Design of a Duty-Cycled Fractional-N ADPLL Based on Instantaneous Start-up LC DCO and High-precision DTCs

Yuan Gao

Master of Science Thesis

Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science

MSCCONFIDENTIAL

Design of a Duty-Cycled Fractional-N ADPLL Based on Instantaneous Start-up LC DCO and High-precision DTCs

Master of Science Thesis

by

Yuan Gao

For the degree of

Master of Science

in

Electrical Engineering

of

DELFT UNIVERSITY OF TECHNOLOGY

December 17, 2014

THIS THESIS IS CONFIDENTIAL AND MAY NOT BE MADE PUBLIC UNTIL DECEMBER 12, 2015

The work in this thesis was in close collaboration with NXP Semiconductor . Their cooperation is hereby gratefully acknowledged.

Printed in The Netherlands

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled

Design of a Duty-Cycled Fractional-N ADPLL Based on Instantaneous Start-up LC DCO and High-precision DTCs

by

Yuan Gao

in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Dated: December 17, 2014

Supervisor(s): prof.dr. Robert Bogdan Staszewski

ir. Frank Leong

Reader(s): assoc.prof.dr.ir. Leo de Vreede

assoc.prof.dr.ir. Nick van der Meijs

Abstract

This thesis deals with the design of a duty-cycled, fractional-N and low-noise Phase Locked Loop (PLL) used for Ultra-Wideband (UWB) applications in 40 nm process. This is the first ever Duty-Cycled PLL (DCPLL) that is designed with an LC-oscillator and breaks the noise record for DCPLLs by more than 1 order of magnitude. The LC oscillator has a tuning range from 4.91 GHz to 7.26 GHz. It features a phase noise level of -105.3 dBc/Hz at 1 MHz frequency offset at the operation frequency of 7.09 GHz and consumes 643 µW when the oscillator is active. Due the special architecture used in this design, the Phase Locked Loop (PLL) in this thesis can support fractional-N operation without difficulty and achieve a much better fractional-N resolution than its ring-oscillator counterpart while requiring little additional hardware and power cost. Furthermore, the latest All-Digital PLL (ADPLL) architecture and techniques are mapped and tailored for this first-ever LC-oscillator based Duty-Cycled All-Digital PLL (DC-ADPLL).

In this project, an inductor is first designed according to the operation frequency and tuning range specifications. Then, the inductor layout is modified and additional circuitry is added to enable instantaneous start-up. Next, the modified inductor layout is simulated with additional circuitry in ADS Momentum. The parameters of the inductor are extracted and modeled with a complex but accurate lumped model. A Digitally Controlled Oscillator (DCO) is designed and simulated with this tailored inductor. Specifications such as operating frequency, tuning range and power consumption are met as well as the instantaneous start-up requirement. Based on that, a tailored ADPLL is designed for duty-cycled operation. Other analog blocks such as DCO buffer, frequency divider, peak detector, comparator and Digital-to-Time Converter (DTC) are designed as well. The DC-ADPLL is described in Verilog. In addition to traditional blocks that can found in continuous-operation ADPLLs, multiple complex Finite State Machines (FSMs), coarse-fine DTCs as well as background DTC gain calibration blocks are also used. Finally, digital part and analog part are simulated together in Verilog Analog-Mixed-Signal (AMS) and the performance of this DC-ADPLL is evaluated.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The author would like to thank his supervisor Frank Leong for his guidance throughout the entire project. It is after countless discussions over various design choices that the author manages to complete the design. The author is also grateful to the tremendous amount of time he spent on reviewing this thesis to help the author make it meet with academic formatting standards.

Prof. Dr. Robert B. Staszewski, as an expert in All-Digital PLL design, gave the author quite some valuable input and the author is grateful to that as well.

Last but not least, the author would also like to thank Tarik Saric, who provided guidance on the design of the special inductor in this thesis, and Salvatore Drago, who shared his his experience in designing DCPLL.

Delft, University of Technology
December 17, 2014

Yuan Gao

Two roads diverged in a wood, and I –
I took the one less traveled by,
And that has made all the difference.

— Robert Frost [1874-1963]

Chapter 1

Introduction

1-1 Motivation

In the past few years, there has been increasing interest in UWB radio. One of the features that can be explored in UWB radio is that the signal pulse is constrained within a very short time window, which is easy to understand from the theory of Fourier transform, i.e., a signal whose energy spreads over a wide band is narrow in time domain. Because of this, the frequency synthesizer can be active during only this short amount of time and then be turned off to save battery power. This characteristic enables the frequency synthesizer used for UWB to operate in duty-cycled mode. To fully utilize this feature, there is a major difference in design constraint between continuous-operation synthesizers and duty-cycled synthesizers, that is, when woken up from off state, the internal states of voltage and current of duty-cycled synthesizers need to be stabilized within a negligible time compared with the duration of a data burst, usually on the order of a few tens of nanoseconds. Just like the ever on-going design optimization in continuous-operation synthesizers, there is also continuous demand of reduction in noise, power and area for duty-cycled synthesizers. Moreover, since the interest in Duty-Cycled PLLs (DCPLLs) began to rise just a few years ago, there is much more room for optimization.

1-2 Introduction to UWB radios

1-2-1 Definition of UWB

Not before long, Federal Communications Commission (FCC) released a new unlicensed spectrum between 0–960 MHz and 3.1–10.6 GHz. The new released spectrum between 3.1–10.6 GHz enables devices to transmit data over a very wide bandwidth. A UWB signal either occupies a bandwidth larger than 500 MHz regardless of fractional bandwidth or has a fractional bandwidth > 0.25, where fractional bandwidth is defined in Eq. (1-1) [1] and f_l and f_h are the upper and lower frequencies of the $-10\,\mathrm{dB}$ emission points.

$$FractionalBandwidth = \frac{(f_h - f_l)}{(f_h + f_l)/2}. \tag{1-1}$$

There are two existing but opposing approaches regarding utilizing UWB spectrum: single-carrier approach and multi-carrier approach. Both of them are within FCC's definition of UWB. In the single-carrier approach, the transmitted signal occupies 3.1–10.6 GHz while in the multi-carrier approach, the available spectrum from 3.1–10.6 GHz is split into multiple smaller and non-overlapping bands, each wider than 500 MHz, and the transmitted signal occupies only one of the sub-bands at a time. An illustration of signal waveforms in narrowband, multi-carrier UWB and single-carrier UWB is shown in Figure 1-1 [2].

1-2-2 Benefits of UWB

UWB radio has certain advantages such as minimal interference to existing narrowband spectrum, low probability of detection & interception and large channel capacity, which will not be explained in detail here. The one advantage that is most relevant to this thesis is the signal waveform in single-carrier UWB approach, shown in Figure 1-1. This special waveform shape brings two advantages. The first one is that the signal energy is concentrated within a very short window in time domain and thus the frequency synthesizer can be deeply duty-cycled to save power. The second advantage is that the operation time of analog circuitry and digital

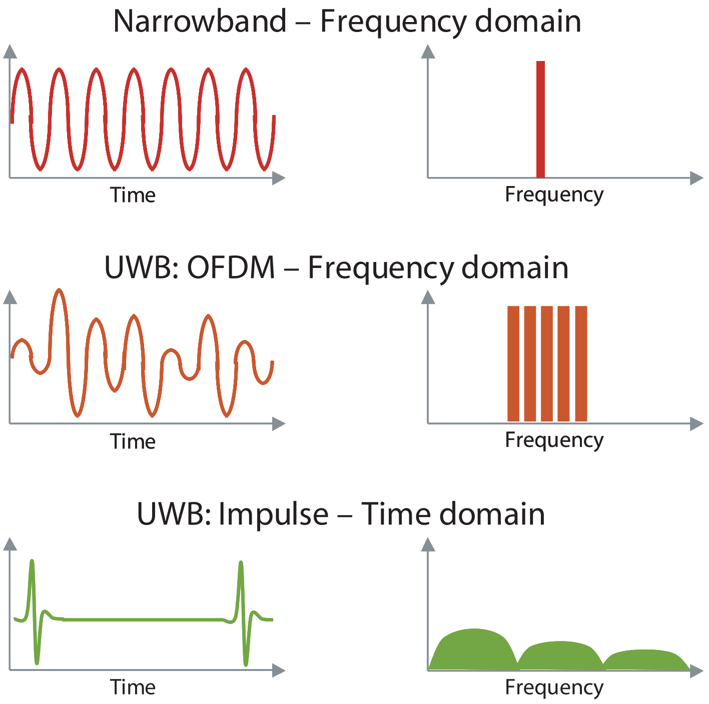

Narrowband – Frequency domain
Time
Frequency
UWB: OFDM – Frequency domain
Time
Frequency
UWB: Impulse – Time domain
Time
Frequency

Figure 1-1: Narrowband, multi-carrier UWB and single-carrier UWB in Time and Frequency [2]

circuitry in a UWB synthesizer can be made non-overlapping, so that interference between analog blocks and digital blocks can be reduced, which is illustrated in Figure 1-2 [2].

1-3 Introduction to UWB Transceivers

The structure of the receiver, in which the PLL of this thesis will be used, is shown in Figure 1-3. Among the two approaches in Ultra-Wideband (UWB), the single-carrier approach, IR-UWB, is adopted. The RF signal f_{RF} is first down converted by an LO oscillating at $\frac{2}{3}f_{RF}$, which brings the spectrum to $\frac{1}{3}f_{RF}$. After this, the spectrum at $\frac{1}{3}f_{RF}$ is further down converted by another clock at $\frac{1}{3}f_{RF}$, which is generated by dividing the previous LO by a factor of 2. Since f_{RF} varies between 3.1–10.6 GHz, the LO oscillating at $\frac{2}{3}f_{RF}$ needs to cover 2/3 of this range, which is 2.0–7.0 GHz. This range can be further separated into a lower band, 2.0–3.5 GHz, and a higher band, 3.5–7.0 GHz. Since the lower band can be covered by dividing the clock at the higher band by a factor of 2, the frequency synthesizer only needs to cover the higher band. However, this range is still too large for a single LC oscillator to

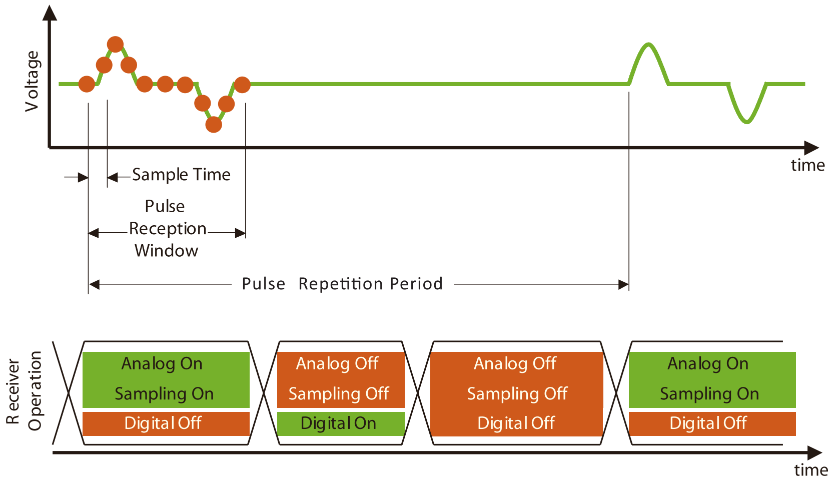


Figure 1-2: Separate operation of analog and digital processing circuitry for UWB radio [2]

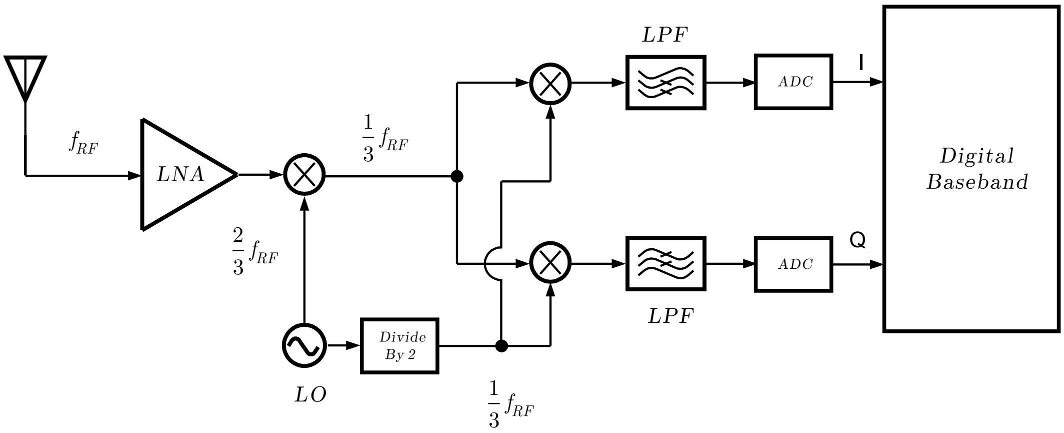


Figure 1-3: Structure of the UWB receiver [1]

cover. Thus, the specification of the tuning range in this thesis is set to 5.0–7.0 GHz to make a demonstration that LC oscillators can be used as duty-cycled oscillators, which have a much lower noise than ring oscillators. A second LC oscillator can be included in future work to cover 3.5–5.0 GHz to complete the tuning range required by the receiver.

1-4 Introduction to PLLs

In this section, introduction to Phase Locked Loops (PLLs) will be given. The introduction starts from old-fashioned analog $\Delta\Sigma$ PLLs, followed by digital $\Delta\Sigma$ PLLs and finally comes to

[1] This specific UWB receiver architecture is drawn according to a conversation about the system architecture with the author's NXP supervisor.

All-Digital PLLs (ADPLLs). While the Duty-Cycled All-Digital PLL (DC-ADPLL) in this thesis is closest in architecture to a (continuous) ADPLL, introduction to analog $\Delta\Sigma$ PLLs is included to show the history of PLL design. Drawbacks of analog $\Delta\Sigma$ PLLs, as well as the reason why digital-assisted structures become more favored, are introduced. Digital $\Delta\Sigma$ PLLs, as a structure that bridges analog $\Delta\Sigma$ PLLs and ADPLLs, are introduced as a transition for readers to understand ADPLLs. For readers that aim to get the basic background of the DC-ADPLL in this design, Subsection 1-4-1 and Subsection 1-4-2 can be skipped. For readers who are also interested to know why an ADPLL structure is chosen in this design, it is recommend to read Subsection 1-4-1 and Subsection 1-4-2 as well.

1-4-1 Introduction to Analog $\Delta\Sigma$ PLLs

It has been a major trend to assist or even replace analog circuitry with digital circuitry in the past decade due to the fact that analog blocks barely scale with technology while digital blocks, with every scaling of technology node, consume less power and can perform more functions within the same silicon area. Moreover, in deep submicron technology nodes, continuously decreasing supply voltage makes it harder and harder for analog intensive blocks to maintain its performance without resorting to new tricks.

This trend applies to PLL design as well. A block diagram of analog $\Delta\Sigma$ PLL is shown in Figure 1-4 [3]. Principle of the loop will be explained later together with digital $\Delta\Sigma$ PLL since they share the same loop structure. What can be noted at this moment is that charge pump, RC filter and VCO are used in analog PLL, all of which are analog intensive blocks. On one hand, the current of charge pump cannot scale down with technology because it is limited by signal-to-noise ratio while it is more and more tough to deliver the same current when supply voltage scales down with technology [4]. On the other hand, the area of RC filter does not scale down with technology and becomes more and more bulky compared with a digital filter approach [4]. Moreover, oscillation frequency of VCO is tuned through variable capacitor (varactor) by the output voltage of RC filter. A wide tuning range requires the voltage-to-frequency gain of the varactor to be high while a high voltage-to-frequency gain

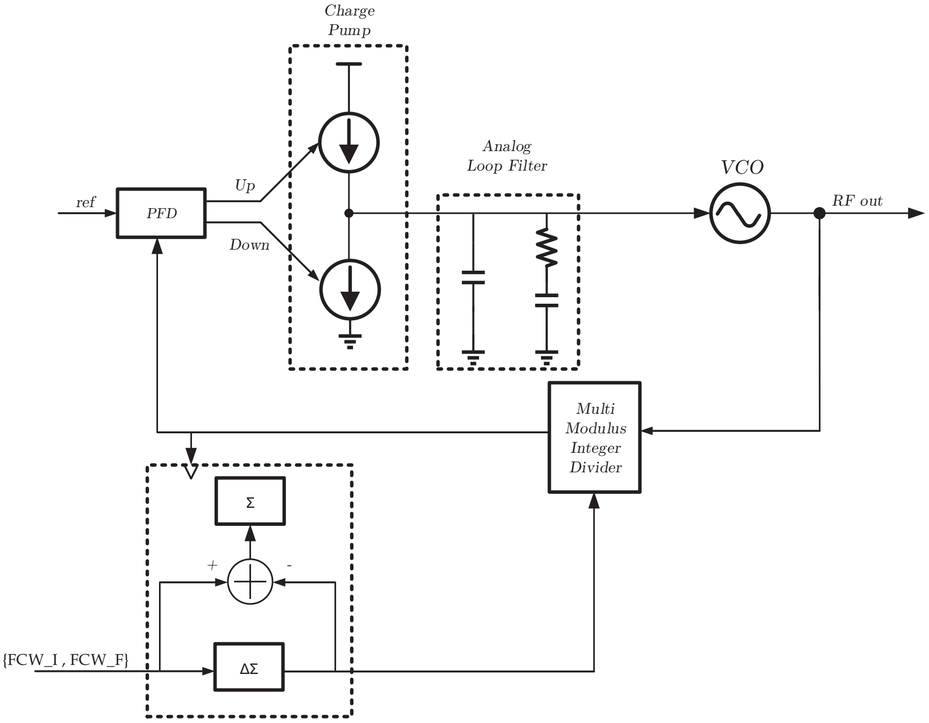


Figure 1-4: Block diagram of an analog $\Delta\Sigma$ PLL [3]

means a high gain from the noise on tuning voltage to output frequency as well [5]. Designers have to make a trade-off between these two metrics and this trade-off becomes even harder in systems where a wide tuning range is required. Due to the reasons explained above, a digital approach is favored over an analog one, especially in deep-submicron technologies.

1-4-2 Introduction to Digital $\Delta\Sigma$ PLLs

Regarding digital PLL, there are two categories: digital $\Delta\Sigma$ PLL, shown in Figure 1-5 [6], and All-Digital PLL (ADPLL), shown later in Figure 1-7 [7]. Detailed description of the simplified blocks in Figure 1-5 is shown in Figure 1-6. Digital $\Delta\Sigma$ PLLs are similar to analog $\Delta\Sigma$ PLLs regarding loop structure and therefore are introduced first. While these two structures look quite different at first glance, it is proven in [4] after thorough analysis that these two structures end up in achieving the same fractional spur level in theory. An elementary introduction to the most basic principle of the loop is given below.

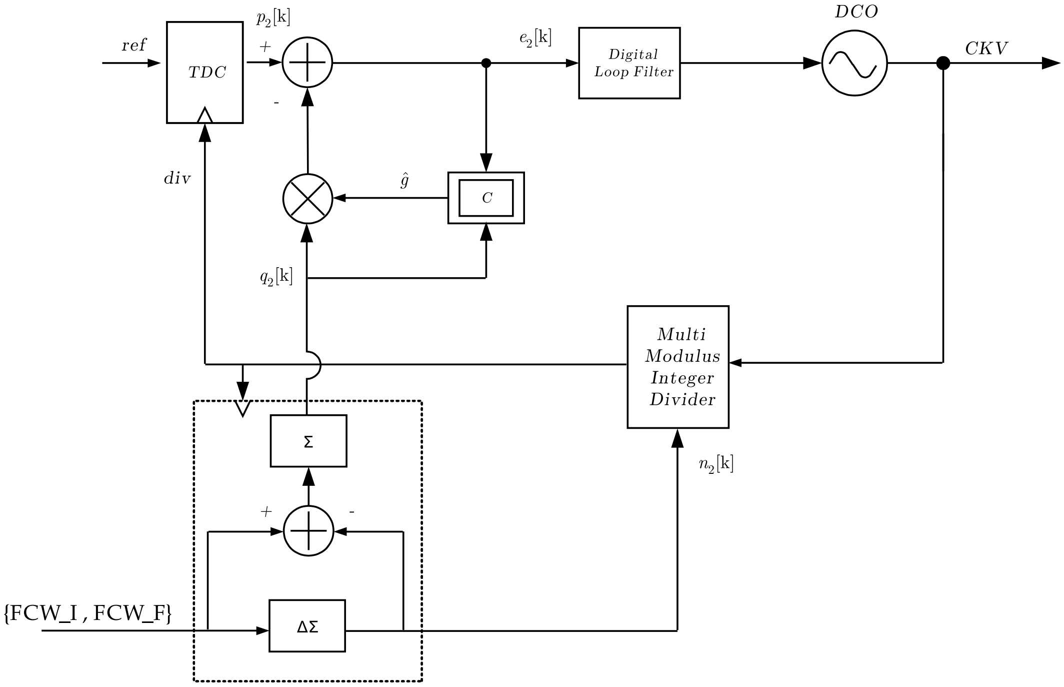
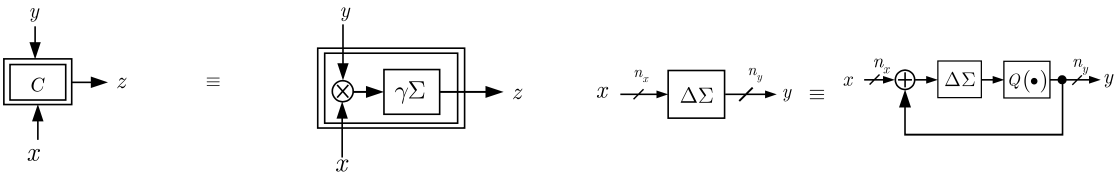


Figure 1-5: Block diagram of digital $\Delta\Sigma$ PLL [6]



(a) **(b)**

Figure 1-6: Description of blocks in Figure 1-5: (a) Correlator (b) First-order $\Delta\Sigma$ Modulator

Figure 1-5 shows a block diagram of digital $\Delta\Sigma$ PLL. The loop in Figure 1-5 receives a frequency control word (FCW), which defines the exact ratio between the desired frequency of RF output and that of reference clock.

Suppose FCW equals $(N + \alpha)$, where N is the integer part and α is the fractional part. Starting from the simple case that FCW is an integer number, thus α being zero, the modulus of divider is set to N so that frequency of divided CKV, div, will be set to f_{CKV}/N. If there is a deviation in either frequency or phase between div and ref, TDC block will be able to detect and quantize the early/late information from its two input clocks. It then sends a correction signal to increase/decrease the control word of Digitally Controlled Oscillator (DCO), which further decreases/increases the frequency of CKV, until both frequency and phase of div is aligned with those of ref, at which point the function of PLL transits from acquisition to tracking. The loop is not disabled after PLL acquires the desired frequency, though. The most fundamental reason for this is that noise is continuously generated within DCO [8] and if not continuously corrected, the phase of CKV will drift without boundary. Moreover, the loop is responsible for detecting and correcting frequency drift due to temperature drift as well.

Now, consider the case when FCW has not only an integer part but also a non-zero fractional part, which means α being non-zero. The straight-forward modification to make from system design point of view is to replace the original integer divider with a fractional divider whose modulus can be set to $(N + \alpha)$. However, as is pointed out in [4], from circuit implementation point of view, integer divider is the straight-forward solution and while fractional divider can be made, it requires additional circuitry and is inevitable to have a poorer performance than its integer counterpart. The general approach so far is to use an integer divider, whose modulus is kept at $(N + 1)$ for α portion of the time and N for $(1 - \alpha)$ of the time so that mean value of the modulus equals $(N + \alpha)$ in the long run. This effectively realizes a $\Delta\Sigma$ quantizer. In some cases, there is a tradeoff between update rate and power consumption, such as $\Delta\Sigma$ quantizers used for updating capacitor banks in DCOs. However, in the case of PLL, the highest possible update clock is reference clock, which usually runs at tens of MHz, thus power consumption is negligible compared with other blocks such as DCO and therefore is not a concern. Thorough and quantitative analysis is carried out in [4]. It is pointed out in [4] that fractional spur arises when a TDC with a much smaller number of bits than that of FCW is used in a PLL with a $\Delta\Sigma$-quantized integer divider.

One difference to note between the loop shown in Figure 1-5 and that shown in Figure 1-4 is that there is an additional correlator in the digital $\Delta\Sigma$ PLL to remove the correlation between the error fed back to loop filter and the quantization error from the $\Delta\Sigma$ quantizer. Thanks to digital implementation, this correlator can be implemented easily. Without this block, fractional spur will be much higher in an digital $\Delta\Sigma$ PLL than in an all-digital PLL [4].

Besides the additional correlator in digital $\Delta\Sigma$ PLL, the phase-frequency detector & charge pump (PFD-CP), RC filter and VCO in analog $\Delta\Sigma$ PLL are replaced by TDC, digital filter and DCO as well. Unlike their analog counterparts, the latter components scale with technology well and thus are more promising approaches in deep sub-micron technology nodes.

1-4-3 Introduction to All-Digital PLLs (ADPLLs)

Figure 1-7 shows a block diagram of an ADPLL introduced in [7]. Some details such as gain normalization blocks are omitted for an intuitive comparison with the other two structures.

While the previous digital $\Delta\Sigma$ PLL resembles its analog counterpart, ADPLL eliminates the multi-modulus divider that appears in both analog and digital $\Delta\Sigma$ PLL. The counter counts the integer number of CKV cycles within one reference cycle while TDC is responsible for quantizing the fractional cycle in fractional-N mode. The DC-ADPLL presented in this thesis can be considered as a duty-cycled counterpart of usually mentioned (continuous) ADPLLs.

1-4-4 Introduction to Duty-Cycled PLLs (DCPLLs)

While PLLs can be categorized into digital and analog, they can also be categorized in terms of whether they operate continuously or in duty-cycled mode. Literature on DCPLLs is much scarer than it is on Continuous-operation PLL (CPLL). Several previous works can be found in [9], [10] and [11]. While it may appear that one can effectively implement a DCPLL by

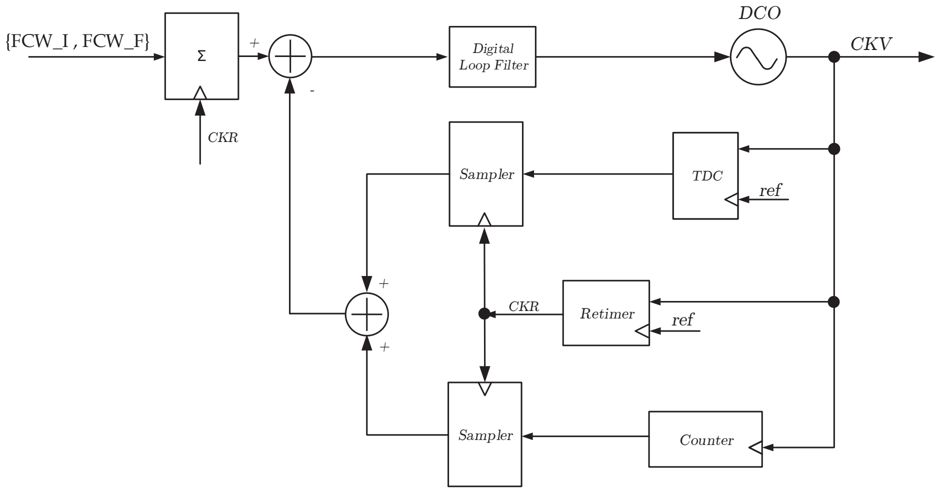


Figure 1-7: Block diagram of an ADPLL [7]

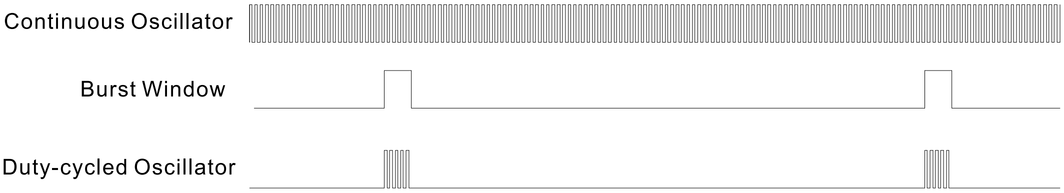


Figure 1-8: Waveform comparison between continuous and duty-cycled oscillator output

borrowing the same circuitry from a CPLLs and leave the duty-cycling control to higher system level, this is not true.

One of the fundamental differences between the design of CPLL and DCPLL lies in oscillator design. In applications where DCPLL is needed, the oscillator is only active during data bursts, which last for only a few tens of ns. Typical waveforms of continuous oscillator output, burst window in duty-cycled applications, and duty-cycled oscillator output are shown in Figure 1-8. The start-up time of an oscillator in a CPLL is a minor concern during its design process, because there is only one off-to-on transition during the entire time that the system is operative and thus is negligible. Design of continuous oscillators focuses on optimizing other metrics such as power and noise rather than the start-up time, which usually ends up with a long start-up time, prohibitive for duty-cycled applications.

While one of the obvious consequences due to excessive start-up time is that the oscillator needs to be turned on much earlier than the burst window, which means a larger power overhead, the not so obvious consequence is that the frequency of the oscillator deviates a lot, compared to the jitter in stable phase, from that specified by its control word during start-up phase. This excessive deviation in frequency leads to jitter and requires additional operation cycles to correct.

Consider the case, where the burst window has a duty-cycle ratio of 10%, then the bottom limit of oscillator duty-cycle ratio is 10% as well. If the oscillator has a long start-up time, e.g., one burst length, then we are looking at a 20% duty-cycle ratio of the oscillator. Furthermore, if the PLL needs 8 cycles to correct the excessive error during start-up phase, we are looking at an oscillator duty-cycle ratio of 100%, at which point there is no benefit in power consumption from duty-cycling. And if the PLL has a low bandwidth and it takes more than 8 cycles to correct the error from start-up phase, then this oscillator cannot be used in this duty-cycled mode altogether and is better left on all the time.

The conclusion is that if an oscillator is to be used in duty-cycled mode, it should have a short start-up time, not just because of the excessive power overhead, but also because of excessive jitter that requires additional cycles to correct. As can be seen in [9], [10] and [11], the utilized oscillators are all ring oscillators that feature a short start-up time and do not require additional cycles to correct the jitter from start-up phase. Furthermore, in cases when the jitter during start-up phase is negligible, there is even benefit from duty-cycling operation in terms of noise, because the RF output phase is re-aligned with the accurate reference phase through the trigger signal that is aligned with reference phase, which is similar to the operation of a delay-locked loop [12]. This is usually done by starting the oscillator from a specified phase instead of letting it establish its amplitude from thermal noise. Also, because of this feature, another difference between DCPLL and CPLL is that, while one of the functions of the CPLL in tracking mode is to keep cancelling the phase error accumulated over time so as to ensure the alignment between the phase of the RF output and the reference clock, the loop of the DCPLL does not need to do this, because the phase alignment is accomplished automatically by periodically triggering the oscillator with the reference clock [12], which will be discussed later in detail.

1-5 Introduction to Wiener Processes

Since the noise in DCPLLs will be heavily analyzed in the time domain, based on theory of stochastic processes, it is better to introduce first the one special process that is most relevant to noise analysis in PLLs, which is the one-dimensional Wiener Process [13, Ch. 4].

A process $W(t)$ satisfying the following properties is called 'Wiener Process':

1. $W(0) = 0$.
2. $W(t) - W(s)$ has a normal distribution with mean 0 and variance $\sigma^2 \cdot (t-s)$ for $s \leq t$.
3. $W(t_2) - W(t_1)$, $W(t_3) - W(t_2)$, ..., $W(t_n) - W(t_{n-1})$ are independent for $t_1 \leq t_2 \leq ... \leq t_n$.

Here, σ^2 is some positive constant. While the definition above is non-intuitive, a Wiener Process can be considered as a continuous-time limit case of a discrete-time random walk [14, Ch. 11]. Further, a white noise process $w(t)$, a process with a flat spectrum density, can be defined by Wiener Process $W(t)$ in Eq. (1-2) [15], where $f(t)$ can be defined in a number of ways, e.g., $f(t) \equiv 1$.

$$\int_a^b f(t)w(t)\,dt = \int_a^b f(t)\,d\,(W(t)). \tag{1-2}$$

According to one of the properties of Fourier transform shown in Eq. (1-3a) [16, Ch. 4], we have Eq. (1-3b). Here $X(j\omega)$ denotes the Fourier transform of a signal $x(t)$ and $\mathcal{F}$ denotes the operation of Fourier transform. Note that the differentiation of a Wiener process is a white noise process, whose spectrum is uniform across all frequencies, shown in Eq. (1-4), where S_0 denotes a positive real random constant. Thus, the spectrum of a Wiener Process, $S(\omega)$, can be derived from Eq. (1-3) and Eq. (1-4), shown in Eq. (1-5).

$$\frac{dx(t)}{dt} \overset{\mathcal{F}}{\longleftrightarrow} j\omega X(j\omega). \tag{1-3a}$$

$$F\{\frac{dW(t)}{dt}\} = j\omega F\{W(t)\}. \tag{1-3b}$$

$$\begin{aligned} F\{\frac{dW(t)}{dt}\} &= F\{w(t)\} \\ &= constant = \sqrt{S_0}. \end{aligned} \tag{1-4}$$

$$\begin{aligned} S(\omega) &= |F\{W(t)\}|^2 \\ &= \frac{\left|F\{\frac{dW(t)}{dt}\}\right|^2}{\omega^2} = \frac{S_0}{\omega^2}. \end{aligned} \tag{1-5}$$

The unconditional probability density function of a Wiener process is derived from random walk in [14, Ch. 11] and re-written in Eq. (1-6), where α is a constant coefficient relating step size and time step. It shows that the distribution follows the form of a normal distribution $N(0, \sigma^2)$ [17] but with its variance σ^2 increasing linearly with time. This feature will be recalled later when analyzing the noise of PLLs.

$$f(W, t) = \frac{1}{\sqrt{2\pi\alpha t}} e^{-\frac{W^2}{2\alpha t}}. \tag{1-6}$$

1-6 Introduction to Circuit Analysis in S-Domain

1-6-1 Introduction to S-Domain Analysis of 2^{nd}-order RLC Tanks

Initial settling behavior of LC oscillators is of great interest in this thesis. The analysis of the settling behavior of LC oscillators can be done in a similar way as that of settling of an RLC tank, thus knowledge on describing RLC tank with a 2^{nd}-order differential equation will be recalled in this section.

Figure 1-9 shows a 2^{nd} order system formed by a parallel RLC tank . The 2^{nd}-order differential equation describing this system can be found in most elementary circuit textbooks, which is repeated in Eq. (1-7). By substituting all state variable in Eq. (1-7) with the state variable of current in inductor, we get Eq. (1-8).

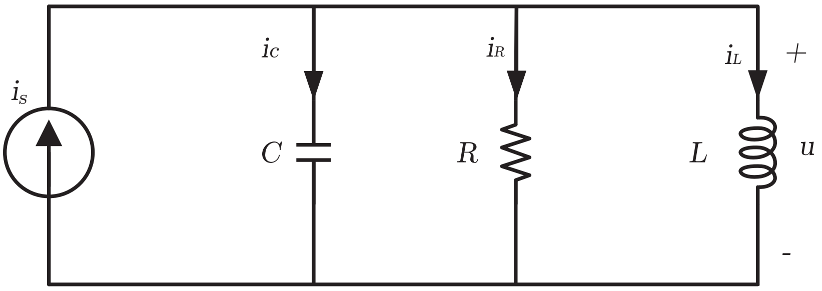


Figure 1-9: A 2^{nd}-order system formed by parallel RLC tank

$$C\frac{du}{dt} + \frac{1}{R}u + i_L(0) + \frac{1}{L}\int_0^t u(\tau)\,d\tau = i_s. \tag{1-7}$$

$$LC\frac{d^2 i_L}{dt^2} + \frac{L}{R}\frac{di_L}{dt} + i_L = i_s. \tag{1-8}$$

By transforming Eq. (1-8) into s-domain, we get:

$$LCs^2 + \frac{L}{R}s + i_L = i_s. \tag{1-9}$$

For a zero-input, non-zero initial state 2^{nd} order system shown in Figure 1-10, its differential equation is shown in Eq. (1-10), where the initial conditions are $i_L(0_+) = 0$ and $\frac{di_L}{dt}(0_+) = U_0/L$. Eq. (1-10) can be solved with the initial conditions just mentioned and the solutions have a general form shown in Eq. (1-11), where $\alpha = 1/2RC$ and $\omega_0 = 1/\sqrt{LC}$.

$$LCs^2 + \frac{L}{R}s + i_L = 0. \tag{1-10}$$

$$s_{1,2} = -\alpha \pm \sqrt{\alpha^2 - \omega_0^2}. \tag{1-11}$$

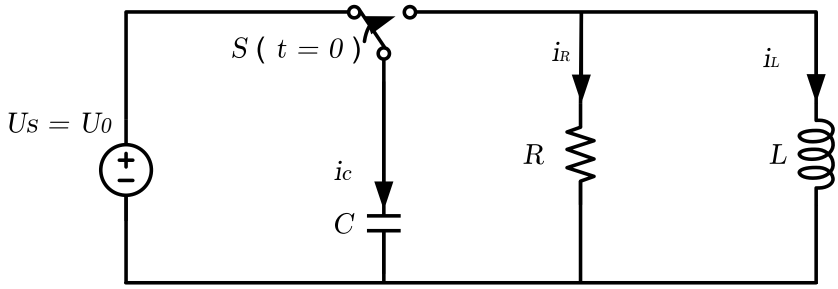


Figure 1-10: A 2^{nd}-order system switched at t = 0

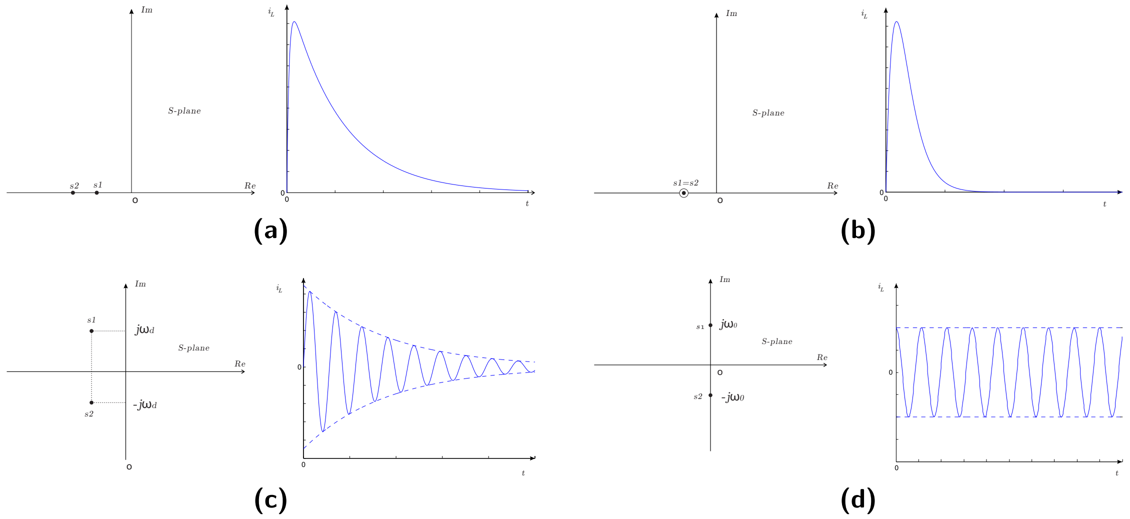


Figure 1-11: Frequency and time domain solutions of parallel RLC tanks: (a) Over-damped (b) Critically-damped (c) Under-damped (d) Non-damped

The roots shown in Eq. (1-11) are known to have 4 different scenarios, depending on different ratios between R,L and C: over-damped, critically-camped, under-damped and non-damped. The time-domain solutions to Eq. (1-10) for these 4 cases, represented by inductor current, are shown in Figure 1-11, along with locations of their corresponding solutions to Eq. (1-11)), s_1 and s_2, in S-plane, where $\omega_d = \sqrt{{\omega_0}^2 - \alpha^2}$.

Scenarios in Figure 1-11c and Figure 1-11d will be seen during start-up of LC oscillator, although there is one scenario missing in this example of zero-input RLC tank. In LC oscillator, where there is external input source, the roots could be located at right-half plane during oscillation build-up phase.

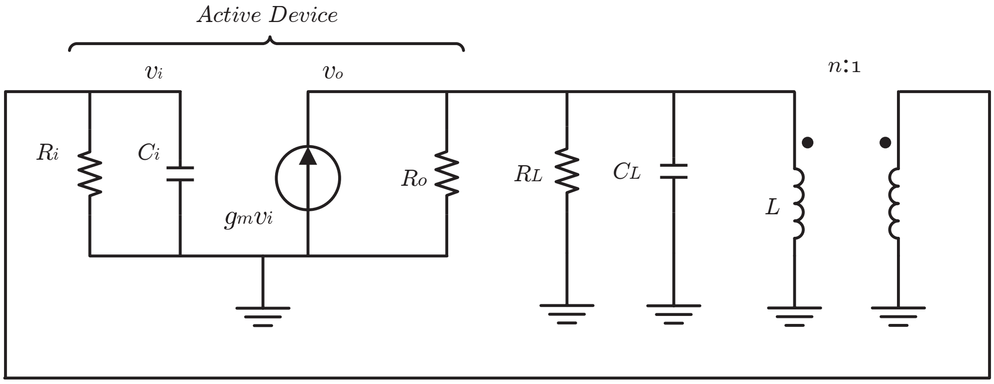


Figure 1-12: Small-signal model of LC oscillator [18]

1-6-2 Introduction to S-Domain Analysis of LC Oscillators

The mostly commonly used start-up mechanism in LC oscillators is to let the oscillator build its amplitude from thermal noise. This process is similar to the zero-input, non-zero initial state RLC tank settling process introduced back at Subsection 1-6-1, except for two major differences. The first one is that the tank in LC oscillator has an initial voltage set by thermal noise rather than being pre-charged by a voltage source. The second one is that the conductance in a simple RLC tank is fixed while the conductance seen by the LC tank in an oscillator is a variable one, due to extra active circuit. The start-up behavior has been analyzed in frequency domain with the small-signal LC oscillator schematic is shown in Figure 1-12 [18]. The system function of this loop is shown in Eq. (1-12), where $A_l = g_m R_T/n$, $R_T = R_o||R_L||n^2 R_i$ and $C = C_L + C_i/n^2$.

$$\frac{v_o(s)}{v_i(s)} = \frac{s \cdot g_m L}{1 + sL/R_T \cdot (1 - A_l) + s^2 LC}. \tag{1-12}$$

The roots of the denominator, as well as the poles of the transfer function, are shown in Eq. (1-13) and Eq. (1-14) [18]. Poles of the transfer function of the loop formed by LC tank and active devices should be located at right half plane during oscillation build-up phase and settle both on imaginary axis in steady state. During this process, the two poles are always conjugate to each other and follow a locus that is a circle with radius equal to the resonant frequency of the tank, which is redrawn in Figure 1-13, where $\omega_0 = 1/\sqrt{LC}$ stands for the resonant frequency of the tank and $\omega_0^{'}$ stands for the initial frequency after start-up, which is close to but a little deviated from ω_0.

$$s_1, s_2 = -(\frac{1-A_l}{2R_TC}) \pm j\sqrt{\frac{1}{LC} - (\frac{1-A_l}{2R_TC})^2}. \tag{1-13}$$

$$|s_1| = |s_2| = \sqrt{1/LC} = \omega_0. \tag{1-14}$$

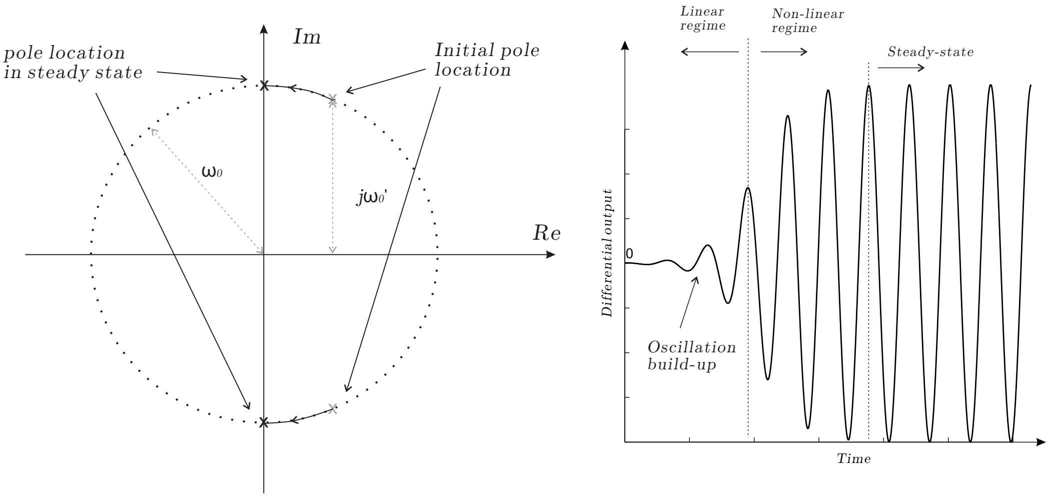


Figure 1-13: Root locus and output waveform of LC oscillator during start-up [18]

1-7 Outline of the Thesis

In Chapter 2, the difference between DCPLL and CPLL is first analyzed, especially their noise accumulation behavior. The phase noise spectrum for DCPLL is derived, as well as the equivalent filtering bandwidth. In Chapter 3, advantages and disadvantages between ring oscillators and LC oscillators are discussed. After that, the possibility of ever building a DCPLL on LC oscillators is examined and while LC oscillators have always been avoided when it comes to DCPLL because of their intrinsic slow start-up behavior, the author notices that LC oscillators do not necessarily need a longer time than ring oscillators to start. In Chapter 4, a novel technique is presented to start LC oscillators as quick as ring oscillators.

