$$

From the closed-loop transfer function (2.6) one can see that the PLL behaves as a second-order system with two poles at the origin (two ideal integrators), opening up the possibility of instability. In order to ensure stability, the PLL must be designed to have a sufficiently high phase margin (ϕ_m), and consequently an adequate damping factor (ζ) and natural frequency (ω_n). Both ω_n (2.7) and ζ (2.8) are crucial parameters in the design of the PLL, as they determine the bandwidth and transient response of the system.

$$\omega_n = \sqrt{\frac{I_p K_{VCO}}{2\pi C_1}} \tag{2.7}$$

$$\zeta = \frac{R_1}{2}\sqrt{\frac{I_p K_{VCO} C_1}{2\pi}} \tag{2.8}$$

It can be proven that the loop bandwidth (ω_u) is related with ω_n and ζ by equation (2.9) [6].

$$\omega_u^2 = (2\zeta^2 + \sqrt{4\zeta^4 + 1})\omega_n^2 \tag{2.9}$$

2.1.3.2 Phase frequency detector

A PLL that has a phase frequency detector (PFD) instead of just a phase detector (PD) is superior because it can track changes in both phase and frequency at the input signal. This approach provides robustness to the system because the PLL locks regardless of the initial value of the output frequency, therefore making the pull-in range equal to the VCO tuning range.

The most common implementation of a PFD is shown in figure 2.7. The operation of this circuit is as follows: When either of the two signals arrive with a rising edge to the corresponding D flip-flop CLK terminal a HIGH state is pass along to its output (Q), this result is maintained until the other signal's rising edge arrives at the second flip-flop because at that moment in time the AND gate would activate and reset both flip-flops. This behavior can be observed in figure 2.8.

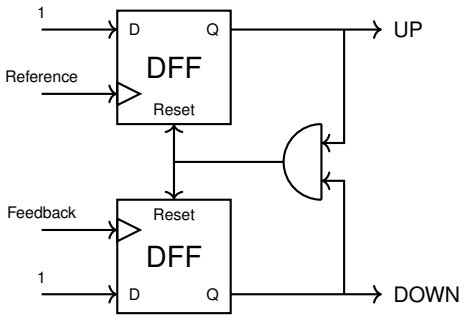

Figure 2.7: Three states PFD implementation.

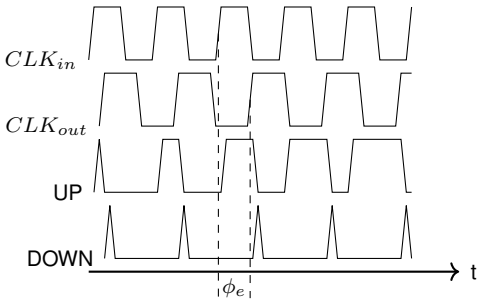

Figure 2.8: Transient response of the PFD.

2.1.3.3 Loop filter

The function of the loop filter is to filter out the high frequency noise from the VCO and to provide a stable control voltage to the VCO. The loop filter is usually a passive RC filter as the one in figure 2.5, but it can also be an active filter. The loop filter is a critical component of the PLL because it determines the bandwidth, the noise performance and the transient response of the PLL. The loop filter is usually designed to be a second-order low-pass filter, but it can also be a higher-order filter if needed. The values of the

components of the filter should be designed first and foremost to achieve stability at a given bandwidth, equations (2.8) and (2.9) are useful for this purpose.

The -3dB closed-loop bandwidth and the open-loop unity-gain bandwidth are fairly close to each other for $\zeta \geq 1$. It is for this reason that both quantities are often used interchangeably. In order to achieve a good transient response and noise performance, the loop bandwidth should be set to a value of $\omega_{ref}/10\,[rad/s]$ or less [6], where ω_{ref} is the reference frequency.

2.1.3.4 Charge pump

Charge pumps (CP) are circuits designed to either source or sink charge to/from a capacitor for a controlled amount of time [7]. There are many different implementations of charge pumps, but all of them consist of a current source, a switch and a capacitor (which is usually part of the loop filter). Most of the PLLs designed today are CPPLLs (or type-II PLLs) because the combination of PFD/CP/LF exhibit an infinite gain for a finite phase error, which means that for the control voltage to be finite, the phase error must be zero. This is a desirable property because it allows the PLL to have a very low phase error. Figure 2.9 shows the diagram of a charge pump, while figure 2.10 shows the schematic of a CMOS drain switched charge pump.

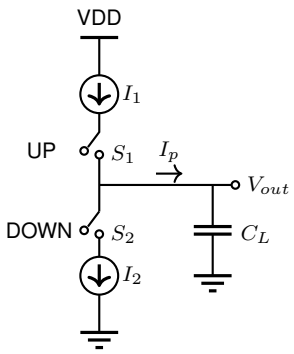

Figure 2.9: Charge pump schematic.

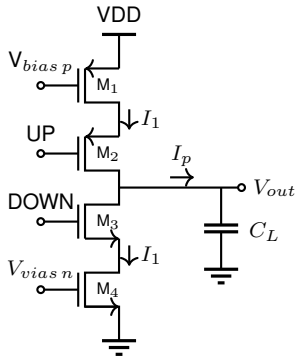

Figure 2.10: CMOS drain switched charge pump.

The greatest challenge in designing a charge pump resides in eliminating any current mismatch between I_1 and I_2 while maintaining a large output

voltage compliance. This task is particularly difficult to achieve because the channel length modulation in the MOSFETs causes the output current to be dependent on the output voltage (V_{DS} modulates the current). This means that both the output voltage compliance and the current mismatch performance trade off with each other.

The current mismatch ($I_{up} - I_{down}$) flows through the loop filter for the time it takes the PFD to reset both the UP and DOWN signals (approximately 5 gate delays) in every input period. The cumulative effect implies that $V_{control} \rightarrow \infty$, which cannot happen because the PLL attempts to lock and keep the output frequency constant [6]. consequently, the loop will settle with a steady-state phase error (Δ_t) and thus, the smaller current will last longer to keep the charge constant. This behavior of the CP is illustrated in figure 2.11.

The static phase error Δ_t can be approximated by equation (2.10) in the steady state because the rise and fall of V_{cntrl} must cancel each other out. The peak-to-peak amplitude ripple in V_{cntrl} on the other hand can be estimated with equation (2.11).

$$\Delta_t = \frac{I_{up} - I_{down}}{I_p} \cdot t_{res} \tag{2.10}$$

Where t_{res} is the time it takes the PFD to reset both the UP and DOWN signals, and I_p is the nominal charge pump current.

$$\Delta V_{cntrl} = \frac{I_{up} - I_{down}}{C_L} \cdot t_{res} \tag{2.11}$$

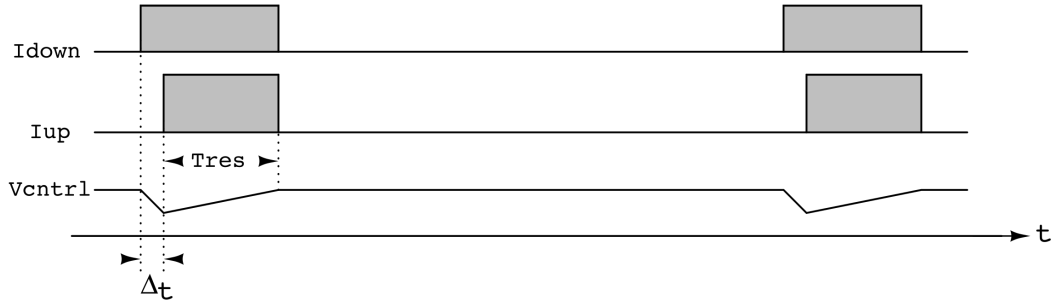

Figure 2.11: Charge pump transient response due to current mismatch.

There are multiple approaches to minimize this problem (some involving more complex topologies dealing also with the issue of clock feedthrough), yet the simplest one is to use wide enough (also large enough in the case of the current sources) transistors to minimize the channel length modulation effect. This solution is far from ideal, but is good enough for some applications.

2.1.3.5 Voltage-controlled oscillator

The VCO is the block that generates the output signal of the PLL. The frequency of the output signal is determined by the control voltage that the loop filter provides. It is a non-linear circuit that converts a voltage into a frequency (though it can be assumed to be linear for the conditions mentioned in the previous section). The VCO response looks like that of figure 2.13, where the gain of the VCO (K_{VCO}) is an important design parameter.

In general, it is desirable to have a relatively small (without compromising the stability of the PLL) K_{VCO} value so that any noise coupled to the previous stage does not translate into a large phase/frequency fluctuation at the output [6]. A good rule of thumb is to keep K_{VCO} below $0.1 f_m Hz/V$ where f_m is the center frequency of the VCO tuning range.

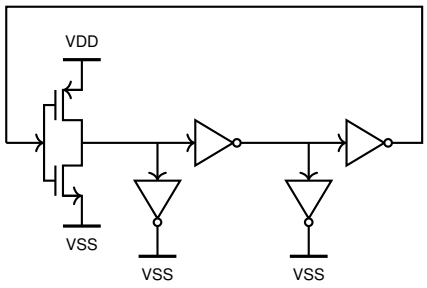

Figure 2.12: CMOS ring oscillator.

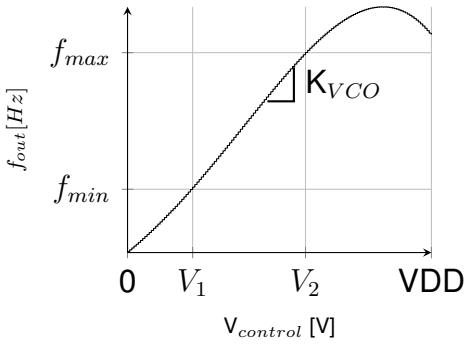

Figure 2.13: VCO characteristic curve.

There is a wide variety of VCOs, but the most common ones are the ring oscillator and the LC oscillator. The ring oscillator can be realized with CMOS inverters (Fig. 2.12) or with differential pairs, usually the former is preferred because is easier to implement, more compact and suffers less from mismatch. According to (2.12), the frequency of the ring oscillator is determined by the number of stages and the delay of each stage.

$$f_o = \frac{1}{2N\, t_p} \tag{2.12}$$

The analog frequency tuning is done by either changing the capacitance of the load (varactor implementation) or by reducing the strength of the inverters (via changing the pull-up or pull-down resistance), figures 2.14 and 2.15 show both tuning methods respectively.

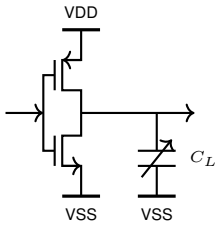

Figure 2.14: CMOS ring oscillator stage with varactor load tuning.

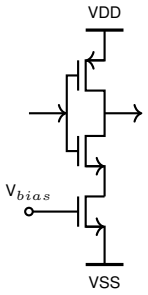

Figure 2.15: CMOS ring oscillator stage with resistive load tuning.

The LC oscillator is preferred for applications that require low phase noise and high frequency operation. This VCO is based on the resonator circuit (LC tank) shown in figure 2.16. The frequency of oscillation of an ideal LC tank is given by (2.13), in practice however, there's a need for an active device to compensate for the losses in the circuit and sustain oscillation. This losses, responsible for dampening the oscillation are primarily caused by the the parasitic series resistance of the inductor.

$$f_n = \frac{1}{2\pi\sqrt{LC}} \tag{2.13}$$

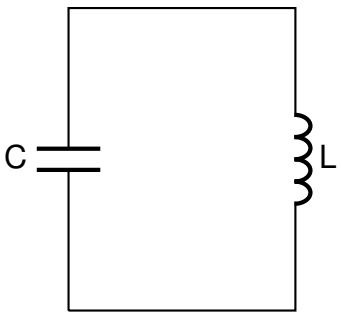

Figure 2.16: Ideal LC tank circuit.

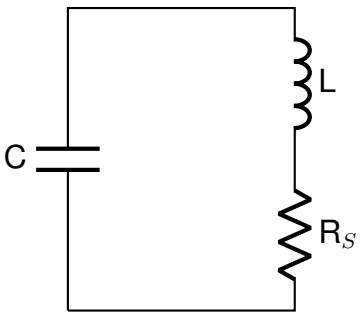

Figure 2.17: Real LC tank circuit.

The LC tank of figure 2.17 can be modeled as a parallel RLC circuit, where the inductor is modeled as an ideal inductor in parallel with a resistor (R_p) and an ideal capacitor; the equivalent circuit is shown in figure 2.18. Equating the transfer functions of both 2.17 and 2.18 yields equation (2.14), which can be used to compute R_p.

$$-L^2\omega^2 + R_S R_P + j\omega L R_S = 0 \tag{2.14}$$

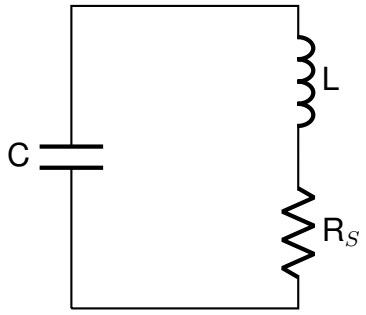

Figure 2.18: Equivalent LC tank circuit.

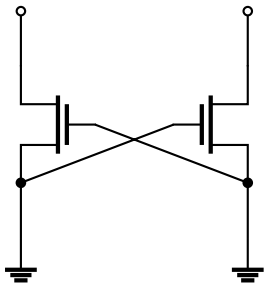

Figure 2.19: Cross-coupled pair.

The active circuit of figure 2.19 is known as a cross-coupled pair, it compensates for R_p with a negative resistance. In fact, by analyzing the small signal model of the cross-coupled pair the designer can arrive at equation (2.15), which allows to solve for the pair's transconductance (g_m). R_{pair} should be chosen $\ll R_p$ so that the former dominates in parallel with the latter (compensating for the losses).

$$R_{pair} = -\frac{2}{g_m} \tag{2.15}$$

The previous discussion focused on sustaining the oscillations of the LC tank, the control of the frequency is done by changing the capacitance of the tank. The most common way to do this is by using a varactor, such a device can be implemented with a MOSFET in the triode region and wiring it as shown in figure 2.20. The capacitance of the MOS varactor doesn't change linearly across all of the applied voltage. 2.21 shows the typical response of a MOS varactor in 65 nm, where it can be distinguished between three regions: Accumulation region, depletion and inversion. The region of interest is the inversion region, where the capacitance changes approximately linearly.

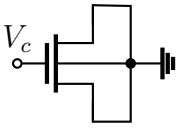

Figure 2.20: MOS varactor.

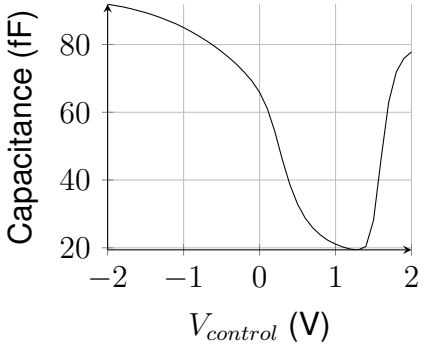

Figure 2.21: MOS varactor characteristic.

The tuning range can be extended by putting two varactors back to back on their gates. The final circuit of the cross-coupled pair VCO is shown in figure 2.22.

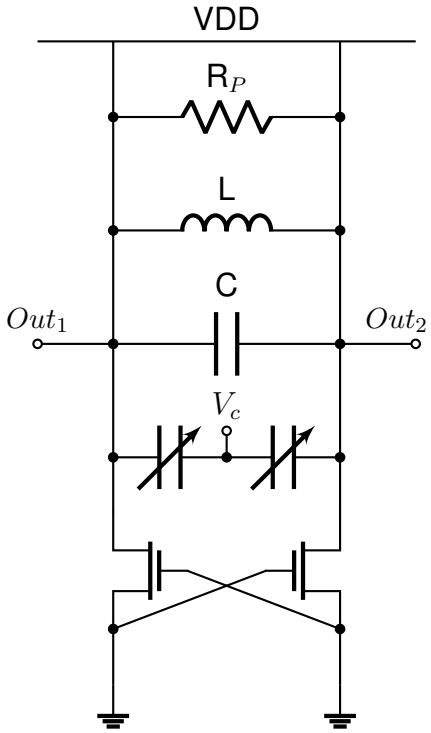

Figure 2.22: Cross-coupled pair VCO schematic.

2.1.3.6 Frequency divider

The frequency divider is a circuit that takes an input signal and produces an output signal with a frequency that is a fraction of the input frequency. The most common frequency divider is the flip-flop (fig. 2.23), which divides

the input frequency by two, although there are other types of dividers that can divide by any integer number. The frequency divider is used in PLLs to reduce the frequency of the input signal to a level that can be processed by the other blocks of the PLL (to match the frequency of the reference signal).

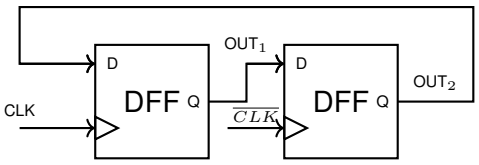

Figure 2.23: %2 frequency divider using D flip-flops.

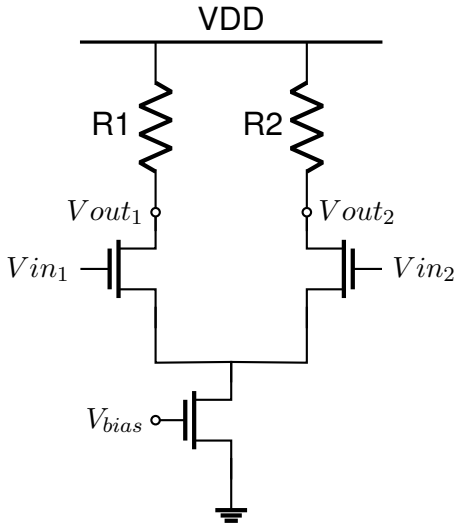

Figure 2.24: CML frequency divider.

Although the flip-flop is the most common frequency divider, there are other types of dividers that can be used in PLLs. One of the most relevant is the CML frequency divider (fig. 2.24), which is a type of frequency divider that uses current-mode logic (CML) to achieve high speed operation. CML dividers are typically used in high-speed applications, such as high-speed data communication systems, where the output of the VCO is too high to be processed by any regualr flip-flop. The CML frequency divider is based on the principle of current steering, where the input signal is used to control the current flowing through a branch of the differential pair.

A CML signal resembles a sinusoidal wave and has a lower harmonic content than a regular square wave, yet its power consumption is higher because of the constant current through the circuit due to the current source of the differential pair.

2.2 Digital phase-locked loop

One of the main problems of analog PLLs is that they are very sensitive to process, voltage and temperature (PVT) variations, which can cause the PLL to become unstable or to fail to lock. To address this issue, digital PLLs (DPLLs) have been developed [8]. DPLLs take advantage of the technology scaling (whereas analog PLLs suffer from it) to achieve a better performance and robustness in modern CMOS processes. To realize a circuit that performs well under this technology nodes it is necessary to avoid the analog domain as much as possible (at least on the critical path). The fundamental block that allows this task is the time-to-digital converter (TDC) [9].

The general architecture of a digital PLL is shown in figure 2.25. The DPLL consists of a TDC, a digital loop filter (DLF) and a digitally controlled oscillator (DCO). The basic operation of this three circuits will be explained in the following sections.

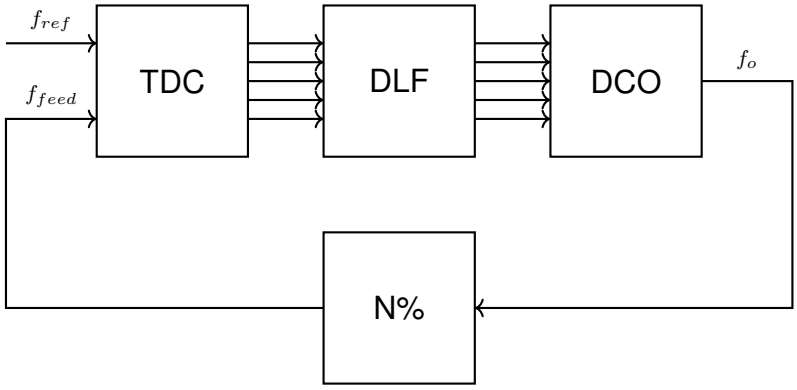

Figure 2.25: Digital PLL architecture.

2.2.1 Time-to-digital converter

The TDC substitutes the phase detector of the analog PLL, it measures the time difference (Δ_t) between two signals and converts it into a digital value that is proportional to that Δ_t, thus it transforms a signal in the time domain into a signal in the digital domain (avoiding the analog domain entirely). The TDC is a critical block in the DPLL, as it determines the resolution and

accuracy of the phase measurement.