This novel instantaneous start-up technique certainly makes LC oscillators a feasible choice for DCPLL, which is a big progress, but comes with certain compromise in performance compared with continuous-operation LC oscillators. The side-effect of this instantaneous start-up technique is analyzed quantitatively in Chapter 5, as well as the design of the split inductor. In Chapter 6, existing LC oscillator structures are examined regarding whether they are compatible with the proposed instantaneous start-up technique and one of them is chosen to implement an instantaneous start-up LC DCO. A first-ever DC-ADPLL is designed and presented in Chapter 7. Design of the DC-ADPLL includes design of the locking principle, the edge-windowing circuitry (for low-power operation), the DTC gain calibration circuitry and the FSMs that control both inner and outer loops. Design of the DTC and other analog blocks, such as peak detector, comparator, buffer and divider, are covered in Chapter 8 and Chapter 9. Simulation results of this DC-ADPLL will be given in Chapter 10 and its performance will be quantified.

Chapter 2

Analysis of Difference in Noise Between DCPLLs and CPLLs

Analysis of noise in Continuous-operation PLL (CPLL) in equilibrium is usually done in frequency domain with transfer functions of system blocks described by s-functions based on the assumption that PLL has been turned on since the negative infinite of time and will continue until positive infinite of time. While this assumption is never going to be met in real world, analysis in frequency domain is still valid and gives quite accurate results, since, given the frequencies of interest, e.g. 10 kHz to 100 MHz in terms of phase noise, the noise spectrum generated by performing a Fourier transform over data points sampled over several milliseconds in locking state provides enough statistical information about the signal.

2-1 Phase Noise and Jitter

2-1-1 Phase Noise

Phase noise is a widely used metric to define performance of either open-loop oscillator or closed-loop PLL. Thorough analysis of phase noise, from two different approaches, can be found in [19, Ch. 8] and will not be repeated here. It is known that phase noise spectrum of an open-loop DCO has the shape shown in Figure 2-1, where both axes are in log scale. The spectrum can be divided into three regions, according to the slope of the curve. It is explained in [20], that the flat part is due to noise contributed by the output buffer and is uncorrelated from cycle to cycle. The 20 dB/dec part is due to accumulated thermal noise and is correlated between cycles. Since thermal noise can be well modeled by a white Gaussian process, this 20 dB/dec rolling can be explained quantitatively by Eq. (1-3) to Eq. (1-5) in Section 1-5. 1/f noise becomes dominant in the long term and results in a 10 dB/dec increase in slope at low frequency offsets in the phase noise spectrum.

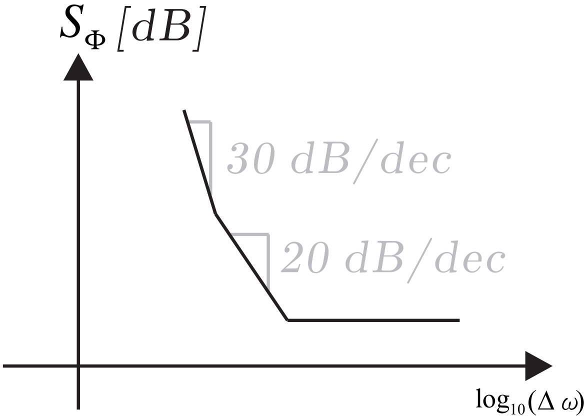


Figure 2-1: Phase noise spectrum of an open-loop DCO

2-1-2 Jitter

While the phase noise spectrum is a representation used to define the noise performance in frequency domain, jitter is used in time domain. Electrical noise in circuit causes an oscillator to deviate a little in its period from that of an ideal clock. RMS value of this deviation per period is called cycle jitter by Herzel while it is called cycle-to-cycle jitter by Demir [21], Hajimiri [22] and A. Zanchi [23]. This jitter will be denoted as Δt_c in this thesis. Another kind of jitter that is used as often is absolute jitter (accumulative jitter). These two kinds of jitter are illustrated in Figure 2-2 by comparing the timestamp of a non-ideal clock with that of an ideal clock, where T denotes the period of an ideal clock, $T[n]$ denotes the n_{th} period of real clock and $\Delta t_{acc}[n]$ denotes the n_{th} accumulative jitter. It is assumed that the real clock starts with a clean edge, same as an ideal clock at $t = 0$.

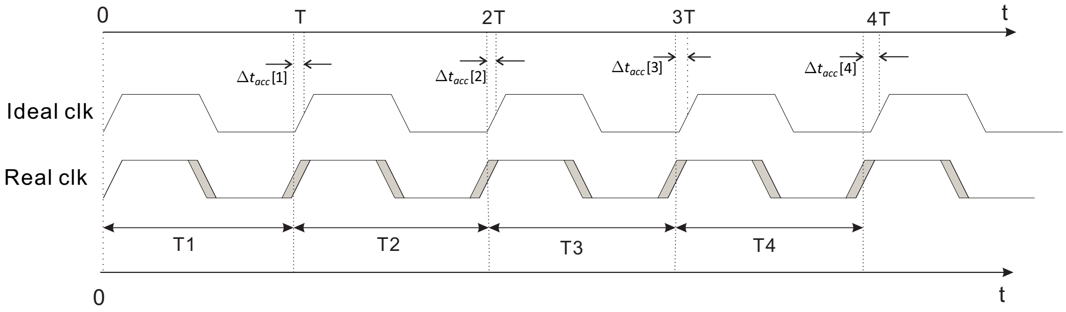


Figure 2-2: Illustration of cycle-to-cycle jitter and accumulative jitter

Δt_c and Δt_{acc} are defined in Eq. (2-1a) and Eq. (2-1b). From Eq. (2-1a) and Eq. (2-1b), it is easy to see that the relationship between $\Delta t_c[n]$ and $\Delta t_{acc}[n]$ follow Eq. (2-2). Eq. (2-2) shows that accumulative jitter of a clock is the integral of its cycle jitter while cycle jitter is the differentiation of accumulative jitter. Phase noise is achieved by first taking the Fourier transform of Δt_{acc} and then multiplying the result with a scaling ratio to transfer power in absolute time to power in phase. The variances of $\Delta t_c[n]$ and $\Delta t_{acc}[n]$ are shown in Eq. (2-3a) and Eq. (2-3b). Another kind of jitter, called cycle-to-cycle jitter by Herzel, measures period variation between two successive cycles, its variance is twice that of cycle jitter [24] and is shown in Eq. (2-3c)

$$\Delta t_c[n] = T[n] - T. \tag{2-1a}$$

$$\Delta t_{acc}[n] = \sum_{i=1}^{n} T[i] - n * T. \tag{2-1b}$$

$$\Delta t_{acc}[n] = \sum_{i=1}^{n} \Delta t_c[i]. \tag{2-2}$$

$$\sigma_c^2 = \lim_{N \to \infty} \left(\frac{1}{N} \sum_{n=1}^{N} (T[n] - T)^2 \right). \tag{2-3a}$$

$$\sigma_{acc}^2[n] \triangleq \sigma_{abs}^2[n] = n \cdot \sigma_c^2. \tag{2-3b}$$

$$\sigma_{cc}^2 = \lim_{N \to \infty} \left(\frac{1}{N} \sum_{n=1}^{N} (T[n+1] - T[n])^2 \right) = 2\sigma_c^2. \tag{2-3c}$$

2-1-3 Colored Noise and Memory in Circuit

In order to understand how duty-cycled operation affects noise behavior, it is important to identify the causes of the above-mentioned 3 kinds of noise on circuit level.

As is pointed out in [8], non-white shaping in noise spectrum means correlation of noise in time. Flat part of the noise spectrum is caused by noise at output buffer. Output buffer adds white thermal noise to the time stamps coming out from oscillator. Progression of time stamp happens at oscillator and is not affected by output buffer. Thus, this part of noise is memoryless and remains flat in the phase noise spectrum.

Noise source for the 20 dB/dec curve is also thermal noise, not at buffer but at oscillator. Although the noise source itself is white, it causes random fluctuation on tank voltage and is remembered by the tank. More specifically, it adds a white fluctuations on tank voltage passing thresholds (zero-crossings), on which period is defined. A noise injection happening at a certain time will shift all subsequent time points from that moment. Because of this, white thermal noise in oscillator circuitry will be transformed into the 20 dB/dec noise in phase noise spectrum.

1/f noise, on the other hand, has a stronger memory effect. The thermal noise that is responsible for 20 dB/dec curve merely adds random fluctuations on the time stamps of tank voltage passing thresholds and does not affect the frequency after that. If a white noise source affects the frequency at the cycle it happens and this fluctuation is remembered by the system and shifts the frequency base of subsequent cycles, then this white noise will be transformed into a 40 dB/dec curve in phase noise spectrum. 1/f noise can be considered as a mid-ground between these two, although its cause is more complicated, which is explained in [25].

2-2 Noise Accumulation and Suppression

2-2-1 Noise Accumulation and Suppression in Continuous-operation PLLs (CPLLs)

Starting from considering the 20 dB/dec region, jitter due to accumulation of thermal noise can be considered as a Wiener process, whose power increases linearly with passage of time while the uncertainty increases with the square root of time.

Since 1/f noise is correlated between its noise samples, accumulation of its power has a larger slope versus passage of time [26], which becomes dominant after a certain amount of time. Jitter accumulation of an open-loop oscillator with 30 dB/dec and 20 dB/dec noise is shown in Figure 2-3. Effect of closing the PLL can be considered as making the original integrator of noise a leaky one. With the passage of time, a certain amount of noise is continuously integrated while the loop periodically subtracts a certain portion of the detected noise in the system. At a certain point, the subtracted noise is equal to the added noise and the system reaches equilibrium. A time constant τ_L can be defined to represent the position where the system reaches equilibrium since it starts accumulating noise from zero or the time it takes for noise added at a certain point to die out [27].

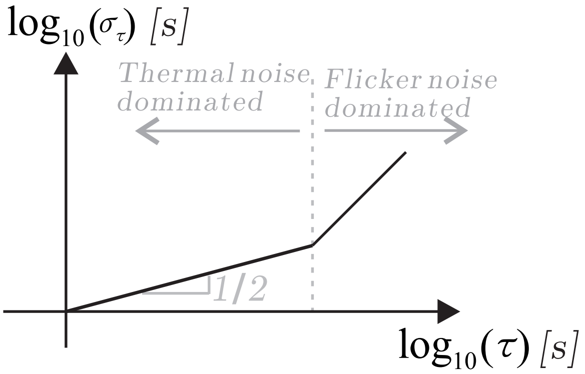


Figure 2-3: Jitter accumulation of an open-loop oscillator

Jitter power for open-loop CPLLs and closed-loop CPLLs are compared in Figure 2-4, where 1/f noise is omitted for simplicity. The effect of filtering noise in the frequency domain is that noise below the loop bandwidth, which corresponds to the loop time constant, is filtered. Open-loop and closed-loop phase noise, assuming a 1^{st}-order loop, are shown in Figure 2-5.

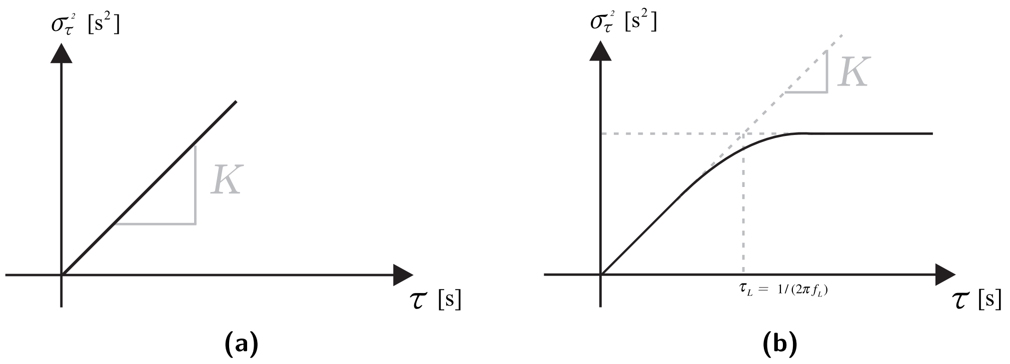


Figure 2-4: Open-loop and closed-loop behavior of noise accumulation for CPLLs: (a) Open-loop (b) Closed-loop

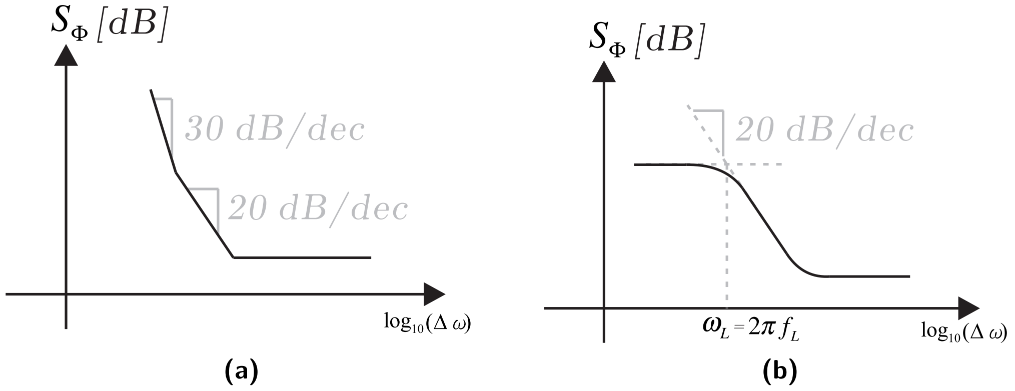


Figure 2-5: Open-loop and closed-loop phase noise spectrum for CPLLs: (a) Open-loop (b) Closed-loop

2-2-2 Noise Accumulation and Suppression in Duty-Cycled PLLs (DCPLLs)

Jitter of DCPLLs

Accumulation of jitter is different in DCPLLs. The difference can be seen in Figure 2-6. Suppose nominal frequency of CKV is N times f_{ref}, where N is an integer, and their 1^{st} rising edges are aligned. For a CPLL, shown in Figure 2-6a, at the end of the 1^{st} ref cycle, there is a jitter of $\Delta t_{acc,CPLL}(n)$, due to, for example, 20 dB/dec thermal noise. Here n means that the 1^{st} ref cycles in Figure 2-6 are the n_{th} ref cycles in continuous ref trains and the subscript $CPLL$ means that $\Delta t_{acc,CPLL}(n)$ is the jitter of the CPLL, in order to be differentiated from the subsequent jitter of the DCPLL. $\Delta t_{acc,CPLL}(n)$ affects the phase of CKV at the beginning of the 2^{nd} ref cycle. The accumulated jitter at the end of the 2^{nd} ref cycle is a summation of $\Delta t_{acc,CPLL}(n+1)$ and the noise accumulated in the 2^{nd} ref cycle. Expressions of $\Delta t_{acc,CPLL}(n)$ and $\Delta t_{acc,CPLL}(n+1)$ are shown in Eq. (2-4), where wgn represents a white Gaussian process, which models the noise in oscillators. It is obvious to see that $\Delta t_{acc,CPLL}(n+1)$ is correlated to $\Delta t_{acc,CPLL}(n)$.

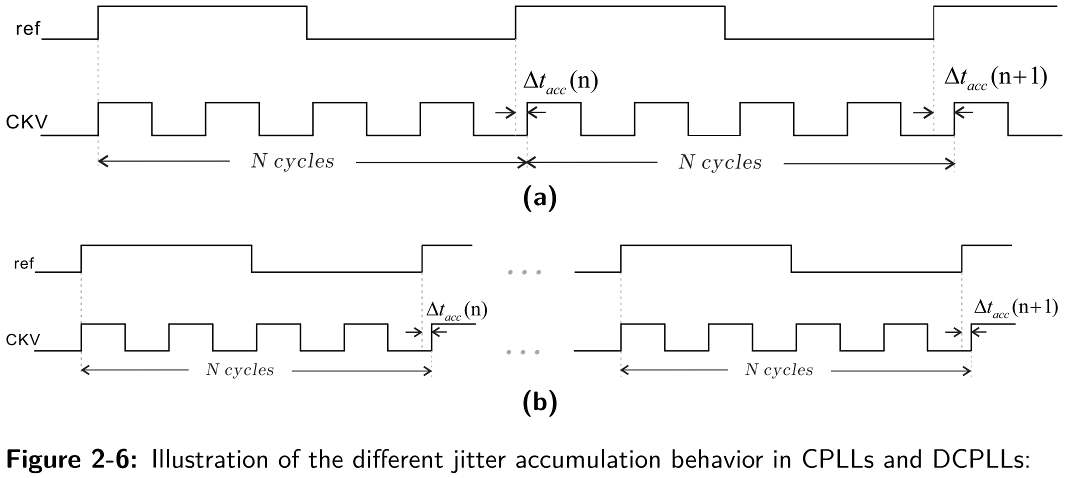


Figure 2-6: Illustration of the different jitter accumulation behavior in CPLLs and DCPLLs: (a) CPLL (b) DCPLL

$$\Delta t_{acc,CPLL}(n) = \sum_{i=1}^{N} wgn(i). \tag{2-4a}$$

$$\Delta t_{acc,CPLL}(n+1) = \sum_{i=1}^{2N} wgn(i) = \Delta t_{acc,CPLL}(n) + \sum_{i=N+1}^{2N} wgn(i). \tag{2-4b}$$

Noise behavior of the DCPLL during the 1^{st} ref cycle is similar to that of the CPLL. However, after the 1^{st} ref cycle, the oscillator in the DCPLL is turned off. When it is turned on again some time later, the phases of ref and CKV are aligned again, shown in Figure 2-6b. Due to this, $\Delta t_{acc,DCPLL}(n+1)$ is uncorrelated with $\Delta t_{acc,DCPLL}(n)$. Expressions of $\Delta t_{acc,DCPLL}(n)$ and $\Delta t_{acc,DCPLL}(n+1)$ for the DCPLL are shown in Eq. (2-5), where m represents the number of ref cycles between two bursts. The meaning of n and the subscript $DCPLL$ follow the definitions in the above analysis of the noise behavior in the CPLL.

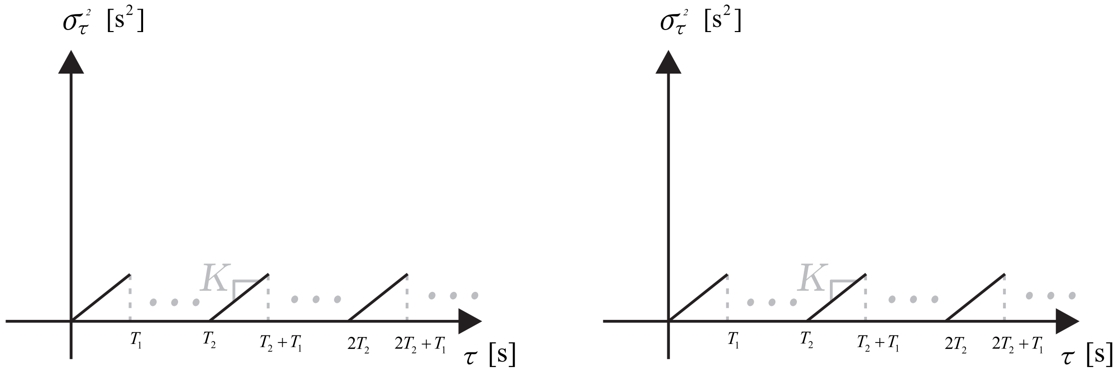
$$\Delta t_{acc,DCPLL}(n) = \sum_{i=1}^{N} wgn(i). \tag{2-5a}$$

$$\Delta t_{acc,DCPLL}(n+1) = \sum_{i=mN+1}^{(m+1)N} wgn(i) = \sum_{i=1}^{N} wgn^{'}(i). \tag{2-5b}$$

Because of this feature, DCO noise suppression occurs independent of the loop. Open-loop and closed-loop jitter behaviors are the same in this regard, which are plotted in Figure 2-7, where T_1 is the period of ref and $T_2 = m \cdot T_1$ is the Pulse Repetition Period (PRF).

Instantaneous Spectrum

The spectrum of DCPLL phase noise is more difficult to calculate than jitter. Unlike the case of CPLLs, there is no equilibrium state where new added noise is equal to the noise subtracted by the loop. However, the shape of the DCPLL spectrum can be roughly estimated through noise memory analysis. The flat noise part will be about the same. 20 dB/dec noise is periodically cleared through phase-realigning. Accumulated 1/f noise is also cleared through the phase-realigning. Although 1/f noise is partially remembered in MOSFET and cannot be completely cleared through phase-realigning, it is pointed out in [25] that large signal



(a) **(b)**

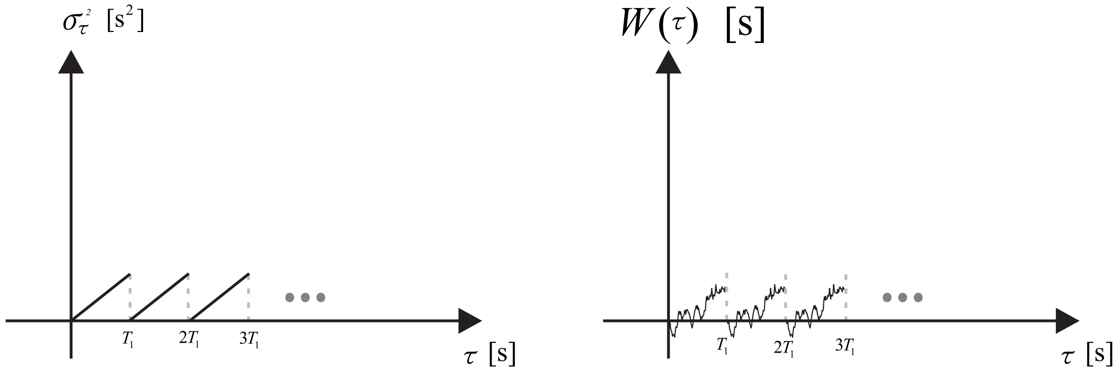
Figure 2-7: Open-loop and closed-loop behavior of noise accumulation for DCPLLs: (a) Open-loop (b) Closed-loop

excitation usually reduces 1/f by a lot. The periodically turning on/off of the DCO in a DCPLL is also a kind of large signal excitation and thus 1/f noise should also be highly suppressed in the phase noise spectrum of DCPLLs.

One derivation of the DCPLL phase noise spectrum can be found in [9]. In [9], jitter of a DCPLL is considered as a process that is identical to that of a CPLL within [0,T_1). Jitter outside of this window is considered as a replica of jitter within the window. Decimation is used to connect noise samples separated by T_2 together, shown by equation (5) in [9]. This decimation is acceptable, since noise between data bursts is not defined anyway. According to equation (5) in [9], the spectrum derived in [9] corresponds to a process shown in Figure 2-8, where W(τ) is identical to a Wiener process within [0,T_1) window. The derivation in [9] is based on an unreasonable assumption that accumulated noise samples separated by T_2 are totally correlated to each other. The actual noise process, with the decimation in [9], is shown in Figure 2-9. Another problem that the spectrum in [9] has is that although equation (5) in [9] is correct, Fig.4 in [9] does not seem match its equation (5) [1]. Equation (5) in [9] is repeated in Eq. (2-6), where $\Phi_n(f)$ is the open-loop phase noise spectrum of an oscillator. Following the same simple phase noise spectrum model of $(\alpha/f)^2$ as in [9], Eq. (2-7) follows. It can be seen from Eq. (2-7) that $\left|\Phi_n^T(f)\right|^2$ does not go to zero as $f \to 0$. Actually $\left|\Phi_n^T(f)\right|^2$ should be a maximum, which means that the Fig.4 in [9] should be closer to what is shown later in Figure 2-11. Note that the latter is drawn in log-log scale while Fig.4 in [9] is drawn in linear scale.

$$\Phi_n^T(f) = \Phi_n(f) \cdot 2i \cdot e^{-2i\pi fT/2} \cdot \sin(\pi fT). \tag{2-6}$$

$$\begin{aligned}
\lim_{f\to 0} \left|\Phi_n^T(f)\right|^2 &= \lim_{f\to 0} \left|\Phi_n(f) \cdot 2i \cdot e^{-2i\pi fT/2} \cdot \sin(\pi fT)\right|^2 \\
&= \lim_{f\to 0} \left[\left(\frac{\alpha}{f}\right)^2 \cdot 4 \cdot 1 \cdot \sin^2(\pi fT)\right] \\
&= 4 \cdot \lim_{f\to 0} \left[\left(\frac{\alpha}{f}\right)^2 \cdot \sin^2(\pi fT)\right] \\
&= 4 \cdot \alpha^2 \cdot \pi^2 \cdot T^2
\end{aligned} \tag{2-7}$$



(a)

(b)

Figure 2-8: Variance and samples of the noise process corresponding to the spectrum derived in [9]: (a) Noise variance (b) Noise samples

Since the derivation in [9] has its problem, the author decided to look for another approach. The problem can be summarized as finding a spectrum that can describe an evolving stochastic process, which is similar to time-frequency analysis in signal processing.

[1] The author realized the difference between Figure 2-11 in this thesis and Fig.4 in [9] during his derivation for the phase noise spectrum of DCPLLs. After that, the author double-checked [9] and came to the conclusion that there seems to be a mistake in Fig.4 of [9], probably due to the overlooking of the $1/f^2$ item when drawing the figure. However, this is the opinion of the author of this thesis. If the reader has a different opinion on this, please feel free to contact and correct the author.

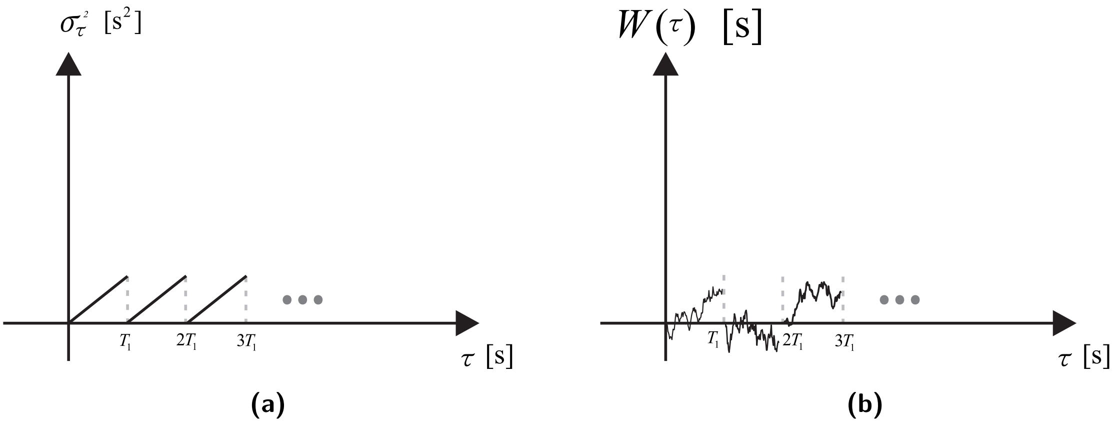


Figure 2-9: Variance and samples of the noise process in actual DCPLLs: (a) Noise variance (b) Noise samples

The Fourier transform of a signal $G(\tau)$ is shown in Eq. (2-8a) [28]. Power in time domain and frequency domain can be related together with *Parseval's theorem* [29], [16, Ch. 4] (also known as *Plancherel's theorem*) [28]. In physics and engineering, *Parseval's theorem* is often written as the form shown in Eq. (2-8b). $g(f)$ represents the energy distribution in frequency of a signal that extends from $-\infty$ to $+\infty$. $G(\tau)$ represents the summation of the signal energy at all frequencies at a specific time t. In [28], an instantaneous power spectrum $\rho(\tau, f)$[1] is defined as the power density of a signal that is distributed both in time and frequency, where $\int_{-\infty}^{T} \int_{-\infty}^{+\infty} \rho(\tau, f) df d\tau$ represents the total signal energy up to time T. The formal definition of $\rho(\tau, f)$ can be found in [28] and is repeated in Eq. (2-9), where $g_t(f)$ can be considered as a traditional Fourier transform of $G_t(\tau)$, which is equal to a continuous function $G(\tau)$ up to time t and is zero ever after. An accumulation of $\rho(\tau, f)$, over τ, within a short time $d\tau$ at $\tau = t$ represents the difference between the power of $G(\tau)$ at t and $t + d\tau$.

$$g(f) = \int_{-\infty}^{+\infty} e^{-2\pi i f t} G(\tau) d\tau. \tag{2-8a}$$

$$\int_{-\infty}^{+\infty} |g(f)|^2 df = \int_{-\infty}^{+\infty} |G(\tau)|^2 d\tau. \qquad (Parseval's\ theorem) \tag{2-8b}$$

[1] In [28], the instantaneous spectrum is labeled $\rho(x, f)$. In this thesis, it is named $\rho(\tau, f)$ to avoid naming conflict with subsequent derivations.

$$\int_{-\infty}^{t} \rho(\tau, f) d\tau = |g_t(f)|^2. \tag{2-9a}$$

$$g_t(f) = \int_{-\infty}^{t} G(x) e^{-2\pi i f \tau} d\tau = \int_{-\infty}^{+\infty} G_t(\tau) e^{-2\pi i f \tau} d\tau. \tag{2-9b}$$

$$G_t(\tau) = \begin{cases} G(\tau)\ , \ \tau < t. \\ 0 \qquad , \tau > t. \end{cases} \tag{2-9c}$$

Instantaneous spectrum definition for stochastic processes is different from that for deterministic processes. One way of defining the instantaneous power spectrum of a stochastic process is the *Wigner spectrum* [30], which is shown in Eq. (2-10). Energy and projection properties of the *Wigner spectrum* are shown in Eq. (2-11) [31], [32]. The Wigner spectrum of a Wiener process is derived in [30] and shown in Eq. (2-12), where N_0 is the variance of the Gaussian noise that the Wiener process is integrated on. The Wigner spectrum corresponding to Eq. (2-12a) is shown in Figure 2-10, the listing of which can be found at List A.1 in Appendix A-1.

$$\overline{W_x}(t, \omega) = \frac{1}{2\pi} \int E\left[x^*(t - \frac{\tau}{2}) x(t + \frac{\tau}{2})\right] e^{-i\tau\omega} d\tau. \tag{2-10}$$

$$|x(t)|^2 = \int_{-\infty}^{+\infty} \overline{W_x}(t, f) df. \tag{2-11a}$$

$$|X(f)|^2 = \int_{-\infty}^{+\infty} \overline{W_x}(t, f) dt. \tag{2-11b}$$

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \overline{W_x}(t, f) df dt = \int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |X(f)|^2 df. \tag{2-11c}$$

$$\overline{W_{x,Wiener}}(t, \omega) = \frac{N_0}{\pi}[1 - cos(2\omega t)] = \frac{N_0}{\pi} \frac{\sin^2 \omega t}{\omega^2}. \tag{2-12a}$$

$$\overline{W_{x,Wiener}}(t, f) = \frac{N_0}{\pi} \frac{\sin^2 2\pi f t}{(2\pi f)^2}. \tag{2-12b}$$

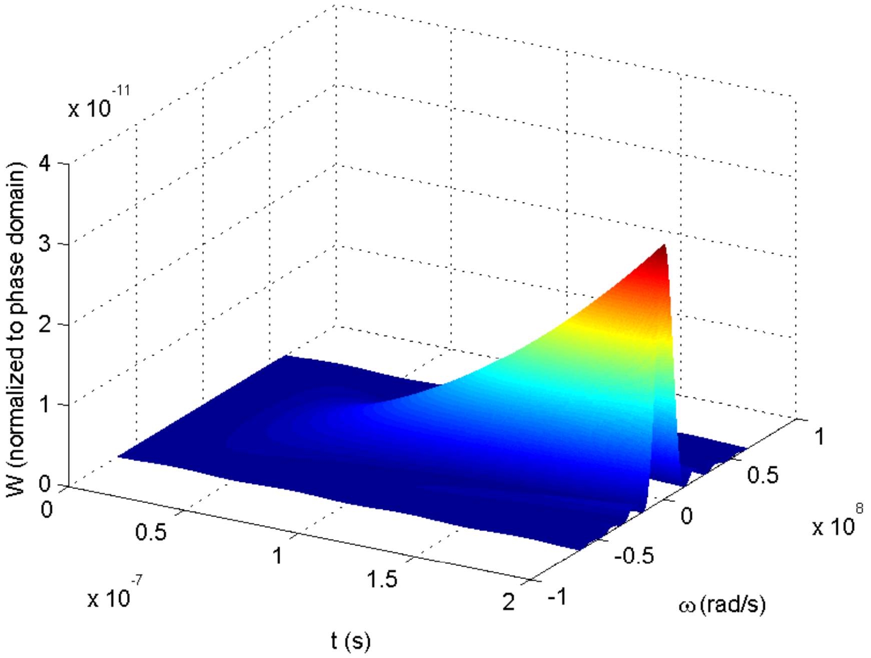


Figure 2-10: Wigner spectrum of a Wiener process

$\overline{W_{x,Wiener}}(t_0, f)$ gives information about the energy distribution in frequency at $t = t_0$. The instantaneous spectrum, with the same parameters as the one in Figure 2-10 at t = 200 ns is shown in Figure 2-11, MATLAB code of which can be found at List A.2 in Appendix A-1. Both axes are in log scale. The Wigner spectrum of the noise process of an open-loop oscillator, which can be modeled as a Wiener process, can be related to the oscillator's open-loop phase noise spectrum through the constant N_0 in Eq. (2-12). According to the derivation in [30], $N_0/(2\pi)$ is the energy increment per unit time while in [24] the relation between variance of the Wiener process and passage of time is derived and shown in Eq. (2-13). By setting $N_0/(2\pi) = c$, the Wigner spectrum $\overline{W_{x,Wiener}}(t, f)$ is made to correspond to the power of jitter, defined in time instead of phase, distributed in both time and frequency domain. When multiplied by $(2\pi/T_{osc})^2$, the power with a dimension of s^2 in time can be transformed to power in phase with a dimension of rad^2. The constant c is related to phase noise spectrum with an expression derived in [24] and repeated in Eq. (2-14). Thus, N_0 can be expressed in terms of phase noise as is shown in Eq. (2-15). It will be shown later that using this value of N_0 gives a value of accumulated jitter at t $= t_0$ which matches with existing theory by integrating the power of $\overline{W_{x,Wiener}}(t_0, f)$ over frequency.

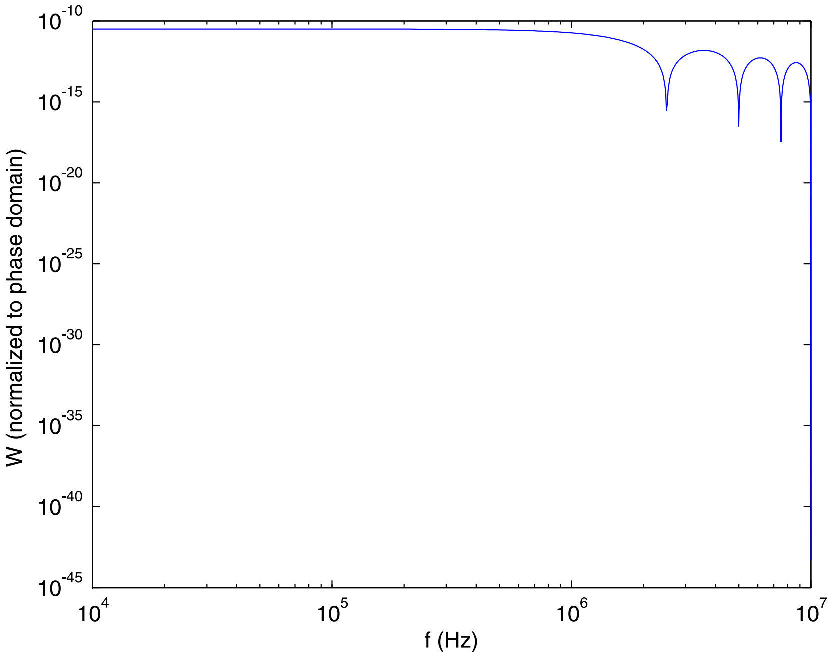


Figure 2-11: Instantaneous spectrum of a Wiener process at t $= 200\,\mathrm{ns}$ (log-log scale)

$$\sigma^2_{abs}(t) = ct. \tag{2-13}$$

$$\sigma^2_c = \sigma^2_{abs}(t = 1/f_{osc}) = \frac{c}{f_{osc}}. \tag{2-14a}$$

$$\mathcal{L}(f) = \frac{\sigma^2_c f^3_{osc}}{f^2}. \tag{2-14b}$$

$$\sigma^2_c = \frac{f^2 \mathcal{L}(f)}{f^3_{osc}}. \tag{2-14c}$$

$$c = \frac{f^2 \mathcal{L}(f)}{f^2_{osc}}. \tag{2-14d}$$

$$N_0 = 2\pi c = 2\pi \cdot \frac{f^2 \mathcal{L}(f)}{f^2_{osc}}. \tag{2-15}$$

According to the definition of Wigner Spectrum and Eq. (2-11a), integration of the spectrum over the f axis at a fixed time $t = t_0$ should give $|x(t_0)|^2$, which is the variance of absolute jitter at that moment. This integration, which is in much the same way as in [9], is shown in Eq. (2-16) [1]. Substituting Eq. (2-15) and Eq. (2-14b) into Eq. (2-16), we get Eq. (2-17). By setting $t = 1/f_{osc}$ in Eq. (2-17), we get Eq. (2-18), which matches exactly the expression given by [24] and shown in Eq. (2-14c), where σ_c^2 is defined as the variance of the absolute jitter at $t = 1/f_{osc}$.

$$\begin{aligned}
|x(t)|^2 &= \int_{-\infty}^{+\infty} \overline{W_{x,Wiener}}(t,f)df \\
&= 2\int_{0}^{+\infty} \overline{W_{x,Wiener}}(t,f)df \\
&= 2\int_{0}^{+\infty} \frac{N_0}{\pi}\frac{\sin^2(2\pi ft)}{(2\pi f)^2}df \\
&= 2\frac{N_0}{\pi}\frac{t}{2\pi}\int_{0}^{+\infty} \frac{\sin^2(2\pi ft)}{(2\pi ft)^2}d(2\pi ft) \\
&= 2\frac{N_0 t}{2\pi^2}\frac{\pi}{2} = \boxed{\frac{N_0 t}{2\pi}}.
\end{aligned} \tag{2-16}$$

$$|x(t)|^2 = \frac{f^2\mathcal{L}(f)}{f_{osc}^2}\cdot t = \sigma_c^2 \cdot f_{osc}\cdot t. \tag{2-17}$$

$$|x(t)|^2\Big|_{t=1/f_{osc}} = \frac{f^2\mathcal{L}(f)}{f_{osc}^3}. \tag{2-18}$$

On the other hand, integration of $\overline{W_{x,Wiener}}(\tau, f)$ over the time axis at $f = f_0$ gives the process's energy at $f = f_0$ since $t = 0$. The integration is shown in Eq. (2-19),where the integration of $sin(x)$ is shown in Eq. (2-20). From Eq. (2-19), it can be seen that, as integration length in time goes to infinity, $X_t(f)$ is infinite at any frequency. This is because $\overline{W_{x,Wiener}}(\tau, f)$ is everywhere positive and this is easy to understand since total energy of an infinitely long noise process is infinite. Infinite number at any frequency is meaningless and does not agree with the conventional phase noise spectrum. Thus, the integration $X_t(f)$ needs to be divided by the integration length t in order to converge [28]. A new spectrum $S_t(f)_{\Delta t}$, which is $X_t(f)$ divided by the integration length, is defined in Eq. (2-21), the subscript Δt

[1] Without specifying, the process x(t) is used to specifically denote the Wiener process in subsequent text.

meaning that $S_t(f)_{\Delta t}$ has the dimension of s^2. In order to be transformed to phase domain, $S_t(f)_{\Delta\Phi}$ is defined in Eq. (2-22).

$$\begin{aligned}
X_t(f) &= \int_{-\infty}^{t} \overline{W_{x,Wiener}}(\tau, f) d\tau \\
&= \int_{0}^{t} \overline{W_{x,Wiener}}(\tau, f) d\tau \\
&= \int_{0}^{t} \frac{N_0}{\pi} \frac{\sin^2(2\pi f\tau)}{(2\pi f)^2} d\tau \\
&= \frac{N_0}{\pi(2\pi f)^3} \int_{0}^{t} \sin^2(2\pi f\tau) d(2\pi f\tau) \\
&= \frac{N_0}{\pi(2\pi f)^3} (\frac{2\pi ft}{2} - \frac{1}{4} sin(4\pi ft)).
\end{aligned} \tag{2-19}$$

$$\int \sin^2 x \, dx = \frac{x}{2} - \frac{1}{4} \sin 2x + C. \tag{2-20}$$

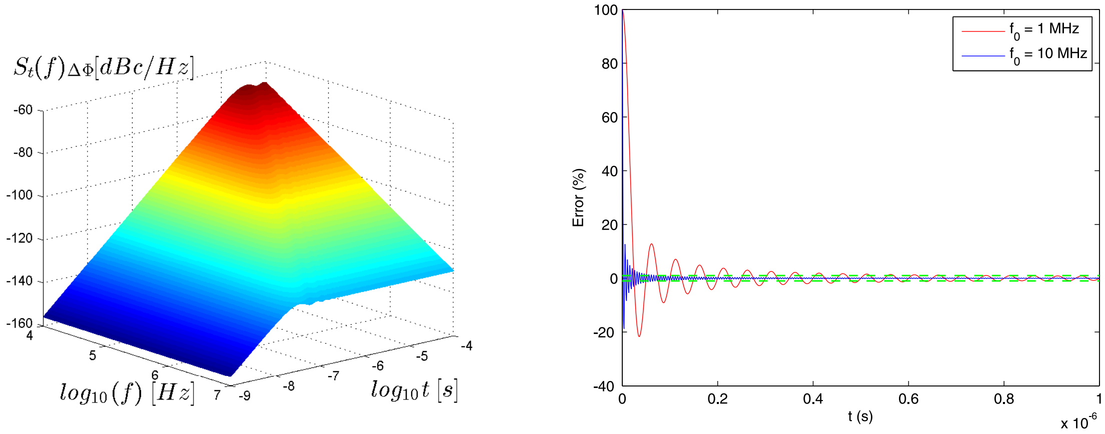
$$\begin{aligned}
S_t(f)_{\Delta t} &= \frac{1}{t} X_t(f) \\
&= \frac{N_0}{2\pi(2\pi f)^2} - \frac{N_0}{4\pi t(2\pi f)^3} sin(4\pi ft).
\end{aligned} \tag{2-21}$$

$$S_t(f)_{\Delta\Phi} = S_t(f)_{\Delta t} \cdot \left(\frac{2\pi}{T_{osc}}\right)^2. \tag{2-22a}$$

$$\begin{aligned}
S(f)_{\Delta\Phi} &= \lim_{t\to\infty} S_t(f)_{\Delta\Phi} \\
&= \frac{N_0}{2\pi(2\pi f)^2} \cdot \left(\frac{2\pi}{T_{osc}}\right)^2 \\
&= \frac{2\pi \cdot f^2 \mathcal{L}(f)/f_{osc}^2}{2\pi(2\pi f)^2} \left(\frac{2\pi}{T_{osc}}\right)^2 \\
&= \mathcal{L}(f).
\end{aligned} \tag{2-22b}$$

It can be seen from Eq. (2-22b) that when taking the limit of integration window to infinity, integration of the Wigner spectrum of a Wiener process over time, divided by integration length, asymptotically converges to the 20 dB/dec part of conventional phase noise spectrum.

However, this does not mean that Wigner spectrum is only useful when integration window is infinite. Actually, the deviation from a conventional $1/f^2$ phase noise spectrum comes from the 2^{nd} item in Eq. (2-21), which dies away when increasing integration length. The ratio between the 2^{nd} and 1^{st} item is $sin(4\pi ft)/(4\pi ft)$. It means that the deviation between the integrated Wigner spectrum and phase noise becomes smaller when the observation window is longer. This is not a defect of the Wigner spectrum, it simply shows the fact that a sufficiently large amount of data samples are needed to estimate accurately enough the power of a statistical process at a certain frequency. Figure 2-12a shows $S_t(f)_{\Delta\Phi}$ in terms of both time and frequency, the listing of which can be found at List A.3 in Appendix A-1. When the time axis goes to infinity, $S_t(f)_{\Delta\Phi}$ converges asymptotically to the conventional phase noise spectrum. It also shows that the longer the measurement window, the lower the frequency whose power can be considered close enough to its value in an infinitely long measurement window. Figure 2-12b shows $sin(4\pi ft)/(4\pi ft)$ in terms of time at two frequencies, 1 MHz and 10 MHz. The residual variation of the power at 10 MHz settles to within 1% after about 0.1 µs while that of the power at 1 MHz needs approximately 1 µs. In this way, the Wigner spectrum can be used to calculate the amount of measurement time needed based on the lowest frequency component of interest and the acceptable residual error.