The main metrics characterizing a TDC are the following [10]:

- Time resolution: It measures the precision of a time interval and represents the smallest time input that a TDC can correctly quantify.

- Dynamic range (also called acquisition-range): It is the maximum time interval that a TDC can measure.

- Conversion time: It is the minimum time required by a TDC to convert an input time interval into a digital output (time counted from the rising edge of the start signal). This parameter is very important for high-speed applications, as a minimum dead time is desired.

- Latency: It is the time interval between the rising edge of the stop signal and the time when the digital output becomes valid.

- DNL (Differential non-linearity): It is the difference between the actual step width and the ideal value of 1 LSB.

- INL (Integral non-linearity): It is defined as the maximum deviation of the actual input-output characteristic from the ideal transfer characteristic.

Figure 2.26 shows an example of DNL and INL in a TDC. The red line represents the ideal transfer function, while the blue line represents the actual transfer function of a TDC.

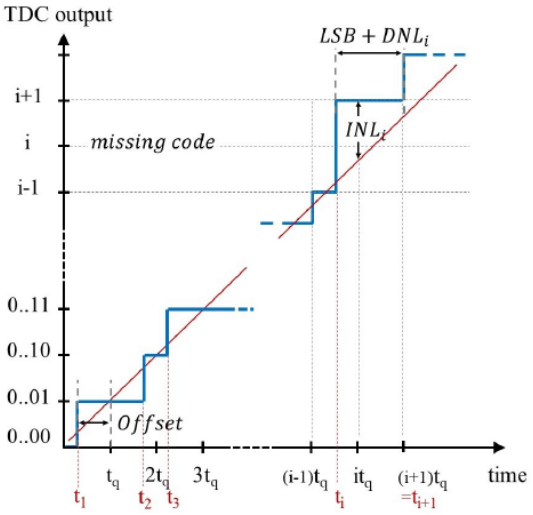

Figure 2.26: Example of DNL and INL in a TDC. Reprinted from "Time-to-Digital Converters: A literature review and new perspectives" by El-Hadbi, 2019 [10].

There are several types of TDCs, but the most common one is the delay line TDC, which uses a series of delay elements to measure the time difference between two signals. A delay line TDC is shown in figure 2.27 and its timing diagram is shown in figure 2.28.

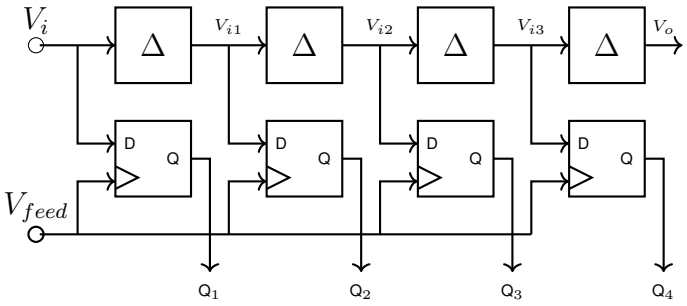

Figure 2.27: Delay line TDC schematic.

The TDC of figure 2.27 consists of a series of delay elements (Δ) that are used to measure the time difference between the input signals V_i and V_{feed}. This delay elements must be identical to each other and have a fixed delay which is the LSB of the TDC. The time difference between the two signals is measured by counting the number of delay elements that are crossed by the input signal V_i before it reaches the output signal V_o (and both signals are in phase after one complete cycle). The way this is done is by an array

of samplers (in this case D flip-flops but can be any type of sampler) that read the output of each Δ and hold it until the next clock cycle. Because the outputs of the samplers are sequentially read, the combination of them represent a digital word in the form of a thermometer code.

The number of delay elements determines the resolution of the TDC and can be calculated as a power of two (2^n), thus the number of bits of the TDC with the waveforms shown in 2.28 is given by $2^3 = 8$, after that number of stages the output of the nth delay element would be in phase with the input signal V_i.

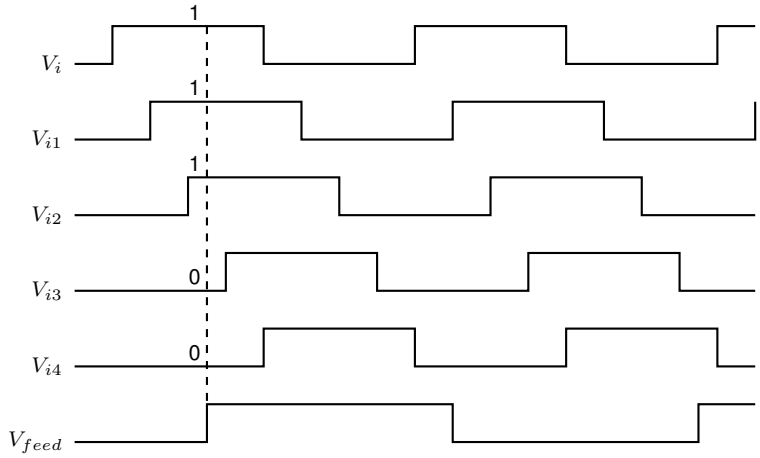

Figure 2.28: 3-bit delay line TDC timing diagram.

It is critically important to design the delay line TDC so that the delays are identical to each other, otherwise the TDC will measure the time difference incorrectly. Furthermore, the delay elements are usually implemented with a chain of CMOS inverters or differential pairs, both of which are sensitive to process variations, thus the complexity of designing this otherwise simple circuits. Several techniques have been proposed to address these challenges, such as calibration methods [11], layout optimization, and the use of delay locked loops (DLLs) to ensure the delay elements are matched even under PVT variations [12]. The next chapter reviews some of these techniques in detail.

2.2.1.1 Delay-locked loop fundamentals

DLLs are used to generate a stable clock signal that is robust to PVT variations due to negative feedback. The basic operation of a DLL is to compare the phase of an input clock signal with a delayed version of itself and adjust the delay of the delayed clock signal until they are in phase. This is done by using a voltage-controlled delay line (VCDL), a phase frequency detector (PFD), a charge pump (CP) and a loop filter (LF). Such an architecture is shown in figure 2.29.

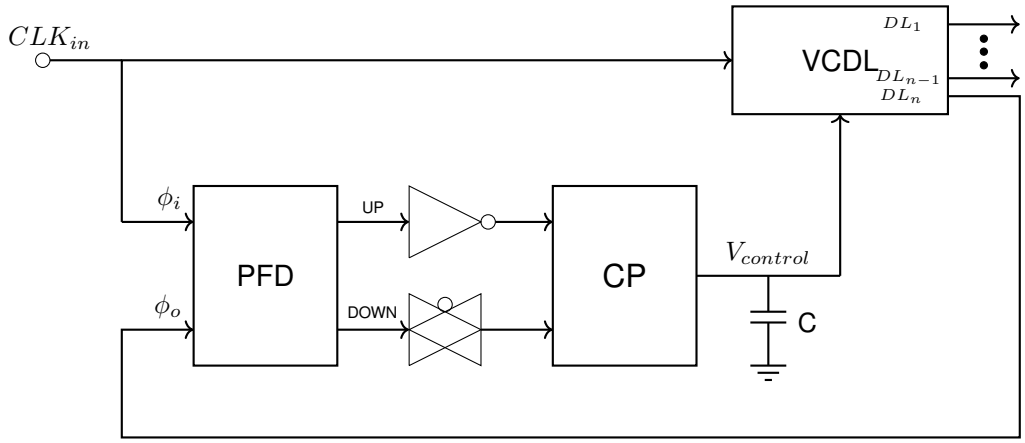

Figure 2.29: DLL architecture.

Since the output of the DLL is a delayed version of the input clock signal, it can be used to synchronize the clock signal with the phase of the input clock signal. The delay of the VCDL is controlled by the output of the CP, which is in turn controlled by the PFD. The PFD compares the phase of the input clock signal with the delayed version of itself and generates an UP or DOWN signal that is used to control the CP. The CP generates a control voltage that is used to adjust the delay of the VCDL until the two clock signals are in phase. This means that the DLL can be interchangeable with the delay line, thus the TDC can be implemented with a DLL.

2.2.2 Digital loop filter

It has already been mentioned that an analog PLL requires a loop filter to stabilize the feedback loop, the same is true for a DPLL. The digital loop filter (DLF) is used to filter the output of the TDC and to generate a control signal that is used to adjust the frequency of the DCO. The DLF is typically implemented as a low-pass filter or a proportional-integral (PI) controller.

Recalling the first order passive LF of figure 2.5 one can arrive at equation (2.16),where R_1 contains the proportional path and $1/S \cdot C_1$ contains the integral path.

$$H(s) = R_1 + \frac{1}{S \cdot C_1} \tag{2.16}$$

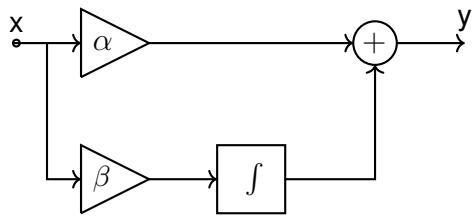

Figure 2.30: Conceptual filter using a proportional and an integral path.

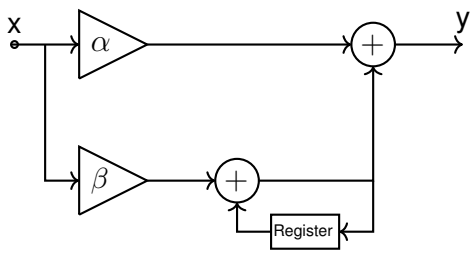

Figure 2.31: DLF implementation using an adder and a register.

Knowing the analog transfer function of the LF, a conceptual topology similar to that of figure 2.30 could be proposed, where $y = \alpha + \beta \int x\,dx$ for continuous-time quantities. To arrive at a digital counterpart, one could change x to D_{in} and y to D_{out}, rewrite this equation as the discrete-time expression (2.17), and take the Z transform of both sides to obtain the transfer function of the DLF (2.18), noting that a discrete-time integrator is represented by $1/(1 - z^{-1})$ [6].

$$D_{out}(k) = \alpha + \beta \sum_{k=0}^{\infty} D_{in}[k] \tag{2.17}$$

$$H(z) = \frac{D_{out}}{D_{in}}(z) = \alpha + \frac{\beta}{1 - z^{-1}} \tag{2.18}$$

Depicted in figure 2.31 is an implementation of the DLF that realizes the discrete-time integrator as an accumulator consisting of an adder and a register. The clock of such register is typically the same as the PLL reference.

2.2.3 Digitally controlled oscillator

Having already discussed the VCO fundamentals and a couple of implementations, the digitally controlled oscillator (DCO) can be viewed as a VCO that is controlled by a digital word instead of an analog voltage. Taking a look back at the VCO schematic of figure 2.22 it can be observed that the frequency is controlled by the bias voltage V_c at the gate of both MOSFET varactors. If the varactors are replaced by either a binary weighted capacitor array or current steering DAC, then the frequency of the circuit can be controlled digitally.

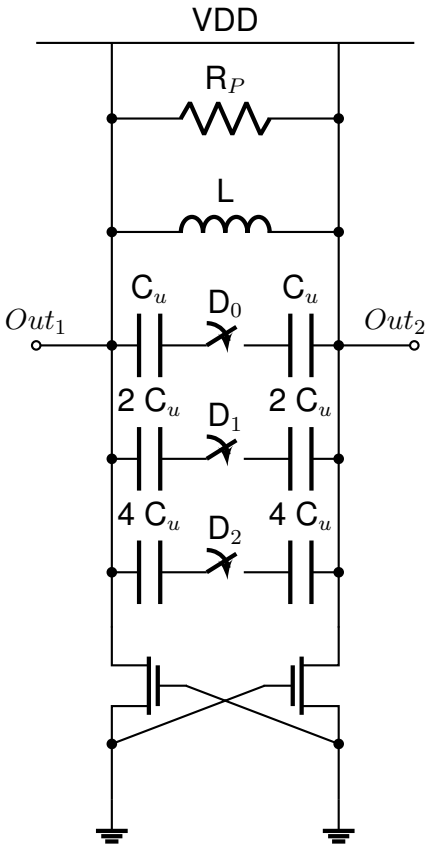

Figure 2.32: Cross-coupled DCO implementation with binary weighted capacitive array.

Figure 2.32 shows a cross-coupled DCO realization that makes use of a binary weighted capacitor array as the control mechanism. This binary weighted branches are controlled by switches that must be transmission gates (TG) because of the differential nature of the circuit. When the switches are closed, the capacitance of the branches is added to the total capacitance of the circuit, thus reducing the frequency of oscillation. It is desirable that the reverse occurs, so the digital word is typically inverted before being applied to the switches, that way if the digital word rises from one cycle to the next the frequency of oscillation will also rise.

It is necessary to pay attention to the design of this capacitive array because the parasitic capacitance of the switches and the interconnects can significantly alter the desired frequency step of the DCO, another factor to take into account is the mismatch between the capacitors, which can cause the

frequency steps to be uneven. To mitigate these issues minimum capacitor values should be avoided if possible.

References

[1] Mike Peng Li. *Jitter, Noise, and Signal Integrity at High-Speed*. Ed. by Mark Taub. Prentice Hall, 2007.

[2] Woorham Bae. "Benchmark Figure of Merit Extensions for Low Jitter Phase Locked Loops Inspired by New PLL Architectures". In: *IEEE Access* 10 (2022), pp. 80680–80694. DOI: 10.1109/ACCESS.2022.3195687.

[3] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill Education, 2003.

[4] Gennady A. Leonov et al. "Hold-In, Pull-In, and Lock-In Ranges of PLL Circuits: Rigorous Mathematical Definitions and Limitations of Classical Theory". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.10 (2015), pp. 2454–2464. DOI: 10.1109/TCSI.2015.2476295.

[5] Dan H. Wolaver. *Phase-Locked Loop Circuit Design*. Part of the Prentice Hall Advanced Reference Series. Englewood Cliffs: Prentice Hall, 1991, p. 262. ISBN: 0-13-662743-9.

[6] Behzad Razavi. *Design of CMOS phase-locked loops: From circuit level to architecture level*. Cambridge University Press, 2020.

[7] Gaetano Palumbo and Domenico Pappalardo. "Charge Pump Circuits: An Overview on Design Strategies and Topologies". In: *IEEE Circuits and Systems Magazine* 10.1 (2010), pp. 31–45. DOI: 10.1109/MCAS.2009.935695.

[8] Yuncheng Zhang, Dingxin Xu, and Kenichi Okada. "Digital Phase-Locked Loops: Exploring Different Boundaries". In: *IEEE Open Journal of the Solid-State Circuits Society* 4 (2024), pp. 176–192. DOI: 10.1109/OJSSCS.2024.3464551.

[9] Stephan Henzler. *Time-to-Digital Converters*. Vol. 29. Springer Series in Advanced Microelectronics. Dordrecht: Springer, 2010.

[10] Assia El-Hadbi, Oussama Elissati, and Laurent Fesquet. "Time-to-Digital Converters: A Literature Review and New Perspectives". In: *2019 5th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. 2019, pp. 1–8. DOI: 10.1109/EBCCSP.2019.8836857.

[11] Wassim Khaddour et al. "Calibration Methods for Time-to-Digital Converters". In: *Sensors* 23.5 (2023). ISSN: 1424-8220. DOI: 10.3390/s23052791. URL: https://www.mdpi.com/1424-8220/23/5/2791.

[12] Behzad Razavi. "The Delay-Locked Loop [A Circuit for All Seasons]". In: *IEEE Solid-State Circuits Magazine* 10.3 (2018), pp. 9–15. DOI: 10.1109/MSSC.2018.2844615.

Chapter 3

Literature review

Recent advancements in Phase-Locked Loop (PLL) design have increasingly incorporated Time-to-Digital Converters (TDCs) as replacements for traditional analog phase detectors. This shift enables higher resolution, better scalability in advanced CMOS nodes, and improved noise performance. This literature review examines key contributions from recent research (2022-2025), focusing on architectural innovations, calibration techniques, and performance metrics in TDC-based PLL designs. Hybrid and Digital-Intensive PLL Architectures Calibrated Dual-Referenced Interpolating TDC

A significant contribution in 28nm CMOS technology demonstrated a fractional-N Digital PLL (DPLL) employing a dual-interpolated TDC (DI-TDC) to address process-voltage-temperature (PVT) induced non-uniform resolution [$\mathbf{di\_tdc\_2023}$]. This design implemented foreground and background calibration to match TDC resolution to the Digitally Controlled Oscillator (DCO) period, achieving -17.5 dBc integrated phase noise at 570 MHz while occupying only 0.019 mm² active area. The architecture eliminates the need for delta-sigma modulators, thereby reducing quantization noise, though it requires complex calibration logic for wide frequency operation (475 MHz to 1.1 GHz). Fast-Locking Hybrid Architectures

A hybrid approach combining TDC with adaptive charge pump in 180nm CMOS demonstrated significant improvements in lock time [$\mathbf{hybrid\_pll\_2022}$]. This design used the TDC to quantize phase errors and dynamically enable a secondary charge pump, achieving 1.11 μs lock time while maintaining -98.07 dBc/Hz phase noise at 1 MHz offset. The architecture presents an effective compromise between analog and digital implementations, though it highlights the persistent trade-off between TDC resolution and power consumption in scaled technologies. Low-Power and IoT-Optimized Designs Calibration-Free RO-Based TDC

A notable innovation for IoT applications appeared in a 40nm ADPLL that reused a ring oscillator (RO) as both the DCO and TDC delay line [$\mathbf{ro\_tdc\_2023}$]. This self-referencing approach automatically tracks the DCO period with the TDC step, achieving 1.4 mW power consumption and -114 dBc/Hz phase noise at 1 MHz offset. The design's calibration-free operation significantly simplifies implementation for Bluetooth Low Energy (BLE) applications, though it requires careful RO tuning to mitigate injection-locking induced differential nonlinearity (DNL) errors. Ultra-Low Power Implementations

Recent divider-less ADPLL architectures have pushed power efficiency boundaries by employing embedded TDCs. A 65nm accumulator-based design achieved 0.53 mW power consumption and 0.87 ps RMS jitter through combination of a class-F^{-1} DCO with narrow-range TDC [$\mathbf{low\_power\_2024}$]. This approach demonstrates the potential for eliminating dividers to reduce quantization noise, though it demands exceptionally high TDC linearity. Advanced Calibration Techniques Noise-Shaping TDCs

Delta-sigma based TDCs have emerged as a powerful technique for quantization noise suppression [$\mathbf{noise\_shaping\_2023}$]. By shaping noise to higher frequencies, these implementations have demonstrated 10-15 dB improvement in in-band phase noise, becoming particularly valuable for high-performance RF synthesizers in 5G and Wi-Fi applications. PVT-Robust

Designs

A 12nm ADPLL implementation addressed PVT variations through digitally controlled delay inverters (DCDIs) that calibrate TDC buffers in real-time [pvt_robust_2024]. This approach maintains sub-picosecond resolution across process corners, though it introduces calibration overhead that increases design complexity. Emerging Topologies and Future Directions Successive-Approximation TDCs

SAR-TDCs have shown promise in reducing conversion latency and power consumption. A 28nm implementation achieved 543 fs RMS jitter at just 0.82 mW power [sar_tdc_2023], demonstrating the efficiency of binary search approaches in time-domain conversion. Time-Amplification Techniques

Time-amplifier-assisted TDCs represent a cutting-edge development, enabling sub-100 fs resolution by amplifying picosecond-scale time differences prior to digitization [time_amp_2024]. These techniques are particularly relevant for next-generation high-precision applications. Performance Comparison

Table 3.1 summarizes key metrics from state-of-the-art implementations:

Table 3.1: Performance Summary of State-of-the-Art TDC-PLLs

Metric	Value	Technology	Technique
Phase Noise	-114 dBc/Hz @1MHz	40nm CMOS	RO-based TDC
Power	0.38 mW	40nm CMOS	Accumulator-based
Jitter	320 fs RMS	65nm CMOS	Noise-shaping
Lock Time	1.11 µs	180nm CMOS	TDC-assisted CP
Area	0.019 mm^2	28nm CMOS	DI-TDC

Challenges and Future Directions

Current research faces several key challenges:

- The linearity versus power trade-off remains problematic for high-resolution TDCs

- Sub-10nm integration requires digitally intensive TDCs with self-healing capabilities

- Wideband applications demand multi-core architectures for GHz-range operation

Future work will likely focus on machine learning for adaptive calibration and 3D-integrated TDCs for next-generation systems.