(a) $S_t(f)_{\Delta\Phi}$

(b) Residual variation at f_0 over time

Figure 2-12: Wigner Spectrum of a Wiener process integrated over time axis: (a) $S_t(f)_{\Delta\Phi}$ (b) Residual variation at f_0 over time

Spectrum of DCPLLs

With the help of the Wigner spectrum, the phase noise spectrum of the noise process shown in Figure 2-9b can be derived. The derivation below uses the same decimation as in [9], for a fair comparison. However, the data interpolation/decimation method is not singular. For example, the undefined noise between T_1 and T_2 in Figure 2-7 can be considered as zero if needed to. The result is the same amount of noise power averaged over a longer measurement window. When the integration time goes to infinity, the achieved phase noise spectrum gets scaled, in y axis, by T_1/T_2 compared to the spectrum achieved with the interpolation shown in Figure 2-9. The phase noise spectrum for a DCPLL noise process is derived below, assuming the DCPLL uses an oscillator with an open-loop phase noise spectrum of $\mathcal{L}(f)$:

Proof.

$$\begin{aligned} S(f)_{\Delta\Phi,DCPLL} &= \lim_{t\to\infty} S_t(f)_{\Delta\Phi,DCPLL} \\ &= \lim_{t\to\infty} \frac{1}{t}\left(\frac{2\pi}{T_{osc}}\right)^2 X_t(f)_{DCPLL} \\ &= \lim_{t\to\infty} \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\int_{-\infty}^{t} \overline{W_{DCPLL}}(\tau,f)d\tau \\ &= \lim_{t\to\infty} \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\int_{0}^{t} \overline{W_{DCPLL}}(\tau,f)d\tau. \end{aligned} \tag{2-23}$$

Define arbitray t: $mT_1 \leq t < (m+1)T_1$, where m is an integer:

$$S(f)_{\Delta\Phi,DCPLL} = \lim_{t\to\infty} \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\left[\int_{0}^{mT_1} \overline{W_{DCPLL}}(\tau,f)d\tau + \int_{mT_1}^{t} \overline{W_{DCPLL}}(\tau,f)d\tau\right].$$

Variance of the noise of DCPLL within $[0, T_1)$ is identical to the noise of the open-loop oscillator, whose noise can be modeled by a Wiener process. Thus, we have:

$$S(f)_{\Delta\Phi,DCPLL} = \lim_{t\to\infty} \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\left[m\int_{0}^{T_1} \overline{W_{x,Wiener}}(\tau,f)d\tau + \int_{0}^{t-mT_1} \overline{W_{x,Wiener}}(\tau,f)d\tau\right].$$

Since $mT_1 \leq t < (m+1)T_1$, $t \to \infty$ is equivalent to $m \to \infty$, thus we have:

$$\begin{aligned} S(f)_{\Delta\Phi,DCPLL} &= \lim_{m\to\infty} \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t} \left[m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau + \int_0^{t-mT_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right] \\ &= \left[\left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{m\to\infty} \frac{1}{t} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right] \\ &\quad + \left[\left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{m\to\infty} \frac{1}{t} \int_0^{t-mT_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right]. \end{aligned} \tag{2-24}$$

Since

$$0 \leq \int_0^{t-mT_1} \overline{W_{x,Wiener}}(\tau, f) d\tau < \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau,$$

where, after substituting $\overline{W_{x,Wiener}}$ with the expression shown in Eq. (2-12b), we have:

$$\begin{aligned} \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau &= \int_0^{T_1} \frac{N_0}{\pi} \frac{\sin^2 2\pi f\tau}{(2\pi f)^2} d\tau \\ &= \frac{N_0}{\pi (2\pi f)^3} \int_0^{T_1} \sin^2(2\pi f\tau) d(2\pi f\tau), \end{aligned} \tag{2-25}$$

which is, obviously, a finite number, thus the 2^{nd} item in Eq. (2-24) converges to 0 when t goes to infinity.

Then, we have:

$$S(f)_{\Delta\Phi,DCPLL} = \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{m\to\infty} \frac{1}{t} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau. \tag{2-26}$$

For Eq. (2-26), since $mT_1 \leq t < (m+1)T_1$, we have:

$$\lim_{m\to\infty} \left[\frac{1}{t} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right] \geq \lim_{m\to\infty} \left[\frac{1}{(m+1)T_1} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right]. \tag{2-27a}$$

$$\lim_{m\to\infty} \left[\frac{1}{t} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right] \leq \lim_{m\to\infty} \left[\frac{1}{mT_1} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right]. \tag{2-27b}$$

Further, according to the definition shown in Eq. (2-12b), where the specific Wiener process is related to the open-loop oscillator through the constant N_0 shown in Eq. (2-15), we have:

$$\begin{aligned}\lim_{m\to\infty}\left[\frac{m}{(m+1)T_1}\int_0^{T_1}\overline{W_{x,Wiener}}(\tau,f)d\tau\right] &= \lim_{m\to\infty}\left[\frac{m}{(m+1)T_1}\int_0^{T_1}\frac{N_0}{\pi}\frac{\sin^2(2\pi f\tau)}{(2\pi f)^2}d\tau\right] \\ &= \frac{1}{T_1}\int_0^{T_1}\frac{N_0}{\pi}\frac{\sin^2(2\pi f\tau)}{(2\pi f\tau)^2}d(2\pi f\tau). \qquad (2\text{-}28)\end{aligned}$$

The integration in Eq. (2-28) is similar to the one performed in Eq. (2-19) and is

$$\lim_{m\to\infty}\left[\frac{m}{(m+1)T_1}\int_0^{T_1}\overline{W_{x,Wiener}}(\tau,f)d\tau\right] = \frac{1}{T_1}\cdot\frac{N_0}{\pi(2\pi f)^3}\left[\frac{2\pi fT_1}{2}-\frac{1}{4}\sin(4\pi fT_1)\right]. \qquad (2\text{-}29)$$

Substituting $N_0 = 2\pi\cdot f^2\cdot\mathcal{L}(f)/f_{osc}^2$ into Eq. (2-29), we have:

$$\lim_{m\to\infty}\left[\frac{m}{(m+1)T_1}\int_0^{T_1}\overline{W_{x,Wiener}}(\tau,f)d\tau\right] = \frac{1}{T_1}\cdot\frac{2\mathcal{L}(f)}{f_{osc}^2\cdot(2\pi)^3 f}\left[\frac{2\pi fT_1}{2}-\frac{1}{4}\sin(4\pi fT_1)\right]. \qquad (2\text{-}30)$$

The derivation for the right item in Eq. (2-27b) follows the almost identical derivation shown from Eq. (2-28) to Eq. (2-30). Its expression is the same as the expression to the right of the equal sign in Eq. (2-30). We have:

$$\lim_{m\to\infty}\left[\frac{m}{mT_1}\int_0^{T_1}\overline{W_{x,Wiener}}(\tau,f)d\tau\right] = \frac{1}{T_1}\cdot\frac{2\mathcal{L}(f)}{f_{osc}^2\cdot(2\pi)^3 f}\left[\frac{2\pi fT_1}{2}-\frac{1}{4}\sin(4\pi fT_1)\right]. \qquad (2\text{-}31)$$

Thus, according to *Squeeze theorem* [33] shown in Theorem 2.1 and Eq. (2-27a), Eq. (2-27b), Eq. (2-30) and Eq. (2-31),

Theorem 2.1 (Squeeze theorem)**.** *Let I be an interval having the point a as a limit point. Let f, g, and h be functions defined on I, except possibly at a itself.*

Suppose that for every x in I not equal to a, we have:

$$g(x) \leq f(x) \leq h(x),$$

and also suppose that:

$$\lim_{x \to a} g(x) = \lim_{x \to a} h(x) = L.$$

Then,

$$\lim_{x \to a} f(x) = L.$$

we have:

$$\lim_{m \to \infty} \left[\frac{1}{t} m \int_0^{T_1} \overline{W_{x,Wiener}}(\tau, f) d\tau \right] = \frac{1}{T_1} \cdot \frac{2\mathcal{L}(f)}{f_{osc}^2 \cdot (2\pi)^3 f} \left[\frac{2\pi f T_1}{2} - \frac{1}{4} \sin(4\pi f T_1) \right]. \quad (2\text{-}32)$$

Substituting Eq. (2-32) into Eq. (2-26), we have:

$$\begin{aligned} S(f)_{\Delta\Phi,DCPLL} &= \frac{2\mathcal{L}(f)}{2\pi f T_1} \left[\frac{2\pi f T_1}{2} - \frac{1}{4} \sin(4\pi f T_1) \right] \\ &= \boxed{\frac{\mathcal{L}(f)}{\pi f T_1} \left[\pi f T_1 - \frac{1}{4} \sin(4\pi f T_1) \right]}, \end{aligned} \quad (2\text{-}33)$$

where $\mathcal{L}(f)$ is the phase noise of the open-loop oscillator used in the DCPLL and T_1 is the burst duration. □

From Eq. (2-33), it can be seen that when T_1 goes to infinity, the phase noise of DCPLL converges to the phase noise of the open-loop oscillator, which agrees with the intuitive fact that when T_1 goes to infinity, the noise of the DCPLL is never cleared.

Area under the phase noise spectrum, called integrated jitter in [34], is often used in literature to measure performance of PLLs. The area under the phase noise spectrum of the DCPLL (considering 20 dB/dec thermal noise only), such as the one in Figure 2-13b, is shown below:

$$\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df = \int_{-\infty}^{+\infty} \left[\lim_{t \to \infty} S_t(f)_{\Delta\Phi,DCPLL} \right] df. \quad (2\text{-}34)$$

Substituting Eq. (2-23) into Eq. (2-34), we have:

$$\begin{aligned}\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df &= \int_{-\infty}^{+\infty}\left[\lim_{t\to\infty}\left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\int_0^t \overline{W_{DCPLL}}(\tau,f)d\tau\right]df \\ &= \int_{-\infty}^{+\infty}\left[\lim_{t\to\infty}\left(\frac{2\pi}{T_{osc}}\right)^2 \frac{1}{t}\int_0^t \overline{W_{DCPLL}}(\tau,f)d\tau\right]df \\ &= \left(\frac{2\pi}{T_{osc}}\right)^2 \int_{-\infty}^{+\infty}\left[\lim_{t\to\infty}\frac{1}{t}\int_0^t \overline{W_{DCPLL}}(\tau,f)d\tau\right]df \\ &= \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\int_0^t\int_{-\infty}^{+\infty} \overline{W_{DCPLL}}(\tau,f)df d\tau. \end{aligned} \tag{2-35}$$

Define arbitray t: $mT_1 \le t \le (m+1)T_1$, where m is an integer,

$$\begin{aligned}\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df &= \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\left[\int_0^{mT_1}\int_{-\infty}^{+\infty} \overline{W_{DCPLL}}(\tau,f)df d\tau\right. \\ &\qquad \left. + \int_{mT_1}^{t}\int_{-\infty}^{+\infty} \overline{W_{DCPLL}}(\tau,f)df d\tau\right] \\ &= \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\left[m\int_0^{T_1}\int_{-\infty}^{+\infty} \overline{W_{x,Wiener}}(\tau,f)df d\tau\right. \\ &\qquad \left. + \int_0^{t-mT_1}\int_{-\infty}^{+\infty} \overline{W_{x,Wiener}}(\tau,f)df d\tau\right]. \end{aligned} \tag{2-36}$$

Substituting Eq. (2-16) into Eq. (2-36), we have:

$$\begin{aligned}\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df &= \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\left[m\int_0^{T_1}\frac{N_0\tau}{2\pi}d\tau + \int_0^{t-mT_1}\frac{N_0\tau}{2\pi}d\tau\right] \\ &= \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\left[m\frac{N_0}{2\pi}\frac{T_1^2}{2} + \frac{N_0}{2\pi}\frac{(t-mT_1)^2}{2}\right]. \end{aligned} \tag{2-37}$$

Since $0 \le (t - mT_1) \le T_1$, the 2^{nd} item inside the square brackets Eq. (2-37) converges to zero when divided by an infinitely large t, thus it follows:

$$\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df = \left(\frac{2\pi}{T_{osc}}\right)^2 \lim_{t\to\infty}\frac{1}{t}\left[m\frac{N_0}{2\pi}\frac{T_1^2}{2}\right]. \tag{2-38}$$

Since $mT_1 \leq t \leq (m+1)T_1$, according to the *Squeeze theorem* shown in Theorem 2.1, Eq. (2-38) follows:

$$\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df = \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{N_0}{2\pi}\frac{T_1}{2}. \tag{2-39}$$

Substituting $N_0 = 2\pi c = 2\pi\sigma_c^2/T_{osc}$ into Eq. (2-39), we have:

$$\begin{aligned}\int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df &= \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{2\pi\sigma_c^2/T_{osc}}{2\pi}\frac{T_1}{2} \\ &= \left(\frac{2\pi}{T_{osc}}\right)^2 \frac{\sigma_c^2}{T_{osc}}\frac{T_1}{2} \\ &= \boxed{2\pi^2 \cdot \sigma_c^2 \cdot T_1 \cdot f_{osc}^3}. \end{aligned} \tag{2-40}$$

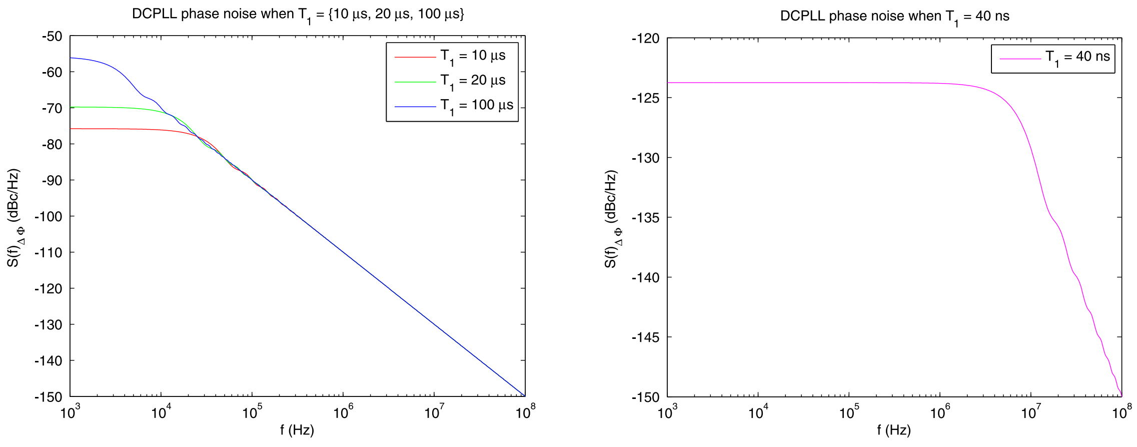
Eq. (2-40) shows the integrated power with a dimension of rad^2. The integrated power with a dimension of s^2 is shown in Eq. (2-41).

$$\begin{aligned}\int_{-\infty}^{+\infty} S(f)_{\Delta t,DCPLL} df &= \left(\frac{T_{osc}}{2\pi}\right)^2 \cdot \int_{-\infty}^{+\infty} S(f)_{\Delta\Phi,DCPLL} df \\ &= \boxed{\frac{\sigma_c^2 \cdot T_1 \cdot f_{osc}}{2}}. \end{aligned} \tag{2-41}$$

Meanwhile, the variance of the jitter at $t = 1/f_{osc}$ can be calculated by setting $t = 1/f_{osc}$ in Eq. (2-17), which is shown in Eq. (2-42). It can be seen from Eq. (2-41) and Eq. (2-42) that integrated jitter with a dimension of s^2 is exactly half of the jitter variance at $t = T_1$.

$$|x(t)|^2\Big|_{t=T_1} = \sigma_c^2 \cdot T_1 \cdot f_{osc}. \tag{2-42}$$

Assuming an open-loop oscillator phase noise of -110 dBc/Hz at 1 MHz offset and oscillation frequency of 5.0 GHz, the DCPLL phase noise when $T_1 = \{10\,\mu\text{s}, 20\,\mu\text{s}, 100\,\mu\text{s}\}$ is shown in Figure 2-13a while the phase noise for the specific burst duration of 40 ns in this design is shown in Figure 2-13b, both listings of which can be found at List A.4 in Appendix A-1. The periodic noise-clearing effect of the DCPLL behaves like a high-pass filter whose bandwidth is proportional to $1/T_1$.



(a) **(b)**

Figure 2-13: DCPLL phase noise for different burst duration: (a) $T_1 = \{10\,\mu\text{s},\, 20\,\mu\text{s}\ 100\,\mu\text{s}\}$ (b) $T_1 = 40\,\text{ns}$

The 3 dB corner frequency as well as the phase noise level at $f \to 0$ Hz can be calculated.

$$\lim_{f\to 0} S(f)_{\Delta\Phi,DCPLL} = \lim_{f\to 0} \frac{\mathcal{L}(f)}{\pi f T_1}\left[\pi f T - \frac{1}{4}\sin(4\pi f T_1)\right]. \tag{2-43}$$

According to the *Taylor Expansion* of $\sin(x)$:

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1},$$

we have:

$$\begin{aligned}\lim_{f\to 0} S(f)_{\Delta\Phi,DCPLL} &= \lim_{f\to 0} \frac{\mathcal{L}(f)}{\pi f T_1}\left[\pi f T - \frac{1}{4}\sin(4\pi f T_1)\right] \\ &= \lim_{f\to 0} \frac{\mathcal{L}(f)}{\pi f T_1}\frac{1}{4}\left[\frac{(4\pi f T_1)^3}{3!} - \frac{(4\pi f T_1)^5}{5!} + \frac{(4\pi f T_1)^7}{7!} - ...\right].\end{aligned} \tag{2-44}$$

Substituting $\mathcal{L}(f) = \sigma_c^2 \cdot f_{osc}^3/f^2$ into Eq. (2-44) we have:

$$\begin{aligned}
\lim_{f\to 0} S(f)_{\Delta\Phi,DCPLL} &= \lim_{f\to 0} \frac{\sigma_c^2 \cdot f_{osc}^3/f^2}{\pi f T_1} \cdot \frac{1}{4} \cdot \left[\frac{(4\pi f T_1)^3}{3!} - \frac{(4\pi f T_1)^5}{5!} + \frac{(4\pi f T_1)^7}{7!} - ... \right] \\
&= \lim_{f\to 0} \frac{\sigma_c^2 \cdot f_{osc}^3}{4\pi f^3 T_1} \cdot \left[\frac{(4\pi f T_1)^3}{3!} - \frac{(4\pi f T_1)^5}{5!} + \frac{(4\pi f T_1)^7}{7!} - ... \right] \\
&= \boxed{\frac{8 \cdot \sigma_c^2 \cdot f_{osc}^3 \cdot \pi^2 \cdot T_1^2}{3}}. \qquad (2\text{-}45)
\end{aligned}$$

For an oscillator with -110 dBc/Hz at 1 MHz frequency offset and oscillation frequency of 5.0 GHz, $\sigma_c = f^2 \cdot \mathcal{L}(f)/f_{osc}^3 = 8.94\,\text{fs}$. Taking $T_1 = 10\,\mu\text{s}$, for example, the same as in Figure 2-13a, the flat phase noise level at low frequency offset calculated from Eq. (2-45) is -75.8 dBc/Hz, which matches exactly what is shown in Figure 2-13a.

The 3 dB corner frequency can be calculated by solving

$$S(f)_{\Delta\Phi,DCPLL} = \frac{1}{2} \lim_{f\to 0} S(f)_{\Delta\Phi,DCPLL}. \qquad (2\text{-}46)$$

There is no analytical solution for it, however, it can be calculated numerically for every T_1. The 3 dB corners detected on spectrum and solved numerically are shown in Figure 2-14. From the slope of the curve in Figure 2-14b, $f_{corner} \approx 0.294 \cdot (1/T_1)$.

Now that the 20 dB/dec noise has been solved, a quick analysis follows on flat noise and 30 dB/dec noise. The flat part of the noise comes from the output buffer and is not filtered by DCPLL phase aligning. In principle, this flat noise can also be derived from the Wigner Spectrum. However, since it is clear that its Wigner spectrum is flat along either time or frequency axis, the derivation is omitted here. An output buffer contributes the same flat noise in both conventional phase noise spectrum and DCPLL phase noise spectrum.

For the 30 dB/dec noise in a conventional phase noise spectrum, its slope is reduced by 20 dB/dec because of the DCPLL's phase-aligning. Although the exact filtering bandwidth is not derived, it should be close to the DCPLL's filtering bandwidth for 20 dB/dec noise, which is approximately $0.294 \cdot (1/T_1)$. If $(1/T_1)$ is 25.0 MHz, then the DCPLL equivalent filtering bandwidth is 8.3 MHz. Unless the DCO is not properly designed, the 1/f corner frequency

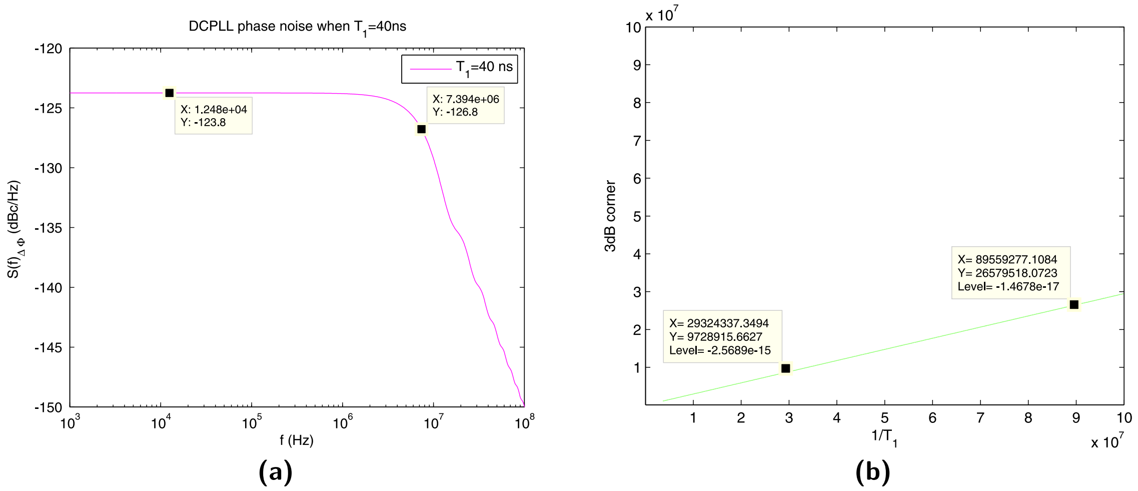
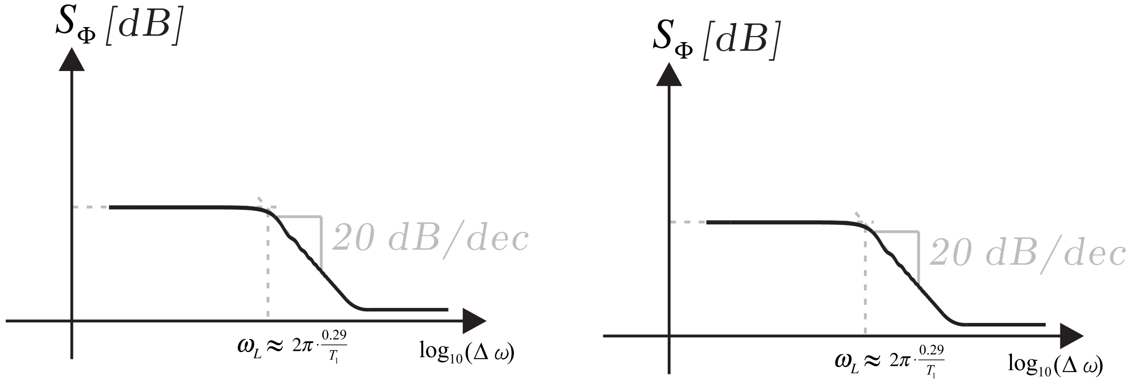


Figure 2-14: Equivalent filtering bandwidth of a DCPLL in terms of burst duration:
(a) $3\,\mathrm{dB}$ corner detected on phase noise spectrum
(b) $3\,\mathrm{dB}$ corner by solving Eq. (2-46)

does not go beyond 8.3 MHz and thus 30 dB/dec curve will not be observed in the DCPLL phase noise spectrum. The remaining 10 dB/dec is suppressed by the DCO's turning on/off operation, which is effectively 'large signal excitation'. Although this mechanism is difficult to analyze quantitatively, it is safe to assume that any trapped charge in MOSFETs is made uncorrelated from burst to burst by turning the DCO off and on. Thus, 1/f noise should not cause noticeable slope rise at low frequency.

Finally, the phase noise spectrum including up-converted thermal noise and white noise at the DCPLL's buffer is shown in Figure 2-15. It should be noted that the differences between a closed-loop and an open-loop DCPLL lie in 2^{nd}-order effects. When considering only the 20 dB/dec thermal noise and the white noise from the output buffer, open-loop and closed-loop behaviors of DCPLLs can be considered the same, especially when compared to the distinctive differences between open-loop and closed-loop behaviors of CPLLs in Figure 2-5. However, in actual implementations, DCPLLs are usually used in closed-loop configurations due to the existence of temperature drift and quantization noise from multiple sources such as the capacitor bank and the DTC.



(a) **(b)**

Figure 2-15: Open-loop and closed-loop spectrum for a DCPLL: (a) Open-loop (b) Closed-loop[1]

Chapter 3

Choice Between Ring and LC Oscillators

Regarding the choice of oscillator type, the author has to choose one out of the several kinds of oscillators in the category. Luckily, there are not many kinds of them. Oscillators can be divided into 3 main categories: LC oscillators, RC oscillators and crystal oscillators.

Crystal oscillators cannot be integrated on-chip and thus are ruled out first. The choice between LC oscillators and ring oscillators is a much more difficult one, which requires further thinking. There is quite some literature, however, on comparison between LC oscillators and ring oscillators in terms of various metrics such as power, area, noise, tuning range, etc. [2], [35] and [36].

3-1 General Metrics

Regarding noise and power, ring oscillators have a much lower quality factor than LC oscillators. They are much noisier when consuming equal power or more power hungry when achieving equal noise performance as LC oscillators. Furthermore, the oscillation period of a ring oscillator is proportional to the RC constant of its unit delay stage when the total number of stages in fixed. Thus, the maximum oscillation frequency of a ring oscillator in a fixed technology node is determined by the delay of an unloaded inverter. Because of this, LC oscillators dominate in applications where oscillation frequency in access of 10 GHz is required.

However, ring oscillators take much less chip area, therefore more integratable, and scale well with technology. In a more advanced technology node, ring oscillators can not only be made smaller but also achieve a higher maximum oscillation frequency. Ring oscillators are also known to be advantageous in tuning range. While the oscillation frequency of a ring oscillator is inversely proportional to the capacitor load and biasing current, the frequency of an LC oscillator is defined by the resonant frequency of its LC tank and the common way of tuning the frequency is by varying the tank capacitance, which usually has a limited range.

3-2 Start-up Time

Another important metric, especially for duty-cycled oscillators, is start-up time. LC oscillators are known to have a longer start-up time than ring oscillators. LC oscillators generally take Q cycles to reach full amplitude from thermal noise, where Q is the quality factor of the inductor used in the LC oscillator. Indeed, the long start-up time of conventional LC oscillators is the deterministic factor that hampers LC oscillators from being used in duty-cycled operations. This has been mentioned repeatedly in previous literature, such as [10], [11] and [9], where duty-cycled oscillators are needed and also where ring oscillators are chosen over LC oscillators.

[37] and [38] derive the transient during start-up though differential equations in time domain while [18] describes the state of the LC tanks during start-up through root locus of the poles of the solutions to the LC tanks' state equations, both showing the fact that both amplitude and frequency of the LC tanks in LC oscillators need to go through a settling phase before they reach steady state. According to the derived transient model in [37], the waveform of an

LC tank's output voltage when established from thermal noise is shown in Figure 3-1. Since the initial amplitude equals the level of thermal noise and thermal noise is a random process, the phase of the LC oscillator when it reaches full amplitude is also random.

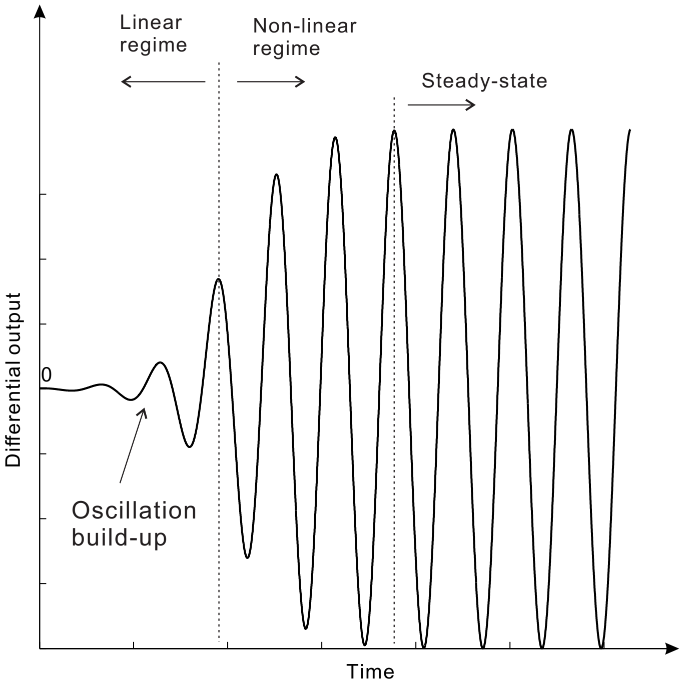


Figure 3-1: Typical LC tank waveform when triggered by thermal noise

While we can observe an amplitude settling process from what is shown in Figure 3-1, hence an implied frequency settling process, which meets neither the requirement of short start-up time nor accurate start-up phase, this is not the end of the story for LC oscillators. It has already been shown in [37] and [38] that, with some additional kick-start circuitry, LC oscillators can reach full amplitude within a few ns from the moment they are triggered. Despite the fact that the start-up time achieved in the above mentioned literature is on the long side compared to the start-up times of ring oscillators reported in [9] and [11] and that the phase of the oscillator is not well defined with regard to the trigger signal, it is shown in literature that there is possible solution to LC oscillators' start-up problem which is the most critical factor that stops them from being used as duty-cycled oscillators. This solution is found by the author and his supervisors, which will be introduced in detail later.

3-3 The Choice Made

As is pointed out, LC oscillators are less noisy than ring oscillators but have always been avoided in duty-cycled radios because conventional LC oscillators have a comparatively smaller tuning range and a longer start-up time, as well as an undefined start-up phase. As long as these three obstacles are dealt with, there is no stopping from using LC oscillators for duty-cycled Ultra-Wideband (UWB) radios.

First, regarding tuning range, several recent works have shown that LC oscillators with up to 2 inductors can cover a rather wide frequency range: 3.24–8.45 GHz [39] and 2.4–5.3 GHz [40]. The literature in [41] even covers 0.85–7.1 GHz with one tunable inductor. Note that in Section 1-3, it is shown that as long as up to 2 inductors can cover 3.5–7.0 GHz, the tuning range specification for Impulse Radio Ultra Wideband (IR-UWB) radio can be met. So, tuning range is not a problem.

Regarding fast start-up and well-defined start-up phase, the author and his supervisors managed to find out a way to qualify LC oscillators for these specifications as well as, if not better than, ring oscillators. While this special technique does not come at no cost, the side effects of this technique will be analyzed quantitatively and it will be shown that the cost is outweighed by the benefits. Judging by the fact that the phase noise of LC oscillators is usually 20 dB better than that of ring oscillators, a cost of a few dBs will still leave LC oscillators with an advantage somewhere near 15 dB.

Another consideration for the author to take the LC oscillator approach is because it is the approach that has never been taken, in Duty-Cycled PLLs (DCPLLs). It is a brand new area to explore and if the author succeeds, this work will be helpful for successors to continue exploring this approach and make big steps in enhancing battery life of UWB radios.

In summary, after analyzing the general pros and cons and discovering the possibility of success on the LC oscillator approach, the author decided to continue the project with this approach.

Chapter 4

Instantaneous Start-up Technique for LC Oscillators

In thesis, an instantaneous start-up technique for LC oscillators is proposed. By instantaneous, it is specifically meant that the start-up technique meets both the requirement of short start-up time and accurate start-up phase. The theory as well as the circuit implementation will be proposed.

4-1 Traditional Start-up Mechanism

As is mentioned in Subsection 1-6-2, the simplest, yet effective way to start an LC oscillator is to let it build its amplitude from thermal noise. This solution is valid for applications where start-up time is a minor design metric. However, it has the drawback of requiring many periods to amplify thermal noise to the final oscillation amplitude. Moreover, the phase of the oscillator after start-up is random. These two drawbacks have stopped LC oscillators from being used in duty-cycled systems where burst duration is on the order of a few tens of ns. [1]

4-2 Existing Fast Start-up Techniques

Prior to this work, there have been several publications aiming to reduce the start-up time of LC oscillators with various kinds or tricks.

In [37], an initial voltage larger than thermal noise is injected into the tank and effectively reduces start-up time. One of the side-effects is that the split current sources at bottom compromises the circuit's Common-Mode Rejection Ratio (CMRR).

In [38], the LC oscillator is deliberately made asymmetric to achieve a shorter start-up time, which compromises the oscillator's CMRRas well.

In [42], the LC oscillator is left to build its oscillation from thermal noise and the dominant component in start-up time is proportional to $\sqrt{Q}$. A short start-up time is achieved by sacrificing Q below 2.

In conclusion, what [38] and [42] have in common is that they depend on sacrificing circuit performance to reduce start-up time: the larger the sacrifice, the less time it takes to start.

In terms of results, what all of the 3 publications mentioned above have in common is that they neither achieve a shorter start-up time than ring oscillators nor guarantee an accurate start-up phase.

4-3 Existing Instantaneous Start-up Technique for Ring Oscillators

Among the little literature on instantaneous start-up ring oscillators, such as [11] and [43], instantaneous start-up is realized by breaking the loop of a ring oscillator and forcing the differential output nodes of each stage to be either supply voltage or ground, before the

[1] In some duty-cycled systems, the burst duration is on the order of tens of ms. For such duty-cycled systems, the long start-up time of traditional LC oscillators is acceptable.

trigger point. With this method, there would be no oscillation build-up process as in LC oscillators and the start-up time, or more precisely, frequency settling time, is limited by the charging time at nodes with large capacitance, such as the gates of current-mirror transistor and the gate delay of switches for duty-cycling operation. It takes only 1–2 ns after an accurate triggering signal, which is aligned with the reference clock, for all every internal nodes to be stabilized and to readily deliver a sufficiently accurate frequency.

4-4 Proposed Instantaneous Start-up Technique for LC Oscillators

The instantaneous start-up technique in ring oscillators can actually be borrowed by LC oscillators.

4-4-1 Theory

First, consider a parallel RLC tank. As a 2^{nd}-order system, the state of the system at any time is completely defined by the combination of the current in the inductor and the voltage across the capacitor (or one of the two variables and its 1^{st}-order derivative). The solutions of the state variables at $t + \Delta t$ depend only on the solutions at t, regardless of the state variables from negative infinity to t. This property reveals that if an LC oscillator, which is effectively a 2^{nd}-order system, is assigned an initial state that is identical to one of its states when it is delivering an accurate frequency, it will be already in steady state from the moment it is turned on. This is better explained in the following example.

Consider two tanks with identical circuitry but different initial states. Figure 4-1 shows such a parallel RLC tank together with a negative conductance modeling the active circuitry part in LC oscillators, according to the negative resistance approach introduced in [18].

A tank with this circuitry, Tank1, is released from a state set by thermal noise. For small amplitude, such as the linear regime dictated in Figure 4-2, $1/R - G(u) < 0$, the poles to Eq. (1-12) are on the Right-Half Plane (RHP) and the oscillation amplitude increases. When the oscillation amplitude continues to increase, transistors start to enter the non-linear regime,

$G(u)$ gradually decreases and the poles gradually move towards the imaginary axis. Finally, in steady state, $1/R - G(u) = 0$ and the poles are located on the imaginary axis. Now, another tank, Tank2, with the same circuitry shown in Figure 4-1, is released with an initial state that is identical to that of Tank1 at $t = t_0$. In the observation window of $t \geq t_0$, the waveforms of Tank2 are identical to those of Tank1. By creating another axis $t^{'} = t - t_0$, we can say that Tank2 is in steady state since $t^{'} = 0$, thus start-up time equals zero.

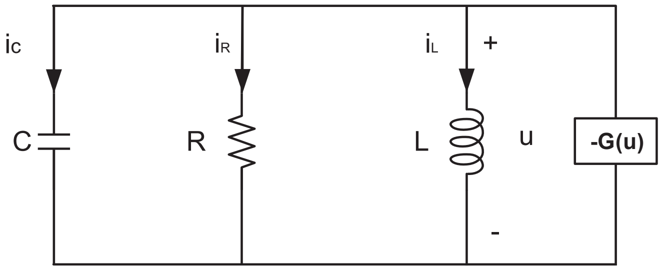


Figure 4-1: Equivalent circuit model for LC tank with active core

The root locuses for Tank1, with implied time axis t, and Tank2, with the implied time axis $t^{'}$, are shown in Figure 4-3. The initial poles location and steady-state pole locations of Tank2 are identical.

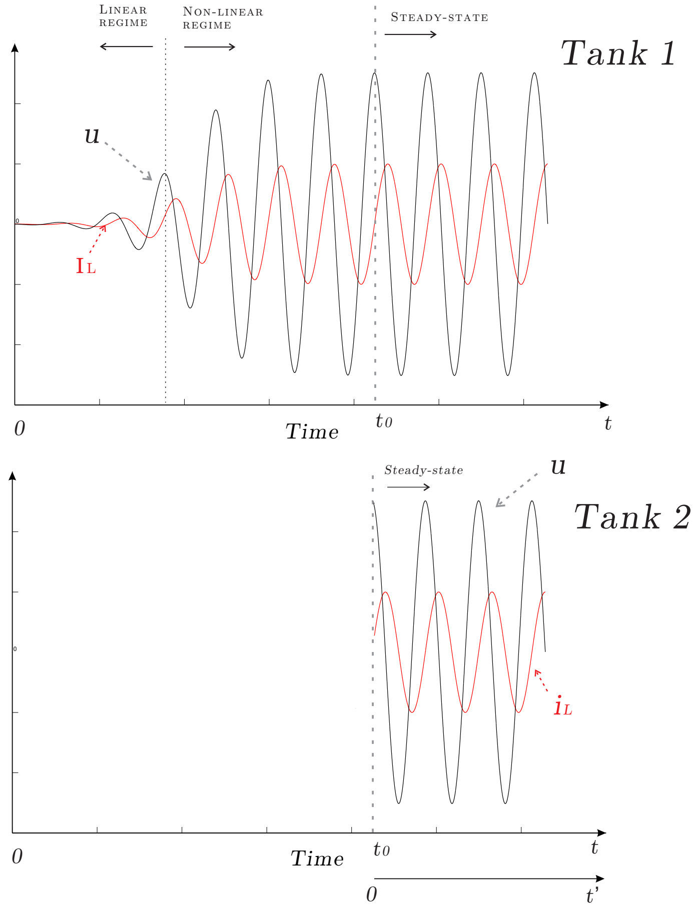


Figure 4-2: Waveforms of two LC oscillators tanks with identical circuitry but different initial states

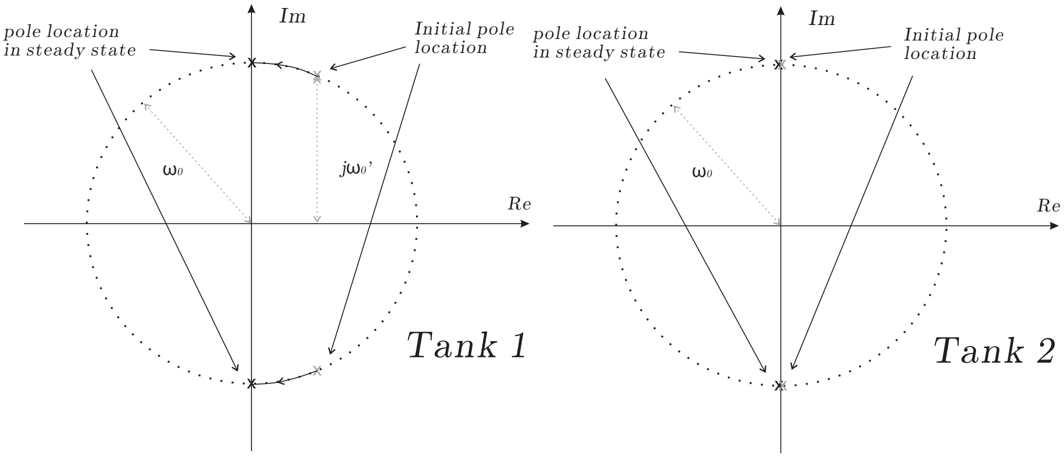


Figure 4-3: Root locuses of two tanks with identical circuitry but different initial states

4-4-2 Circuit Implementation

Regarding circuit implementation, the basic principle is to store the LC tank's state variable. The easiest state to store is when all energy is in electrical form and inductor current is zero, which is illustrated in Figure 4-4. While there are multiple approaches to implement this principle, the circuitry shown in Figure 4-5 is used in this design.

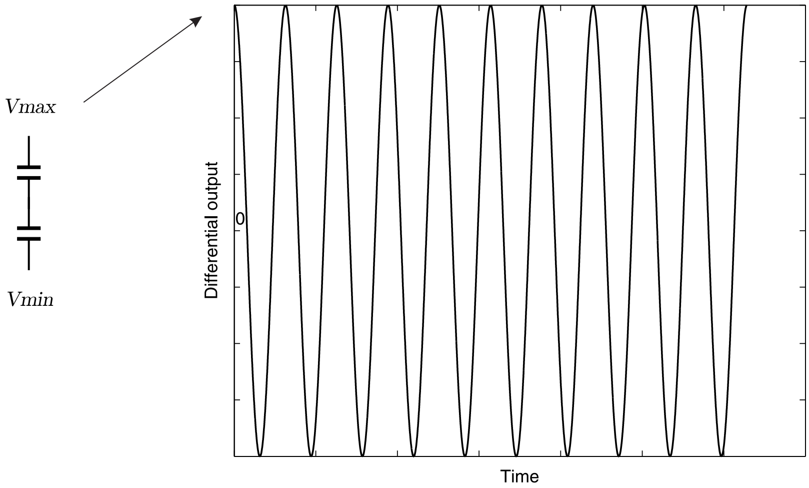


Figure 4-4: Basic principle of implementing instantaneous start-up feature in circuit

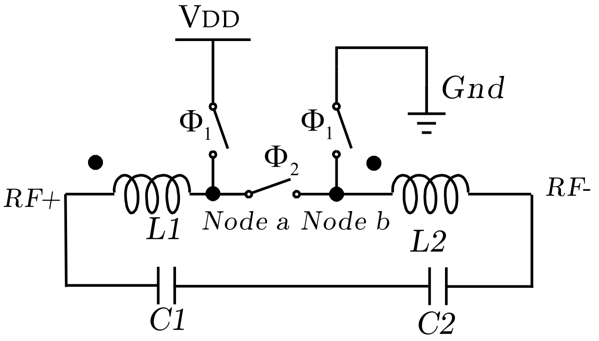


Figure 4-5: Simplified representation for the instantaneous start-up circuitry

Before start-up, the switches belonging to Φ_1 are closed, bringing *Node a* to Vdd and *Node b* to Gnd. In this way, energy is stored in the capacitor bank. At the instant of switching on the DCO, switches belonging to Φ_1 are opened while the one belonging to Φ_2 is closed. In this way, the tank will be instantaneously connected.

The waveform in Figure 4-4, however, is a bit too idealistic. Chances are that the start-up amplitude does not 100% match the steady-state amplitude. Scenarios of starting from a smaller or larger amplitude than that at steady state are shown in Figure 4-6. The larger the deviation between initial amplitude and steady-state amplitude is, the longer it takes for the tank to reach steady state within certain error. Because of this, to ensure that the steady-state amplitude is close enough to the initial amplitude, which is about half of Vdd, special DCO design together with an extra amplitude detector is needed, which will be covered later.

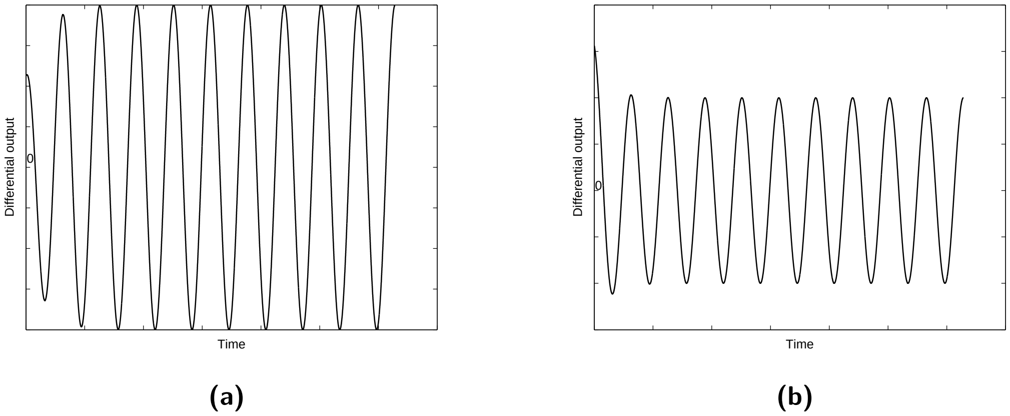


Figure 4-6: Scenarios when starting from a smaller or larger amplitude:
(a) Starting from a smaller amplitude (b) Starting from a larger amplitude

Chapter 5

Design of the Inductor

The implementation of the instantaneous start-up circuitry is a bit more complicated than suggested by Figure 4-5. It requires design of a split inductor that can be either opened or closed.