TDC-Based PLLs with Full-Range Phase Acquisition ($\pm 2\pi$) 1. Hybrid PFD-TDC Architectures for Extended Range

A key innovation in CN101753142B addresses the limited pull-in range of conventional TDCs by integrating:

- A coarse phase-frequency detector (PFD) for initial frequency acquisition

- A fine-resolution TDC for phase tracking

- A frequency detector that converts frequency errors to digital codes 4

This hybrid approach enables operation across the full $\pm 2\pi$ range by:

- Using the PFD to handle large frequency offsets (beyond the TDC's native range)

- Switching to the TDC for high-resolution phase alignment once the PLL is near lock

- Employing a delay-line TDC with intermediate node taps to detect both phase and frequency errors 4

2. Dual-Loop Gain Techniques

The bang-bang PLL in 18 demonstrates an alternative method using:

- Two discrete loop gains (inner and outer loops)
- A quaternary phase detector that toggles between gains based on phase error magnitude
- Outer loop optimized for pull-in range ($\pm 2\pi$ capability)
- Inner loop optimized for jitter performance during lock 18

Key equations derived show the pull-in range depends primarily on the outer loop gain, freeing the inner loop for noise optimization 18. 3. State-of-the-Art Performance Tradeoffs

Recent designs face critical challenges:

- Power vs. Resolution: High-resolution TDCs (<1ps) in 4 consume more power during full-range operation
- Linearity: Delay-line mismatches in TDCs introduce nonlinearities that complicate wide-range operation 4
- Lock Time: Hybrid architectures in 418 show 20-30% faster locking than pure TDC-PLLs

4. Emerging Solutions

Innovations to enhance $\pm 2\pi$ capability include:

- Time Amplifiers: Stretch small time differences before TDC digitization 4
- Noise-Shaping TDCs: Improve in-band resolution while maintaining wide range 4
- Adaptive Calibration: Background calibration of TDC nonlinearities during operation 4

5. Comparative Analysis [1]

Table 3.2: Comparison of Full-Range ($\pm 2\pi$) TDC-Based PLL Architectures

Design	**Technology**	**Range**	**Resolution**	**Lock Time**
Hybrid PFD-TDC [**hybrid_patent**]	65nm CMOS	$\pm 2\pi$	$<$5ps	$<2\mu$s
Dual-Loop BB-PLL [**dual_loop**]	180nm CMOS	$\pm 2\pi$	Coarse/Fine	1.5μs
RO-TDC ADPLL [**ro_tdc**]	40nm CMOS	$\pm \pi/2$	1.2ps	5μs
SAR-TDC PLL [**sar_tdc**]	28nm CMOS	$\pm 2\pi$	543fs	3μs

Chapter 4

Methodology

4.1 Time to Digital Converter

4.1.1 specifications

The main specifications for the TDC are the following:

- Corner set up: TT, 60°C, 1.2 V.
- Static phase error (Δ_t) < 1%.
- Clock reference frequency: 100 MHz.
- Number of bits: 4.
- Acquisition range: -2π to 2π.

The TDC implements a 4-bit flash architecture for a time resolution of 625 ps (Δ) in each half of the clock period. The acquisition range is 20 ns, which corresponds to a phase range of -2π to 2π for a clock frequency of 100 MHz.

4.1.2 Architecture

The TDC architecture is based on the replica delay line presented in [1]. The core of the TDC consist of a voltage-controlled delay line (VCDL), a set of D flip-flops to sample the state of the VCDL, and a thermometer-to-binary decoder that process the sampled data to produce a workable digital output. This arrangement is used twice, once for the case when the feedback signal has a lower frequency (or delayed phase) than the reference clock, and the other for the case when the feedback signal has a higher frequency (or advanced phase) than the reference clock. The two outputs are then combined to produce the final output of the TDC. Such an architecture (Fig. 4.1) would be the conventional way to implement a TDC that reads both cases for the phase difference between the reference clock and the feedback signal.

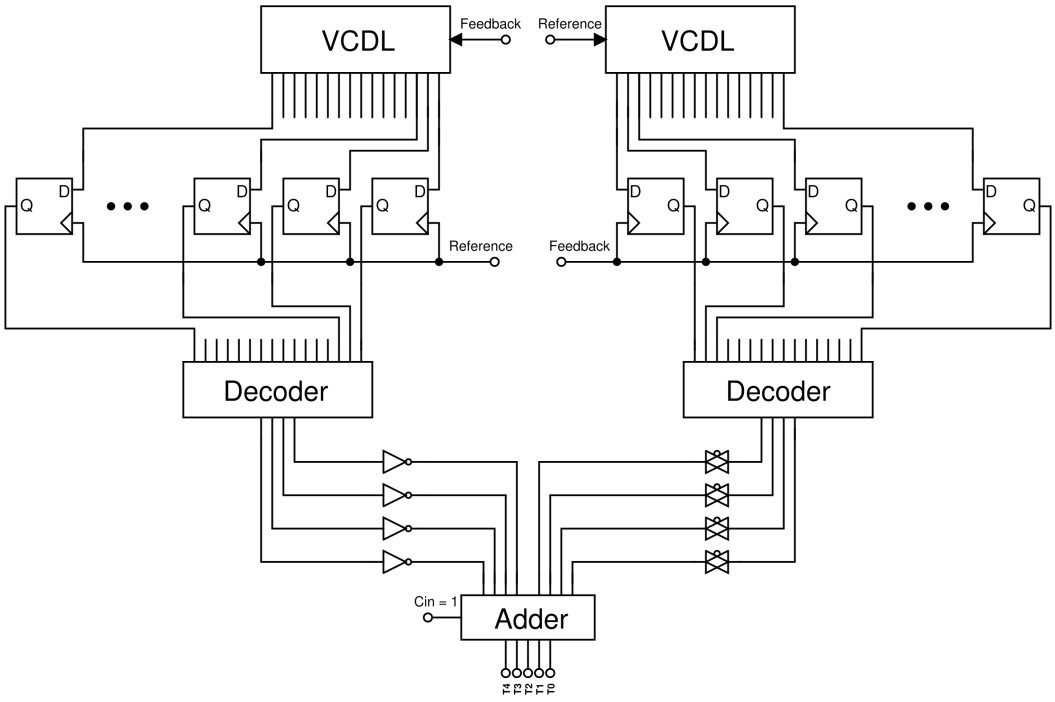

Figure 4.1: Conventional TDC architecture

The architecture of figure 4.1 has shortcomings, such as the fact that it does not allow for a phase difference of more than 2π between the reference clock and the feedback signal, meaning that the TDC will not be able to read the phase difference beyond half cycle for each clock period. This is

limiting for the PLL, as it would not be able to correct the phase difference if it exceeds π for each case. To overcome this limitation and achieve the desired acquisition range set in the specifications, the TDC architecture of figure 4.2 is proposed.

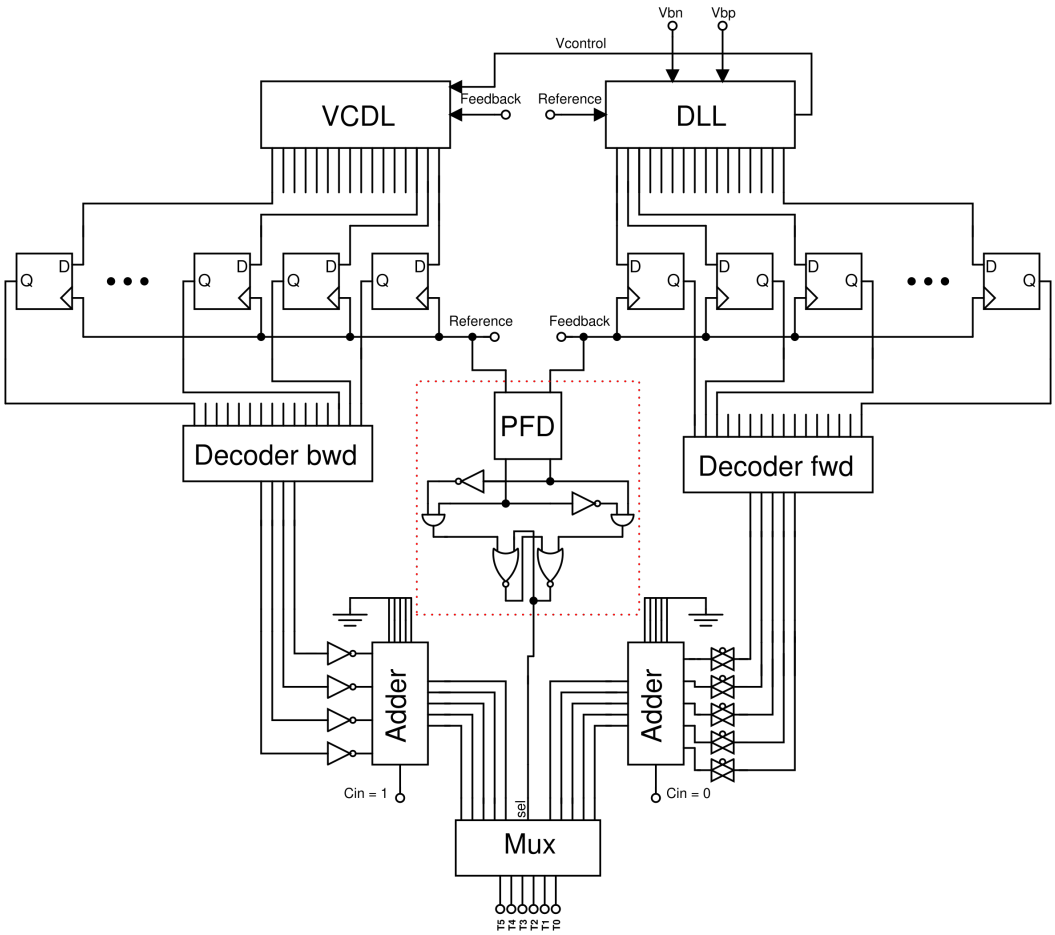

Figure 4.2: Proposed TDC architecture

The main differences between the conventional architecture and the proposed one are that the latter uses a DLL instead of a VCDL to implement the delay line, a slight modification to the thermometer-to-binary decoder, and a different way to combine the outputs of the two TDCs.

The DLL allows for a more robust and stable delay line as it is less sensitive to PVT variations. This is important for the PLL, as it needs to be able to operate under different conditions and still be able to correct the phase difference between the reference clock and the feedback signal.