5-1 Design Procedure

Design of the split inductor follows the following procedures. Firstly, the inductor as a whole is designed to meet RF specifications such as inductance, quality factor and Self-Resonant Frequency (SRF). Inductor parameters such as inner diameter (Din), width (W) and spacing (S) can be determined at this step. After this, the inductor layout is generated with an automatic tool and a part of the conductor in the center is replaced with a properly designed semiconductor switch. The resulting 2-port switchable inductor can be considered as a 4-port inductor, with its center 2 ports connected by a semiconductor switch. The 4-port inductor is simulated in ADS Momentum and a lumped model is used to represent it for ease of PSS simulation in Cadence Spectre. The effect of the semiconductor switch on inductor metrics as well as the accuracy of the lumped model will be quantified.

5-2 Design for RF Specifications

5-2-1 Self-resonant Frequency

The specification of tuning range is 5.0 GHz to 7.0 GHz, which means the inductor should follow the design rules of a wide-band inductor. Usually, SRF of narrow-band inductors are chosen to be twice the operation frequency. Thus, the wide-band inductor is designed to have a SRF at three times its maximum operation frequency, which is 21.0 GHz.

5-2-2 Inductance

The choice of inductance depends on the power budget. The power required by a DCO is, to a first order, inversely proportional to $R_p \approx \omega LQ$. As is mentioned in [44, Ch. 5], it is easier to increase R_p by increasing L instead of increasing Q. While a larger inductance would reduce the power consumption of a DCO, it not only lowers SRF of the inductor but also requires a smaller resonant capacitance at the same resonant frequency, leading to a smaller acceptable parasitic capacitance from the remaining part of the circuit. The power budget of this DCO is recommended to be within 1 mA. After a first-round simulation, it was discovered that an inductance of 1.0 nH requires a power consumption that is much larger than the power budget. In order to meet the power budget, an inductance of 2.0 nH is chosen to be a starting point.

5-2-3 Number of Turns

Usually, the number of turns increases with the inductance needed. An inductance value of 2.0 nH can be possibly implemented in 2 turns with a large diameter or 4 turns with a small diameter. After a discussion with his supervisor, the author decided to avoid an odd number of turns due to the consideration that the inductor will later be modified to include a semiconductor switch at the center. With an even number of turns, the center of the inductor will be at the same side as its RF outputs, thus it will be easier to route control signals to the switch. Without the special constraint of staying at an even number of turns, 3 turns is probably advantageous over either 2 turns or 4 turns. A list of metrics under different combinations of parameters of a 4-turn inductor is shown in Table 5-1, where Din stands for inner diameter, S stands for spacing between windings, W stands for conductor width, SRF stands for self-resonant frequency, Q stands for quality factor, L stands for inductance and Rs stands for series resistance. FOM_1 and FOM_2 are two FOMs measured at 5.0 GHz, which are defined in Eq. (5-1). The larger ωLQ is, the less the power required from the DCO. FOM_2 complements FOM_1 by taking into consideration the resistance from the switch that will be included in later, which is estimated to be 1.5 Ω during the early part of the design phase.

Table 5-1: List of metrics under different parameters of a 4-turn inductor

Din (µm)	S (µm)	W (µm)	SRF (GHz)	Q	L (nH)	Rs (Ω)	FOM_1	FOM_2
85	3.0	4.0	19.2	13.85	2.46	5.6	34.1	26.8
80	2.5	4.5	19.1	14.07	2.29	5.1	32.2	24.9
80	2.5	4.0	19.5	13.60	2.30	5.3	31.3	24.4
80	3.0	4.0	20.2	13.60	2.29	5.3	31.1	24.3
75	2.0	4.5	19.4	13.73	2.14	4.9	29.4	22.5
75	2.0	4.0	19.8	13.31	2.15	5.0	28.6	22.0

$$FOM\_1 = \omega LQ = \frac{(\omega L)^2}{Rs}. \tag{5-1a}$$

$$FOM\_2 = \frac{(\omega L)^2}{Rs + R_{switch}}. \tag{5-1b}$$

It can be seen that when continuously decreasing the diameter of the inductor, the FOMs drop linearly. This is easy to understand because, at some point, the inductor windings are so close to each other that they behave like a single conductor. For relatively reasonable diameters such as 85 µm, the inductance value is on the large side of the estimated 2.0 nH. Thus, a 2-turn inductor is tried and it turns out to meet the requirements.

5-2-4 Diameter

Since the inductance strongly depends on the number of turns and the diameter while weakly depending on conductor width and spacing, a rough scope of needed diameter can be determined once the inductance and the number of turns have been determined. Table 5-2 shows inductor metrics of a 2-turn inductor from schematic simulation when sweeping Din around 200 µm. After simulating the inductor layout in ADS Momentum, it was discovered that the schematic simulation gave an optimistic estimation of Rs by approximately 30%, which is why the final chosen parameter set does not have a highest FOM in Table 5-2. Moreover, from later simulations with other parts such as the active core and the extracted capacitor bank, it was discovered that an inductance of 2.0 nH would be too large for the oscillator to oscillate at 7.0 GHz, which is why the final chosen value is smaller than 2.0 nH. More accurate metrics of the inductor, from simulating it in ADS Momentum will be shown later.

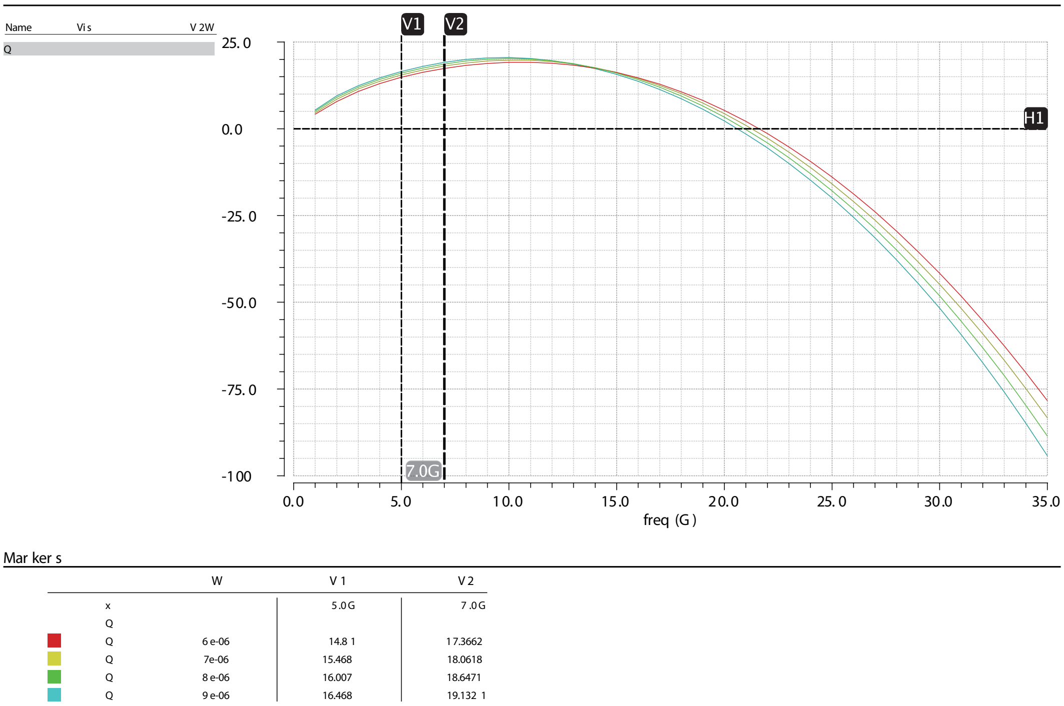
Table 5-2: Sweeping Din of a 2-turn inductor locally

	W = 8 µm, S = 1.85 µm		
	Din = 200 µm	Din = 205 µm	Din = 210 µm
L (nH)	1.72	1.77	1.83
Q	15.9	16.0	16.1
Rs (Ω)	3.39	3.48	3.57
SRF (GHz)	21.46	20.94	20.42
FOM_1	27.3	28.3	29.5
FOM_2	19.0	19.8	20.8

5-2-5 Width and Spacing

The FOM of the inductor is not a monotonic function of either W or S. Increasing W reduces the dc resistance of the inductor, which does not benefit the quality factor at high frequency much, since loss at high frequency is dominated by skin effect and current crowding effect. Figure 5-1 shows the effect on quality factor from varying W.

At the same time, increasing W reduces inductance and SRF. Reducing S would result in more inductance with the same length of conductor wire but the coupling between windings



Markers

		W	V1	V2
	x		5.0G	7.0G
	Q			
🟥	Q	6e-06	14.81	17.3662
🟨	Q	7e-06	15.468	18.0618
🟩	Q	8e-06	16.007	18.6471
🟦	Q	9e-06	16.468	19.1321

Figure 5-1: Sweeping W locally

increases significantly beyond a certain point. The minimum spacing allowed by the process needs to be taken into consideration as well.

Basically, W and S are chosen after sweeping their values around the estimated value and making adjustments after extracting the layout with ADS Momentum. The chosen parameter set is the Din = 205 µm set shown in Table 5-2. Inductor layout is shown in Figure 5-2.

5-3 Design for Instantaneous Start-up

5-3-1 Design of the Center Switch

The design of the switch needs to take into consideration aspects such as resistance, parasitic capacitance, area and charge injection. In order not to substantially deteriorate inductor quality factor, on resistance of the switch needs to be no more than a few Ωs. Before this work, there was also literature where low-ohmic semiconductor switches are used with inductors [45],

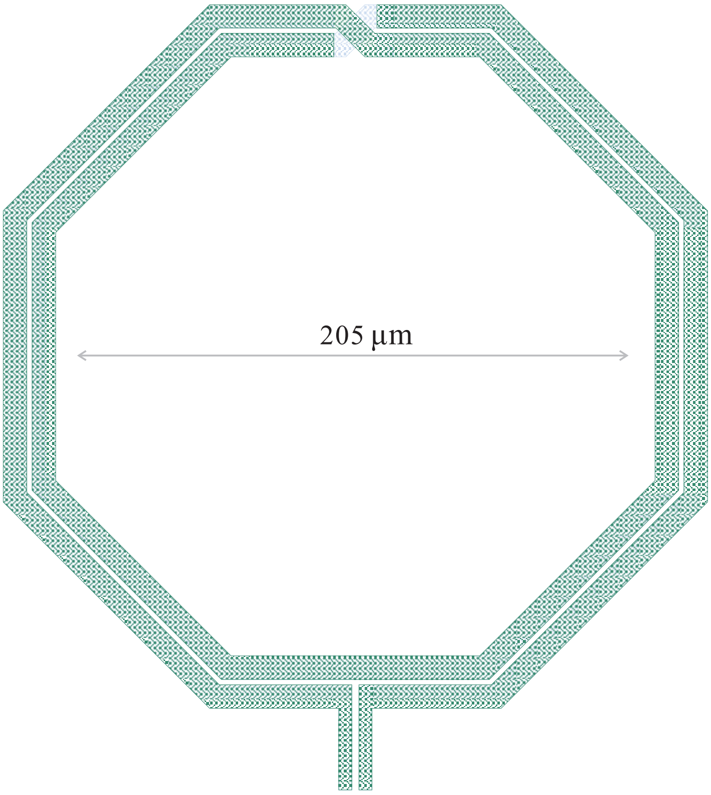


Figure 5-2: Original inductor layout

[46], [47], [48], [49], [50], although purposes of the switches in those publications are to tune the inductance. In a 0.18-µm process in [50], the switch resistance was made as low as 0.4 Ω by over-sizing the transistors.

Thanks to the fact that the switch is located at the center of the inductor, there is almost no voltage swing across the low-impedance switch and thus the switch can be sized very large without worrying loading the tank with parasitic capacitance. Judging by the fact that estimation of the inductor series resistance is 3.48 Ω, the switch is designed to have an impedance a little below 2 Ω during on state. The area needed at this impedance level is relatively small compared to the area of the inductor so that when the switch is added to the inductor layout, the inductance change, which is difficult to model accurately, is expected to be negligible.

In general, NMOS is favored over PMOS in realizing switches. However, using NMOS only would cause quite large charge injection because the switch is huge. Thus, an NMOS and

PMOS combination is used. The switches sizes are 576 µm/40 nm for NMOS and 192 µm/40 nm for PMOS. The occupied area is 30.0 × 37.0 µm^2, which is much smaller than the area of the inductor. When used in the DCO, the switch is gated by 1.1 V while its common voltage is half of the DCO supply voltage, which is 0.4 V. Figure 5-3 shows the RF impedance of the switch. The layout of the switch is extracted and the extracted model is shown in Figure 5-4.

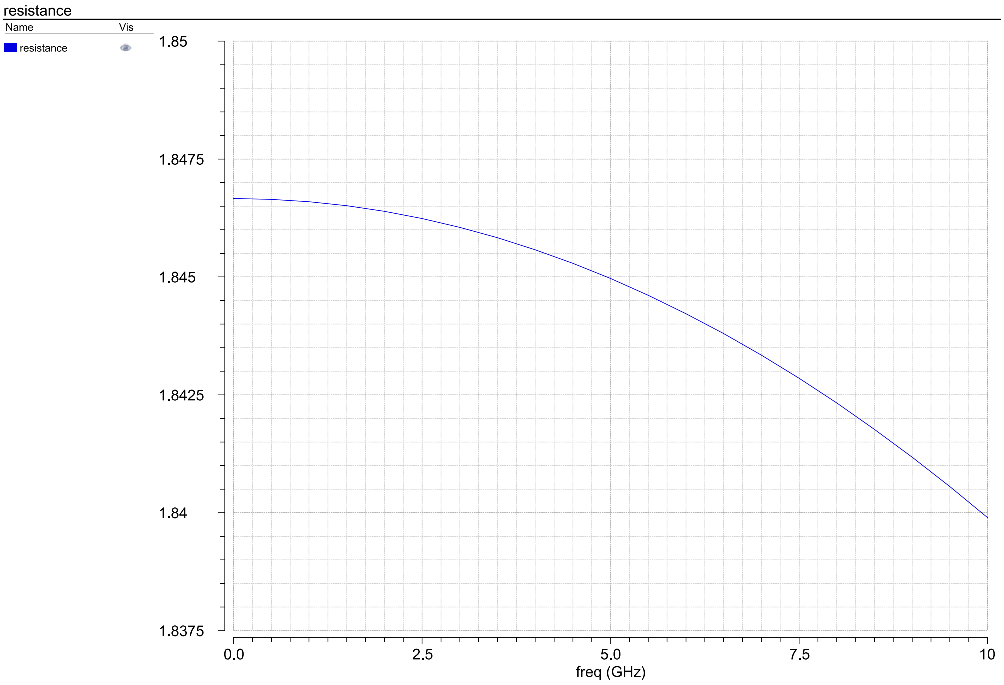


Figure 5-3: RF impedance of the switch

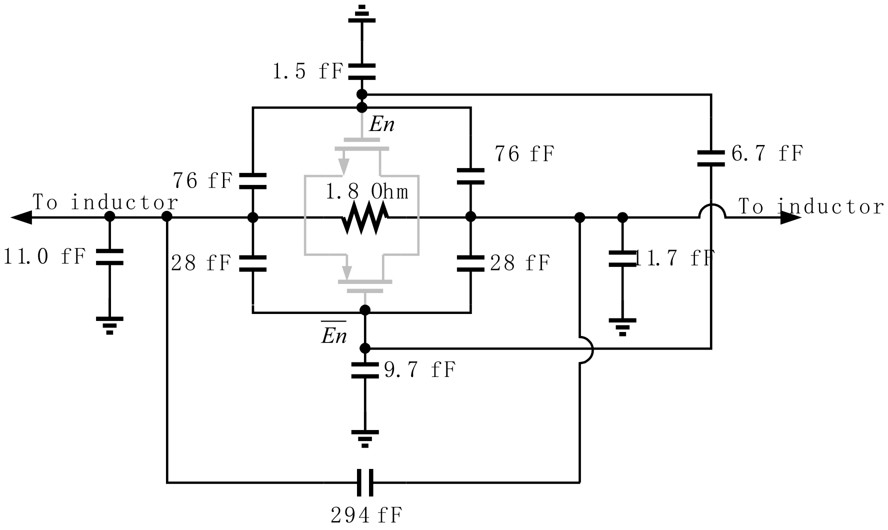


Figure 5-4: Extracted model of the switch

5-3-2 Modification to the Inductor Layout

To accommodate the switch at the center, the original inductor layout in Figure 5-2 is modified and shown in Figure 5-5. S-parameters of this 4-port inductor is extracted by ADS Momentum. The inner pins of this 4-port inductor are set on M1 so that the via resistance between M1 and inductor metal layer is also taken into account.

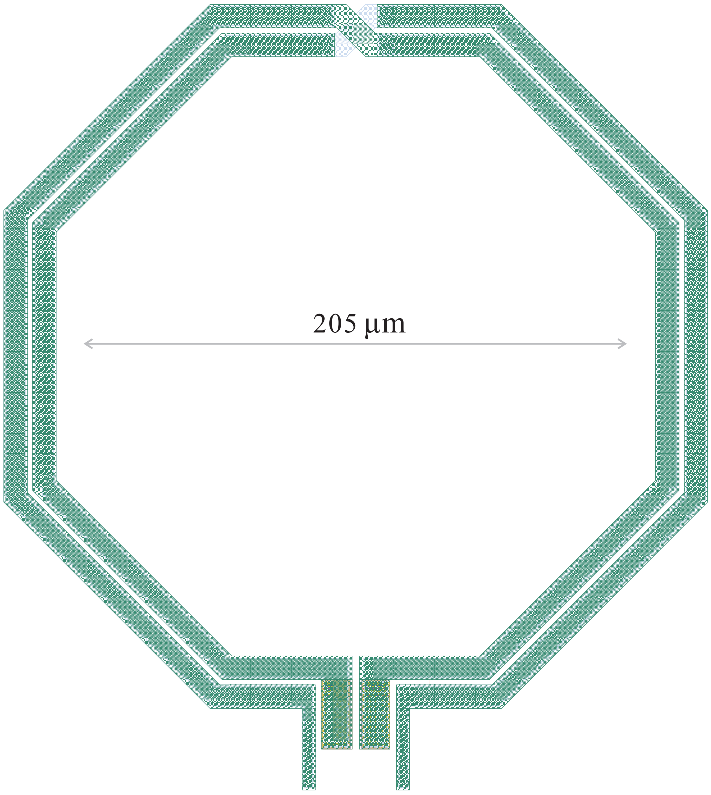
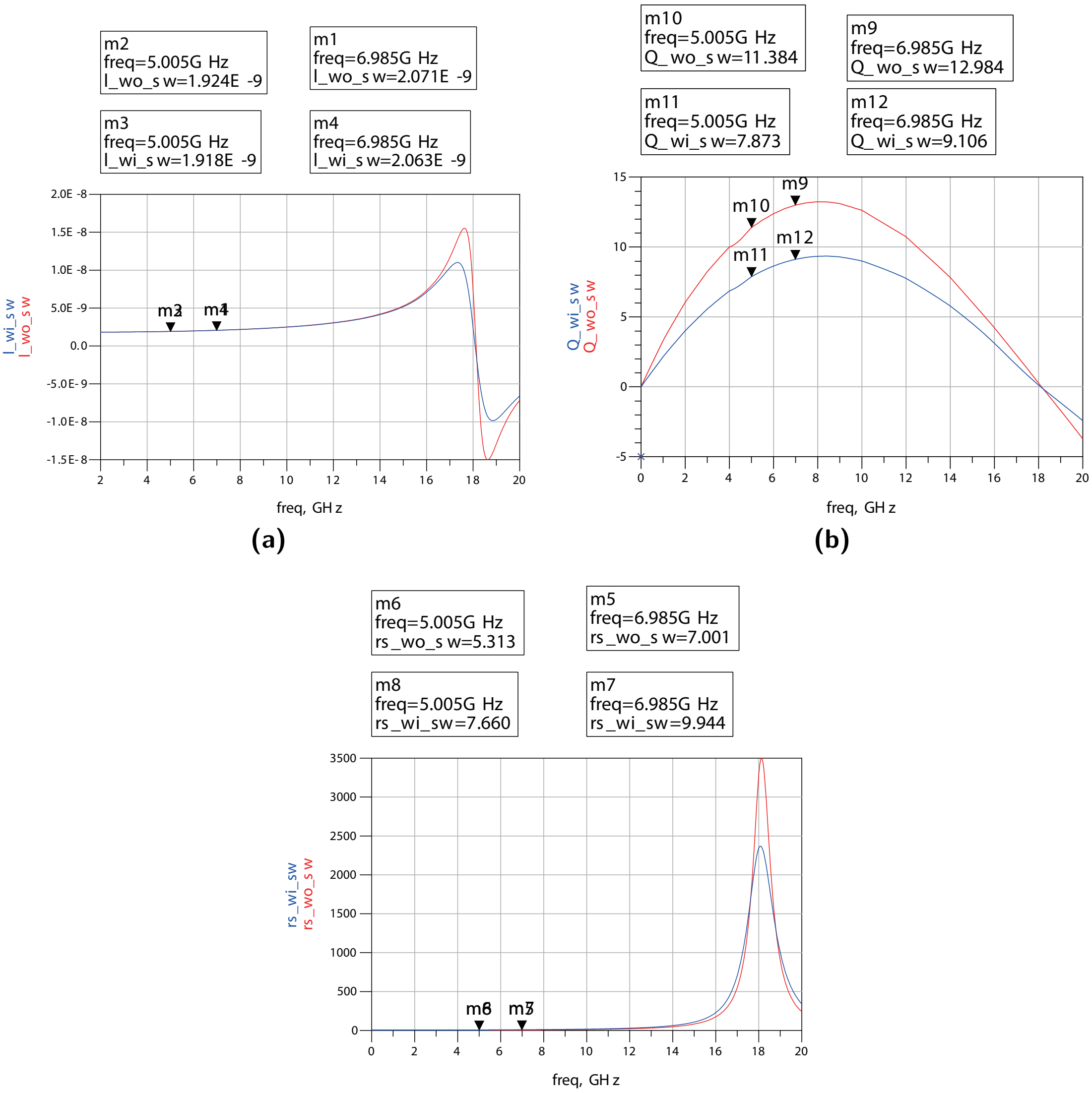


Figure 5-5: Inductor layout cut at the center

5-4 Simulation of the Inductor wi/wo Center Switch

S-parameters for the 4-port inductor shown in Figure 5-5, as well as the schematic of the switch, are imported into ADS. Two simulation sets are performed: one with the inner two pins shorted, the other with the inner two pins connected by the switch. Inductance, quality factor and series resistance for these two configurations are shown in Figure 5-6. It can be seen that there is no loss in tuning range from the parasitic capacitance of the switch while there is an approximately 30% loss in quality factor. Now, it can be safely assumed that LC oscillators are feasible in duty-cycled applications.



(a)

(b)

(c)

Figure 5-6: Inductor metrics wi/wo switch: (a) Inductance (b) Quality factor (c) Resistance

5-5 Lumped Model for the 4-Port Inductor

Since the S-parameter black box representation is not convenient for PSS simulation in Cadence Spectre, a lumped model is used to fit the parameters of the 4-port inductor, shown in

Figure 5-7. Several models, from simple to complex, have been tried until a complex model showed a quite will fitting curve. Model fitting is done with the automatic optimization tool in ADS. This lumped model is shown in Figure 5-7 while value of the parameters are listed in Table 5-3.

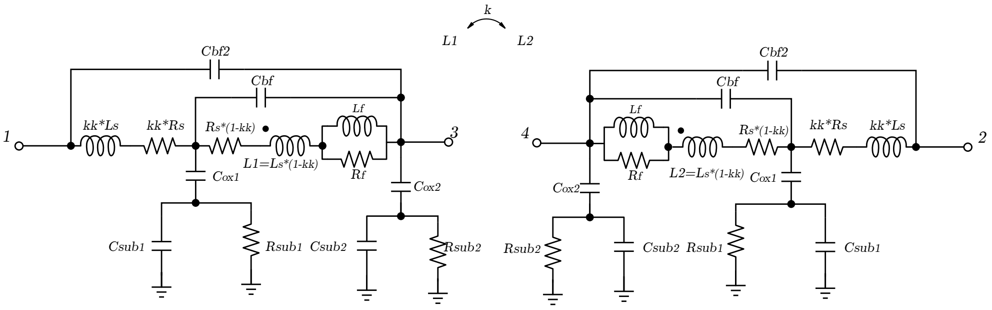


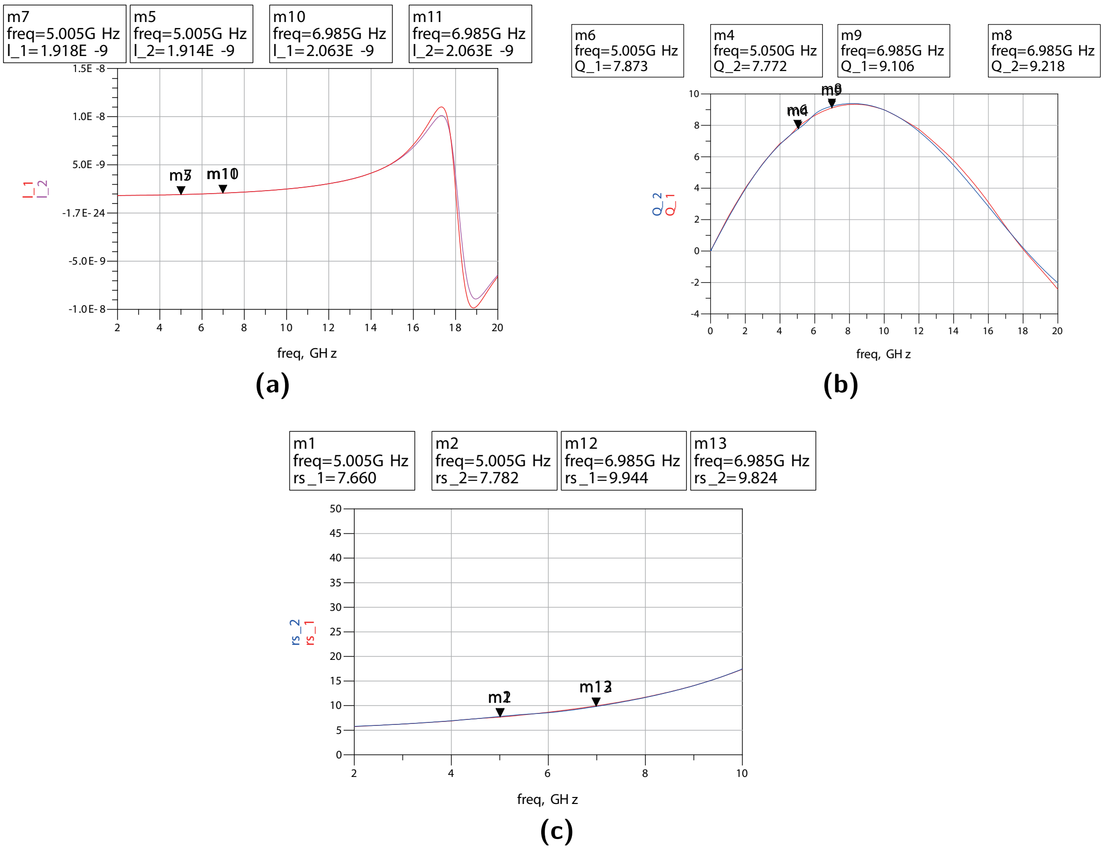
Figure 5-7: Lumped model for the 4-port split inductor

Table 5-3: Parameters of lumped inductor model

Cbf2	0.079693 pF	Cox1	1.08095 pF
Cbf	0.0234492 pF	Csub1	0.184208 pF
Rs	1.76907 Ω	Rsub1	100.515 Ω
Ls	0.855323 nH	Cox2	0.931917 pF
kk	0.834133	Csub2	0.188675 pF
Rf	1.17281 Ω	Rsub2	6055.83 Ω
Lf	0.0279408 nH	k	0.123128

The lumped model gives a very good matching within the frequency range of interest. After connecting the two center pins by the switch, the inductor is simulated as a 2-port device. Figure 5-8 shows simulated inductor metrics with S-parameter box and lumped model respectively. The red curve represents the S-parameter-box simulation while the blue curve represents the lumped-model simulation.

Finally, the lumped model is imported to Cadence Spectre and used for further design and simulations. Inductor metrics for the 2-port inductor, composed of a 4-port lumped inductor model and the switch, are verified in Cadence Spectre and can be seen in Figure 5-9. The inductor layout, together with the switch at the center, can be seen in Figure 5-10.

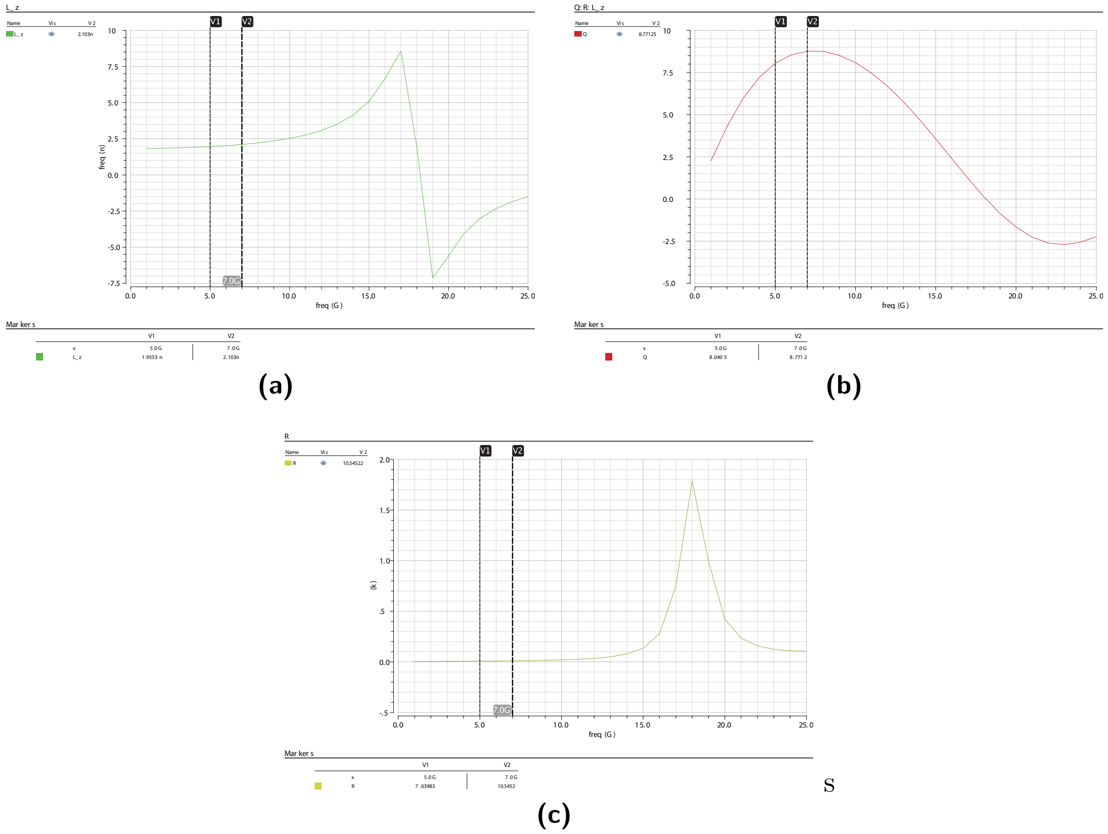


(a)

(b)

(c)

Figure 5-8: Inductor metrics from S-parameter box and lumped model: (a) Inductance (b) Quality factor (c) Resistance



Mar ker s

		V1	V2
	x	5.0G	7.0G
🟩	L_ z	1.9553 n	2.103n

(a)

Mar ker s

		V1	V2
	x	5.0G	7.0G
🟥	Q	8.040 5	8.771 2

(b)

Mar ker s

		V1	V2
	x	5.0G	7.0G
🟨	R	7 .63983	10.5452

(c)

Figure 5-9: Inductor metrics after importing the lumped model into Cadence Spectre: (a) Inductance (b) Quality factor (c) Resistance

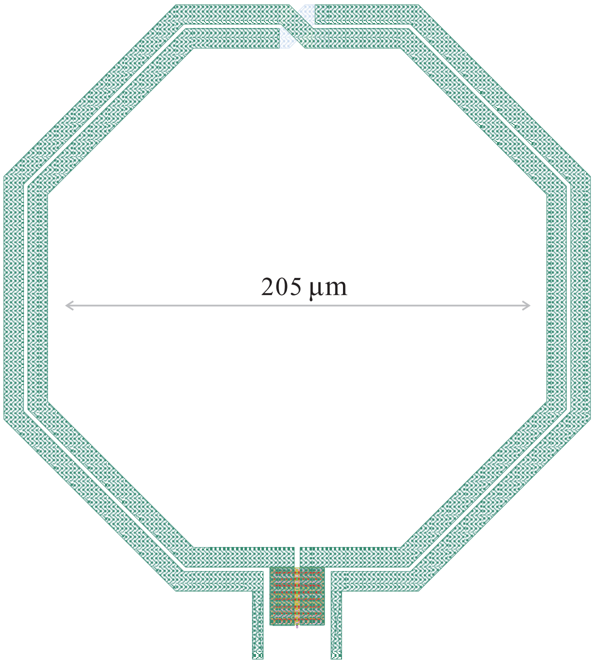


Figure 5-10: Layout of split inductor with switch

Chapter 6

Design of the DCO

6-1 Choice of Oscillator Architecture

The first step in DCO design is to choose a proper oscillator architecture. Figure 6-1 lists some of the most commonly used oscillator architectures, covering Class B, Class C and Class D. Their compatibilities with the instantaneous start-up circuitry will be discussed in this section.

Figure 6-1a is a tail-biased Class B LC oscillator. It is not compatible with the instantaneous start-up circuitry because the common mode voltage during oscillation is Vdd, which means the maximum voltage at one of the output nodes would exceed the supply voltage. The instantaneous start-up circuitry needs to charge one of the output nodes to a level close to the maximum voltage during oscillation. Charging the output node above supply requires extra boot-strapping circuitry. Moreover, it brings reliability concerns of semiconductor devices. Thus, it is out of the list.

Figure 6-1b is similar to Figure 6-1a, except that the original biasing is moved to the top. Compared with Figure 6-1b, it has a large improvement in that the oscillation voltage is now

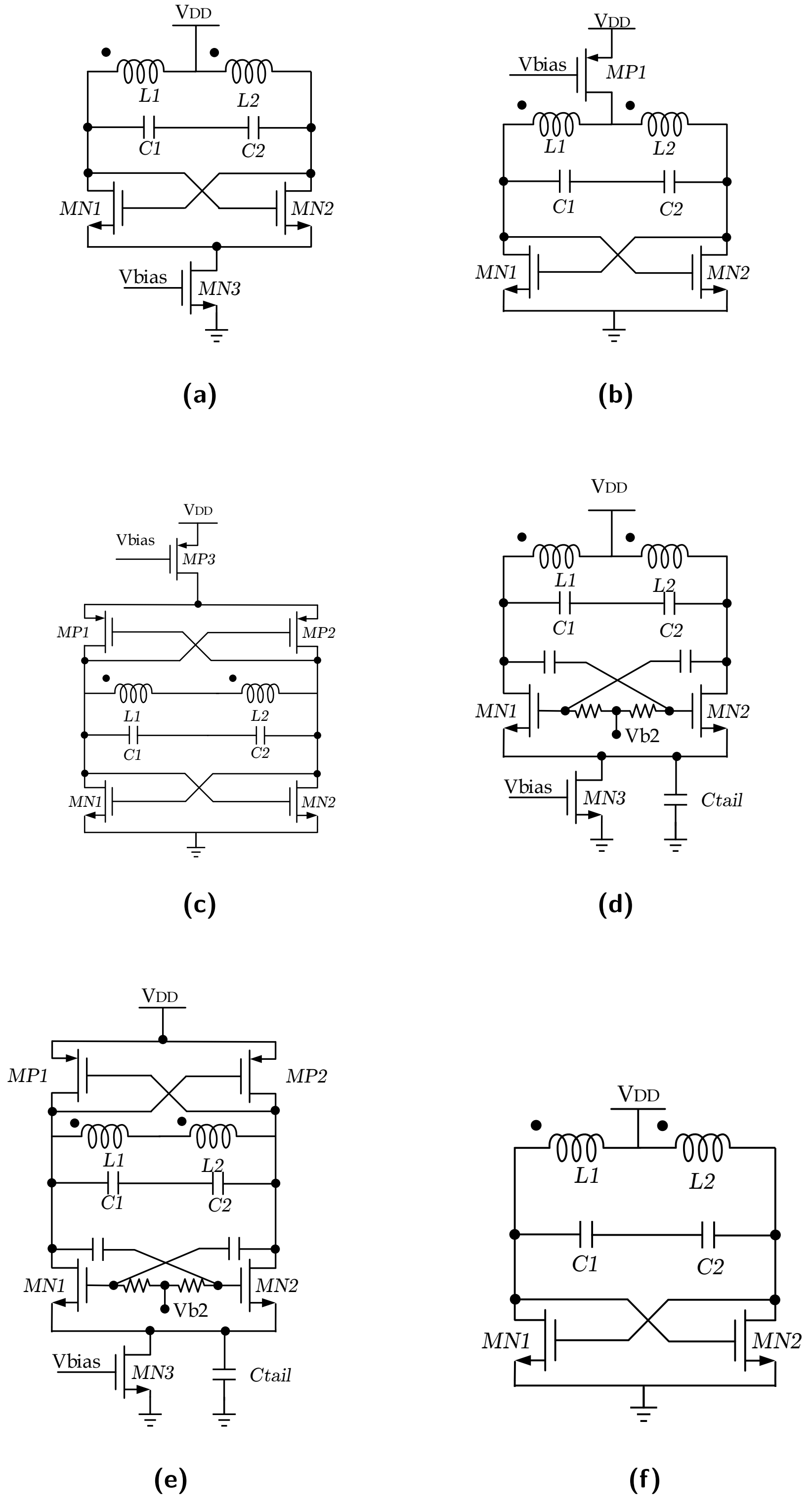


Figure 6-1: List of existing commonly used LC oscillators: (a) Tail-biased Class B (b) Top-biased Class B (c) Complementary Class B (d) Class C [51], [52] (e) Push-pull Class C [53] (f) Class D [54]

within Gnd and Vdd. Thus, it is possible to pre-charge one of the oscillation nodes to Vdd before start-up. However, taking into account the fact that a switch needs to be placed at the center of the inductor, the structure in Figure 6-1b not only requires multiple switches at the conjunction point between inductor and biasing transistor, but also the over-drive voltage of the switch, presumably NMOS, would be quite small. Thus, it is not suitable for the instantaneous start-up circuitry.

Figure 6-1c is better than Figure 6-1b in two aspects. Firstly, center of the inductor is not connected to biasing transistors, which makes the center switch for instantaneous start-up easy to implement. Secondly, the common-mode level of the inductor center node is about half Vdd, thus the over-drive voltage for the switch at the inductor center would be larger than that in Figure 6-1b. Moreover, as is pointed out in [18], this complementary structure can provide twice the oscillation amplitude with the same power consumption compared to either Figure 6-1a or Figure 6-1b. The catch is that it reaches the voltage limited region earlier than the above two structures.

Figure 6-1d shows the schematic of a Class C LC oscillator. The most obvious advantage of Class C oscillator over a Class B oscillator is a higher efficiency in converting DC power to RF power. However, transistor gates in Class C oscillators are usually dc decoupled from the RF outputs and have a separate dc biasing voltage, which can be seen in both Figure 6-1d and Figure 6-1e. This dc decoupling characteristic is not favored by an instantaneous start-up DCO, which will be explained in detail later in Section 6-5. In short, the ac coupling couples a transition of twice the amplitude of oscillation during the first few cycles while it only couples one oscillation amplitude during steady state. Thus, Class C is not a first choice. It does not mean, however, that Class C is incompatible with instantaneous start-up circuitry. As can be seen later in Chapter 6-5, some work-around is found to largely alleviate the problem caused by dc decoupling. Furthermore, the instantaneous start-up circuitry can be a potential solution to ensure proper start-up of the Class C LC oscillator, which has been tried to solve in various publications with feedback loops, such as [55] and [56]. However, as this is the first-ever design to use the proposed instantaneous start-up technique, the author has tried to keep design complexity at a controllable level.

Next, the Class D LC oscillator is shown in Figure 6-1f [54]. It can achieve an even higher efficiency than Class C. Its oscillation amplitude is also pretty well defined, which is a bonus point for instantaneous start-up circuitry. However, what also increases with the efficiency is its design complexity. Other than that, it is very promising to use it with the instantaneous start-up circuitry.

Finally, the complementary Class B structure shown in Figure 6-1c is chosen as a starting point of the DCO design due to its compatibility with start-up circuitry and moderate complexity in implementation. Structures in Figure 6-1a and Figure 6-1b are avoided because of compatibility issues while structures in Figure 6-1d, Figure 6-1e and Figure 6-1f are not chosen due to complexity concerns in implementation.

6-2 Current Biasing vs. Voltage Biasing

While complementary Class B is chosen as the starting point for the DCO design, some modifications are needed to get the best performance out of the instantaneous start-up circuitry. The first step is to change the commonly used current-biased scheme to a voltage-biased scheme. A comparison between current-biased DCO and voltage-biased DCO is shown in Figure 6-2. The scheme in Figure 6-2a forces the dc current of the tank to be fixed while Figure 6-2b forces the inner supply voltage to be fixed. Extra LC filters, such as the ones used in [57], can be added at both the top and the bottom common-source nodes to filter the noise at twice the oscillation frequency and boost the impedances at these two nodes. Although current biasing is used more often in continuous LC DCOs, it is necessary to use voltage biasing here.

Firstly, with the current biasing scheme, the common source node of MP1 and MP2 is not fixed. The steady-state oscillation swing is always smaller than Vdd since the top-biasing transistor MP3 needs voltage margin to operate in the saturation region, which is about 300 mV in a 40 nm technology. Thus, there is a 300 mV difference between initial voltage swing and steady-state voltage swing, which could require as much as 10 ns to settle.

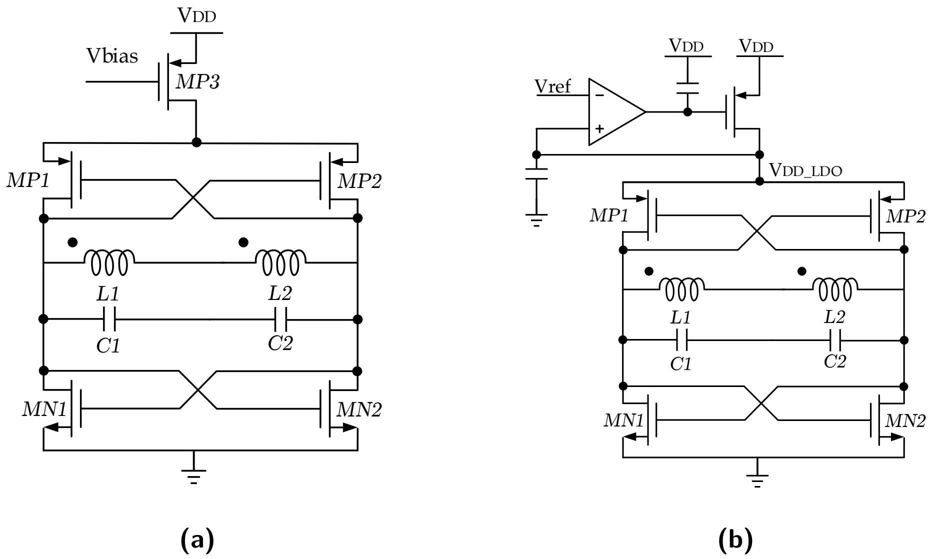


Figure 6-2: Class B complementary push-pull LC DCOs: (a) Current-biased DCO (b) Voltage-biased DCO

Secondly, the purpose of the active part of the DCO is to compensate for the energy lost every cycle in the LC tank. The loss mechanism in the LC tank can be modeled through a parallel resistor R_p across the tank. The definition of R_p can be found in Eq. (6-1), assuming the loss is dominated by inductor loss, where r_s is the series resistance of the inductor. The smaller R_p is, the more loss there is in the tank. It can be seen from Eq. (6-1) that R_p is smaller at a lower frequency, which means the DCO needs to drain more current at lower frequencies, assuming the amplitude is to be kept the same. In a current biasing scheme, this is usually done by varying either the control voltage of MP3 or its width. However, in a wideband LC DCO, the value of R_p can easily vary by more than 50% from lowest to highest operation frequency. Such a large change in biasing current would substantially change the dc biasing points of MN1, MN2, MP1 and MP2. Even if the amplitude can be kept the same, the maximum and minimum voltage during oscillation would vary together with the biasing point.

$$R_p \approx \frac{(\omega_0 L)^2}{r_s}. \tag{6-1}$$

Because of the above-mentioned factors, it is necessary that a voltage-biasing scheme as shown in Figure 6-2b is used. By fixing the voltage at the common-source node of MP1 and MP2, the DCO can be designed to have a swing from Gnd to VDD_LDO both at start-up and in steady state. As for the method to change the impedance of active core to keep the oscillation amplitude constant while R_p varies with frequency, an approach other than varying dc biasing current as in Figure 6-2a is used.

6-3 Impedance Tuning Method

While a voltage-biasing scheme makes it possible to design an LC DCO that has a same swing during start-up and in steady state, the voltage-biasing scheme in Figure 6-2b loses the feature of varying biasing current. However, there is still a way to ensure an almost constant amplitude across the wide operation frequency range. The solution can be found in [58]. Basically, multiple active cores which are similar to the main active core in Figure 6-2b, but smaller in size, are connected in parallel to the tank. At lower frequencies, where more power is needed to keep a certain oscillation amplitude, more active cores are turned on, as if the width of the transistors in the main core is increased. The schematic of a single unit of the impedance-tuning core is shown in Figure 6-3. In the designed DCO, a total of 15 impedance-tuning cores are connected in parallel to the main core to ensure an almost constant oscillation amplitude in steady state from 5.0 GHz to 7.0 GHz.

6-4 Design of the Active Core

Design of the active core, or sizing of the transistors, is usually an important part during oscillator design to achieve low noise. Design rules for low noise can be found in many publications, such as [52] and [59]. Basically, for a fixed W/L ratio, transistors are sized a little larger to reduce 1/f noise. However, as the analysis in Chapter 2 suggests, 1/f noise is less a concern in this duty-cycled DCO since 1/f noise only accumulates to a level higher than thermal noise in the long run, which is not the case when the DCO is turned on for only 40 ns each time. Moreover, the DCO design in this thesis aims to make LC oscillators

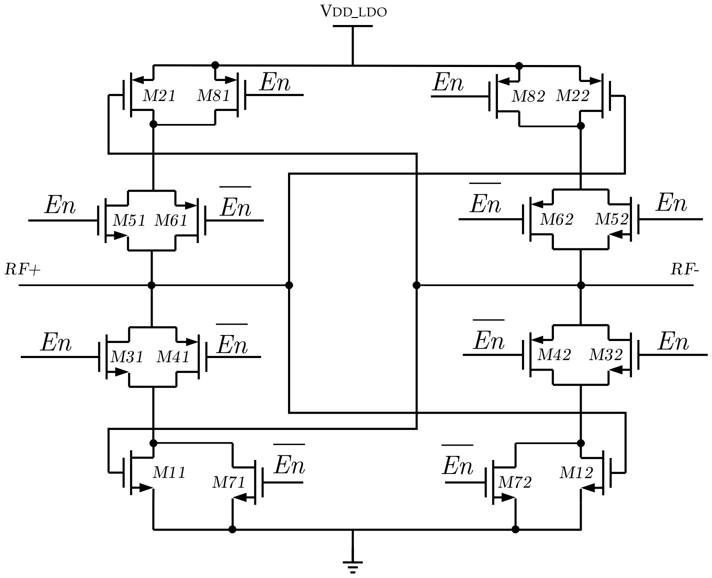


Figure 6-3: Schematic of an impedance-tuning active core unit

a valid choice for duty-cycled DCOs rather than competing noise performance with existing LC DCOs. The fact that an LC oscillator instead of a ring oscillator is used already reduces the noise (both flicker noise and thermla noise) by more than 1 order of magnitude.

In summay, the transistor sizes are kept at minimum in this design in order to ensure that the oscillator can oscillate at the highest frequency of 7.0 GHz.

6-5 Design of the Capacitor Bank

6-5-1 Choice Between MOS Capacitors and Fringe Capacitors

It has been a trend to tune the frequency of an LC DCO with switched capacitor banks rather than a varactor because the former approach is fully digital and eliminates the need to generate an analog control voltage with a Digital-to-Analog Converter (DAC), which does not scale well with technology. However, regarding switched capacitors, a choice needs to be made between MOS capacitors and fringe capacitors.

Wide tuning range favors fringe capacitors over MOS capacitors because the former has a larger on/off capacitance ratio. Morever, MOS capacitors are very sensitive to oscillation amplitude [60]. In this design, initial amplitude has a non-zero settling phase and when the capacitor is ideal, this amplitude deviation from steady state causes deviation in frequency becasue impedance of active core is amplitude-sensitive. Amplitude sensitivity of MOS capacitors is probably larger than that of active core and noticeably deteriates the frequency settling behavior. Thus, fringe capacitors are chosen over MOS capacitors.

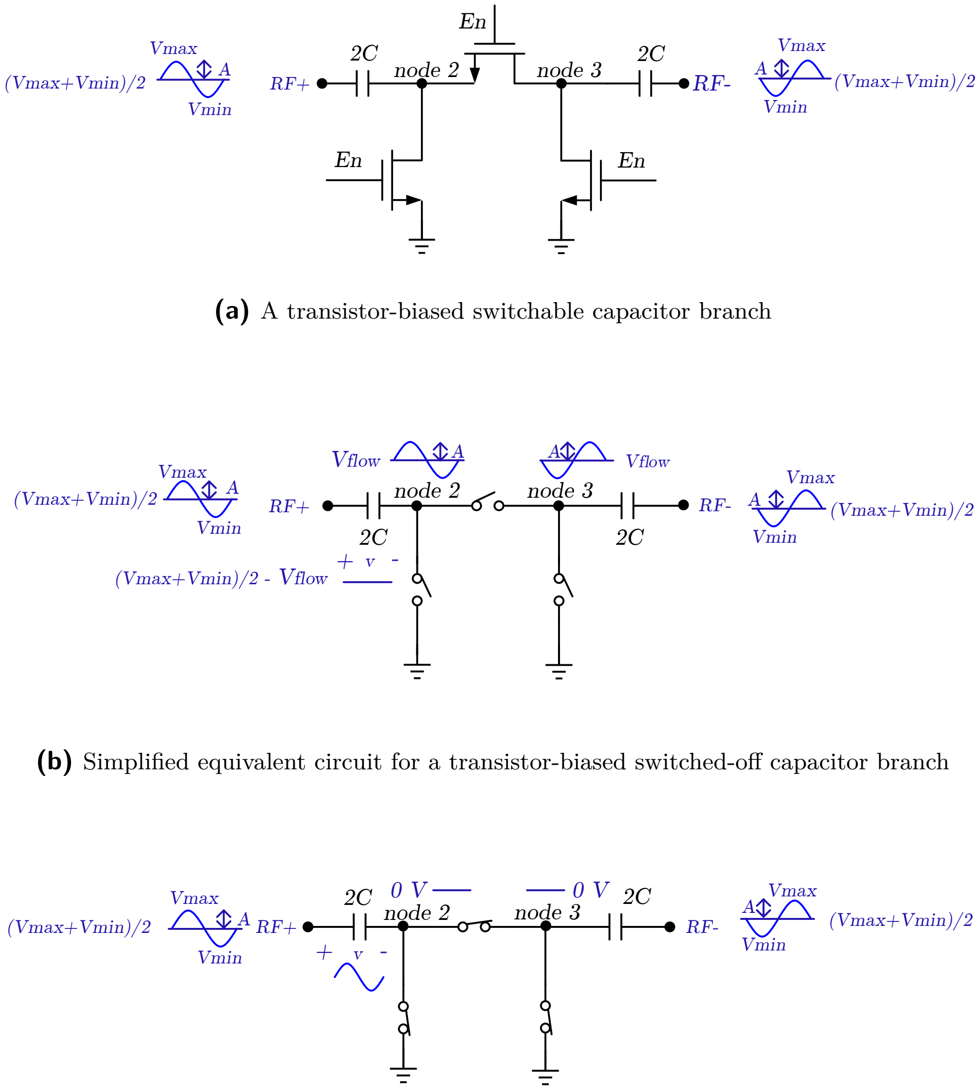
6-5-2 Attention to Start-up Behavior

Due to the special start-up phase of duty-cycled DCO, biasing network of the capacitor bank needs special attention. In terms of biasing of the switches that control switchable capacitor branches, there are two commonly used implementations.

The first implementation is the one used in [61]. A pair of controlled transistors are used to bias the switch at the center. When the capacitor branch is enabled, both of the biasing transistors are turned on, providing low-impedance paths from drain and source of the center switch to ground. When the capacitor branch is disabled, both of the biasing transistors are turned off, leaving the common mode voltage of the center switch floating. Schematic of this implementation, together with simplified equivalent circuits at both on and off states, are shown in Figure 6-4. For a simple qualitative analysis, parasitic capacitors to substrate, mainly contributed from fringe capacitors, are not taken into account.

It can be seen from Figure 6-4b that when the capacitor branch is turned off, voltage across either fringe capacitor is a constant and thus no current flows through the capacitor. This agrees with the intention of de-loading the capacitor branch. On the other hand, when the branch is turned on, RF output voltage directly modulates the voltage across the capacitor, thus the capacitor is periodically charged and discharged, shown in Figure 6-4c. In this way, this capacitor effectively loads the rest of the tank.

While this implementation works well for most applications, it is noted that the common mode voltage of the switch during off state is left undefined, which has been discussed in detail in [62]. The undefined common-mode voltage of the switch might cause the switch to be weakly turned on twice during oscillation.



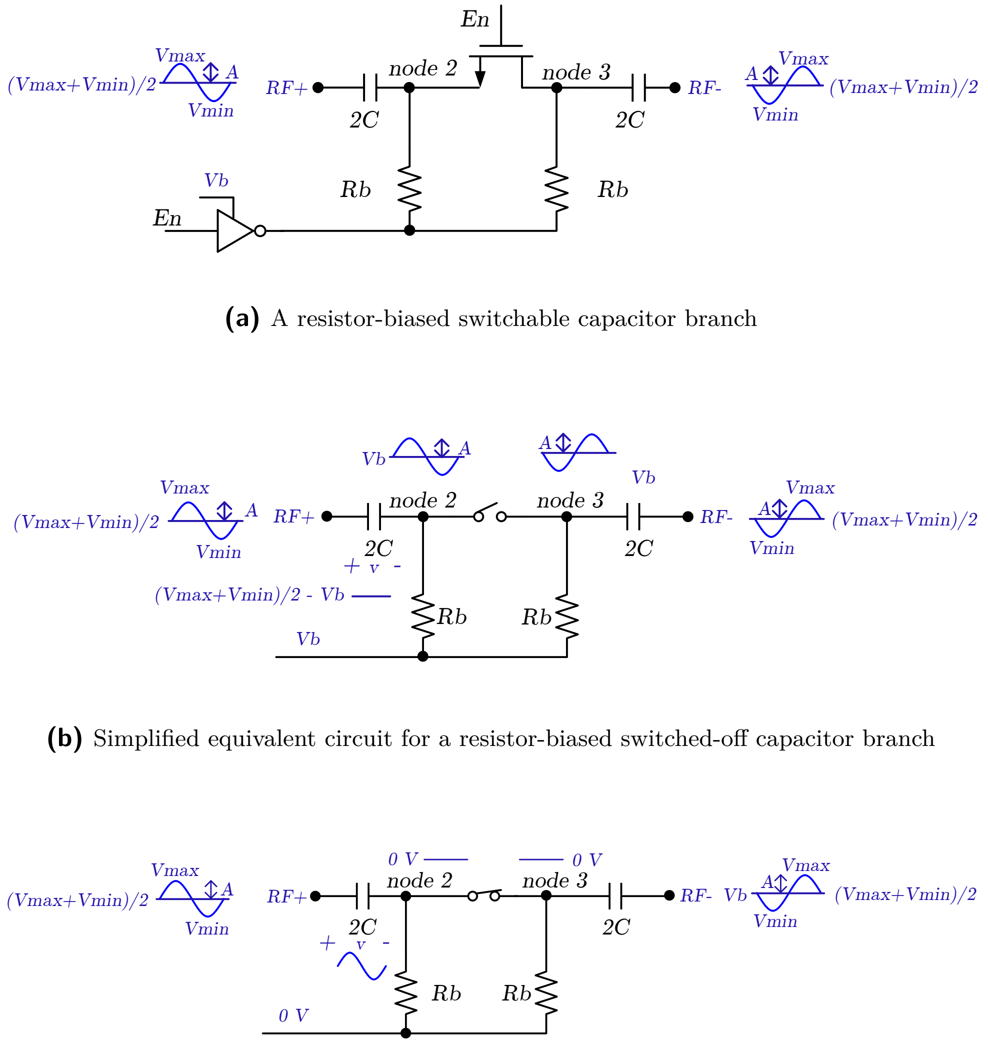
(a) A transistor-biased switchable capacitor branch

(b) Simplified equivalent circuit for a transistor-biased switched-off capacitor branch

(c) Simplified equivalent circuit for a transistor-biased switched-on capacitor branch

Figure 6-4: A transistor-biased capacitor branch: (a) General case (b) Switched-off case (c) Switched-on case

In order to have a clearly defined common-mode voltage at off state, one commonly used approached is to bias drain and source of the center switch with two larger resistors, such as the configuration used in [54]. The biasing voltage is set to Gnd when the capacitor branch is enabled and set to about half Vdd when disabled. A schematic of a resistor-biased capacitor branch, together with simplified equivalent circuits at on and off states, is shown in Figure 6-5.



(a) A resistor-biased switchable capacitor branch

(b) Simplified equivalent circuit for a resistor-biased switched-off capacitor branch

(c) Simplified equivalent circuit for a resistor-biased switched-on capacitor branch

Figure 6-5: Resistor-biased capacitor branch: (a) General case (b) Switched-off case (c) Switched-on case

As long as the biasing resistors are large enough not to deteriate the capacitor quality factor at off state, behavior of the resistor-biased configuration is almost identical to transistor-biased configuration, except clearly defining the common-mode voltage at off state to be V_b.

However, neither of the above-mentioned configurations works for the capacitor bank used in a duty-cycled LC DCO. The ultimate reason is that both of the biasing networks are time-invariant and it is assumed when using these circuits that rules of steady-state circuit analysis apply.

Consider the situation if a resistor-biased capacitor bank is used in a duty-cycled LC DCO. While it works fine when it is enabled, a problem occurs when it is disabled. This is illustrated in Figure 6-6.

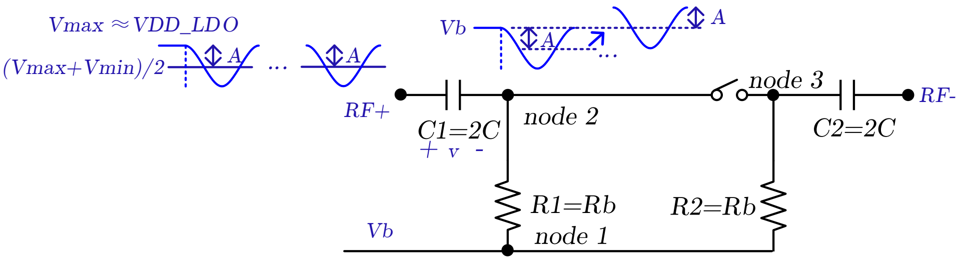


Figure 6-6: Behavior of a resistor-biased capacitor branch at off state when used in a duty-cycled LC DCO

RF+ is pre-charged to *VDD_LDO*, which is about equal to the maximum voltage during oscillation. During idle state, *node* 2 is biased by the resistor at V_b. Once the oscillator is released from idle state, neglecting the right part of the branch, C1 together with R1 forms a high-pass transfer function from $RF+$ to *node* 2. The first cycle of the waveform at *node* 2 will mimic the change happening at $RF+$. The voltage across C1 is $(V_{max} - V_b)$. However, as the system continues to run, the circuit will reach steady state. The DC component in $RF+$ will be filtered by the high-pass network formed by C1 and R2. In steady state, the waveform at *node* 2 will have a DC component equal to V_b while the AC component follows the AC component at $RF+$. Therefore the voltage across C1 would be $(V_{max} + V_{min})/2 - V_b$. During this slow varying of voltage across the capacitor C1, C1 is seen as a time-variant load on the tank.

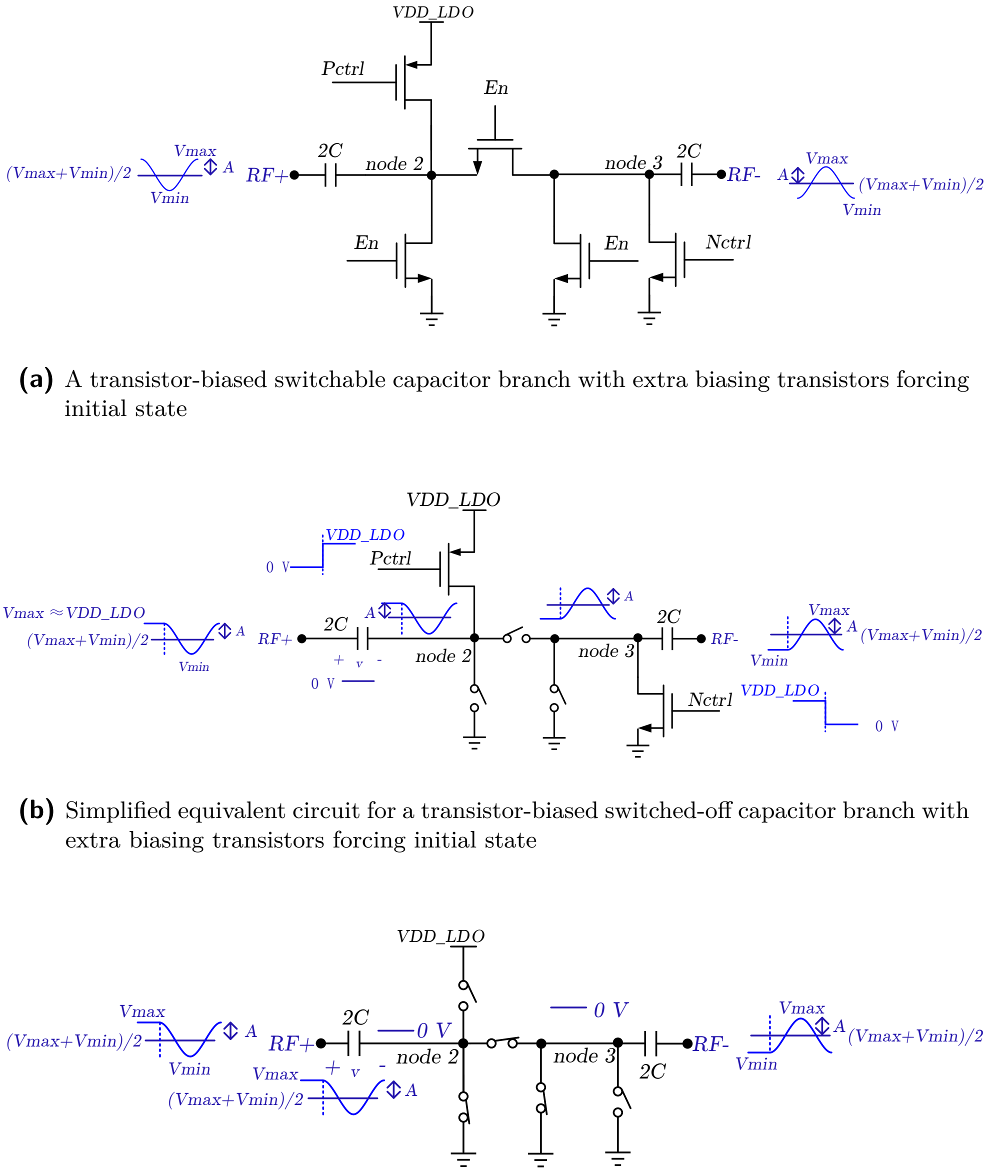
This problem is even worse when the RF amplitude is almost rail-to-rail. If V_b is set to about half Vdd, as in the common case, then during the first few cycles, *node* 2 will be pulled below Gnd and reverse-biased diodes might be turned on and end up causing circuit failure.

Apparently, instead of a time-invariant biasing network, a time-variant biasing network is needed here. The work-around that the author came up with is shown in Figure 6-7. Two additional transistors are added to the biasing network in case the capacitor branch is turned off.

The two additional transistors do not take effect if the branch is enabled. However, if the branch is disabled, they are responsible for charging one node to *VDD_LDO* while pulling the other to Gnd before start, which is shown in Figure 6-7b. The two additional transistors are turned off together with the release of the DCO. In this way, the waveforms at the inner nodes of the capacitors during the initial cycles will resemble those during steady state. The truth table for the control signals P_{ctrl} and N_{ctrl} is shown in Table 6-1, where *En* represents enable signal for the capacitor branch and *start_stop* represents the enable signal the for the DCO.

Table 6-1: Truth table for Pctrl and Nctrl

En	start/stop	Pctrl	Nctrl
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



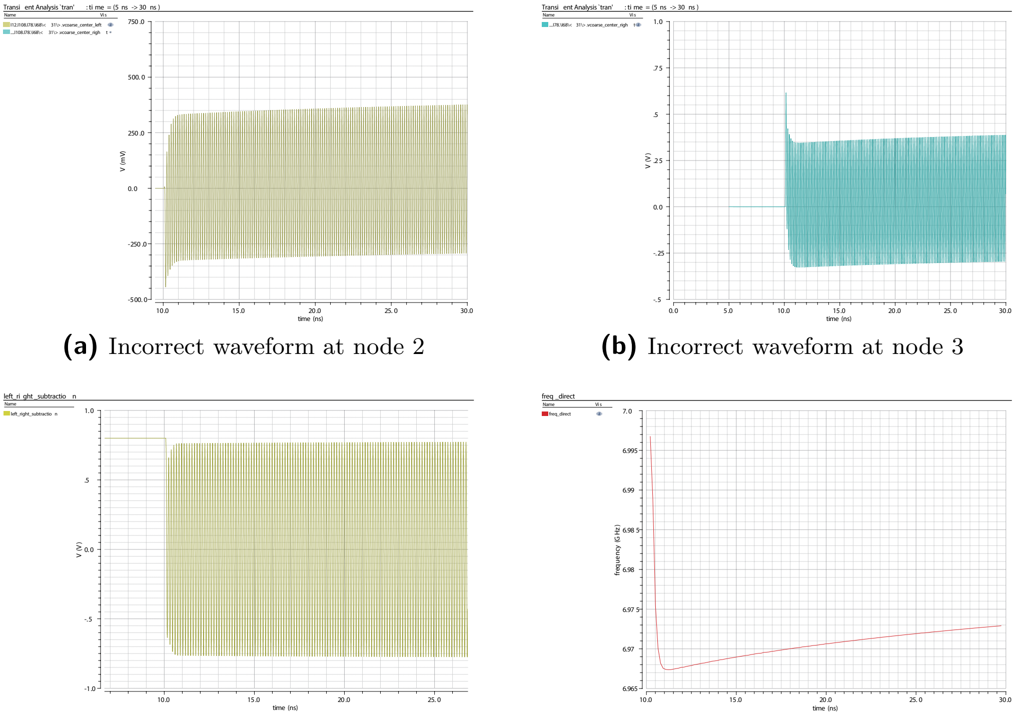
(a) A transistor-biased switchable capacitor branch with extra biasing transistors forcing initial state

(b) Simplified equivalent circuit for a transistor-biased switched-off capacitor branch with extra biasing transistors forcing initial state

(c) Simplified equivalent circuit for a transistor-biased switched-on capacitor branch with extra biasing transistors forcing initial state

Figure 6-7: A transistor-biased switchable capacitor branch with extra biasing transistors setting initial state: (a) General case (b) Switched-off case (c) Switched-on case

The above analysis shows what would happen if a resistor-biased network is used for the capacitor bank in a duty-cycled LC DCO. Figure 6-8 shows the incorrect waveforms for the transistor-biased case. Since the malfunction of a capacitor branch only happens when it is disabled, the incorrect waveforms shown in Figure 6-8 is drawn from a simulation of the DCO at 7.09 GHz, where almost all of the capacitor branches are turned off so that the effect of this problem is most obviously seen. The oscillator is released by a trigger signal at 10.0 ns.



(a) Incorrect waveform at node 2

(b) Incorrect waveform at node 3

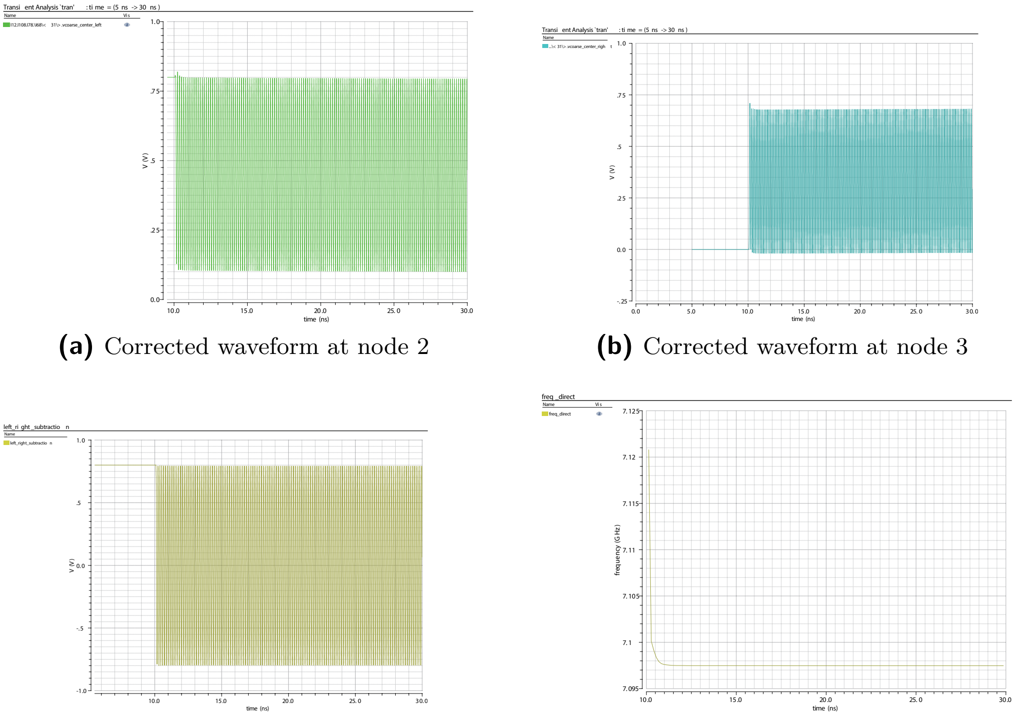
(c) Incorrect DCO output

(d) Incorrect frequency settling

Figure 6-8: Incorrect waveforms of a transistor-biased capacitor branch used for a duty-cycled LC DCO @ 7.09 GHz

The settling process at node 2 and node 3 that has just been analyzed can be clearly identified from Figure 6-8a and Figure 6-8b. A following consequence of this is incorrect settling behavior in both time domain and frequency domain waveforms, which are plotted in Figure 6-8c and Figure 6-8d respectively.

The effectiveness of the work-around is shown in Figure 6-9.



(a) Corrected waveform at node 2

(b) Corrected waveform at node 3

(c) Corrected DCO output

(d) Corrected frequency settling

Figure 6-9: Corrected waveforms of a transistor-biased capacitor branch with extra biasing transistors setting initial state @ 7.09 GHz

6-5-3 Design of the Medium and Fine Bank

Capacitor bank dimensioning starts from the quantization noise specification. The quantization noise of the capacitor banks is determined by the step size of the fine capacitor bank, if no time domain modulation like ΣΔ is used. A rule of thumb is to make the accumulated time difference due to finite capacitor bank resolution over one ref cycle 1/10 of the accumulated jitter during the same time. These two error sources makes up the total phase error at the end of a burst, as is explained in [9].

From a simulation of the DCO in the early design phase with an ideal capacitor bank and an estimated inductor quality factor, the DCO has a phase noise at 1 MHz frequency offset of -108 dBc/Hz @ 7.0 GHz and -111 dBc/Hz @ 5.0 GHz, which translates into an accumulated jitter of 0.09 ps and 0.11 ps, respectively, over the 40 ns burst, according to the equation shown in [9]. The amount of accumulated jitter is equivalent to an accumulated timing error due to a quantization step of 31.5 kHz @ 7.0 GHz and 27.5 kHz @ 5.0 GHz. The capacitance resolution needed can be calculated from Eq. (6-2). The corresponding capacitance resolution needed is 2.32 aF @ 7.0 GHz and 5.57 aF @ 5.0 GHz.

$$\Delta f = \Delta C \cdot 2\pi^2 \cdot L \cdot f^3. \tag{6-2}$$

A capacitance resolution of 2.32 aF is already stringent, and this is even before dividing it by a factor of 10. The capacitance of the smallest fringe capacitor from the standard library is on the order of 10 fF to 50 fF. Sizing it aggressively might keep the value at about 1 fF. However, there is no way to reach 2.32 aF by only reducing size. To keep the system simple and avoid increasing capacitor resolution with $\Sigma\Delta$ modulation, which needs to take care of limit-cycle behavior, a capacitor scaler is used, similar to those used in [63] and [64]. With the capacitance scaler, the resolution on the order of 2.32 aF can be achieved. However, it is still not able to achieve a capacitance resolution of 1/10 of 2.32 aF. Since the specification does not take into account the jitter from the ref path and an accumulated jitter of 100 fs seems to be on the optimistic side, the specification on the capacitance resolution is kept at 2.32 aF at this moment and will be reviewed later when a more accurate simulation result from the DCO is available.

The capacitance scaler used for the fine bank is shown in Figure 6-10. A capacitance change of ΔC in the inner part of the switched fringe capacitor is translated to a smaller capacitance ΔC_{eq}. The relation between ΔC_{eq} and ΔC is shown in [63] and repeated in Eq. (6-3), where C_0 includes the intrinsic capacitance at the center, which increases with codes, and all other parasitic capacitances of fringe capacitors, which is irreverent to the change in codes. When designed correctly, $C_1 \gg max\,(C_2, C_0)$ and the scaling ratio is about $(C_1/(C_1 + C_2))^2$.

$$\Delta C_{eq} = \left(\frac{C_2}{C_0 + C_1 + C_2}\right)^2 \Delta C. \tag{6-3}$$

Although as large a capacitance division ratio as possible is desired from the resolution point of view, the division ratio deviates more and more from the designed value if the number of bits of the fine bank keeps increasing, since the capacitance in the inner part of the capacitance

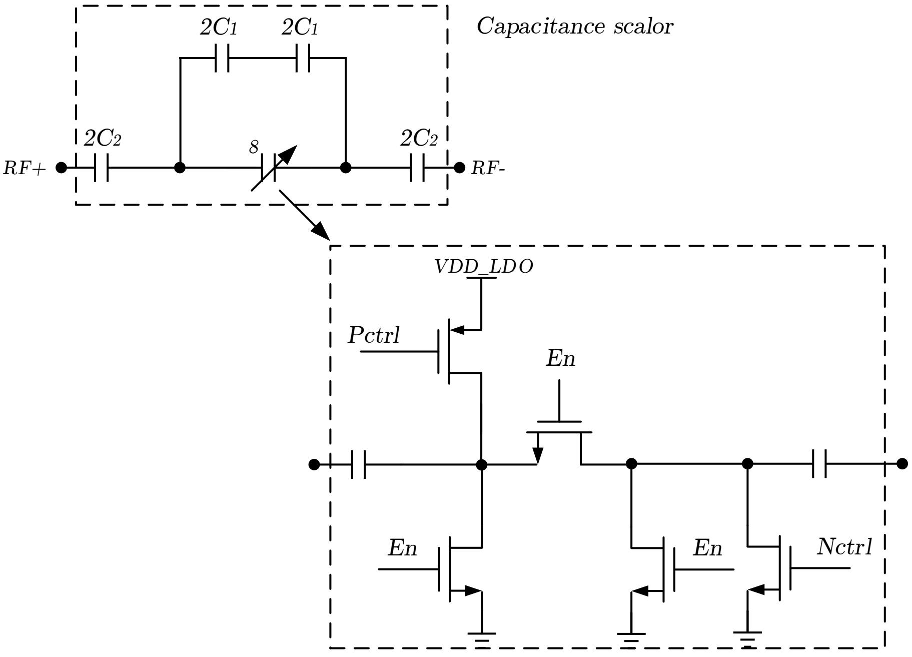


Figure 6-10: The capacitance scaler used for the fine bank in this design

divider would be a non-negligible factor in Eq. (6-3) by then. Suppose the equivalent fine bank needs to cover 2 LSBs of the medium bank and the LSB of the medium bank is ΔC_m and its minimum value is limited by matching performance. Then, the summation of the inner capacitance of the capacitance scaler is shown in Eq. (6-4). Also, assume that C_2 is much smaller than C_1, then its ratio to C_1 is shown in Eq. (6-5).

$$\begin{aligned}\Sigma\left(\Delta C_f\right) &= \Delta C_f \cdot \frac{2\cdot\Delta f_m}{\Delta f_{c,equi}} \\ &= \Delta C_f \cdot \frac{2\cdot\Delta C_m}{\Delta C_f\cdot\left(C_2/C_1\right)^2} \\ &= 2\cdot\Delta C_m\cdot\left(C_1/C_2\right)^2. \qquad (6\text{-}4)\end{aligned}$$

$$\frac{\Sigma\left(\Delta C_f\right)}{C_1} = \frac{2\cdot\Delta C_m\cdot\left(C_1/C_2\right)}{C_2}. \qquad (6\text{-}5)$$

$\Sigma\left(\Delta C_f\right)$ in Eq. (6-5) is the dominant component of C_0 in Eq. (6-3). The larger $\Sigma\left(\Delta C_f\right)/C_1$ is, the larger the deviation between the actual division ratio and the designed $(C_1+C_2)/C_2)^2$ is. Since parasitic capacitance imposes an upper limit on C_2 , therefore from Eq. (6-5), it can be seen that an attempt to increase the division ratio, C_1/C_2, would make $\Sigma(\Delta C_f)$ a larger portion of C_1. It further means, observing Eq. (6-3), that the ratio would be more dependent on the present code and the capacitance curve would become more and more non-linear across the range of the fine bank.

In order to achieve a capacitance resolution of 2.32 aF, a C_1/C_2 of 15 is chosen. The choice of C_2 is a trade-off between the required linearity of division ratio and acceptable parasitic capacitance from the capacitance scaler, since the parasitic capacitance of the scaler would be about equal to C_2. The scaling ratios of the capacitance scaler at several different C_2 values are shown in Table 6-2, where S(off) represents the scaling ratio when all the inner part of the scaler is turned off and S(on) represents the ratio when all the inner part is turned on. When C_2 is 27.52 fF, the ratio varies by more than 100% between all off state and all on state. By increasing C_2 to 67 fF, the ratio variation is kept at 50%. It should be emphasized that linearity of capacitance is not a critical specification, since whatever frequency error caused by nonlinearity can be tracked and corrected by the PLL. Finally, the chosen values for C_2 and C_1 are 67 fF and 1005 fF. The unit-branch capacitances, which are half of the fringe capacitances, for medium and fine bank are sized to 631 aF and 1.32 fF respectively. In this way, an 8-bit fine bank, after scaling, will cover 2 LSBs of the medium bank. The capacitance versus code curve of the fine bank, after using the capacitance scaler, is shown in Figure 6-11. A slight decrease in the slope at larger codes due to the self-loading effect can be observed.

Table 6-2: Choice of C_2 in capacitance scaler

C_1 (fF)	C_2 (fF)	S(off)	S(on)
412.8	27.52	336	799
825.6	55.04	294	490
1005	67	287	442

The unit branch for the medium bank is shown in Figure 6-12. The choice of center switch size is known to depend on the trade-off between quality factor and parasitic capacitance. The trade-off is in the context of the relationship between quality factor of the tank and individual quality factors of inductor and capacitor, shown in Eq. (6-6), where Q, Q_L and Q_C represent quality factor of the tank, inductor and capacitor respectively. Q is dominated by

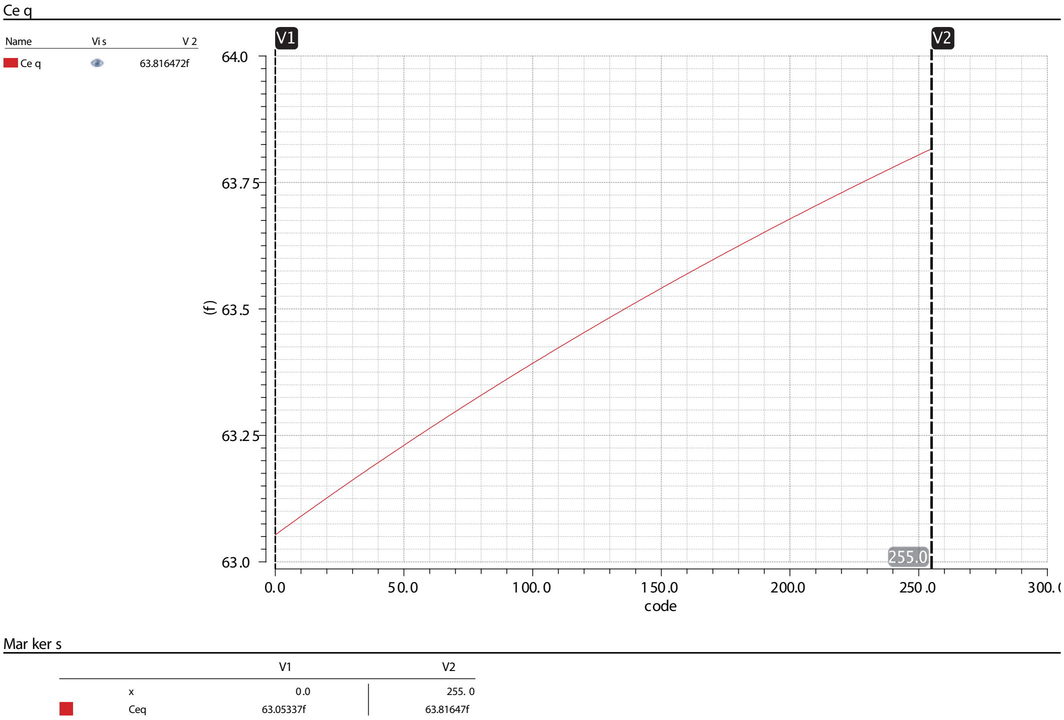


Figure 6-11: Capacitance vs. code for the fine bank

the smaller one between Q_L and Q_C. A rule of thumb is to size the switch so that Q_C is 4–6 times Q_L. Thus, the sizing of the switch for the medium bank is relatively easy, judging by the fact that Q_L in this deign is about 8. The designed quality factor for the medium bank is 29 @ 5.0 GHz and 20 @ 7.0 GHz.

$$\frac{1}{Q} = \frac{1}{Q_L} + \frac{1}{Q_C}. \tag{6-6}$$

The choice for the switch size for the fine bank is not so straight-forward. Analysis is needed on how the quality factor changes after using the capacitor scaler. The equivalent circuit for a half branch of the capacitance scaler is shown in Figure 6-13. Under the assumption that $C_1 \gg max(C_2, C_0)$, Eq. (6-7) follows.

$$|V \cdot I| = \frac{C_1^2}{C_0 \cdot C_2} |V_1 \cdot I_1| \, . \tag{6-7}$$

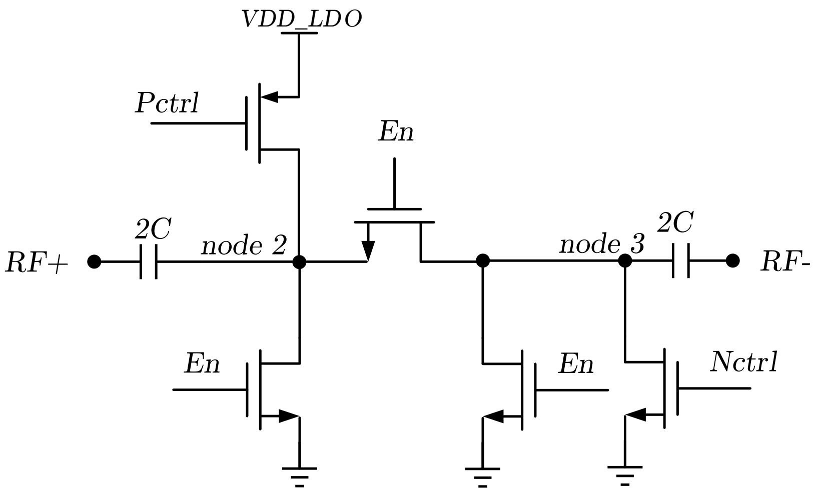


Figure 6-12: Schematic for a unit of the medium bank

What Eq. (6-7) suggests is that the imaginary part of the energy of the equivalent tank is $C_1^2/(C_0 \cdot C_2)$ times the imaginary energy of the inner tank, which further means that the quality factor of the equivalent tank is $C_1^2/(C_0 \cdot C_2)$ times the quality factor of the inner part of the tank. Since (C_1/C_2) is set to 15 and $C_1 \gg C_0$, the switch size for the inner part of the tank can be safely set to the minimum.

6-5-4 Design of the Coarse Bank

The design of the coarse bank needs to take one more factor into consideration. The LSB of the coarse bank should bring a change in frequency less than one f_{ref} to ensure absolute loop stability, which is explained in [11]. The unit branch of the coarse bank has a capacitance of 3.7 fF at on state and the switch is sized to achieve a capacitor quality factor of 30 @ 5.0 GHz and 21 @ 7.0 GHz. Medium bank is designed to have 5 bits to enure it covers 2 LSBs of the coarse bank. The coarse bank is made 6 bits to cover the required tuning range with enough margin for process variations.

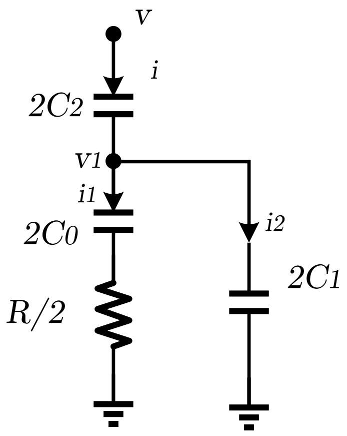


Figure 6-13: Illustration for the analysis of change in quality factor after using capacitance scaler

6-5-5 Summary of the Capacitor Bank

Since turning on more active cores at lower frequency will also add parasitic capacitance, which will contribute to the capacitance tuning range,the coarse bank is adjusted a bit to ensure a tuning range from 5.0 GHz to 7.0 GHz. The parasitic capacitances of fringe capacitors for the fine and medium bank are extracted while the parasitic capacitance of the coarse bank is estimated with linear approximation, since there were multiple iterations after initial design. Summary of the capacitor bank is shown in Table 6-3 and more detailed parameters can be found in Table 6-4.

Table 6-3: Summary of the capacitor bank

C_{min}	C_{max}	C_1	C_2
109 pF	316 pF	1.005 pF	67 fF

Table 6-4: List of parameters for the capacitor bank

	Coff (F)	Con (F)	ΔC (F)	$\Sigma\,(\Delta C)$ (F)
Inner fine bank	63.8 f	337.5 f	1.07 f	273.7 f
Equivalent fine bank	63.1 f	63.8 f	2.99 a	763 a
Medium bank	8.47 f	19.5 f	356 a	11.03 f
Coarse bank	38.0 f	233 f	3.10 f	195 f

6-6 Summary of the DCO

A more complete view of the DCO can be seen in Figure 6-14. Minimum and maximum oscillation frequency for the DCO are 4.91 GHz and 7.26 GHz. When the whole coarse bank is turned on or off while half of the medium and fine bank are enabled, corresponding oscillation frequencies are 4.94 GHz and 7.09 GHz. Since the number of enabled active cores is affected by offset in peak detector and comparator, the frequency range that the PLL can safely lock to is 5.05 GHz to 7.01 GHz. The DCO metrics, based on schematic level simulation, are summarized in Table 6-5, where the definition of FOM is shown in Eq. (6-8). Compared to the most advanced FOMs such as the one reported in [65], the large degradation in FOM is directly due to the degradation in inductor quality factor. From the phase noise equation in [19, Ch. 8], which is repeated in Eq. (6-9), a 50% decrease in Q would directly cause 6 dB degradation in phase noise.