The thermometer-to-binary decoder is modified to allow for the reading of both cases of the phase/frequency difference between the signals. This is the main way that the TDC is able to read a phase difference of more than 2π each cycle and as far as the author is aware, this is the first time that such a modification has been proposed in the literature.

Finally, the outputs of the two TDCs are combined using a multiplexer that selects the output based on the frequency state of the feedback signal. The selector needs to be able to determine whether the feedback signal is leading or lagging the reference clock, which is done by retrieving the information lost because of the decoder modification.

All of this changes done to the conventional TDC architecture will be further explained in the following sections of this chapter, where the design of the TDC will be presented in detail.

4.1.3 System behavior

Both architectures have the same principle of operation and each of their blocks have the same behavior. Both architectures have some sort of delay line that measures the phase difference between the reference clock and the feedback signal, a thermometer-to-binary decoder that processes the sampled data to produce a workable digital output, and both have the same number of D flip-flops to sample the state of the delay line which is equal to 2^n where n is the number of bits of the TDC.

The operation of the TDC of 4.1 is as follows: When the rising edge of the reference clock occurs and a new cycle starts, the circuit in charge of reading the case where the feedback signal is lagging (forward) with respect to the reference will start its operation, meaning that its VCDL will be reset and start counting the delay until the next rising edge of either the reference clock or the feedback signal occurs. If the former occurs first, then the circuit in charge of reading the case where the feedback signal is leading (back-

ward) will never start its operation and the TDC output will be the same as the last cycle. If the latter occurs first, then the backward circuit will start its operation and count the remaining delay until the next rising edge of the reference clock and the TDC output will be the subtraction of the forward output minus the backward output.

Because the decoders of the TDC of figure 4.1 are thermometer-to-binary decoders, their output is equal to 0 after half of the reference clock period time has passed, meaning the result of the TDC will be either the forward output or the 2's complement of the backward output depending on which of the half cycle of the clock period the feedback signal occurs. This feature is what allows the TDC to detect whether the feedback signal is leading or lagging the reference clock, but it is also what limits the TDC to a phase difference of $\pm\pi$ each cycle.

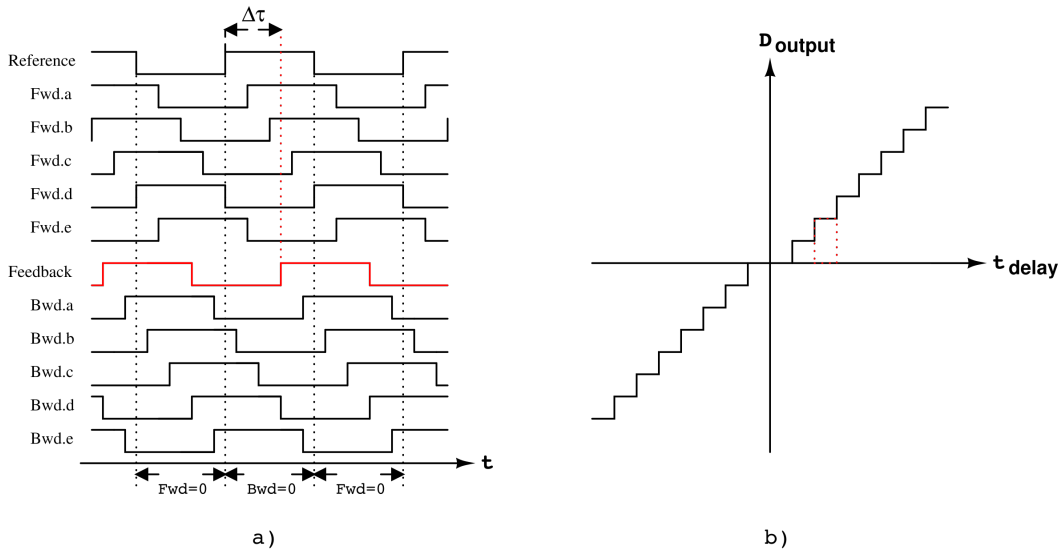

Figure 4.3: a) Conventional TDC timing diagram for a positive phase error, and b) its ideal characteristic.

Figure 4.3 further elaborates on this point, showing the timing diagram and the ideal characteristic of the conventional architecture. Notice that there is a 0 gain zone around zero delay where the TDC will not be able to detect small phase differences, also notice that the output will only take values

between 7 and -7 for a 4-bit TDC. Both of these issues are addressed in the proposed architecture.

4.1.4 Design

The design for the TDC proposed in this work is approached in a modular way, where each block is designed separately and then integrated into the final TDC. Even though differences exist between both paths (forward and backward) of the TDC, the blocks are the exact same for both, meaning that the design can be done in a single step and then replicated for each path with slight modifications.

4.1.4.1 DLL

The implementation of the DLL is that of figure 2.29, which is based on [2]. The DLL is composed of a voltage-controlled delay line (VCDL) that is used to measure the phase difference between the reference clock and the feedback signal, a charge pump that is used to control the delay of the VCDL, and a phase frequency detector (PFD) that is used to determine whether the feedback signal is leading or lagging the reference clock.

VCDL The delay elements of the VCDL are implemented using a series of CMOS inverters whose delay is controlled by two variable resistors at the source of each transistor (see Fig. 4.4). Each one of this elements must have a delay equal to 625 ps for a total of 10 ns (the period of the reference clock) after 16 of them. 625 ps is a relatively high amount of delay for a CMOS inverter of minimum dimensions in 65 nm, there are multiple approaches towards meeting such a high delay, but the preferable method from the perspective of optimizing the phase noise according to [3] is to increase the number of stages.

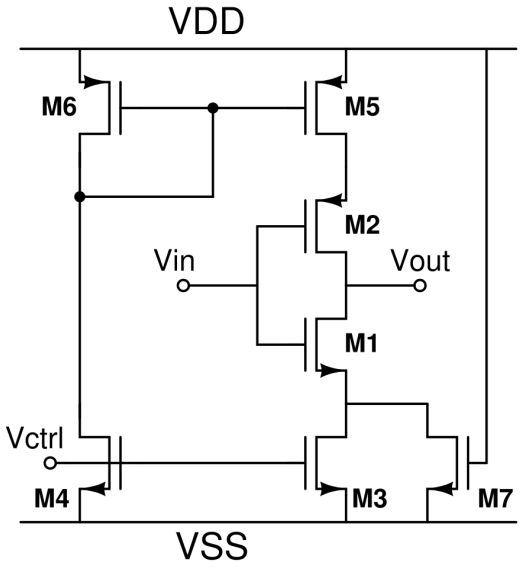

Figure 4.4: CMOS inverter with variable resistors at the source of each transistor to control its delay.

Therefore after simulating the circuit of figure 4.4 and measuring its delay (for a load of another identical inverter) it was determined that the number of stages necessary to achieve 625 ps in each delay element is 38. The dimensions for all 6 MOSFETs are annotated in table 4.1.

Transistor	**Length (μm)**	**Width (μm)**
M1	0.06	0.08
M2	0.06	0.215
M3	0.3	0.3
M4	0.3	0.3
M5	0.3	6
M6	0.3	6
M7	0.3	0.3

Table 4.1: Dimensions of the VCDL delay elements

PFD The PFD implemented for the DLL is the one presented in 2.7. The function of the PFD is to compare the phase and frequency of the reference clock and the feedback signal, and to generate the appropriate control signals (UP and DOWN) for the charge pump. The resettable D flip-flops are implemented using the latches shown in figure 4.5, where the output of each latch drives one input of the other.

The time it takes from the rising edge of the RESET signal to the point where both outputs (UP and DOWN) are low (T_{res}) is the dead zone of the PFD, which is a critical parameter as it determines the minimum phase difference that the DLL can detect and it is equal to approximately 64.34 ps for the design presented in this work.

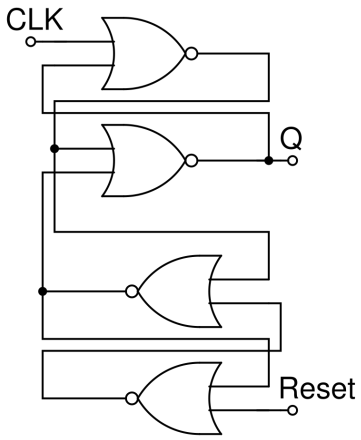

Figure 4.5: Simple resettable D flip-flop consisting of two cross-couped latches.

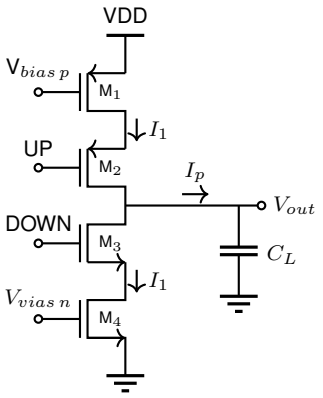

Figure 4.6: CMOS drain switched charge pump.

Charge pump The charge pump implemented for the DLL is the drain switched CP described in chapter 2 and shown again in figure 4.6. As stated before, the most challenging aspect of designing the CP is to minimize the current mismatch (Δ_I) between both branches, the approach taken in this work towards achieving this goal is to use relatively long current sources (M1 & M4) and low resistance switches (M2 & M3) to minimize the effect of the channel length modulation.

The static phase error (Δ_t) to which the DLL will lock onto is determined not only by (Δ_I) but also by the nominal charge pump current (I_p) as described in equation 2.10. Hence, with the purpose of minimizing (Δ_t) and making the TDC as accurate as possible, the current sources of the CP in figure 2.10 are designed to have a current of 150 μA, which is a good compromise between the static phase error, the power consumption and the area of the CP.

The specifications set Δ_t percentage to be less than 1%, which corresponds to a maximum Δ_I of 14.6 μA for a t_{res} of 64.34 ps according to equation (2.10). The designed dimensions that meet this requirements are shown in table 4.2. All that is left is to design the loop filter, which for the DLL it consist of a single capacitor (C_L) connected to the output of the CP. There is no need for a second order filter because the DLL doesn't have a VCO that contributes with a pole at the origin, making the choice of C_L less critical than in a PLL [2]. The considerations for choosing C_L are mainly the stability of the loop and the locking time, for this design a value of 3 pF is chosen as it provides a good trade-off between both parameters.