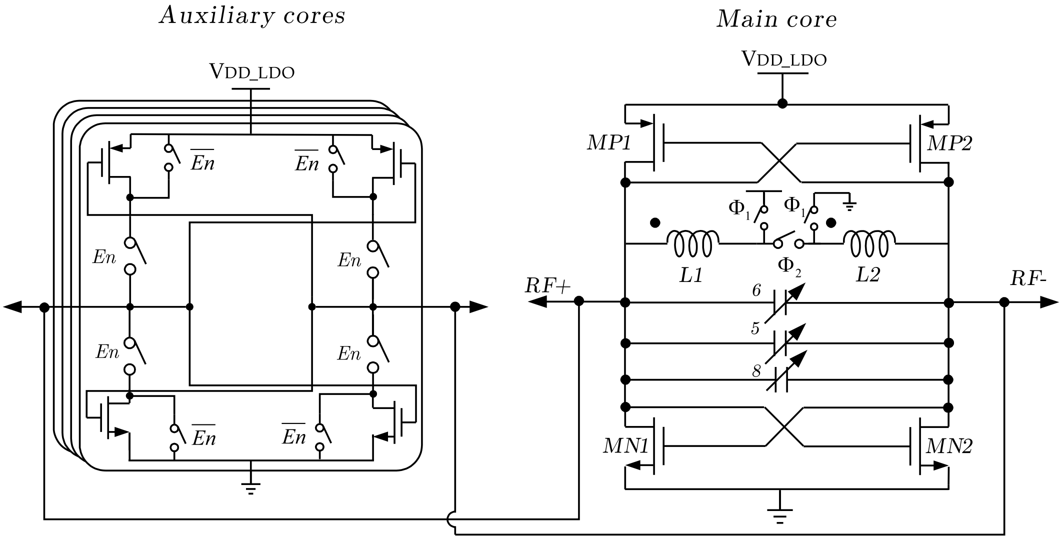


Figure 6-14: shematic for the DCO including main core and auxiliary cores

$$FOM = |PN| + 20 * log_{10}(f_0/\Delta f) - 10 * log_{10}(P_{DC}/1\,\text{mW}). \tag{6-8}$$

$$S\left(\Delta\omega\right) = \frac{\pi^2}{R_p}\frac{kT}{I_{SS}^2}\left(\frac{3}{8}\gamma + 1\right)\frac{\omega_0^2}{4Q^2\Delta\omega^2}. \tag{6-9}$$

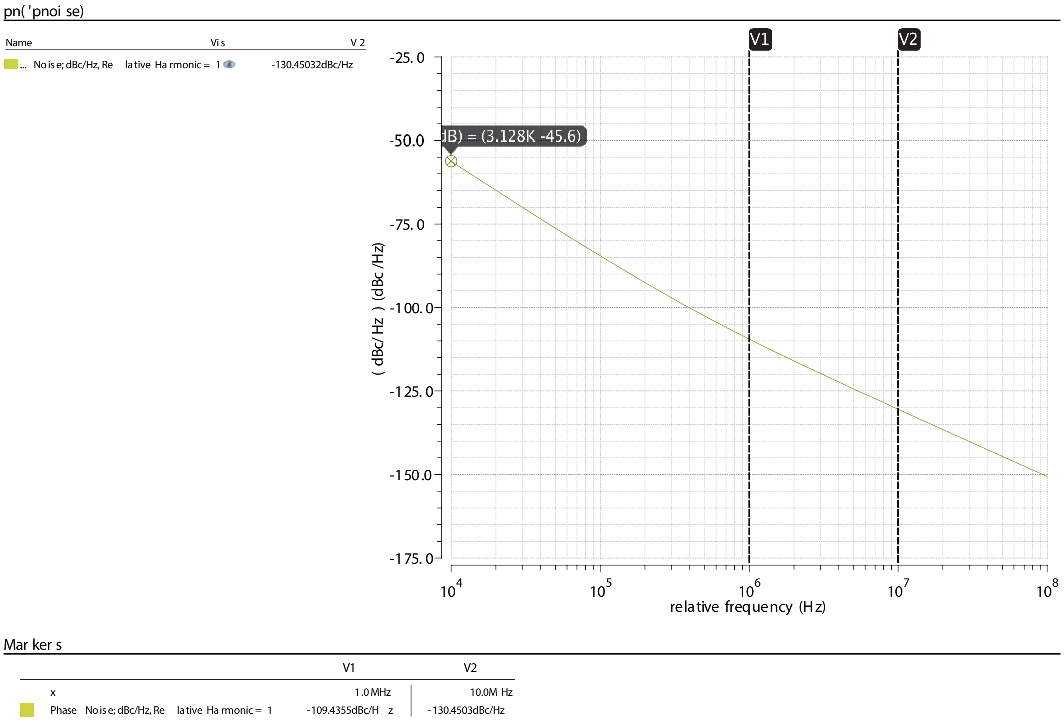
Table 6-5: Metrics of the DCO

	Frequency (GHz)	
	4.94	7.09
Power (µW)	1290 (auxiliary: 571)	643
PN (dBc/Hz)	-109.4 @ 1 MHz -130.4 @ 10 MHz	-105.3 @ 1 MHz -127.3 @ 10 MHz
FOM (dB)	182.2	184.2

Degradation of the quality factor in this design is two-fold. On one hand, in order to cover the required wide tuning range from 5.0 GHz to 7.0 GHz, the parasitic capacitance of the inductor itself needs to be kept lower, which means a higher Self-Resonant Frequency (SRF). SRFs for narrow-band inductors are usually set at twice the operation frequency, while it is set to three times the maximum operation frequency in this design. As is mentioned in [44, Ch. 5], maximum Q of an inductor appears at approximately half the SRF while the maximum of $R_p \approx \omega LQ$ generally occurs at a higher frequency than where Q_{MAX} occurs. Setting a high SRF compromises both Q and R_p at the operation frequency. On the other hand, the instantaneous start-up circuitry introduces an extra series resistance in the inductor, which further degrades the inductor quality factor. The combination of these two factors results in an inductor quality factor of 8 in this design, which is about half of that of a standard narrowband inductor.

However, since the voltage biasing for the DCO is simplified as an ideal voltage source, noise from biasing circuitry is not taken into account, which makes the FOM in Table 6-5 on the optimistic side in this regard. According to the phase noise level shown in Table 6-5, accumulated jitter is equivalent to timing error caused by 16.9 kHz and 27.1 kHz offset at 4.94 GHz and 7.09 GHz respectively. Thus, the minimum required capacitor bank resolution is 33.9 kHz and 54.2 kHz at 4.94 GHz and 7.09 GHz respectively, putting a requirement of 3.94 aF on capacitance resolution. The designed capacitor bank has a resolution of 2.99 aF. Although it is not 10 times smaller than 3.94 aF, this is pretty much how far it can go without using $\Sigma\Delta$ modulation and the design has to live with it.

Phase noise plots are shown in Figure 6-15 and Figure 6-16 while far-out views of settling behaviors, both time domain and frequency domain, can be found from Figure 6-17 to Figure 6-20. In the testbench, the DCO is triggered by an enable signal at 10.0 ns. This signal passes through multiple inverter buffers and finally turns on the switch at the center of the inductor, at which point the DCO starts to oscillate from 90° phase. Thanks to the specially designed start-up circuitry, either initial offset or settling time is smaller than those of the ring oscillator in [11].

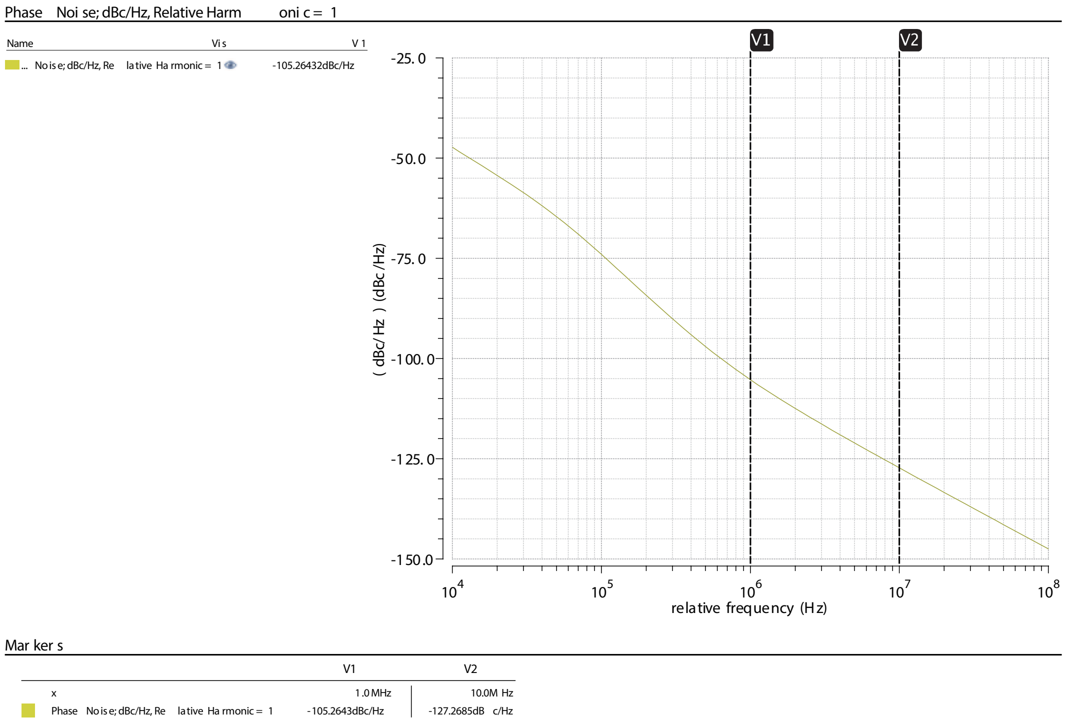

pn('pnoise)

Name	Vis	V2
🟨 ... Noise; dBc/Hz, Relative Harmonic = 1	👁	-130.45032dBc/Hz

Markers

		V1	V2
	x	1.0 MHz	10.0M Hz
🟨	Phase Noise; dBc/Hz, Relative Harmonic = 1	-109.4355dBc/H z	-130.4503dBc/Hz

Figure 6-15: Phase noise @ 4.94 GHz



Name	Vis	V1
... Noise; dBc/Hz, Relative Harmonic = 1	👁	-105.26432dBc/Hz

	V1	V2
x	1.0MHz	10.0M Hz
Phase Noise; dBc/Hz, Relative Harmonic = 1	-105.2643dBc/Hz	-127.2685dB c/Hz

Figure 6-16: Phase noise @ 7.09 GHz

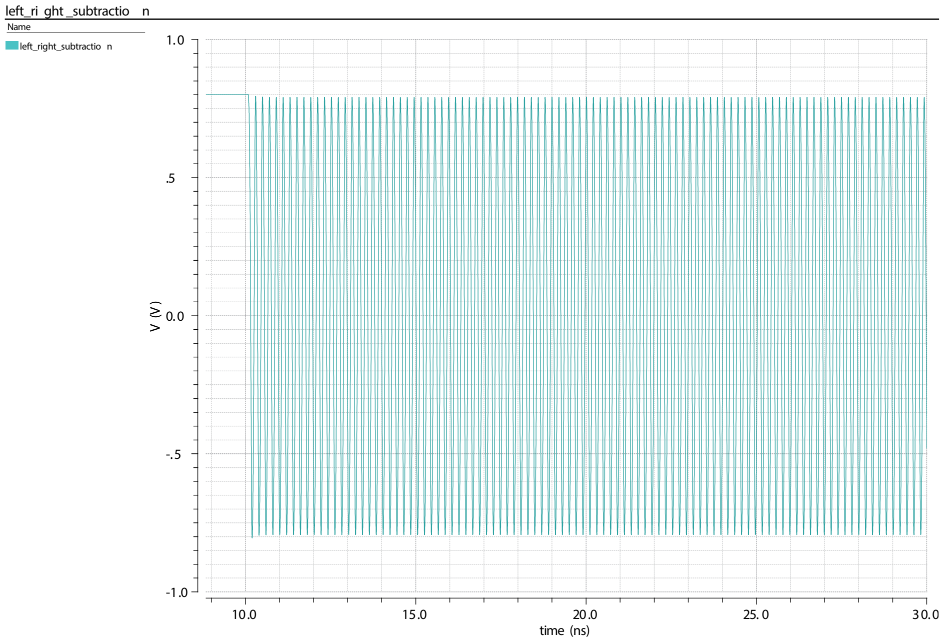


Figure 6-17: Far-out view of the DCO output @ 4.94 GHz

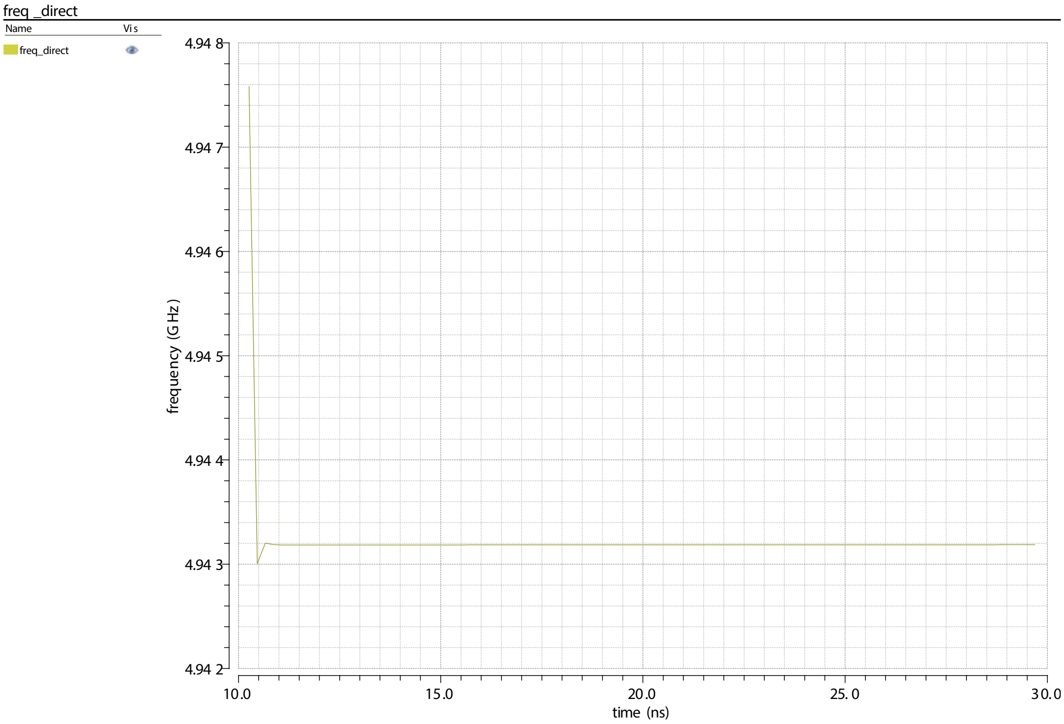


Figure 6-18: Far-out view of frequency settling @ 4.94 GHz

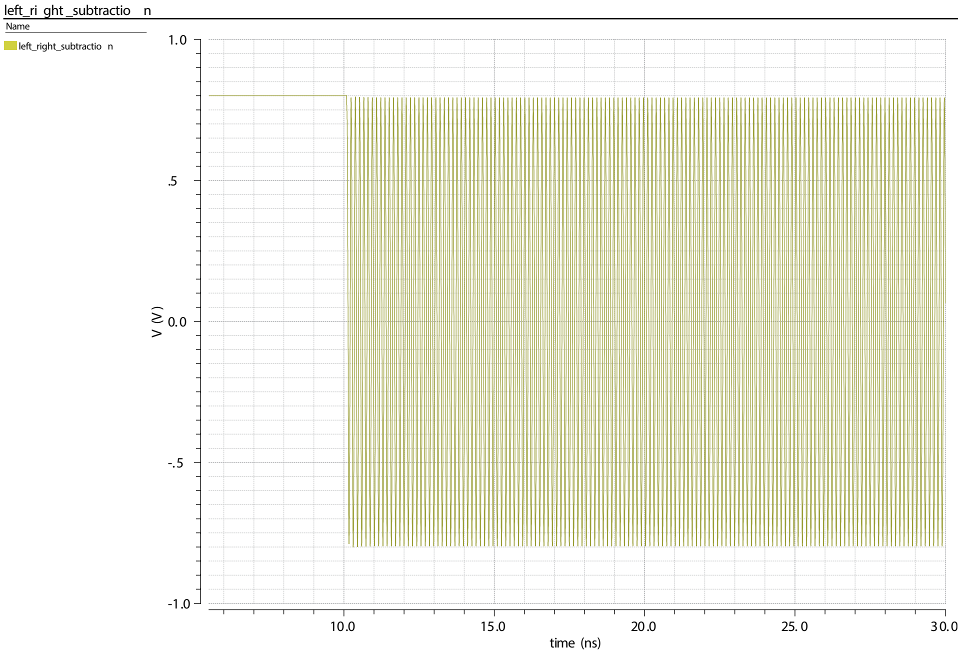


Figure 6-19: Far-out view of the DCO output @ 7.09 GHz

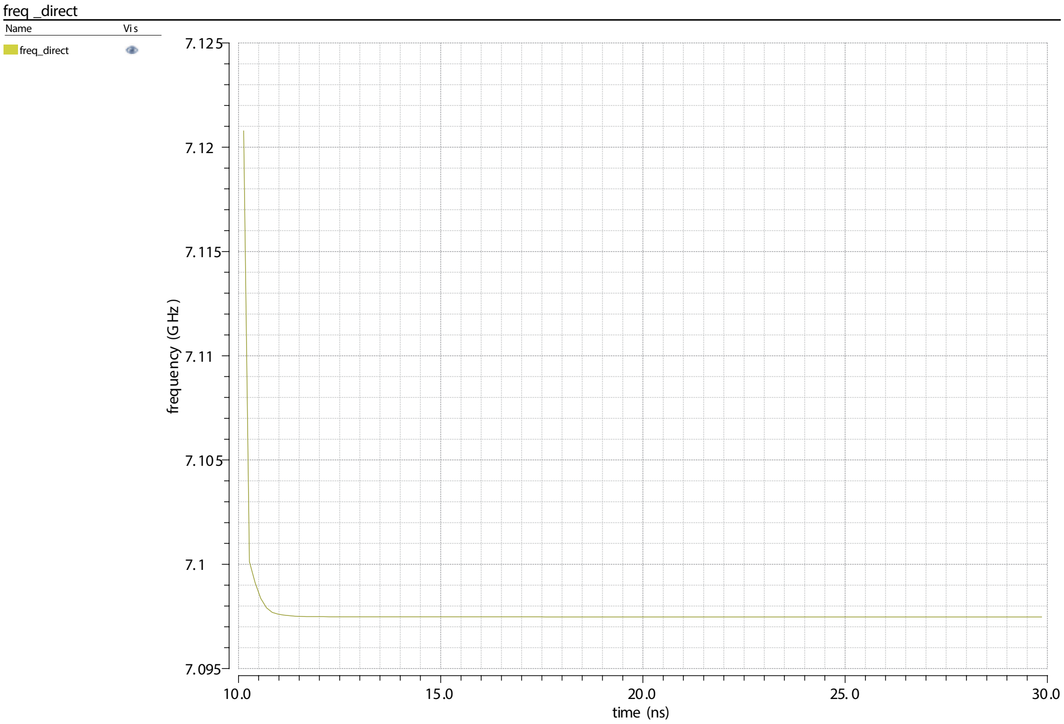


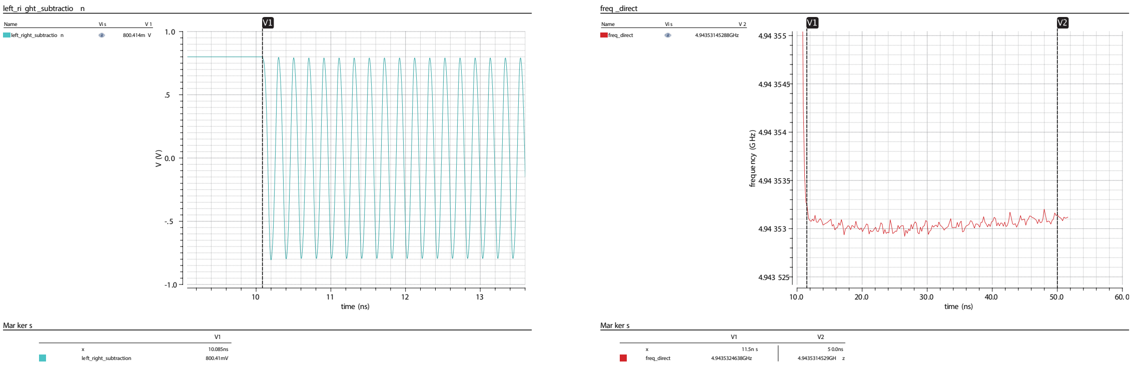
Figure 6-20: Far-out view of the frequency settling @ 7.09 GHz

While the settling behavior shown from Figure 6-17 to Figure 6-20 lives up to the name of instantaneous, there are ultimate limitations for the settling time. Besides the delay of the inverter chain, the DCO needs a non-zero settling time after its inductor is closed. This non-zero settling settling time is composed of the following several sources:

1. Limited accuracy of amplitude calibration: While the amplitude can be kept almost constant by turning on/off the 15 auxiliary cores. There is residue deviation between the start-up amplitude and the steady-state amplitude, due to the dead region of the peak detector and quantization step of the auxiliary cores. This initial amplitude settling can be seen from the first and second cycle of DCO output in Figure 6-21a and Figure 6-22a.

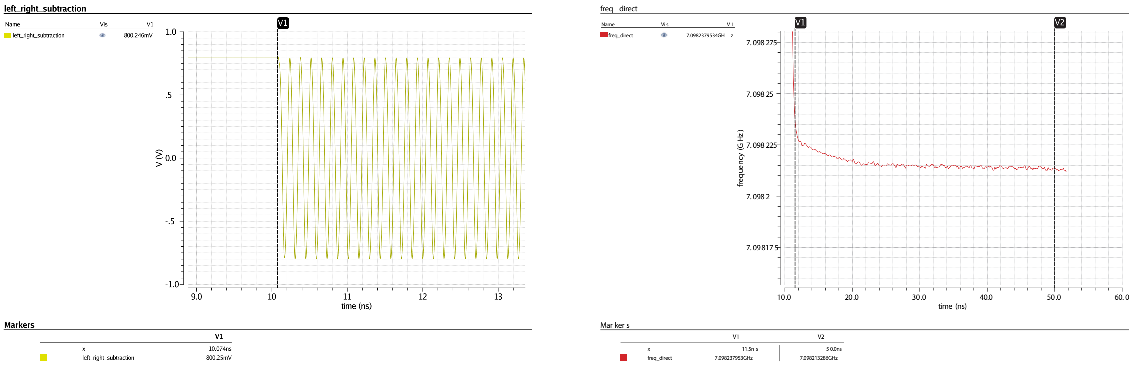
2. Slow settling component: The work-around in capacitor bank corrects the previously incorrect waveform. However, it just alleviates the issue to a large extent instead of solving it completely. No matter how close it is between initial amplitude and steady-state amplitude, there is a slow settling component in the frequency settling curve, which is more severe at 7.09 GHz than at 4.94 GHz, which can be seen Figure 6-21b

and Figure 6-22b. At 7.09 GHz, this slow settling component varies by 20 kHz during the 40 ns burst. It is probably due to the common level drift of the differential output, since the instantaneous start-up circuity sets the initial state by storing the energy in the capacitor bank. It can ensure the start-up amplitude, but not so much the common-mode level. A carefully designed CML buffer should alleviate this slow settling component to a large degree. A better yet modification that aims to solve the source of this problem would be to use a time-variant resistor-biased capacitor bank, switching the biasing network at the right time. Both of these can be explored in future work.



(a) **(b)**

Figure 6-21: Close-in view of the settling behavior @ 4.94 GHz: (a) DCO output (b) Frequency settling



(a) **(b)**

Figure 6-22: Close-in view of the settling behavior @ 7.09 GHz: (a) DCO output (b) Frequency settling

Chapter 7

Design of the Fractional-N DC-ADPLL

Duty-Cycled PLLs (DCPLLs) are usually turned on for only one reference cycle within a pulse repetition period. A DCPLL's frequency is locked on the period defined by this reference cycle, which, in other words, is the time between two successive rising edges belonging to this reference cycle. In the following section, the locking principle of a Digital-to-Time Converter (DTC)-based continuous All-Digital PLL (ADPLL) [61], and two previous DCPLLs, [11] and [9], will be introduced and compared. After that, the locking principle of the Duty-Cycled All-Digital PLL (DC-ADPLL) presented in this thesis will be shown. The difference between the locking principle of presented DC-ADPLL and the other three previous Phase Locked Loops (PLLs), either continuous or duty-cycled, will be obvious by then.

There are a few things to clarify, though, before going on with the comparison and the analysis. Firstly, DCPLLs, by definition, are not constrained to be turned on for only one reference cycle. They can be turned on for multiple reference cycles within a pulse repetition period, in which case, the duty-cycling ratio will be larger. The DC-ADPLL presented in this thesis and the two references to be talked about, [11] and [9], are all turned on for one reference cycle only. Because of this, noise accumulated within this reference cycle is completely cleared and does not affect the timing base of the next burst. Due to this feature, this kind of DCPLL does not need to correct for the jitter contributed by the Digitally Controlled Oscillator (DCO). The reason why the loop in this kind of DCPLL is not turned off after locking is not to correct for phase drift but to correct for temperature drift and quantization errors from sources such as capacitor banks and DTCs. After locking, this kind of DCPLL usually jumps between several codes around the optimum one and the average value changes slowly to correct for temperature drift. DCPLLs that are turned on for multiple reference cycles need not only correct for temperature drift and quantization errors but also phase drift, as in the case in Continuous-operation PLLs (CPLLs). The longer a DCPLL is turned on, the more this type of loop resembles a CPLL. Proper loop design and analysis of multi-cycle DCPLL is more complicated than those of either single-cycle DCPLLs or CPLLs and therefore is beyond the scope of this thesis.

Secondly, while it may appear that analyzing the loop behavior with what happens within two successive references edges is only valid for single-cycle DCPLLs, it will be shown in the following section that this analysis is valid for CPLLs as well, as long as it is assumed that the CPLL is in locking state, when the phase between the rising edge of ref and that of CKV is well defined, which is similar to the phase-aligning property of DCPLLs.

Finally, for simplicity, jitter is not taken into account in the following analysis. As notation convention, in this chapter, without specifying, FCW is defined as frequency control word, which follows Eq. (7-1).

$$FCW = f_{CKV}/f_{ref} = N + \alpha, \tag{7-1a}$$

$$N = \lfloor FCW \rfloor \ . \tag{7-1b}$$

7-1 General Locking Principle of PLLs

While PLL design is a very wide topic and has hundreds and thousands of variants when it goes down to detailed implementations, all PLLs share a common and simple principle.

The function of a PLL is to generate a signal at RF range whose frequency is a multiple of and phase-locked to a low-frequency reference clock, usually at a few tens of MHz. The ratio between the desired frequency RF and that of the reference clock would be a positive number, usually referred to as FCW (or N). The first step is to lock the frequency. This is done by either adjusting the number of counted RF cycles within one reference cycle to be equal to FCW or adjusting the frequency of the RF-divided-by-FCW signal to be equal to that of the reference signal. Once frequency is locked, phase can be aligned by first adjusting the frequency up and down a bit until the rising edges are measured to be aligned and then bringing the frequency back to the locked value. Note that this explanation makes it easy for understanding but is not totally correct since frequency locking and phase locking are not isolated in a PLL.

Anyway, the explanation above gives the basic principle, except neglecting one important fact that both counter and divider are limited to integer operations at circuit level. When FCW has a non-zero fractional part, while the integer part can still be done with an integer-counter and an integer-divider, the fractional part needs some other ways to be taken into account. Early structures such as analog and digital $\Delta\Sigma$ PLLs used $\Delta\Sigma$-quantized multi-modulus integer dividers to mimic a fractional divider [3], [66]. Early ADPLLs used multi-bit Time-to-Digital Converters (TDCs) to quantize the fractional cycle [67]. More recent structures, either counter based or multi-modulus-divider based, add this fractional cycle through a DTC at either CKV edge , [68], or ref edge, [61]. In this way, the period of CKV is locked on the period of ref plus (or minus) a fractional cycle. The plus or minus depends on whether the fractional cycle is added at the CKV or the ref edge. Both approaches are valid but require two opposite translations from the fractional part of FCW to the fractional-cycle delay. The use of DTCs is consistent with early structures based on full-flash TDCs [67]. While a TDC functions like an ADC to quantize an analog value, a DTC reduces the dynamic range of this 'analog' value before feeding it to the TDC. Or, it can be equivalently understood as the TDC being responsible for quantizing the residue error that is beyond the LSB of the DTC. Basically, part of the burden of the TDC is shifted to the DTC. As is mentioned in [4], an $(m + r)$-bit TDC in theory has comparable performance to a combination of an m-bit DTC and an r-bit TDC. However, when taking circuit implementation issues such as element mismatches into account, the combination of these two is advantageous over a full-flash TDC structure. While it may seem that equally distributing the number of bits between TDC and DTC is the optimum approach, a bias towards more bits in the DTC is favorable. In addition to the advantages mentioned in [4], another advantage of DTCs, mentioned in [69], is that a fine-resolution DTC can be implemented by cascading a coarse DTC after a fine stage, while a fine-resolution TDC requires a fine stage at every coarse unit. The relative weighting of DTC and TDC usage can be chosen at will according to the designer. [67] is an extreme end that uses only a TDC. [61] uses a combination of many-bit DTC and few-bit TDC. [68] is another extreme case that reduces the number of TDC bits to unity and puts the burden of fractional support completely on its DTC.

The different combinations of frequency locking and fractional cycle approaches are summarized in Table 7-1. It should be noted that although a divider is used in [61], it is a fixed-modulus divider, simply to divide the RF signal which would be too fast if fed to the counter directly. The mechanism of locking across different values of FCW still relies on the counter.

Table 7-1: Comparison between existing Fractional-N PLL structures and this work

Frequency locking method	Fractional cycle method	Literature
Multi-modulus Divider	$\Delta\Sigma$-quantized Divider	[66]
Counter	Full-flash TDC	[67]
Multi-modulus Divider	DTC at ref edge + 1-bit TDC	[68]
Counter + Prescaler	DTC at ref edge + 4-bit TDC	[61]
Counter + Prescaler	DTC at ref edge + 1-bit TDC	This work

7-2 Locking Principle of a DTC-based Continuous ADPLL

Figure 7-1 shows a simplified block diagram of the ADPLL in [61], for ease of illustrating the principle. CKR is a non-critical sampling clock with an average period equal to that of ref. Its locking principle, illustrated by signal waveforms, is shown in Figure 7-2. Although the structure in [61] actually aligns the phase between ref and CKV-divided-by-2, $CKVD2$, it is assumed for ease of qualitative analysis here that ref is aligned with CKV, instead of CKV-

divided-by-2. The effect of this change will only come into play when it goes down to the actual amount of DTC delay and will be shown later during quantitative analysis. It should also be noted that the word ‘align’ here has a broader definition which not only includes the case when two rising edges happen exactly at the same time, which is the usually understood meaning of ‘align’, but also cases when the two rising edges do not happen at the same time but have a determined phase difference. Also, it is assumed that the structure in [61] is modified to have only 1-bit TDC, thus the fractional support is completely fulfilled by the DTC, which requires the DTC to have a very small quantization error and this quantization error is neglected during qualitative analysis.

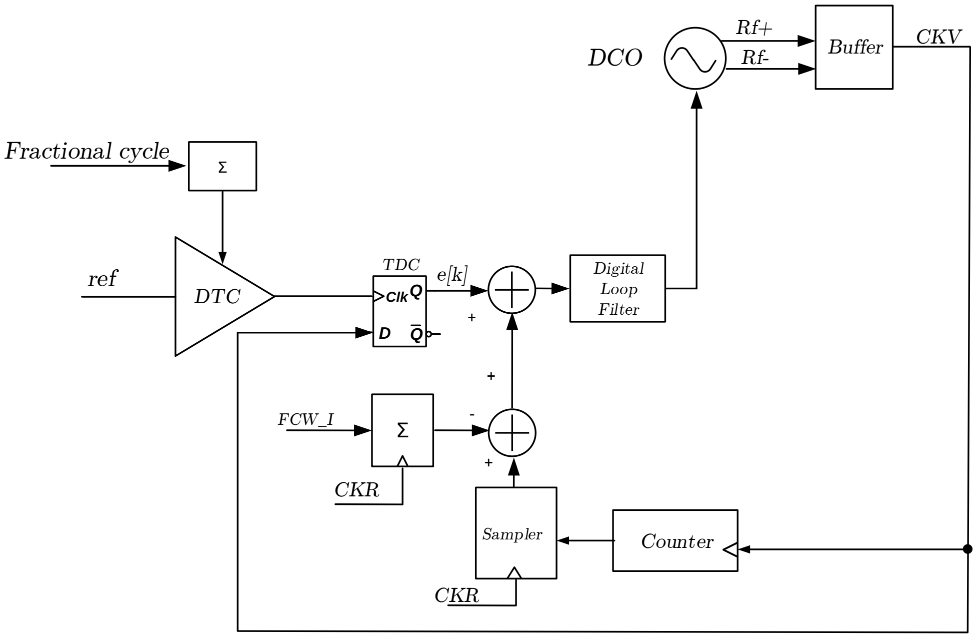


Figure 7-1: Simplified and modified system block diagram to illustrate locking principle of the ADPLL in [61]

The signal $ref\_windowed$ in Figure 7-2 does not have a corresponding signal in Figure 7-1 since there is no actual circuit performing windowing operation in CPLLs. However, this naming is helpful during the comparison with the other two DCPLLs. $ref\_windowed$ can be considered as the signal ref windowed with the observation window of this ref cycle. $ref\_windowed[k]$ $(k = 1,\ 2)$ are two virtual signals that have identical periods as ref but

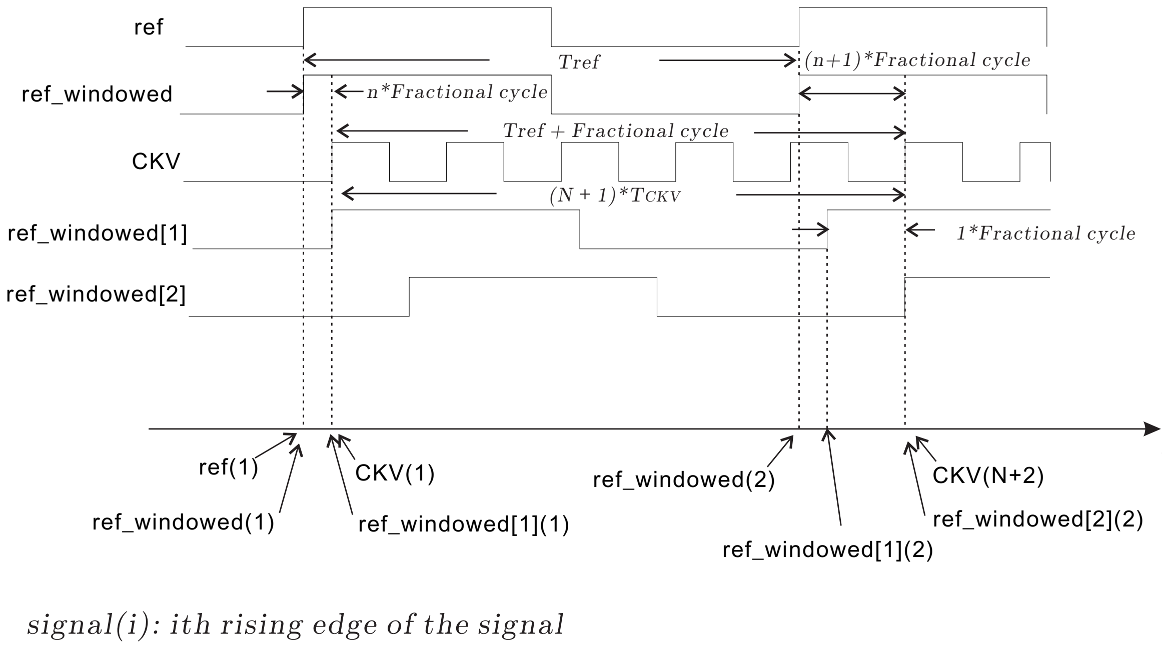


Figure 7-2: ref and CKV between two successive rising ref edges of a continuous ADPLL in [61]

their individual k_{th} rising edge, $ref\_windowed[1](1)$ and $ref\_windowed[2](2)$, are aligned, respectively, to the 1^{st} CKV edge after the i_{th} $(i = 1,\ 2)$ $ref\_windowed$ rising edge.

In equilibrium state, $ref\_windowed\,[1]\,(1)$ is aligned with $CKV(1)$. For $ref\_windowed\,[1]\,(1)$, a match can be found in the circuit, which is $ref(1)$ delayed by a DTC with a delay of $(n * Fractional\ cycle)$. The factor of n is here because we are arbitrarily choosing the observation window from the continuously running ref train. The relative phase between ref and CKV, when neglecting jitter, is a periodic process and it is assumed that we are observing the n_{th} sample within this period, without loss of generality. Since the fractional support completely relies on the DTC, the DTC is responsible for creating a fractional cycle with a very small residual quantization error so that the delay between the two edges sent to the 1-bit TDC is still dominated by the useful information of the jitter instead of being flooded by the residual quantization error of the DTC. As is seen in Figure 7-2, if the delay of the DTC at $ref\_windowed(2)$ is still its value at $ref\_windowed(1)$, which is $(n * Fractional\ cycle)$, then the edges sent to the 1-bit TDC would be $ref\_windowed\,[1]\,(2)$ and $CKV(N+2)$. The quite large $(1 * Fractional\ cycle)$ delay between these two edges would usually flood any other information of jitter contained in this delay. Thus, the code of the DTC needs to be switched before $ref\_windowed(2)$ arrives. An additional delay weighting of $(1 * Fractional\ cycle)$ needs to be added to the DTC so that the total delay of the DTC is $((n+1) * Fractional\ cycle)$ at $ref\_windowed(2)$. In this way, the edges sent to the 1-bit TDC would be $ref\_windowed[2](2)$ and $CKV(N+2)$. Without jitter, the delay between these two edges is zero. Thus, jitter information can be retrieved by comparing these two edges.

To make the analysis a bit more detailed, the value of $(Fractional\ cycle)$ will be derived here. A similar derivation can be found in [70]. Suppose FCW equals $(N+\alpha)$, where N is the integer part, α is the fraction part, and the desired period of CKV is named $\overline{T_{CKV}}$, then we have Eq. (7-2)

$$\overline{T_{CKV}} = \frac{T_{ref}}{N+\alpha} = \frac{T_{ref}}{(N+1)-(1-\alpha)}. \tag{7-2a}$$

$$T_{ref} + \overline{T_{CKV}} \cdot (1-\alpha) = \overline{T_{CKV}} \cdot (N+1). \tag{7-2b}$$

From Eq. (7-2b), it can be seen that an additional delay of $\left(\overline{T_{CKV}} \cdot (1-\alpha)\right)$ needs to be added to $ref(k+1)$ by the DTC so that the timing shown in Eq. (7-3) is met and an equivalent signal whose period equals $\left(T_{ref} + \overline{T_{CKV}} \cdot (1-\alpha)\right)$ is created.

$$t(ref\_windowed[k+1](k+1)) = t(ref\_windowed[k](k)) + T_{ref} + \overline{T_{CKV}} \cdot (1-\alpha). \tag{7-3}$$

Now, back to the assumption made at the beginning of this section, that the fed-back signal to be aligned with ref is CKV instead of *CKVD2*. There is no difference in the operation principle between these two scenarios, but the additional delay added every ref cycle needs to be modified a bit, as shown in Eq. (7-4)

$$N^{'} = \lfloor (N+\alpha)/2 \rfloor .$$

$$\alpha^{'} = (N+\alpha)/2 - N^{'} = \begin{cases} \alpha/2, \ for \ even \ N. \\ \alpha/2 + 0.5, \ for \ odd \ N. \end{cases}$$

$$(N+\alpha)/2 = N^{'} + \alpha^{'} .$$

$$FCW = (N+\alpha) = 2 \cdot (N^{'} + \alpha^{'}).$$

$$\overline{T_{CKVD2}} = \overline{T_{CKV}} \cdot 2 = \frac{2 \cdot T_{ref}}{(N+\alpha)} = \frac{T_{ref}}{(N'+\alpha')} = \frac{T_{ref}}{(N'+1)-(1-\alpha')}.$$

$$T_{ref} + \overline{T_{CKVD2}} \cdot (1-\alpha^{'}) = \overline{T_{CKVD2}} \cdot (N^{'}+1).$$

$$T_{ref} + \overline{T_{CKV}} \cdot 2 \cdot (1-\alpha^{'}) = \overline{T_{CKVD2}} \cdot (N^{'}+1). \tag{7-4}$$

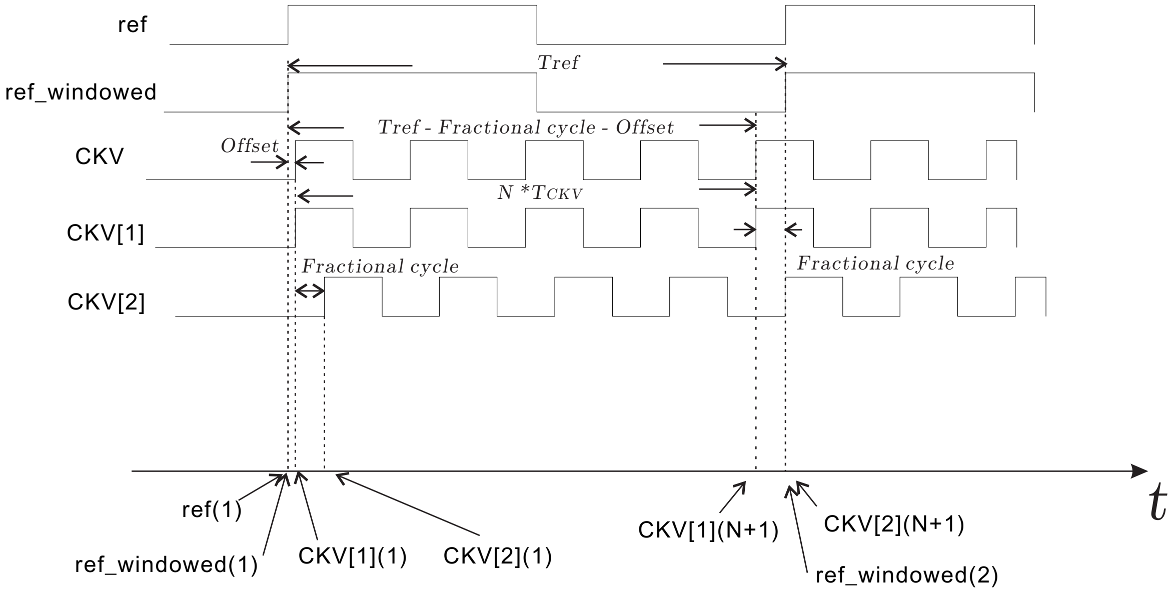
It can be seen that the additional delay that needs to be added to ref edge each cycle is $\left(\overline{T_{CKV}} \cdot 2 \cdot (1-\alpha^{'})\right)$, where $\alpha^{'}$ is the fractional part of $FCW/2$.

7-3 Locking Principles of Existing DCPLLs

7-3-1 Locking Principle of a 25% Fractional Resolution Ring Oscillator Based DCPLL

Now that we have gone through the locking principle of a (digital) CPLL, the analysis of DCPLLs will be easier to understand. Figure 7-3 shows the waveforms of ref and CKV between two successive rising edges of ref for the DCPLL presented in [11]. The naming conventions follow from those used in the last section. The difference is that $ref\_windowed[k]$ are missing while two new signals named $CKV[k], (k = 1, 2)$ take place. This is because the fractional-N support in [11] is achieved by adding the fractional delay at CKV instead of $ref\_windowed$. $CKV\,[k]$ $(k = 1, 2)$ are two virtual signals that have periods identical to CKV. $CKV[1](1)$ is aligned with $CKV(1)$ while $CKV\,[2]\,(N+1)$ is aligned with $ref\_windowed(2)$. In principle, in order to lock the frequency of CKV to that of ref with either an integer or an integer-plus-fraction ratio, measures need to be taken to align

$CKV\,[1]\,(1)$ with $ref\_windowed(1)$. However, this is impossible to achieve with the architecture in [11]. Firstly, the architecture in [11] adds the fractional cycle at the CKV edge, which means that the resulting $CKV[1](1)$ can only be physically later than $CKV(1)$, or at the same time as $CKV(1)$ if no measures are taken. However, the DCO itself is triggered by $ref\_windowed(1)$, so $CKV(1)$ is physically later than $ref\_windowed(1)$, therefore it is impossible to align $CKV[1](1)$ with $ref\_windowed(1)$. The design in [11] therefore abandons the aligning of the 1^{st} edge, which is why neither $CKV[1](1)$ nor $CKV(1)$ is aligned with $ref\_windowed(1)$.



*Fractional cycle = 0.25*Tckv*

signal(i): ith rising edge of the signal

Figure 7-3: ref and CKV between two successive rising ref edges of the DCPLL in [11]

In integer mode, the DCPLL in Figure 7-3 locks CKV with an accurate $ref\_windowed(2)$ and a not so accurately defined edge, which is $ref\_windowed(1)$ delayed by an offset. It is reported in [11] that this delay offset is about 1.2 ns while the period of the reference clock is 50 ns. This delay includes error sources such as turning-on time of the switch and the delay between the turning-on of the switch and the 1^{st} RF output edge.

The DCPLL in [11] achieves fractional support by exploiting the quadrature-phase property of the ring oscillator. $CKV[2](N+1)$, which is $0.25 \cdot T_{CKV}$ later than $CKV(N+1)$, is used as the alignment edge with $ref\_windowed(2)$. Because of this mechanism, the fractional resolution is limited to $0.25 \cdot f_{ref}$. Actually, $CKV[2]$ has a corresponding signal in the circuit. If CKV is defined as the zero-phase output, then $CKV[2]$ is the 90° shifted node in the ring oscillator.

7-3-2 Locking principle of an Integer-N Ring Oscillator Based DCPLL

Figure 7-4 shows ref and CKV waveforms of the DCPLL reported in [9]. Just like other DCPLLs, the DCPLL in [9] also has an initial offset, which is shown in Figure 7-4. However, this offset between $ref\_windowed(1)$ and $CKV[1](1)$ is compensated for. Firstly, the delay between $ref\_windowed(1)$ and $CKV[1](1)$ is used to control the charging time of a capacitor, which translates the delay between them into the voltage domain. Next, the same is done for the delay between $ref\_windowed(2)$ and $CKV[2](N+1)$. Then, the loop compares these two voltages and forces them to be equal. In this way, the frequency of CKV will be locked to integer multiples of f_{ref} without residual error caused by the offset shown in Figure 7-4. This offset cancellation technique resembles the one used in analog PLLs, where charge-pumps are used. It can be considered as an analog version of the full-flash TDC approach. As is explained in Subsection 1-4-1, this technique is analog insensitive and does not scale well with technology.