Transistor	**Length (μm)**	**Width (μm)**
M1	0.3	3.6
M2	0.06	1.2
M3	0.06	1.2
M4	0.3	3.6

Table 4.2: Dimensions of the charge pump transistors

4.1.4.2 Sampling flip-flops

The sampling flip-flops are implemented using the same resettable D flip-flops shown in figure 4.5. The number of flip-flops used is equal to 2^n where n is the number of bits of the TDC, which for this design is 16 (one extra flip-flop is added at the output of the last stage in order to have the same charge at the output of each stage of the VCDL). The flip-flops are connected in parallel and their clock input is driven by the feedback signal for the forward path and by the reference clock for the backward path. The reset input of all flip-flops is set to low because the control is done at the multiplexer block, not at the flip-flops.

4.1.4.3 modified thermometer-to-binary decoders

The standard thermometer-to-binary decoder for a 4-bit flash TDC has the truth table shown in table 4.3, where the inputs A-P are the outputs of the DLL sampled by the flip-flops. Figure 4.7 shows the timing diagram of the

n. delays	INPUTS															OUTPUTS			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1
6	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1
8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
9	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
10	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
11	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
12	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
13	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
14	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Table 4.3: Truth table of the standard thermometer-to-binary decoder for a 4-bit TDC.

DLL output, here A is the output of the last stage of the delay line (when exactly 10 ns have passed since the rising edge of the reference clock) and P is the reference clock itself (before the first stage of the delay line).

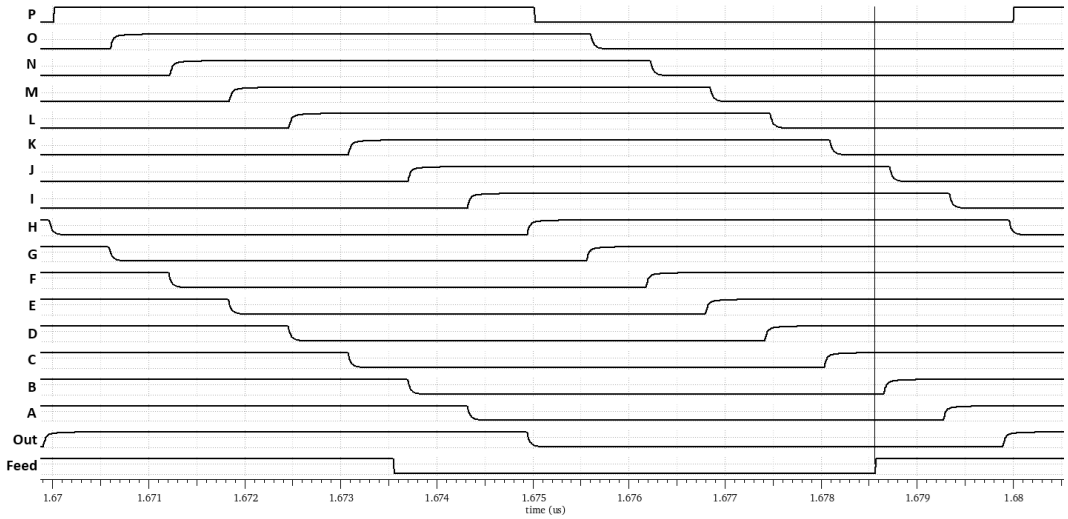

Figure 4.7: Timing diagram of the DLL output.

Two modifications are made to the decoder. First, each path of the TDC (forward and backward) has its own decoder. This is necessary to solve the issue of the 0 gain zone around zero delay (see Fig. 4.3) and is achieved by making the forward decoder output a minimum value of 1 instead of 0 when the delay is small. Such a modification essentially shifts the output

characteristic (Fig. 4.8) of the TDC up by 1, eliminating the 0 gain zone.

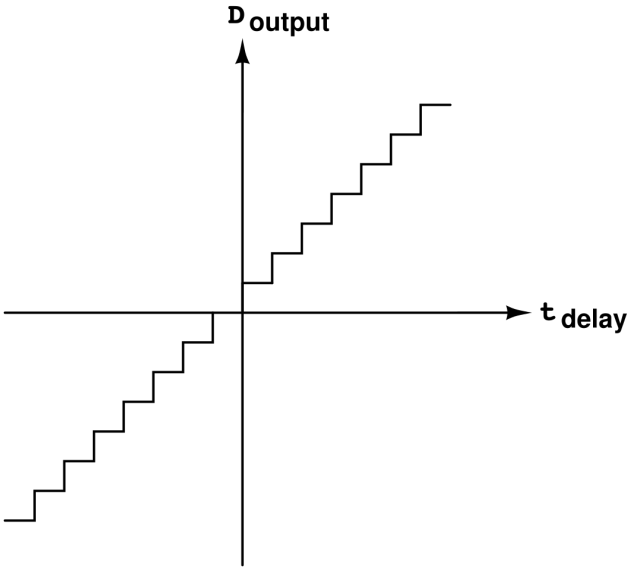

Figure 4.8: Ideal characteristic of the TDC with the modified decoders.

The second modification is what allows the TDC to read a phase difference of $\pm 2\pi$ each cycle. The modified truth tables for the forward and backward decoders are shown in tables 4.4 and 4.5 respectively.

N delays	INPUTS																OUTPUTS				
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	S4	S3	S2	S1	S0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
2	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1
2	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1
3	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0
3	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	0
4	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1
4	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1
5	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0
5	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0
6	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1
7	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	1
9	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0
9	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	1	0	1	0
10	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
10	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	1	0	1	1
11	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0
11	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0
12	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	1
12	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1
13	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0
13	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0
14	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1
14	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
15	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0
15	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0

Table 4.4: Truth table of the modified forward thermometer-to-binary decoder for a 4-bit TDC.

N delays	INPUTS																OUTPUTS				
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	S3	S2	S1	S0	
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	
1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	
1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	
2	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	
2	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	
3	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	
3	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	
4	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	
4	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	
5	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	
5	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	
6	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	
7	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	
8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0	
9	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	0	0	1	
9	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	
10	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	
10	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0	
11	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	
11	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	1	
12	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	0	
12	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	
13	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	0	1	
13	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	
14	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	
14	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	
15	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	
15	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	

Table 4.5: Truth table of the modified backward thermometer-to-binary decoder for a 4-bit TDC.

Recognizing that the only two conditions defining each output of both decoders are the high state of its decimal equivalent DLL output (e.g L=high for the number 4 in decimal) and the low state of the next number (e.g K=low), it is possible to obtain the bool equations for this otherwise large truth tables. The set of bool equations for the forward decoder are shown in 4.1, while those for the backward decoder are shown in 4.2.

$$\begin{aligned}
S_0 &= A\overline{B} + C\overline{D} + E\overline{F} + G\overline{H} + I\overline{J} + K\overline{L} + M\overline{N} + O\overline{P} \\
S_1 &= A\overline{B} + B\overline{C} + E\overline{F} + F\overline{G} + I\overline{J} + J\overline{K} + M\overline{N} + N\overline{O} \\
S_2 &= A\overline{B} + B\overline{C} + C\overline{D} + D\overline{E} + I\overline{J} + J\overline{K} + K\overline{L} + L\overline{M} \\
S_3 &= A\overline{B} + B\overline{C} + C\overline{D} + D\overline{E} + E\overline{F} + F\overline{G} + G\overline{H} + H\overline{I} \\
S_4 &= A\overline{P}
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
S_0 &= (O + \overline{P})(B\overline{C} + D\overline{E} + F\overline{G} + H\overline{I} + J\overline{K} + L\overline{M} + N\overline{O} + A\overline{P}) \\
S_1 &= (O + \overline{P})(A\overline{B} + D\overline{E} + E\overline{F} + H\overline{I} + I\overline{J} + L\overline{M} + M\overline{N} + A\overline{P}) \\
S_2 &= (O + \overline{P})(A\overline{B} + B\overline{C} + C\overline{D} + H\overline{I} + I\overline{J} + J\overline{K} + K\overline{L} + A\overline{P}) \\
S_3 &= (O + \overline{P})(A\overline{B} + B\overline{C} + C\overline{D} + D\overline{E} + E\overline{F} + F\overline{G} + G\overline{H} + A\overline{P})
\end{aligned} \tag{4.2}$$

4.1.4.4 PFD selector circuit

Because of the modifications done to the decoders, the TDC is now able to read a phase difference of $\pm 2\pi$ each cycle, but it has lost the ability to determine whether the feedback signal is leading or lagging the reference clock. This information is necessary for the PLL to be able to correct the phase difference, therefore a circuit that retrieves this information is needed. The approach taken in this work is to use another PFD identical to the one of the DLL but with additional circuitry to determine whether the feedback signal is leading or lagging.

Figure 4.9 shows the schematic of the PFD selector circuit. The additional circuitry consists of two AND and two NOT gates that process the UP and DOWN signals of the PFD and eliminate the glitches that occur when both signals are high. The resulting signals are then used as the set and reset inputs of a SR latch, whose output (Q) indicates whether the feedback signal is leading or lagging the reference clock.

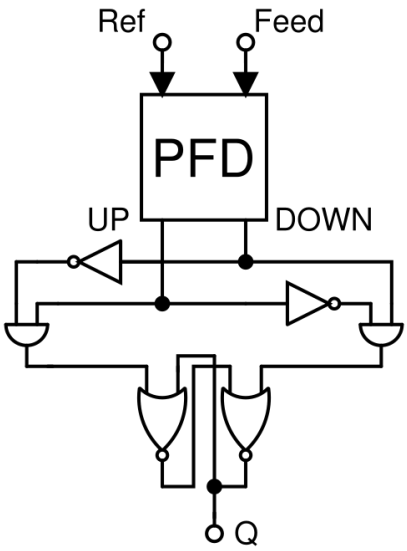

Figure 4.9: PFD selector circuit schematic.

4.1.4.5 Adder

The adder is only necessary for the backward path of the TDC, as it needs to do the 2's complement of the backward decoder output before it is sent to the multiplexer. however, because this block adds a significant amount of delay to the backward path, it is also added to the forward path (only adding zeros) to balance the delays of both paths.

The adder is implemented using a carry look-ahead (CLA) architecture to minimize the delay as much as possible. Because the forward path has an extra bit (S_4) that the backward path doesn't have, the adder is a 5-bit adder for both cases.