The fact that the delay offset is only measured and there is no additional delay added to either the ref path or the CKV path results in CKV, $CKV[1]$ and $CKV[2]$ being three identical signals after locking. Finally, the DCPLL in [9] only supports integer operation.

7-4 Locking Principle of the DC-ADPLL

After the above analysis of one continuous ADPLL and two DCPLLs, it finally comes to the DC-ADPLL of this thesis. The DC-ADPLL borrows the fractional support method from continuous ADPLLs while its duty-cycling operation is borrowed from DCPLLs. The result

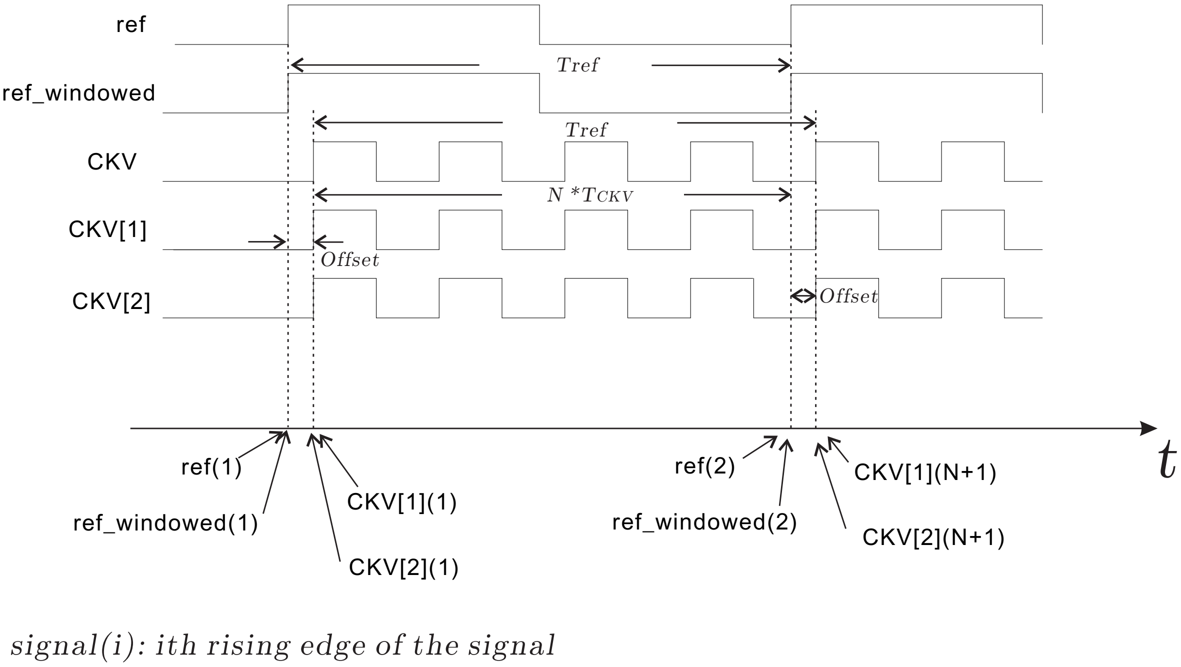
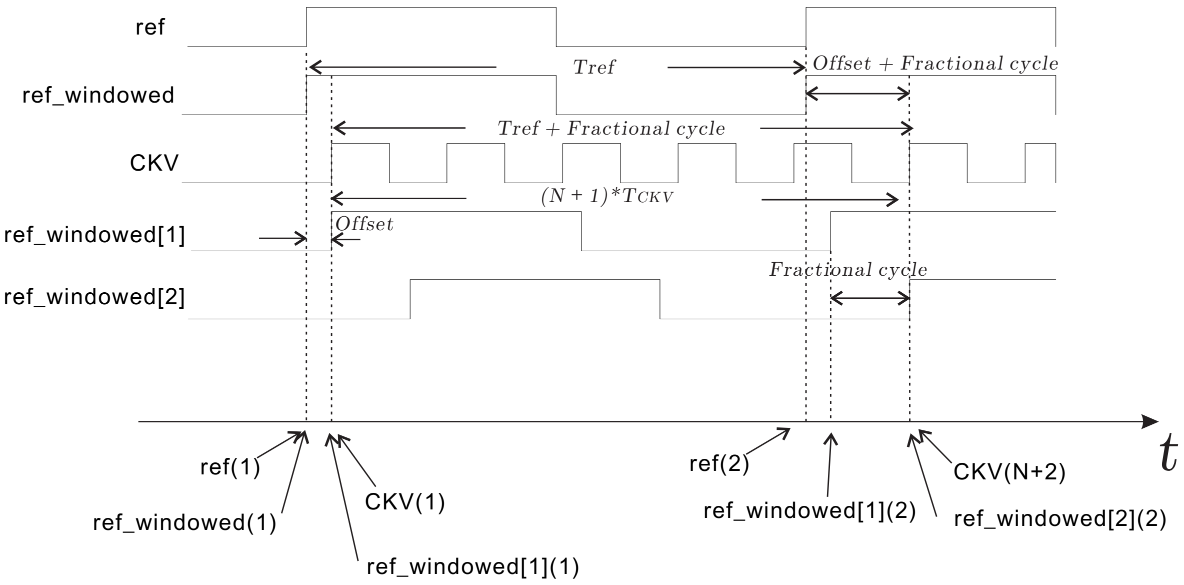


Figure 7-4: ref and CKV between two successive rising ref edges of the DCPLL in [9]

would be a first-ever hybrid DC-ADPLL that is extremely low in noise, thanks to the LC oscillator, supports high fractional resolution and cancels initial offset with digital implementation when compared to existing DCPLLs. Compared to existing continuous ADPLLs, the DC-ADPLL is extremely low in average power. The duty-cycling ratio of this DC-ADPLL is 4.4%, which means the average power would be more than 20 times lower than a comparable ADPLL that is always on.

Figure 7-5 shows ref and CKV waveforms of the DC-ADPLL presented in this thesis. $ref\_windowed[1](1)$ is aligned with the 1^{st} CKV edge at which time the frequency is considered to have fully settled. Here, for simplicity, it is assumed that this edge is the 1^{st} edge of CKV, $CKV(1)$. $ref\_windowed[2](N+2)$ is aligned with the CKV edge that is $(N+1)$ CKV cycles later than the 1^{st} CKV edge at which the frequency is considered to be fully settled. Here, since the previously chosen edge is $CKV(1)$, this edge is considered to be $CKV(N+2)$ and thus $ref\_windowed[2](2)$ is aligned with $CKV(N+2)$. In later analysis, even if the chosen CKV edges are not $CKV(1)$ and $CKV(N+2)$, the relationship still follows that $ref\_windowed[2](2)$ is later than and separated with $ref\_windowed[1](1)$ by $(N+1)\cdot T_{CKV}$.



signal(i): ith rising edge of the signal

Figure 7-5: ref and CKV between two successive rising ref edges of the DC-ADPLL in this design

Several similarities can be identified in Figure 7-5 right away. Firstly, there is a delay offset between $CKV(1)$ and $ref\_windowed(1)$, which also appeared in the waveforms of the two DCPLLs introduced above. Furthermore, it can be seen that the delay between $ref\_windowed[2](2)$ and $ref\_windowed[1](2)$ is $(Fractional\ cycle)$, which resembles the situation in a continuous ADPLL. The basic idea is to add a delay to the $ref\_windowed$ signal through a DTC, whose delay is set to *Offset* at $ref(1)$ and *(Offset+Fractional cycle)* at $ref(2)$. In this way, initial offset is cancelled and fractional support is achieved. The comparison of features of the DC-ADPLL in this thesis and the above three PLLs analyzed is shown in Table 7-2.

Table 7-2: Comparison of features with previous works

	This work	[61]	[11]	[9]
Duty-cycling	Yes	No	Yes	Yes
Fractional support	Yes	Yes	Yes	No
Fractional-N approach	Digital	Digital	Digital	None
Fractional resolution	High	High	0.25*f_{ref}	None
Oscillator Type	LC	LC	Ring	Ring
Noise	Low	Lowest	High	High
1^{st} edge aligning/ quantizing	Digital	Digital	None	Analog
Residue offset	No	No	Yes	No

7-5 The CKV Edge-sampling Circuitry

7-5-1 Motivation and Design Procedure

A quite advanced technique called 'snapshot' is implemented in [61]. It is realized in [61] by adding a customized circuit block called 'Snapshot & CKR gen'. While this block may seem to be a band-aid at first glance, it turns out to be quite useful in this DC-ADPLL after a few modifications.

The 'snapshot' circuitry aims to solve two issues [70]. In earlier ADPLL structures, the ref clock is re-sampled by the CKV clock [71] and the re-sampled clock CKR is compared with CKV through a many-bit TDC to retrieve information about the fractional cycle. Doing this introduces two issues [70]. The first one is that if ref edge is exactly aligned with the CKV edge, the sampling runs into a meta-stability problem and may fail. The second one is that presenting this high frequency CKV at the clock input of a register burns unnecessary power when it comes to gate-level analysis, although the output of this sampler is charged only once every ref cycle. The 'snapshot' aims to mitigate these two issues by first sampling CKV with a clock, $FREF_{dly}$ [61], whose period is $(Tref + Fractional\ cycle)$ so that both edges sent to the TDC are at low frequency and thus the above power issue is mitigated. $FREF_{dly}$ is generated in a similar way as was discussed in Section 7-2, which is achieved by adding a DTC after ref, whose delay is increased by $(Fractional\ cycle)$ every ref period. The difference between $FREF_{dly}$ and the one described at Section 7-2 is that the rising edges of $FREF_{dly}$

is deliberately made mis-aligned with the rising edges of CKV. Basically, $FREF_{dly}$ is earlier than CKV by a delay that is equal to the offset of the TDC and is used to sample CKV. In this way, the meta-stability problem is solved as well. The residual delay between $FREF_{dly}$ and the sampled CKV, $CKVD2_s$ [61], is forwarded to a multi-bit TDC to be quantized.

However, this technique cannot be directly applied to cases where a many-bit DTC and a single-bit TDC are used. As explained previously, when the TDC is single-bit, residual delay between the two signals sent to it must not flood the information of the jitter. Ideally, without jitter, the input signals at the 1-bit TDC should be perfectly aligned. In [61], however, the two input signals of the TDC are $FREF_{dly}$ and $CKVD2_s$, where $CKVD2_s$ itself is sampled by and definitely later than $FREF_{dly}$. Thus, residual error from the DTC in [61] is not only large but also always has the same sign and thus cannot be used in single-bit TDC applications.

However, the idea of this circuitry is still valid and the circuitry can be modified a bit and turns out to be quite useful in DC-ADPLLs.

Recalling Figure 7-5, which is redrawn as Figure 7-6 for readers' convenience, in order to achieve fractional support with a single-bit TDC, a signal needs to be created whose 1^{st} rising edge is aligned with $CKV(1)$ and whose 2^{nd} edge is aligned with $CKV(N+2)$. Furthermore, the period of this signal needs to be $(Tref + Fractional\ cycle)$. This signal is named $ref2$, which is drawn in Figure 7-6 as well. While fractional-N operation can be achieved by feeding $ref2$ and CKV into a single-bit TDC, $ref2$ itself is a terrible clock to sample CKV, if the low-power 'snapshot' circuitry in [61] is to be implemented.

However, it is noted in Figure 7-6 that $ref\_windowed(1)$ would be a quite good edge to sample $CKV(1)$ since it is physically before $CKV(1)$ and the delay between these two edges, i.e., *Offset*, is pretty well known. The default value of *Offset* is a sum of the delay of the inverter line to drive the large switch in the inductor and $(1/4) \cdot T_{CKV}$. The former term has a length of about 100 ps to 200 ps with negligible variation, which is the jitter level of a ring oscillator, when it comes to avoiding the meta-stability problem. The latter term varies according to the operation RF frequency. For the supported frequency range from 5.0 GHz to 7.0 GHz, the variation is 14 ps. Assume that the nominal delay of the inverter chain is

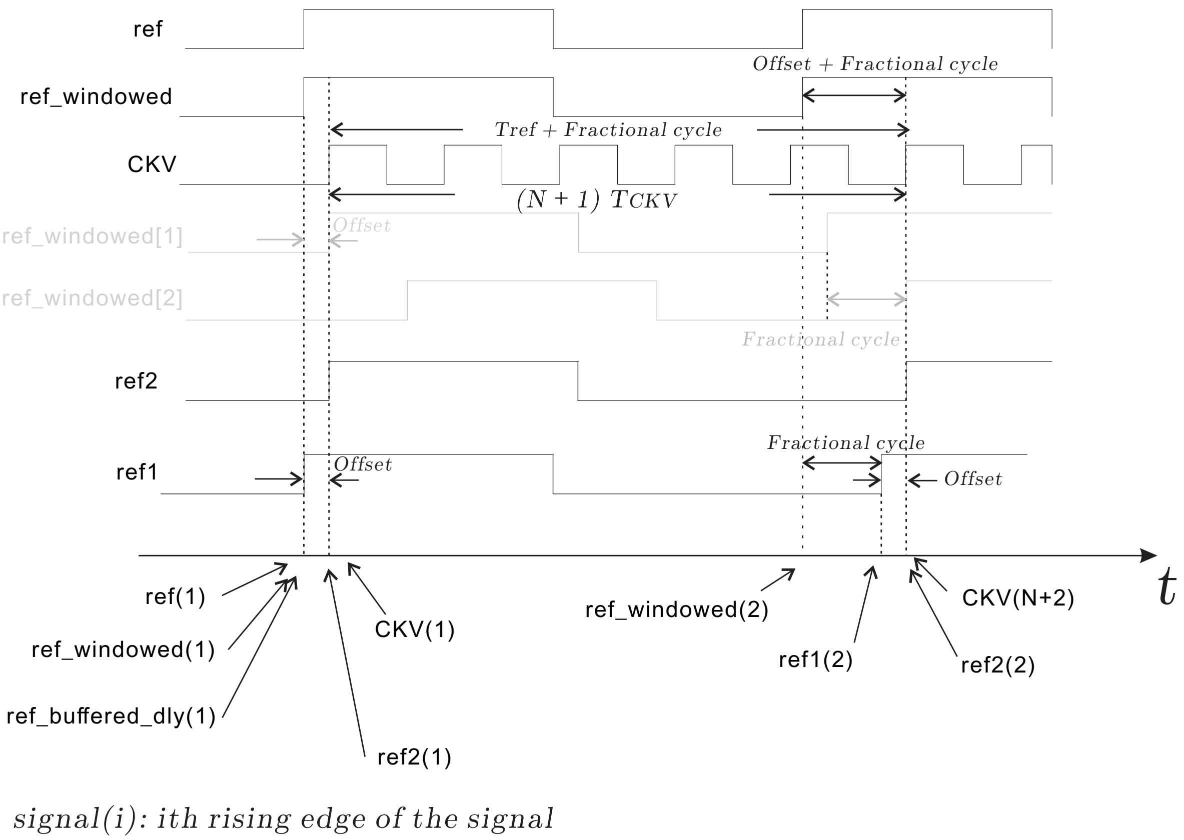


Figure 7-6: Redrawing of Figure 7-5 for the design of CKV edge-sampling circuitry

150 ps, then *Offset* has a default average delay of 193 ps and a variation of 14 ps across the entire operation frequency range. It means that if the signal $ref1$ meets the following timing requirements shown in Eq. (7-5), then it will be a pretty good clock to sample CKV.

$$t(ref1(1)) = t(ref\_windowed(1)). \tag{7-5a}$$

$$t(ref1(2)) = t(ref\_windowed(2)) + (Fractional\ cycle)\,. \tag{7-5b}$$

Meanwhile, it can be noted from Figure 7-6 that a signal that satisfies the conditions specified by Eq. (7-5) also satisfies the condition specified in Eq. (7-6)

$$ref1(i) = ref2(i) - \mathit{Offset},\ \ (i = 1,\ 2). \tag{7-6}$$

What Eq. (7-6) suggests is that if $ref1$ satisfies Eq. (7-5), then $ref2$, whose 1^{st} edge is aligned with $CKV(1)$ satisfying Eq. (7-7):

$$ref2(1) - ref\_windowed(1) = \mathit{Offset}, \tag{7-7}$$

and also whose 2^{nd} edge is later than $ref1(2)$ by *Offset* as well, as shown in Eq. (7-8a), will have its 2^{nd} edge aligned with $CKV(N+2)$, as shown in Eq. (7-8b).

$$ref2(2) = ref1(2) + \mathit{Offset}. \tag{7-8a}$$

$$ref2(2) = CKV(N+2). \tag{7-8b}$$

The relation between $ref1$ and $ref2$ can be presented from another aspect. If $ref2$ is aligned with CKV at both edges and that $Offset = ref2(1) - ref\_windowed(1)$ is recorded, then a signal earlier than $ref2$ by a delay equal *Offset* at both edges can be a quite good clock to sample CKV. This signal, namely $ref1$, satisfies Eq. (7-5).

7-5-2 Circuit Implementation and Timing Constraints

The above analysis suggests the implementation of $ref1$ and $ref2$ with the circuitry shown in Figure 7-7, with several differences as compared to the qualitative waveforms shown in Figure 7-6.

Firstly, CKV is divided by 4 before being sampled. This pre-scaling does not change the locking principle and only matters in quantitative analysis, as is shown back at Section 7-2. The reason for this division is because the counter is not fast enough to count a signal up to 7.0 GHz.

Secondly, the edge to be aligned would be a sampled and delayed version of $CKVD4$, which is named $CKVD4S$. This is a direct result from the sampling of $CKVD4$, due to t_p, the clock-to-output delay of a flip-flop. Since the two $CKVD4$ edges, one at the 1^{st} and the other at the 2^{nd} rising edge of ref, are delayed by the same amount, the principle of locking remains the same. The edge-sampling circuitry is shown in Figure 7-8. The signal $ref\_windowed$ is generated by an FSM for the outer loop, which sets the positions of the DC-ADPLL control

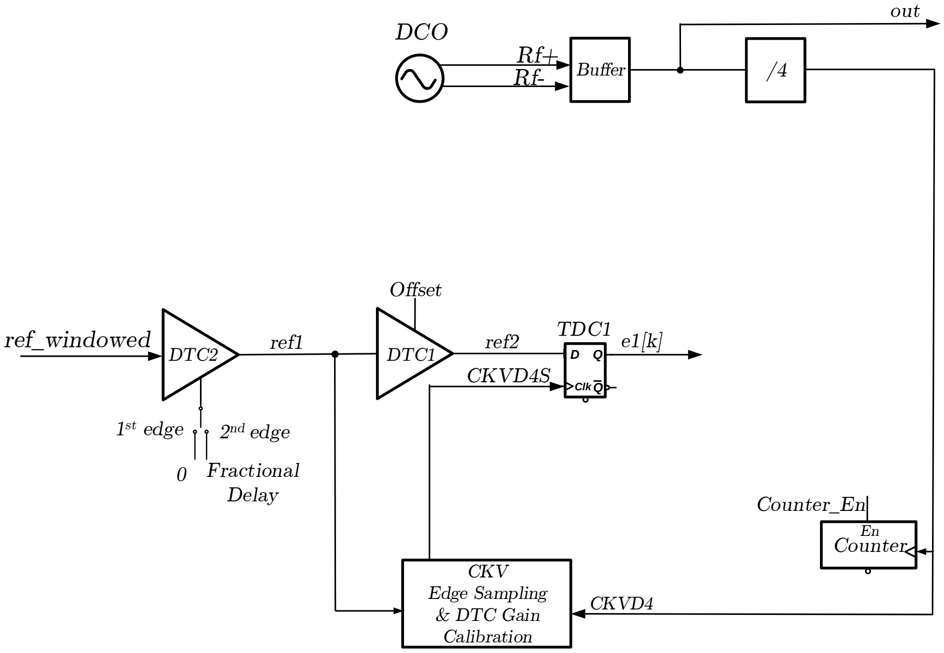


Figure 7-7: System block diagram including only DTCs and the edge-sampling circuitry

signals within a pulse repetition period. Basically, it windows the continuously running ref clock during the existence of a burst. DTC3 is an auxiliary DTC for gain calibration and can be overlooked at this moment.

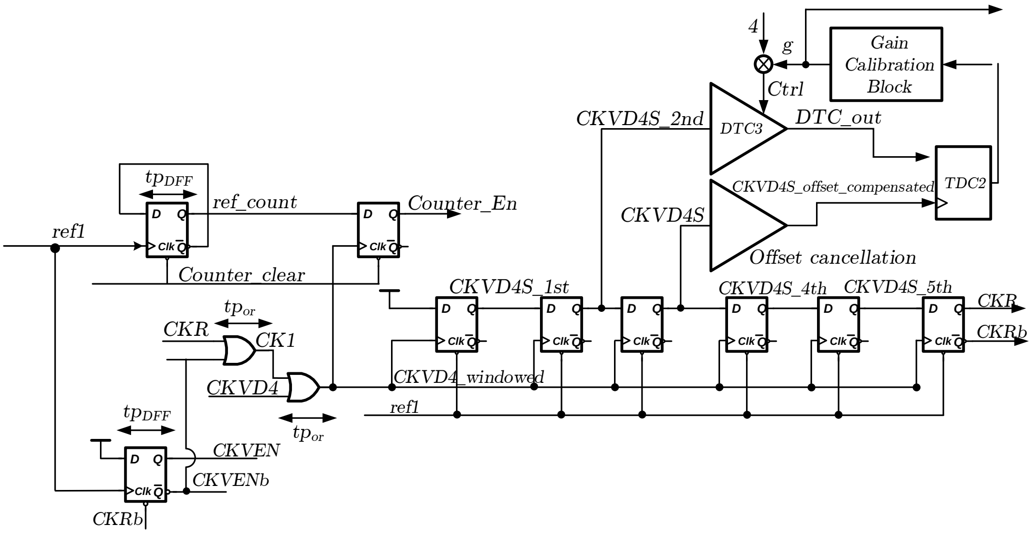


Figure 7-8: Block diagram for CKV edge-sampling circuitry

A far-out view of waveforms of signals in edge-sampling circuitry, covering a complete burst duration, is shown in Figure 7-9, while two close-in waveforms, covering, respectively, only a short time of interest after the 1^{st} and 2^{nd} $ref\_windowed$ rising edge, are shown in Figure 7-10 and Figure 7-11.

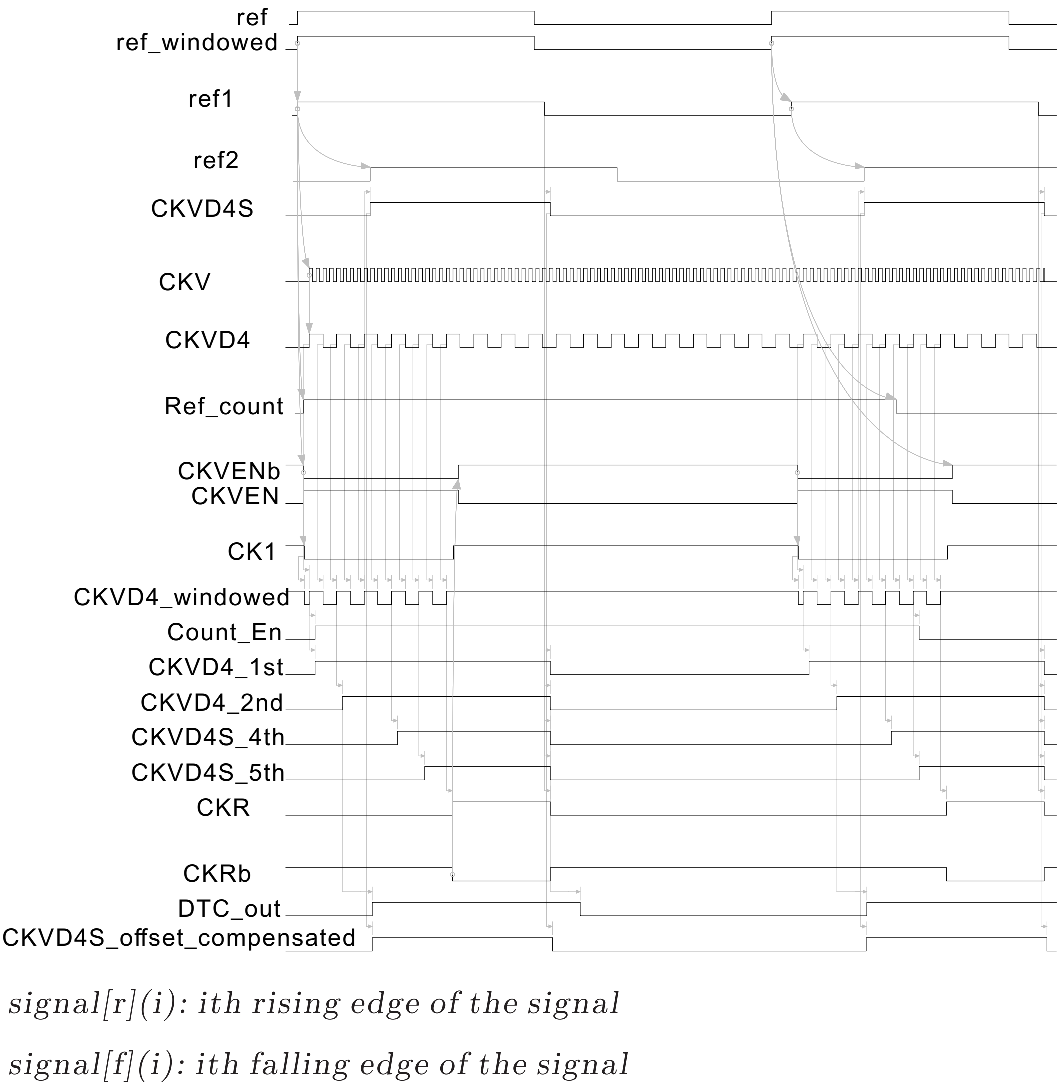


Figure 7-9: Far-out signal waveforms of edge-sampling circuitry within one burst window

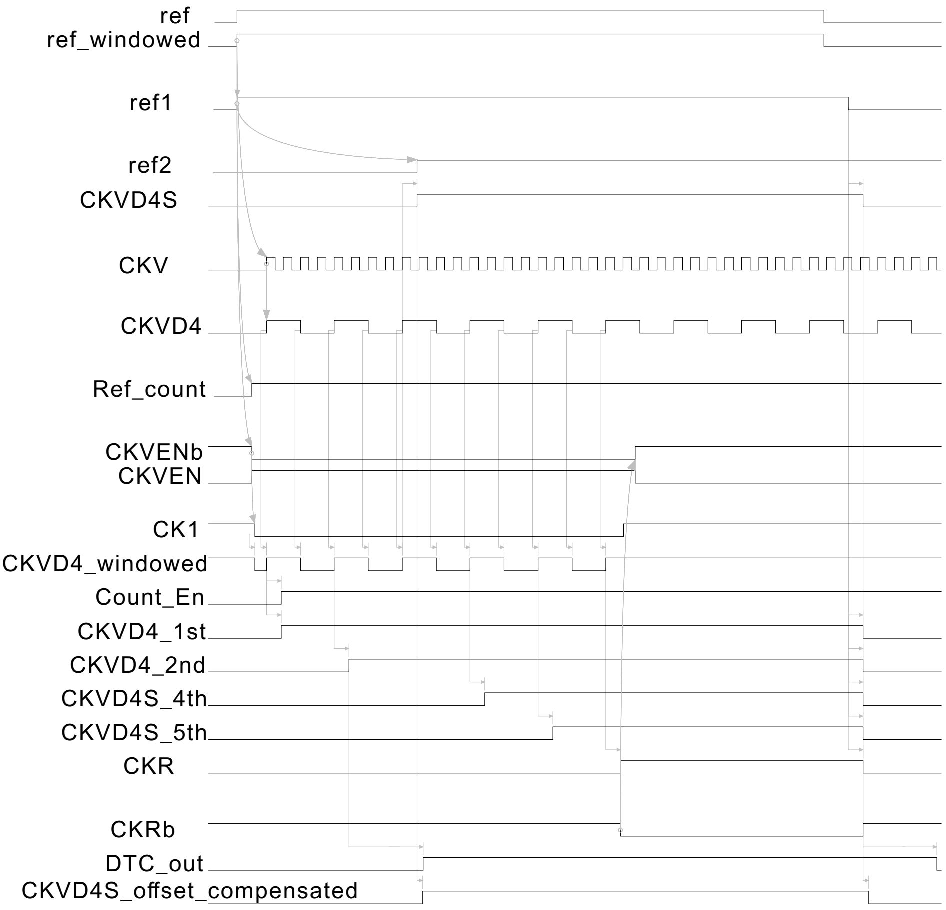
The effectiveness in reducing power of this edge-sampling circuitry can be seen by observing $CKVD4\_windowed$ in Figure 7-9, which gates the clock pins of the DFFs. Since it makes much fewer transitions than $CKVD4$, dynamic power consumption of those DFFs is reduced.

1^{st} **rising edge (Figure 7-10):** The DCO is triggered by $ref\_windowed\,[r]\,(1)$ and the 1^{st} CKV divided-by-4 output, $CKVD4$, appears after some delay. This delay is mainly composed of the following 3 sources:

1. The propagation delay of the inverter chain to drive the huge low-ohmic switch in the split inductor.

2. $(1/4) \cdot T_{CKV}$ due to the fact that the DCO starts from a 90° phase.

3. The propagation delay from the output buffer and the divider.



signal[r](i): ith rising edge of the signal

signal[f](i): ith falling edge of the signal

Figure 7-10: Close-in signal waveforms of edge-sampling circuitry at 1^{st} rising edge of $ref\_windowed$

The discharge of $CK1$ is triggered by $CKVENb\,[f]\,(1)$ at $t(CKVENb\,[f]\,(1))$. The timing between $CKVENb\,[f]\,(1)$ and $ref1\,[r]\,(1)$ follows:

$$t(CKVENb\,[f]\,(1)) - t(ref1\,[r]\,(1)) = t_{p_{DFF}}. \tag{7-9}$$

In order to catch $CKVD4\,[r]\,(1)$, the timing constraint on $CK1\,[f]\,(1)$ follows:

$$t\,(CK1\,[f]\,(1)) < t\,(CKVD4\,[r]\,(1))\,. \tag{7-10}$$

This timing constraint can be translated into a constraint between $ref1$ and $CKVD4$:

$$t(ref1\,[r]\,(1)) + t_{p_{DFF}} + t_{p_{or}} < t(CKVD4\,[r]\,(1)). \tag{7-11}$$

While Eq. (7-5a) sets an upper bound for $t(ref\_buffered\_dly\,[r]\,(1))$, it does not show the lower bound of it. If there is a variable $Slack$ defined as

$$Slack = t(CKVD4\,[r]\,(1)) - t(ref1\,[r]\,(1)) + t_{p_{DFF}} + t_{p_{or}}, \tag{7-12}$$

then the definition shown in Eq. (7-12) requires $Slack > 0$. However, it has been shown in Figure 7-6 that:

$$t(CKVD4\,[r]\,(N^{''} + 2)) - t(ref1\,[r]\,(2)) = t(CKVD4\,[r]\,(1)) - t(ref1\,[r]\,(1)), \tag{7-13}$$

$$N^{''} + \alpha^{''} = FCW/4. \tag{7-14}$$

Basically, the same $Slack$ applies to the sampling at the 2^{nd} $ref1$ edge as well. In order to correctly sample $CKVD4\,[r]\,(N^{''} + 2)$ instead of $CKVD4\,[r]\,(N^{''} + 1)$, $Slack$ must also satisfy:

$$Slack < T_{CKVD4}. \tag{7-15}$$

In total, the constraint for $Slack$ is:

$$0 < Slack < T_{CKVD4}. \tag{7-16}$$

Instead of specifying the measuring window by sending an Enable/Disable signal to the divider as in [7], the counting window of CKV is specified through an Enable/Disable signal of the counter in this design. The signal $Counter\_En$ is generated by sampling ref_count with $CKVD4$. In this way, $Counter\_En$ makes transitions just a little after $CKVD4\,[r]\,(1)$ and $CKVD4\,[r]\,(N^{''}+2)$, so that no meta-stability problem occurs at the counter. After locking, the counter should count to $(N^{''}+1)$.

The timing constraint for proper operation is given in Eq. (7-17):

$$t(ref1\,[r]\,(1)) + t_{p_{DFF}} + t_{hold} < t(CKVD4\,[r]\,(1)) + t_{p_{or}}. \tag{7-17}$$

There are multiple choices regarding which $CKVD4$ edge to be sampled by the PLL. The rule of thumb is that the frequency of $CKVD4$ should have settled within the defined specification at the chosen edge. In [7], the 1^{st} edge used is 2 ns after the oscillator is switched on. From the simulation of the DCO at Chapter 6, the frequency settles within specification after 1.5 ns since the initial switching signal, at both LB and HB. Subtracting an inverter chain delay of about 300 ps from this 1.5 ns, it shows that the frequency settles after about 1.2 ns since the 90° starting edge. Thus, $CKVD4\,[r]\,(3)$ is chosen. It is later than $CKVD4\,[r]\,(1)$ by 1.6 ns at 5.0 GHz and 1.1 ns at 7.0 GHz, which can be considered accurate.

Another 2 DFF stages are inserted between $CKVD4S$ and CKR. The purpose of this is to ensure proper sampling and readout of the TDC results. The sampling window closes as soon as $CKVD4\,[r]\,(1)$ propagates through the DFFs and pulls up CKR.

2^{nd} **rising edge (Figure 7-11):** The principle of the edge-sampling circuitry after the 2^{nd} edge of ref resembles that at the 1^{st} edge of ref and therefore does not need to be repeated again. However, one more thing to note is that another function of CKR is to turn off the

DCO after the 2^{nd} rising edge of ref. In this way, the DCO is turned on for a bit longer than one reference cycle plus the fractional cycle[1], but proper sampling at the 2^{nd} edge can be assured.

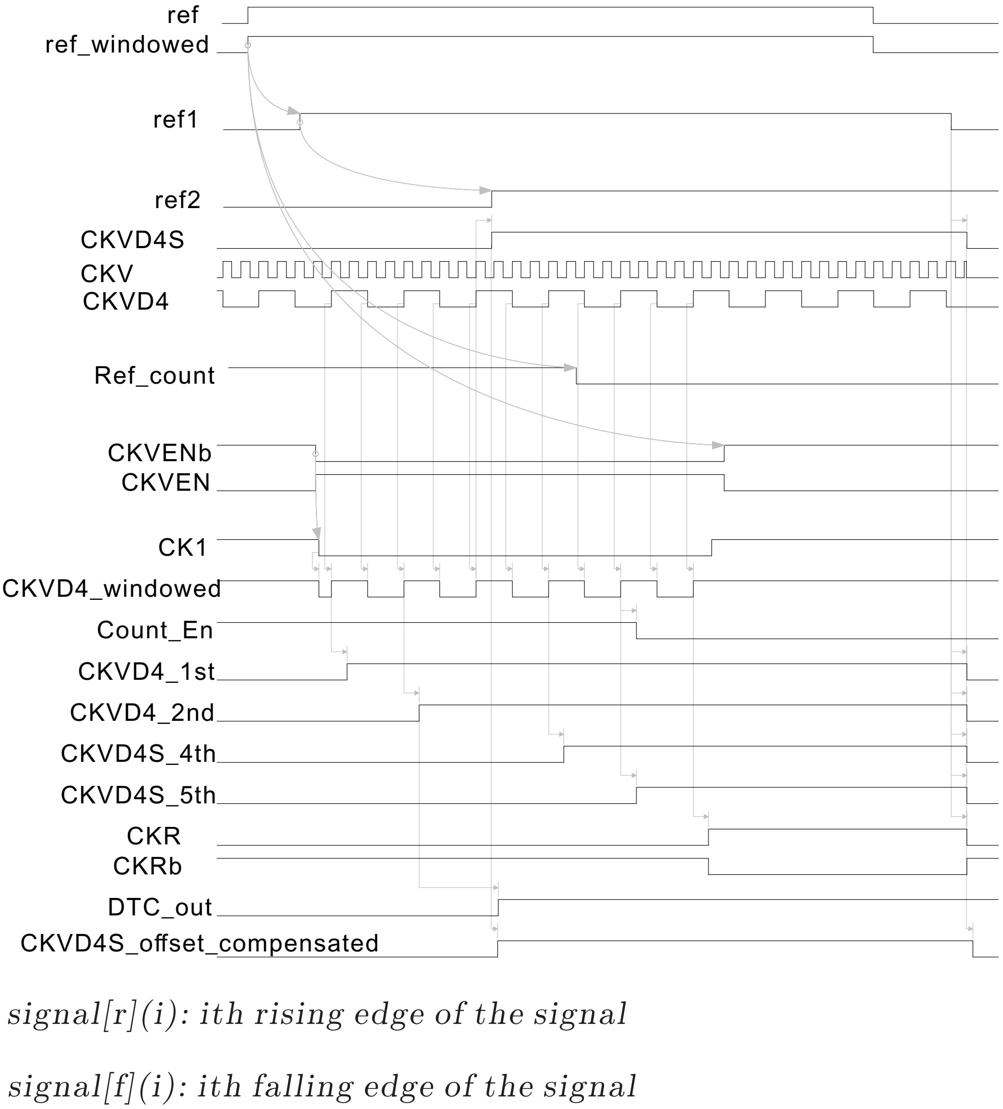


Figure 7-11: Close-in signal waveforms of edge-sampling circuitry at 2^{nd} rising edge of $ref\_windowed$

The ultimate duty-cycling ratio that can be achieved is calculated from the 25 MHz reference frequency and the 1 MHz pulse repetition rate, which is 4%. Turning on the DCO for a few more cycles increases this ratio by 10% and the resulting duty-cycling ratio is 4.4%.

[1] In Figure 7-11, the number of CKV cycles that the oscillator is turned on for extra is accurate, while the scale between ref and CKV is off in order to make CKV observable. Thus, it may seem from Figure 7-11 that the oscillator is turned on for almost an extra of half the ref period. Actually, this extra time is only 10% of the ref period.

Finally, it is realized that several timing constraints need to be met to ensure proper functionality, and the default occurrence of $CKVD4\,[r]\,(1)$ might not meet the constraint. Thus, two optional delays are added to the reference path and the DCO triggering path to bias the slack in either positive or negative direction. A more detailed system block diagram with further modifications, compared to Figure 7-8, is shown in Figure 7-12. A simplified version of the timing diagram showing relative timing relations between critical signals is shown in Figure 7-13. This time, zero-code offsets of the DTCs are also taken into consideration.

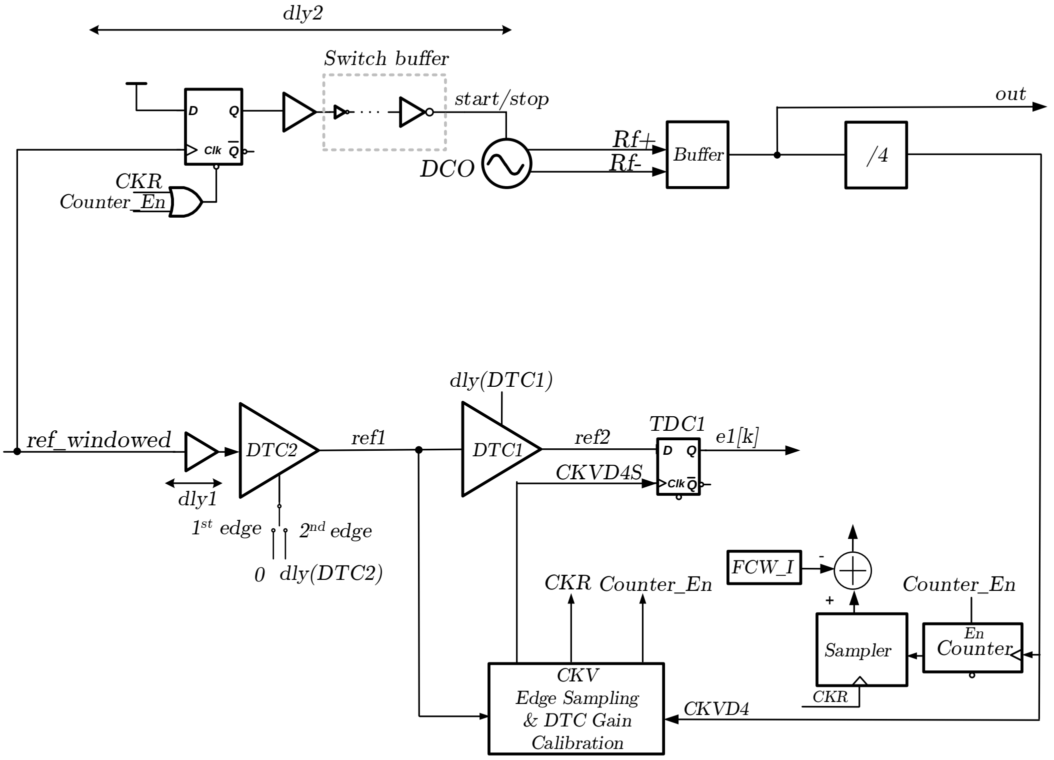


Figure 7-12: System block diagram after adding controls from edge-sampling block and optional delays

The signal $CKVD4S$ in Figure 7-7 is replaced with $CKVD4S\_offset\_compensated$ in Figure 7-12, both of which can be found in Figure 7-8. $CKVD4S\_offset\_compensated$ is used to be compared with the reference clock for the following considerations:

1. The DTC used in this design has an offset of about 140 ps at zero code. This means that $ref2$ is later than $ref1$ by at least a delay equal this offset. If the sampled $CKVD4$, i.e.,

$CKVD4S$, is earlier than that, then the loop cannot be locked. Although $CKVD4S$ in this design is later than $ref1$ by at least $8 \cdot T_{CKV}$ (1.6 ns @ 5.0 GHz and 1.1 ns @ 7.0 GHz), which is safe enough to eliminate this concern, to increase the robustness of the design, the signal $CKVD4S\_offset\_compensated$ is used to completely eliminate this constraint.

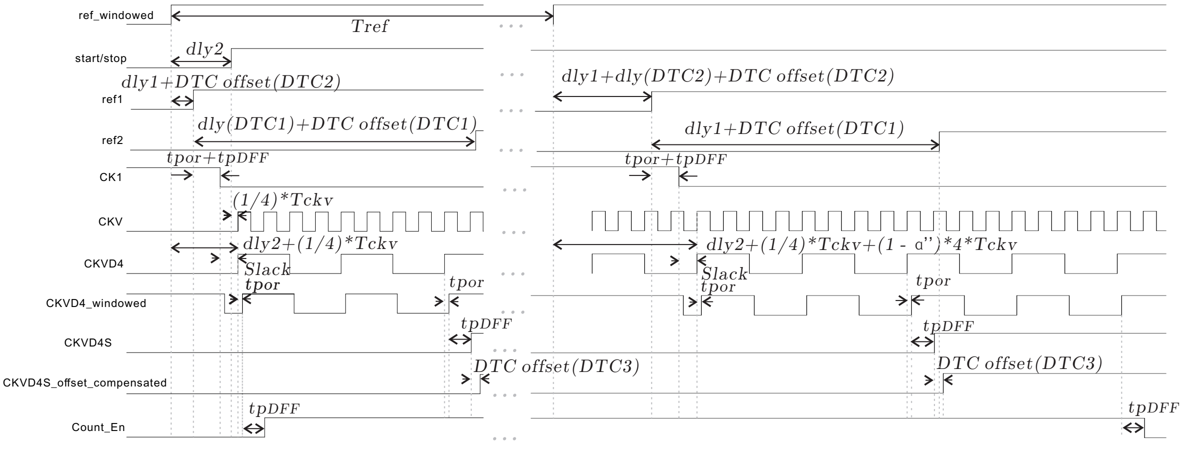
2. One of the TDC inputs, $ref2$, is the output of DTC2. The DTC used in this design employs a coarse-fine structure and the fine stage is realized by loading a capacitor bank at the output of the coarse stage. The slope of the waveform at that node becomes more and more gentle when the delay of the fine stage increase. Although 2 inverters are inserted between the output of the fine stage and the output of the DTC, the slope variation still has residue effect on the signal waveform at the DTC output node. The TDC might be triggered at an incorrect value when its inputs are close enough. Thus, $CKVD4S\_offset\_compensated$, which is also the output of a DTC and has a much closer slope to $ref2$, is used to be compared with $ref2$.

3. $CKVD4S$ is a direct output from a DFF, which is implemented by a standard digital cell and thus has a low driving capability. Extra loading of the TDC might cause variation of the slope of $CKVD4S$ and results in detection error. Thus, the signal $CKVD4S$ is sent to the DTC input, which has a low input capacitance, and is then buffered by several inverters in the DTC until the output signal has a higher driving capability.

Among all the timing constraints mentioned in Subsection 7-5-2, it is assumed that the timing constraint between $CK1$ and $CKVD4$, rather than Ref_count and $CKVD4\_windowed$, is most stringent, under the reasonable assumption that $t_{hold} < 2 \cdot t_{p_{or}}$.

Furthermore, the delay contributed from the DCO output buffer and the divider is not taken into consideration for simplicity of analysis. During the design phase, these two delays can be absorbed into $dly2$ as far as timing constraints are concerned.