4.1.4.6 multiplexer

The multiplexer is a simple 2-to-1 multiplexer that selects between the outputs of the forward and backward decoders based on equation 4.3 and according to the output of the PFD selector circuit. If Q=1, then the feedback signal is lagging the reference clock and the output of the TDC is that of the forward decoder. If Q=0, then the feedback signal is leading the reference clock and the output of the TDC is that of the backward decoder.

$$\text{TDC}_n = Q \cdot \text{FWD}_n + \overline{Q} \cdot \text{BWD}_n \tag{4.3}$$

4.2 Digitally controlled oscillator

The digitally controlled oscillator (DCO) is implemented using an NMOS cross-coupled pair with an LC tank load. The frequency control is done by switching in and out different branches with binary weighted capacitors. The implementation of the DCO is shown in figure 4.10.

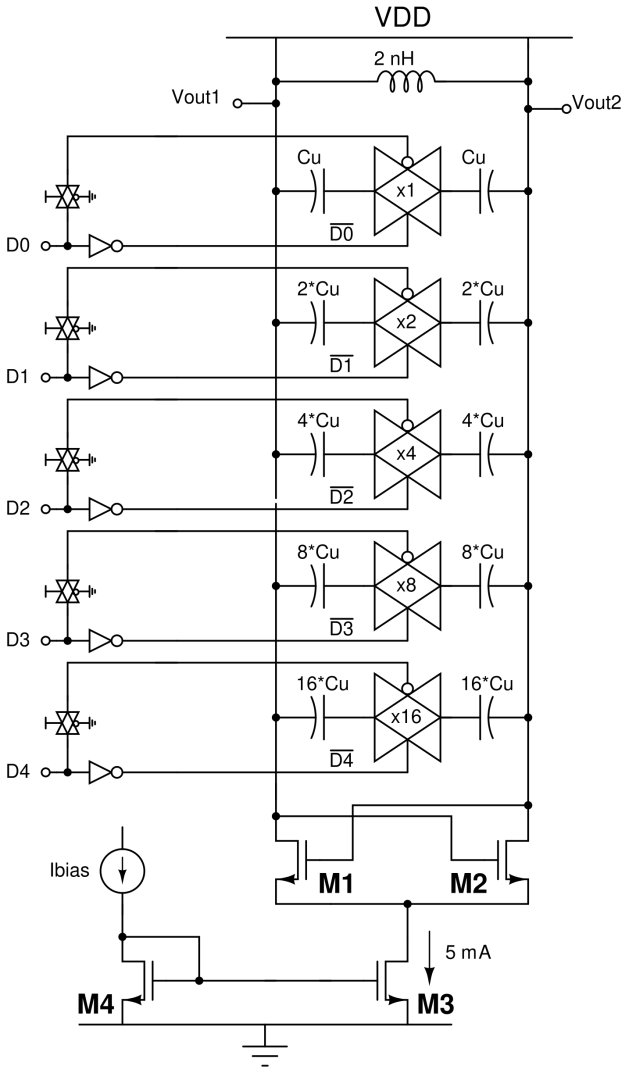

Figure 4.10: DCO implementation schematic.

Notice that there are CMOS inverters between the output of the digital loop filter and the DCO, this is done because the frequency of oscillation of the DCO decreases when the control voltage increases (adding capacitance) and it should be the opposite, otherwise the loop of the PLL would have positive feedback.

4.2.1 System behavior

4.2.2 Specifications

The specifications for the DCO are the following:

- Corner set up: TT, 60°C, 1.2 V.
- Center frequency: 8 GHz.
- Tuning range: 7.5 GHz to 8.5 GHz.
- Number of bits: 5.
- Frequency resolution: 31.25 MHz.
- Phase noise: < -80 dBc/Hz at 100 kHz offset from the carrier.
- Tail current = 5 mA.

All of the above specifications are set to be met at the nominal corner considering a layout routing parasitic contribution of 50 fF (so a proper layout for this design should have equal or less capacitance).

4.2.3 Design process

The design of the DCO is based on the LC resonator, which natural frequency of oscillation is given by equation (2.13). In order to correctly design the DCO, it is first necessary to characterize the inductor to know its parasitic series resistance and its quality factor (Q). Once both parameters are known, the equivalent parallel resistance (R_p) can be calculated with equation (2.15) to give dimensions to the cross-coupled pair. After that, the capacitor array can be designed. This procedure is explained in detail below.

4.2.3.1 Inductor

The inductor's Q is related to its equivalent parallel resistance (R_p) and can be approximated as $L\omega Q$. Considering the worst case for R_p (at the minimum frequency of the tuning range) and setting the inductance to 2 nH yields a Q of 13.3 for the inductor model used ($L\_SYCT30K\_RFVIL$). Equation (4.4) is the constraint that will determine the cross-coupled dimensions in this design.

$$Rp = L\omega Q = 2\pi \cdot 2x10^{-9} \cdot 7.5x10^{9} \cdot 13.3 = 1.17k\Omega \tag{4.4}$$

The inductor also has parasitic capacitance in parallel, this model adds approximately 38.66 fF at the resonant frequency (18.1 GHz) and shall be taken into account for an accurate design.

4.2.3.2 Cross-coupled pair

In order to compensate for the inductor losses and sustain oscillation, the negative resistance generated by the cross-coupled pair must be much less than 1.17 kΩ. The minimum transconductance necessary to achieve such a resistance is therefore 1.71 mS according to (2.15), but it will be designed to be 10 times bigger at 17.1 mS. The dimensions needed to achieve that *gm* are L = 120 nm and W = 9 μm.

By doing the small signal analysis of the circuit it can be shown that the cross-coupled pair adds an additional capacitance in parallel to the inductor and its value is given by equation (4.5). Only the gate-drain capacitance (C_{gd}) and the gate-source capacitance (C_{gs}) are considered for this analysis.

$$C_{pair} = \frac{1}{3}WL \cdot C_{ox} + \frac{5}{2}C_{ov}W \tag{4.5}$$

Where C_{ox} is the oxide capacitance per unit area (13.275 fF/μ^2 for this technology) and C_{ov} is the overlap capacitance per unit width (0.212 fF/μ^2).

Notice that the previous analysis assumes that the cross-coupled pair is perfectly matched and at the zero common-mode output voltage point to take into account the worst case for the noise performance.

For this design the cross-coupled pair adds an additional 9.55 fF in parallel with the inductor.

4.2.3.3 Capacitor array

The capacitor array is the method used to control the frequency of oscillation, turning on or off the branches containing the binary weighted capacitors to either increase or decrease the output frequency. Each branch connecting both output nodes of the DCO is composed of 2 identical capacitors in series with a bidirectional switch between them, the values of such capacitors are binary weighted multiples of a unit capacitor (C_u) like in the schematic of figure 4.10.

The unit capacitance is designed to be 9 fF, so accounting for all five branches there is a total of 558 fF in the array, though this is not the load in parallel with the inductor even if all branches are switched on. The analysis for both cases (either switched on or off) is done below.

4.2.3.4 Switches

The switches themselves add parasitic capacitance to the circuit,depending on whether theyre are turned on or off. Figure 4.11 a) shows the equivalent circuit of a single branch in figure 4.10 for the case in which the switch is turned on, while figure 4.11 b) shows the case for which the switched is turned off.

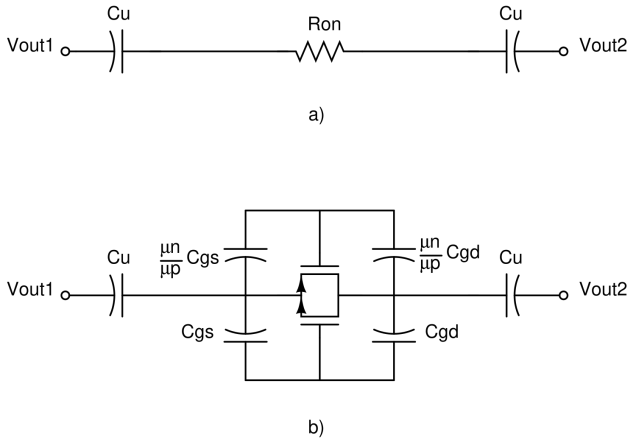

Figure 4.11: Capacitor array branch: a) turned-on and b) turned-off

If the switches are designed in such a way that the on-resistance can be

assumed to be negligible, then an approximation can be made for the equivalent capacitance of each branch for the case in which the switch is on. This is shown in equation (4.6), where the equivalent capacitance of each branch per unit capacitor is half the value of the unit capacitor (C_u) itself. Therefore, if all branches are turned on, the total capacitance of the array is 139.5 fF. If the analysis is done for the case in which the switches are turned off, then the equivalent capacitance per unit capacitor is dominated by the parasitic capacitances of the NMOS and the PMOS, so assuming $C_{gs} = C_{gd}$ and that the PMOS device is 2.5 times wider than the NMOS the expression for the off case is given by equation (4.7).

$$C_{on} \approx \frac{C_u}{2} \tag{4.6}$$

$$C_{off} \approx 1.75 \cdot C_{gs,NMOS} \tag{4.7}$$

4.2.3.5 Summary

Table 4.6 is a summary of all the different capacitance contributions for the DCO as well as its estimated frequency range (f_{max} & f_{min}). Table 4.7 shows the dimensions of all the transistors used in the DCO design.

Table 4.6: DCO capacitance contributions.

Contribution	Capacitance (fF)	Frequency (GHz)
Inductor parasitic	38.66	-
Cross-coupled pair parasitic	9.55	-
Capacitor array (all on)	139.5	-
Capacitor array (all off)	67.44	-
Routing parasitic	50	-
Total (all on)	237.71	7.3
Total (all off)	165.65	8.74

Table 4.7: DCO transistor dimensions.

Transistor	L (nm)	W (μm)
M1, M2 (cross-coupled pair)	120	9
M3 (tail current source)	300	100
M4 (biasing diode)	300	12.5
NMOS switch (unitary)	120	5.5
PMOS switch (unitary)	120	13

4.3 Digital loop filter

References

[1] C. Priyanka and P. Latha. "Design and implementation of time to digital converters". In: *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 2015, pp. 1–4. DOI: 10.1109/ICIIECS.2015.7193116.

[2] Behzad Razavi. "The Delay-Locked Loop [A Circuit for All Seasons]". In: *IEEE Solid-State Circuits Magazine* 10.3 (2018), pp. 9–15. DOI: 10.1109/MSSC.2018.2844615.

[3] Behzad Razavi. *Design of CMOS phase-locked loops: From circuit level to architecture level*. Cambridge University Press, 2020.

Chapter 5

Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque

cursus luctus mauris.

Chapter 6

Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque

cursus luctus mauris.

Chapter 7

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque

cursus luctus mauris.

Appendix A

Appendix

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.