It is noted from Figure 7-13 that, in cases where $dly2 + (1/4) * T_{CKV} > \alpha'' * 4 * T_{CKV}$, the 1^{st} $CKVD4$ edge sampled after $ref\_windowed[r](2)$ is not the 1^{st}, but the 2^{nd}, $CKVD4$



FCW/4 = N'' + α''

*dly(DTC2) =(1 - α'')*4*Tckv*

DTC offset(DTC1)= DTC offset(DTC2)=DTC offset(DTC3)

signal[r](i): ith rising edge of the signal

signal[f](i): ith falling edge of the signal

Figure 7-13: Redrawing of Figure 7-9 with only critical signals including timing constraints

after $ref\_windowed\,[r]\,(2)$. However, it is always $CKVD4\,[r]\,(N'' + 2)$. When the gain of the DTC is correctly calibrated, DTC2 will have a delay of

$$dly(DTC2) = (1 - \alpha'') \cdot 4 \cdot T_{CKV}. \tag{7-18}$$

The delay of DTC1 is dynamically aligned with a feedback loop and when aligned, its value is:

$$dly(DTC1) = dly2 - (dly1 + DTC\ offset) + \frac{1}{4} \cdot T_{CKV} + 8 \cdot T_{CKV} + t_{p_{or}} + t_{p_{DFF}}. \tag{7-19}$$

Slack, assuming DTC2 gives a correct fractional-cycle delay, is given as:

$$Slack = dly2 - dly1 - \frac{1}{4} T_{CKV} + DTC\ offset - (t_{p_{or}} + t_{p_{DFF}}). \tag{7-20}$$

The timing constraint for *Slack* is:

$$0 < Slack < 4 \cdot T_{CKV}. \tag{7-21}$$

The delay of DTC1 can be expressed in terms of $Slack$ as well:

$$dly(DTC1) = Slack + 2 \cdot (t_{p_{or}} + t_{p_{DFF}}) + 8 \cdot T_{CKV}. \tag{7-22}$$

It can be seen that while the delay of DTC1 is quite large, actually the largest among all 3 DTCs, its dynamic range is very small. During operation, it only needs to dynamically align $ref2\,[r]\,(1)$ with $CKVD4S\_offset\_compensated\,[r]\,(1)$. In principle, it can be implemented with a constant delay line plus a narrow range, high resolution DTC. However, for ease of design, it is implemented with almost the same structure as the other two DTCs, except being given one more coarse bit. It should be emphasized that while DTC1 has the largest number of bits among the 3 DTCs, it is not the design bottleneck for overall DTC linearity.

7-6 Controls of DTC1 and DTC2

In this design, a total of 3 DTCs are used. Two are shown in Figure 7-8 and the 3^{rd} is shown in Figure 7-8. The parameters and functions of these 3 DTCs are summarized in Table 7-3, where $\alpha^{''}$ is the fractional part of $FCW/4$:

$$N^{''} + \alpha^{''} = FCW/4. \tag{7-23}$$

Table 7-3: Parameters and functionalities of DTCs used in this design

	DTC1	DTC2	DTC3
Number of bits	7b coarse + 8b fine	6b coarse + 7b fine	6b coarse + 8b fine
Functionality	Cancellation of initial offset	Creation of fractional cycle delay	Background calibration of DTC gain
Delay (excl. zero-code offset)	$Slack$ $+2 \cdot (t_{p_{or}} + t_{p_{DFF}})$ $+8 \cdot T_{CKV}$	$4 \cdot \left(1 - \alpha^{''}\right) \cdot T_{CKV}$	$4 \cdot T_{CKV}$

While Subsection 7-5-2 focuses on the timing constraints for proper sampling and leaves the controls of the DTCs untouched, this section will go into detail about the controls of DTC1 and DTC2. A figure including signal waveforms similar to those shown in Figure 7-6, after adding details and modifications of this design, is shown in Figure 7-14.

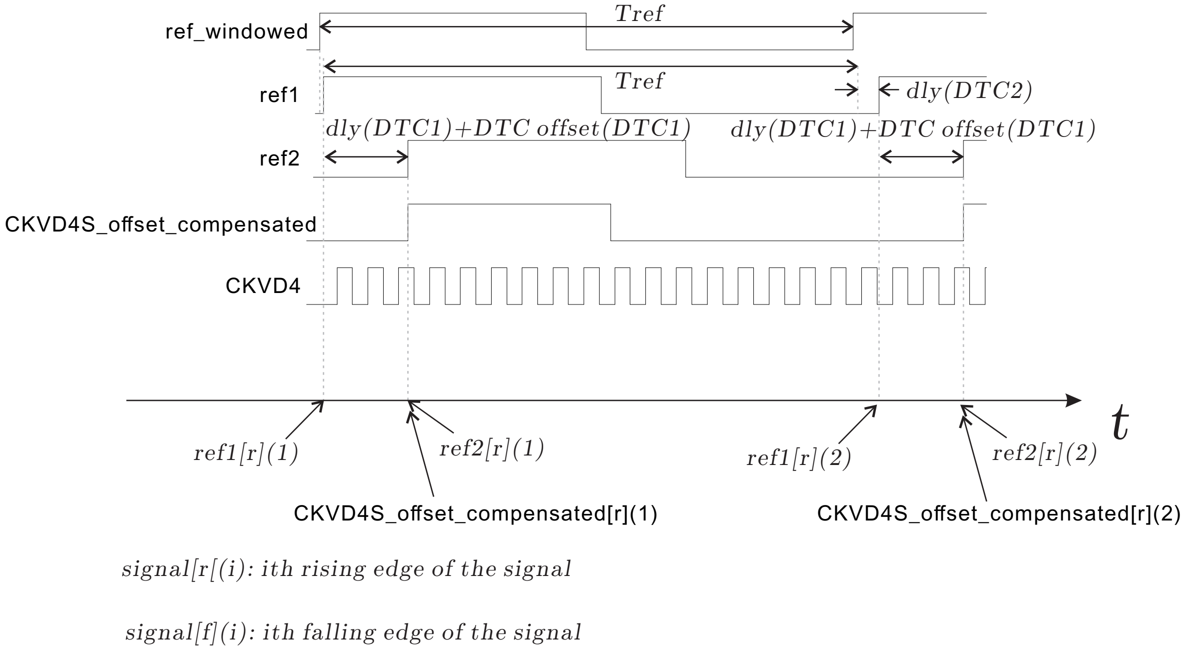

ref_windowed
Tref
ref1
Tref
dly(DTC2)
dly(DTC1)+DTC offset(DTC1)
dly(DTC1)+DTC offset(DTC1)
ref2
CKVD4S_offset_compensated
CKVD4
t
ref1[r](1)
ref2[r](1)
ref1[r](2)
ref2[r](2)
CKVD4S_offset_compensated[r](1)
CKVD4S_offset_compensated[r](2)
signal[r[(i): ith rising edge of the signal
signal[f](i): ith falling edge of the signal

Figure 7-14: Signal waveform illustration for DTC1 and DTC2 control

Since the purpose of DTC1 is to align $ref2\,[r]\,(1)$ with $CKVD4S\_offset\_compensated\,[r]\,(1)$ dynamically, it can be controlled in a successive approximation manner by detecting the 1^{st} outcome of the TDC. The 2^{nd} outcome of the TDC, on the other hand, is used for the feedback control of the frequency.

As for DTC2, its purpose is to insert a fractional delay at the 2^{nd} edge. Its control word can be set to 0 at the 1^{st} edge and set to the fractional control word at the 2^{nd} edge, where the fractional control word is $(1-\alpha'')\cdot g$. While α'' can be directly calculated from FCW, g, the relative DTC delay to T_{CKV}, needs extra calibration, especially when the operation frequency range is wide. The gain calibration is done in the edge-sampling circuitry since there is a well-defined T_{CKVD4} in it, against which the DTC gain can be calibrated.

The system block diagram is shown in Figure 7-15, showing the extra output of the calibrated gain from the edge-sampling block and the controls for DTC1 and DTC2.

The control words for DTC1 and DTC2 are updated at the rising edge of *dco_update*. The counter is cleared after each burst and before the next one by a signal *Count_clear*. Both of these two signals are generated by the outer-loop FSM, which is shown in Figure 7-15 as well.

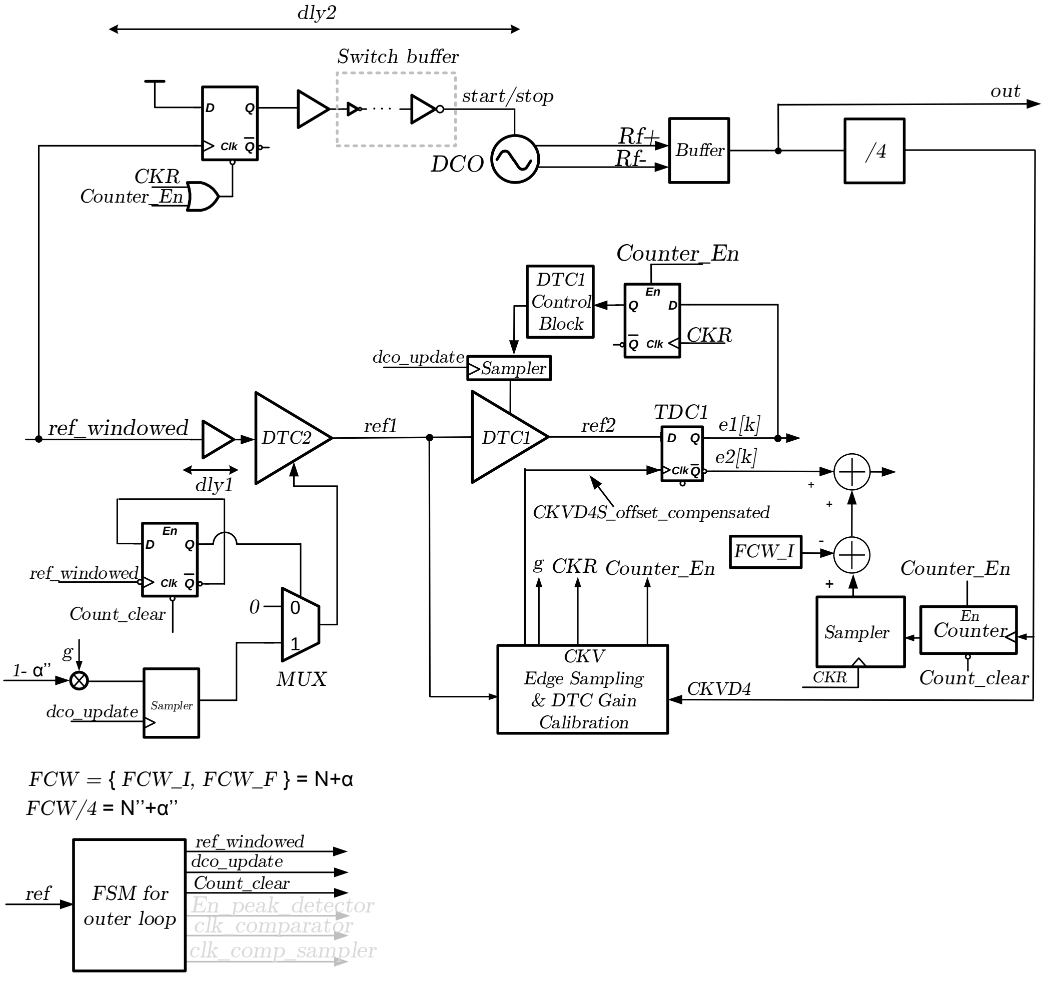


Figure 7-15: System block diagram including controls for DTC1 and DTC2

7-7 The DTC Gain Calibration Mechanism

7-7-1 Motivation

As explained in 7-6, an additional delay that equals $(1-\alpha'')\cdot 4\cdot T_{CKV}$ needs to be added by DTC2 at the 2^{nd} rising edge of the reference clock. This added delay represents the fractional part of FCW and directly affects the frequency of the PLL, thus it needs to be accurate.

Suppose the LSB of DTC2 is T_{DTC2}, then $(1-\alpha'')\cdot 4\cdot T_{CKV}/T_{DTC2}$ would be the control word to DTC2. While $(1-\alpha'')\cdot 4$ can be calculated from FCW with more than enough accuracy, neither T_{DTC2} nor its relative ratio to T_{CKV} is accurate enough. T_{DTC2} varies with PVT and T_{CKV}/T_{DTC2} scales with the frequency of CKV. Thus, a calibration is required to get an accurate estimation of T_{CKV}/T_{DTC2}.

7-7-2 Algorithm

One example of coarse-fine DTC gain calibration circuitry can be found in [34] and is drawn in Figure 7-16. The theory is explained in [4]. It calibrates both coarse and fine DTC gain by exploiting an important fact that the relative phase between ref and CKV is a periodic process. In a continuous fractional-N PLL, the phase difference between ref and CKV will accumulate until the accumulated fractional cycle reaches unity, at which point the phase between ref and CKV would be aligned again and also the extra delay created dy the DTC can be returned to zero. If the gain estimation of the DTC is exact, then the extra delay created by the DTC would equal exactly T_{CKV} before clearing and it is obvious that subtraction of exactly one T_{CKV} from the delayed ref signal would not change its relative phase to CKV. Since the phase between the delayed ref signal and that of CKV is aligned by the loop before clearing, it can be expected that their phase should also be aligned after the clearing, if the gain used for the DTC is accurate. Thus, the comparison result from the TDC after the clearing can be used to correct the gain with the help of a correlator.

However, in a DC-ADPLL, specifically one that is turned on for only one ref cycle, starting from a fixed phase between ref and CKV, the above mentioned property is gone. Another

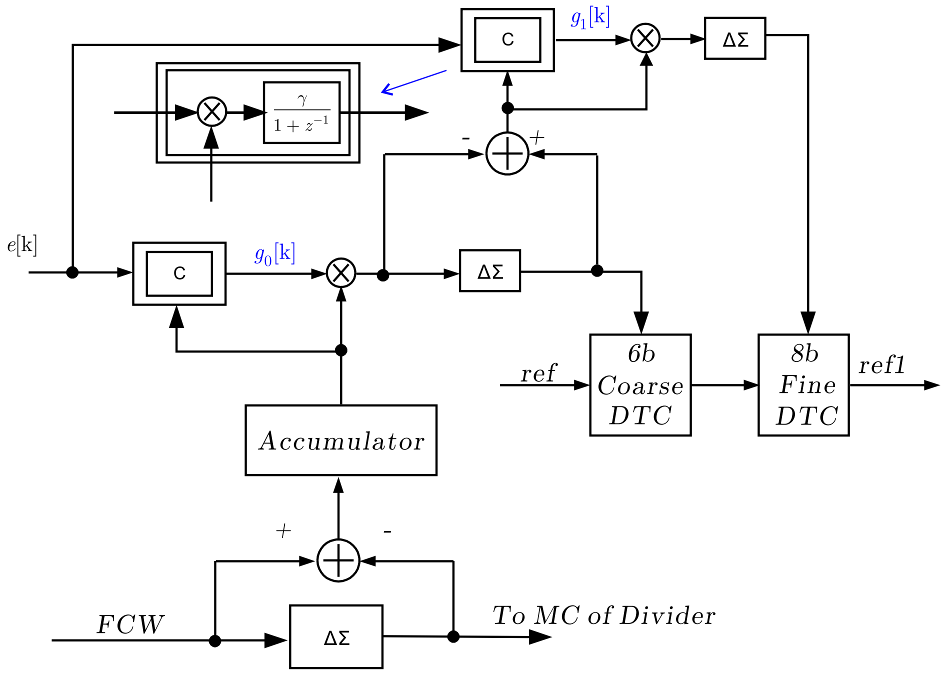


Figure 7-16: DTC gain calibration mechanism in [34]

method needs to be found to calibrate the delay of the DTC against T_{CKV}. One straightforward approach is mentioned in [4], where the DTC is composed of a delay line where each delay stage in this delay line is regulated in an analog way and the total delay of this delay line is calibrated against T_{CKV}, easily created with the help of a DFF. This mechanism can work in principle but has two major disadvantages. Firstly, the delay line is regulated in an analog way, which is not desirable. Secondly, it requires to access 2^r phases simultaneously for an r-bit DTC, which would require a large multiplexor. The mismatch between each channel can be another problem.

The problem can be solved by looking at the delay line in another way. While the delay of a delay line can be regulated by changing the delay of each unit simultaneously while keeping the number of delay stages unchanged, the same function can be achieved by keeping the delay of each unit unchanged while changing the number of enabled delay stages. One example of such kind of DTC can be found in [34]. Combining the DTC regulation method presented in [4] while replacing the DTC structure with the one presented in [34], the gain of the DTC can be calibrated against T_{CKV} in a digital manner. Furthermore, an auxiliary DTC, DTC3, which is designed to be matched with DTC2, is used to calibrate the DTC gain against T_{CKV} and the calibrated gain is used by DTC2 to correctly set the fractional-cycle delay. The DTC calibration mechanism used in this design is shown in Figure 7-17. It should be mentioned that this mechanism relies on layout matching between DTC2 and DTC3. Detailed block diagram for the correlators in Figure 7-16 and Figure 7-17 are drawn in Figure 7-18.

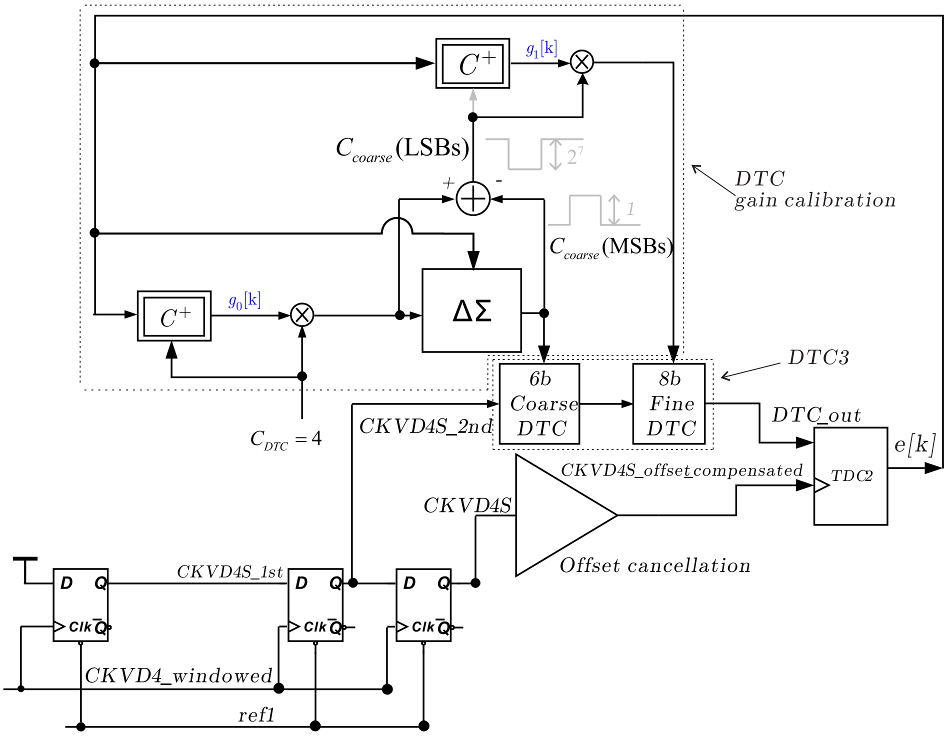


Figure 7-17: DTC3 for gain calibration

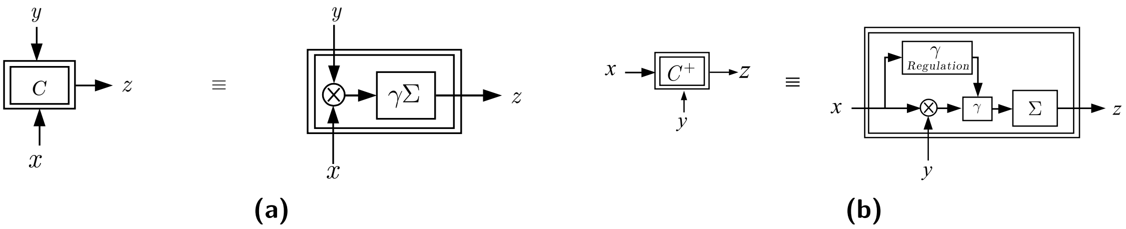


Figure 7-18: Block diagram for correlator and modified correlator: (a) Correlator (b) Modified correlator

The gain calibration circuitry shown in Figure 7-17 resembles the one shown in Figure 7-16 in most part but with some necessary modifications. Firstly, the DTC calibration and fractional-cycle setting were done within the same DTC in [34]. The delay of the DTC can be represented as Eq. (7-24). As long as $delay[k]$ is calibrated to match the desired delay, the gain calibration loop won't make further calibration on $g_0[k]$ and $g_1[k]$, simply because there is no need to. It means that there are multiple solutions for $g_0[k]$ and $g_1[k]$ to create a correct $delay[k]$. However, for the gain calibration mechanism used in this design, not only the delay of DTC3 but also the value of both $g_0[k]$ and $g_1[k]$ need to be calibrated. Only in this way, DTC2 can create a correct delay with a coarse-fine DTC structure. Actually, $g_0[k]$ represents the ratio between the LSB of the coarse DTC and T_{CKV} while $g_1[k]$ represents the ratio between the LSB of the fine DTC and that of the coarse DTC. By forcing an additional constraint to calibrate $g_1[k]$, $g_0[k]$ will be forced to represent the correct gain of the coarse DTC. And $g_1[k]$, the gain relating the coarse DTC and the fine DTC is calibrated by occasionally subtracting/adding 2^{r-1}, where $r = 8$, from the uncalibrated control word of the fine DTC while increasing/decreasing the control word of the coarse DTC by 1. In this way, $g_1[k]$ is forced to follow a relation shown in Eq. (7-25). Due to this calibration mechanism of $g_1[k]$, DTC3 needs to have 1 more bit in terms of fine stage than that of DTC2, which is shown in Figure 7-17 as well.

$$delay[k] = (C_{\text{DTC}} * g_0[k])\{MSBs\} * \Delta t_{coarse} + [C_{\text{DTC}} * g_0[k]]\{LSBs\} * g_1[k] * \Delta t_{fine}. \tag{7-24}$$

$$g_1[k] * 2^{r-1} * \Delta t_{fine} = \Delta t_{coarse}. \tag{7-25}$$

Secondly, since gain calibration and fractional-cycle setting in this design are done at two separate DTCs, it is possible that the fine control word for the gain-calibration DTC, DTC3, is zero while it is not for the fractional-cycle setting DTC, DTC2. Thus, it is important to make sure that $g_1[k]$ is calibrated even when $C_{coare}(LSBs)$ for DTC3 is zero. In this design, the calibration of $g_1[k]$ is triggered by detecting consecutive +1/-1/+1 transitions from TDC2, which indicates that $delay[k]$ is calibrated, and by subtracting/adding 2^{r-1} from $C_{coare}(LSBs)$ while increasing/decreaing $C_{coare}(MSBs)$ by 1, the next comparison result from TDC2 will be most correlated to the gain estimation error of $g_1[k]$. This error will be corrected over multiple cycles until $g_1[k]$ follows the relation shown in Eq. (7-25). The state transfer for the coarse and fine gain calibration is plotted in Figure 7-20.

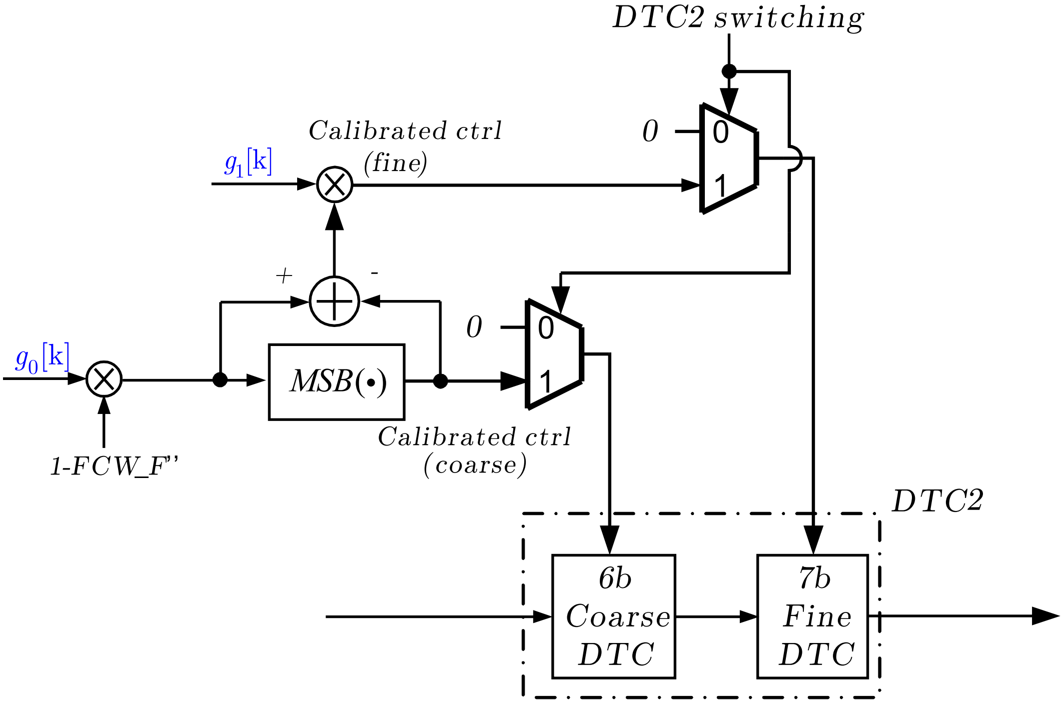


Figure 7-19: DTC2 for setting fractional cycle

Another change to the original gain calibration circuitry shown in Figure 7-16 is at the correlator. The correlator in Figure 7-16 effectively implements an LMS algorithm with a fixed step size. The choice of step size is a trade-off between the settling time and the accuracy after settling [72]. While the settling time is not a major concern for a continuous PLL, it is of critical importance in a DCPLL. Since the DCPLL in this design is enabled for only 1 cycle among the 25 cycles within a pulse repetition period, it means that the DCPLL settles slower than a continuous PLL in terms of absolute time even if the number of cycles needed to settle between these two are about the same. While the settling time of the LMS filter is still acceptable for a continuous PLL, it is no longer acceptable after being amplified by 25 times. The direct consequence, if no measures are taken, would be an excessive overhead before data can be transmitted. To alleviate this problem, a variable-step LMS, explained in [73], instead of a fixed-step LMS algorithm, is used, which considerably reduces the number of cycles needed to settle. On circuit implementation level, as is shown in Figure 7-18b, the correlator step size γ is regulated by detecting previous feedback errors. After γ settles to the minimum step, both $g_0[k]$ and $g_1[k]$ are low-passed before feeding to DTC2. The settling curves for both coarse and fine DTC gain is shown in Figure 7-21.

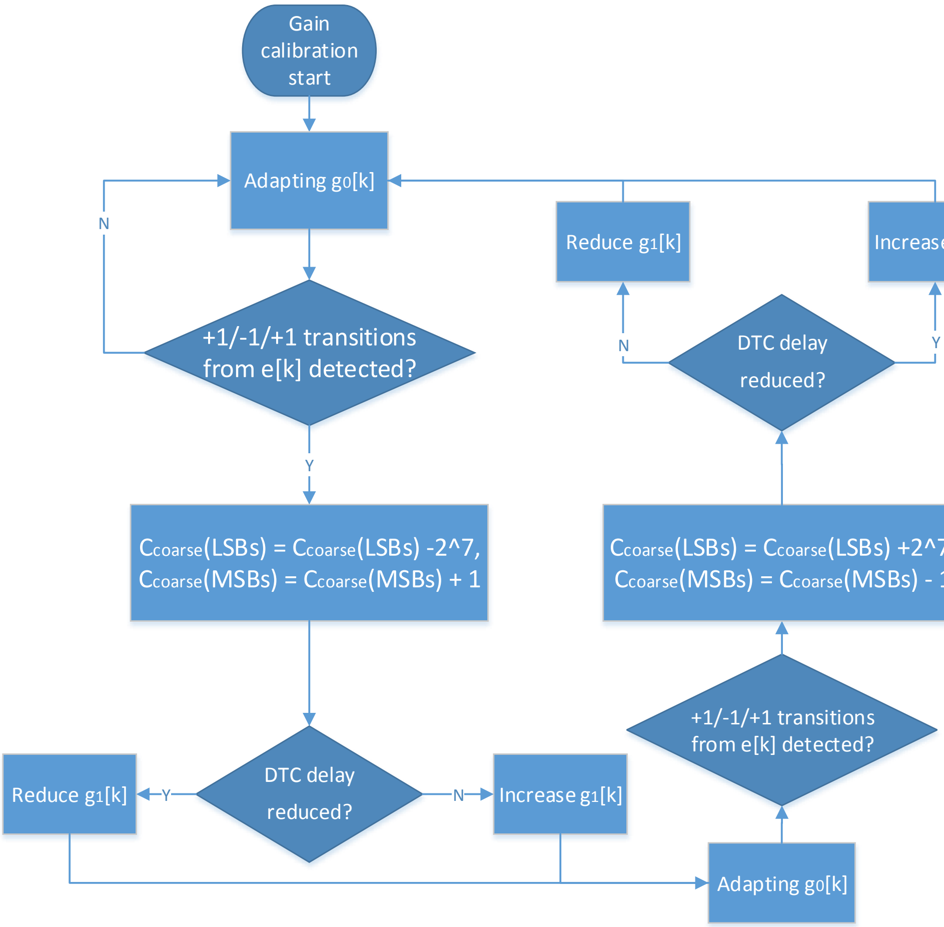

Gain calibration start
Adapting g0[k]
N
+1/-1/+1 transitions from e[k] detected?
Y
Ccoarse(LSBs) = Ccoarse(LSBs) -2^7, Ccoarse(MSBs) = Ccoarse(MSBs) + 1
DTC delay reduced?
Y
Reduce g1[k]
N
Increase g1[k]
Adapting g0[k]
+1/-1/+1 transitions from e[k] detected?
Ccoarse(LSBs) = Ccoarse(LSBs) +2^7, Ccoarse(MSBs) = Ccoarse(MSBs) - 1
DTC delay reduced?
N
Reduce g1[k]
Y
Increase g1[k]

Figure 7-20: State transfer diagram for interactive coarse and fine gain calibration

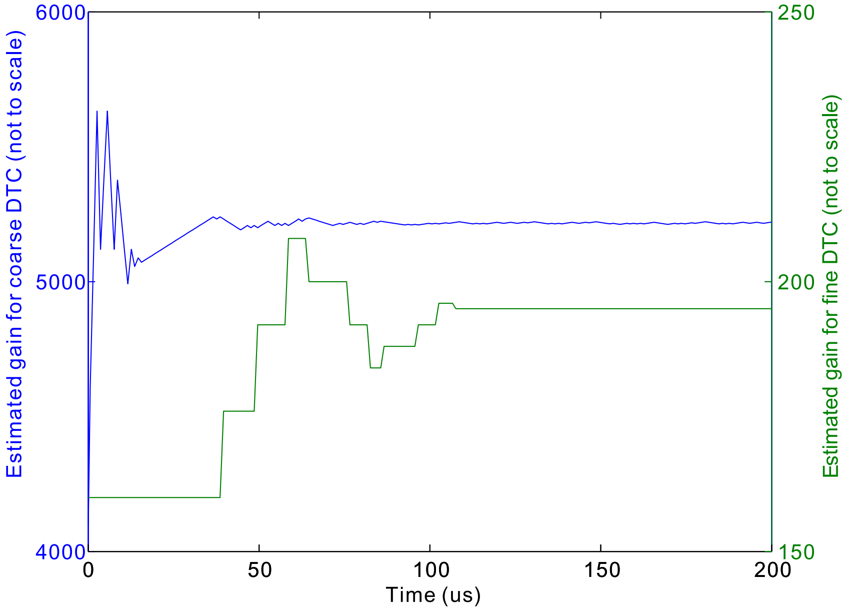


Figure 7-21: DTC gain settling curve for both coarse and fine gain

7-8 Optimization for Hardware Cost

While it may seem that multiple many-bit multipliers are needed from what is drawn in Figure 7-17 and Figure 7-19, actually the most hardware-hungry multipliers needed are the two 8-bit multipliers used for the multiplication of $g_1[k]$ in Figure 7-17 and Figure 7-19. The multiplication function in both of the modified correlators shown in Figure 7-17 can be eliminated. The 16-bit multiplication needed by $g_1[k]$ in Figure 7-19 is implemented with a sequential multiplier, by exploiting the fact that post digital processing can make full use of the vacancy between two successive bursts.

7-8-1 Modified Correlator

None of the modified correlators used in this design need actual multipliers at hardware level. Since one of the inputs to the two modified correlators is $e[k]$, which is only 1-bit, multiplication will be reduced to sign detection. Furthermore, since the step sizes for the modified correlators are automatically regulated, they do not depend on C_{DTC} and $C_{coarse}(LSBs)$.

For the multiplication between $g_0[k]$ and the constant number 4 in Figure 7-17, it is inherently done during the coding stage by re-ordering the index of the signal. The only multiplier needed in Figure 7-17 is an 8-bit multiplier to multiply $g_1[k]$ with $C_{coarse}(LSBs)$.

7-8-2 Booth-coded Radix-4 Multiplier

Both of the multipliers in Figure 7-19 cannot be simplified out. However, the hardware cost can still be reduced considerably by implementing the multiplier in a sequential way instead of a combinatorial way. Thanks to the duty-cycled operation of the DC-ADPLL, spreading post-digital processing over multiple cycles is possible. There are 25 ref cycles between two successive bursts, and taking into account the cycles needed for burst operation, DCO update, bandwidth regulation, etc., there is a vacancy of about 10 cycles left to implement a sequential multiplier.

A conventional m-bit sequential multiplier costs only an m-bit adder in hardware but needs m cycles to generate a final result. The multiplication between $g_0[k]$ and $\left(1 - FCW\_F''\right)$ in Figure 7-19 is a multiplication between two 14-bit numbers. If implemented with a conventional sequential multiplier, it requires 14 cycles to generate the result, which will exceed the temporal budget. Thus, some mid-ground multiplier between a fully sequential and a fully combinatorial multiplier is needed here. One of the solutions is to use a high-radix multiplier [74, Ch. 4]. The commonly used high-radix multipliers are radix-4 and radix-8. While conventional sequential multipliers shift partial results by 1 bit every time and add them to previous results with an m-bit adder, radix-4 multipliers shift 2 bits every time and partial results are added after being coded. Although the hardware cost will be a little more than that of a fully sequential one, the direct advantage is that it can reduce the number of calculation cycles by half. In the author's case, the multiplication of two 14-bit numbers with a radix-4 multiplier requires only 7 cycles, which fits well within the temporal budget. Finally, an open-source 16-bit Booth-coded radix-4 multiplier [75] is used, leaving some margin to allow for a higher accuracy for fractional control. A temporal budget of 8 calculation cycles is left for this multiplication, which will be seen later in the FSM for the outer loop. The 8-bit multiplier required by $g_1[k]$ in Figure 7-19 is left untouched and assumed to be implemented with a fully combinatorial multiplier. On one hand, the hardware cost of an 8-bit combinatorial multiplier is acceptable and the prorogation delay can safely fit into two cycles of ref. On the other hand, there is no more temporal budget of 4 cycles to implement it with a radix-4 multiplier.

7-9 FSM for the Outer Loop

Similar to the DCPLL control in [11], an outer-loop FSM is used in this design to set the positions of control signals within a pulse repetition period. The complete list of signals from this FSM is shown in Figure 7-22. Positions of the control signals are indexed with their relative positions within a pulse repetition period that lasts for 25 ref cycles.

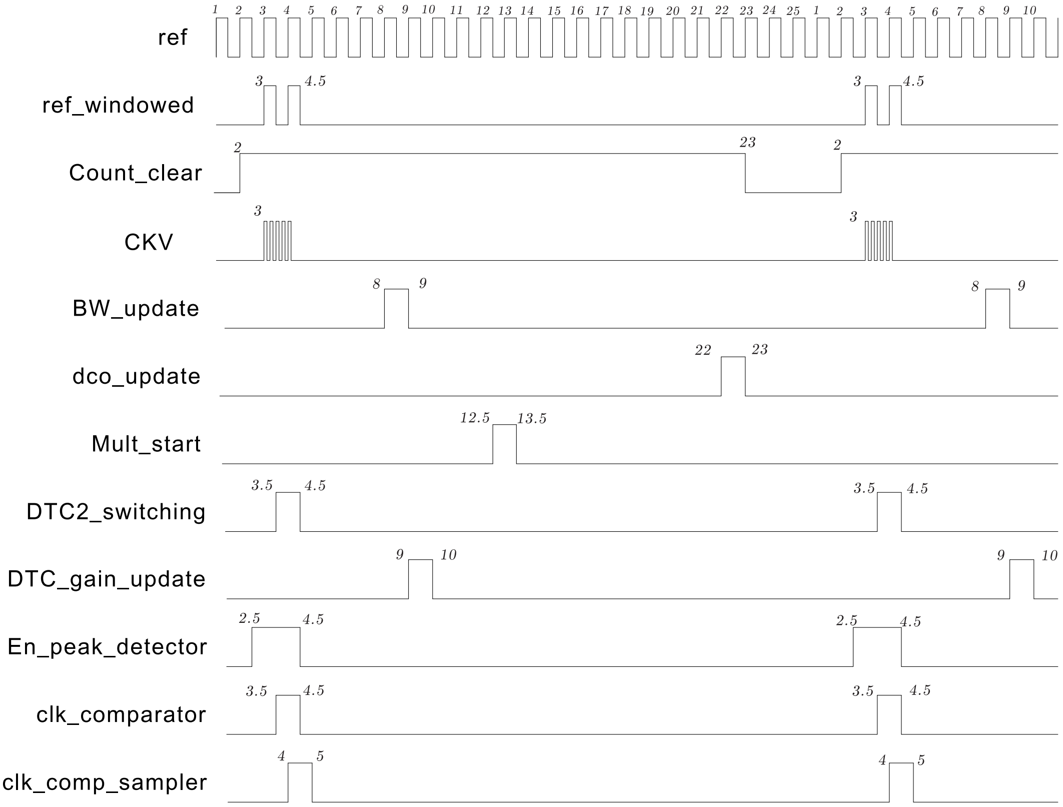


Figure 7-22: Control signals from the outer-loop FSM

Explanations for the control signals are given below:

ref ref signal is the reference clock from an off-chip crystal oscillator, expected to run at 25 MHz. It is used to drive the outer-loop FSM as well as most of the other digital blocks.

ref_windowed Since the DC-ADPLL is turned on for only about 1 ref period within a pulse repetition cycle, the continuously running ref train is windowed to include only 2 rising edges, which is just enough to define the period of ref.

Count_clear The counter for CKV is cleared after one burst and before another. This is done by the signal $Count\_clear$ which resets the counter after the update of the DCO is completed. $Count\_clear$ is also used to reset some other blocks such as the divider and the edge-sampling circuitry.

CKV CKV is a representation for the RF output signal, which is not generated by this FSM itself. It is drawn here to illustrate the burst position in terms of the position of other signals.

BW_update As is mentioned, both the update of the capacitor bank and the DTC gain are automatically regulated in bandwidth. The bandwidth update is done at the rising edge of BW_update, which precedes both the update of capacitor bank and DTC gain.

Mult_start $Mult\_start$ is used for triggering the 16-bit Booth-coded radix-4 multiplier in this design. The multiplication takes 8 cycles and is completed before the rising edge of dco_update.

DTC2_switching The code of DTC2 needs to be different at the 1^{st} and 2^{nd} rising edge of $ref\_windowed$. $DTC2\_switching$ is used to switch the control word of DTC2 at the 1^{st} falling edge of $ref\_windowed$.

DTC_gain_update The gain of the DTCs are updated after each burst, following the bandwidth update of the correlator, which is triggered by BW_update.

En_peak_detector *En_peak_detector* specifies the window in which the peak detector needs to be active. The peak detector is turned on before the burst begins and turned off after the burst ends, so that the switching does not cause interference to the RF signal. It should be noted that while *En_peak_detector* specifies the position of the window, it is windowed by another control signal from the inner-loop FSM which specifies whether the PLL is in amplitude calibration phase or not. It is only during the amplitude calibration phase and within the window specified by *En_peak_detector* that the peak detector is enabled. The situation is similar for the following $clk\_comparator$ and $clk\_comp\_sampler$.

clk_comparator $clk\_comparator$ is used to trigger the clocked comparator to compare the results from the peak detector. The rising edge of $clk\_comparator$ comes 20 ns later than the beginning of the burst, which puts a design constraint on the settling time of the peak detector.

clk_comp_sampler The rail-to-rail results from the comparator is sampled at the rising edge of *clk_comp_sampler*, which arrives 20 ns after the rising edge of *clk_comparator*, requiring also the comparator to settle within 20 ns.

Finally, as a side note, the outer-loop FSM has an extra feature to change the position of the burst from 3 to 6. The position of the burst is programmed at power on and positions of other control signals are automatically adjusted to ensure proper functionality.

7-10 FSM for the Inner Loop

While the outer-loop FSM controls the positions of control signals, the inner-loop FSM controls the locking process of the DC-ADPLL. The summarized locking flow chart is shown in Figure 7-23.

After power on, an estimation of the required number of active cores is made, according to whether FCW specifies the frequency to be at HB or LB. After that, the 1^{st} coarse frequency calibration begins. After the 1^{st} coarse frequency calibration, there is a better estimation of the frequency than that at power on. Thus, an amplitude calibration phase follows the 1^{st} coarse frequency calibration and adjusts the amplitude to be between 395–400 mV. After the amplitude calibration, the frequency might be off a bit due to the change in parasitic capacitance when turning on/off auxiliary active cores. Thus, a 2^{nd} coarse frequency calibration follows the amplitude calibration to lock the frequency within one LSB of the coarse bank. After that, medium and fine frequency calibration is performed until the DC-ADPLL locks. Finally, the DC-ADPLL switches to 2^{nd}-order during locking.

The flow chart for the amplitude calibration phase is shown in Figure 7-24. The calibration is implemented as part of the inner-loop FSM. The inner-loop FSM receives two amplitude comparison results from the peak detector and comparator, which detects and compares the

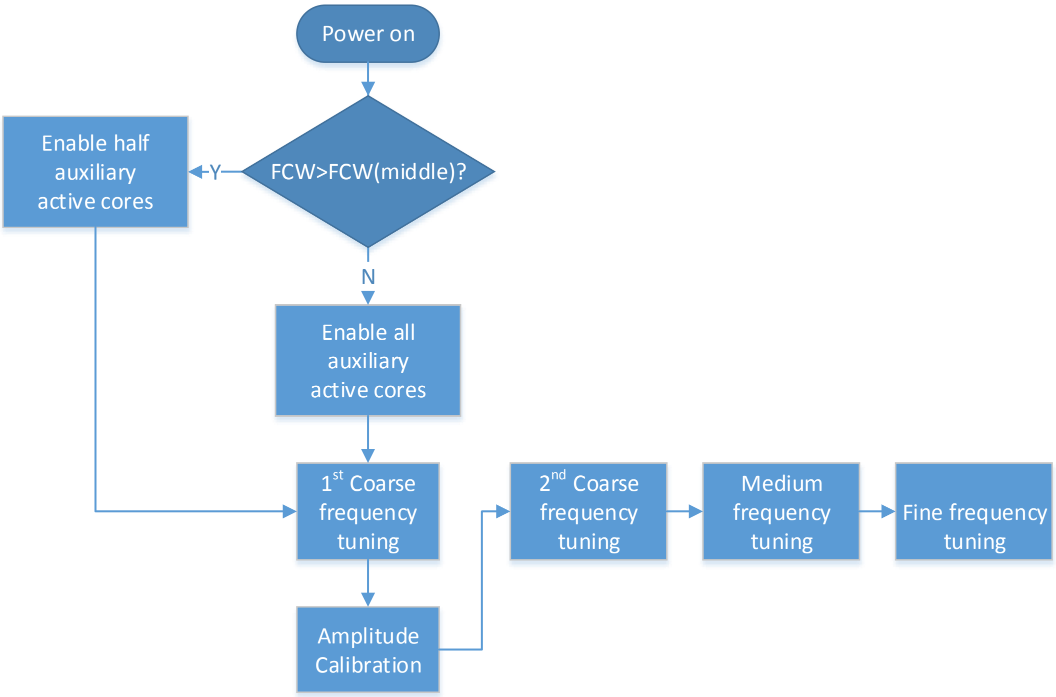

Power on
FCW>FCW(middle)?
Y
Enable half auxiliary active cores
N
Enable all auxiliary active cores
1st Coarse frequency tuning
Amplitude Calibration
2nd Coarse frequency tuning
Medium frequency tuning
Fine frequency tuning

Figure 7-23: Flow chart for the locking of the DC-ADPLL

amplitude of the RF output belonging to the last cycle with two programmable thresholds. The higher threshold is set at 400 mV while the lower one is set at 395 mV. The inner-loop FSM will increase/decrease the number of auxiliary active cores until the amplitude is calibrated within 395–400 mV. After that, the amplitude calibration phase completes and the peak detector and the comparator are turned off.

The locking process at 7.0 GHz is shown in Figure 7-25. The pulse repetition period is set to 1.0 µs, which is the update period for digital controls signals as well. The first row shows the number of enabled impedance-tuning cores, the next three rows show the control words for coarse, medium and fine bank. The number of active cores is set to maximum, 15, at the falling edge of the global reset signal and is changed to a closer estimation of 8 at the onset of a synchronous reset signal, judging by the fact that 7.0 GHz is at higher band, where probably only few active cores need to be turned on.

Three cycles after turning on, the loop detects an overflow of the coarse bank on the lower end that even when all coarse units are turned off, the oscillation frequency is still not high

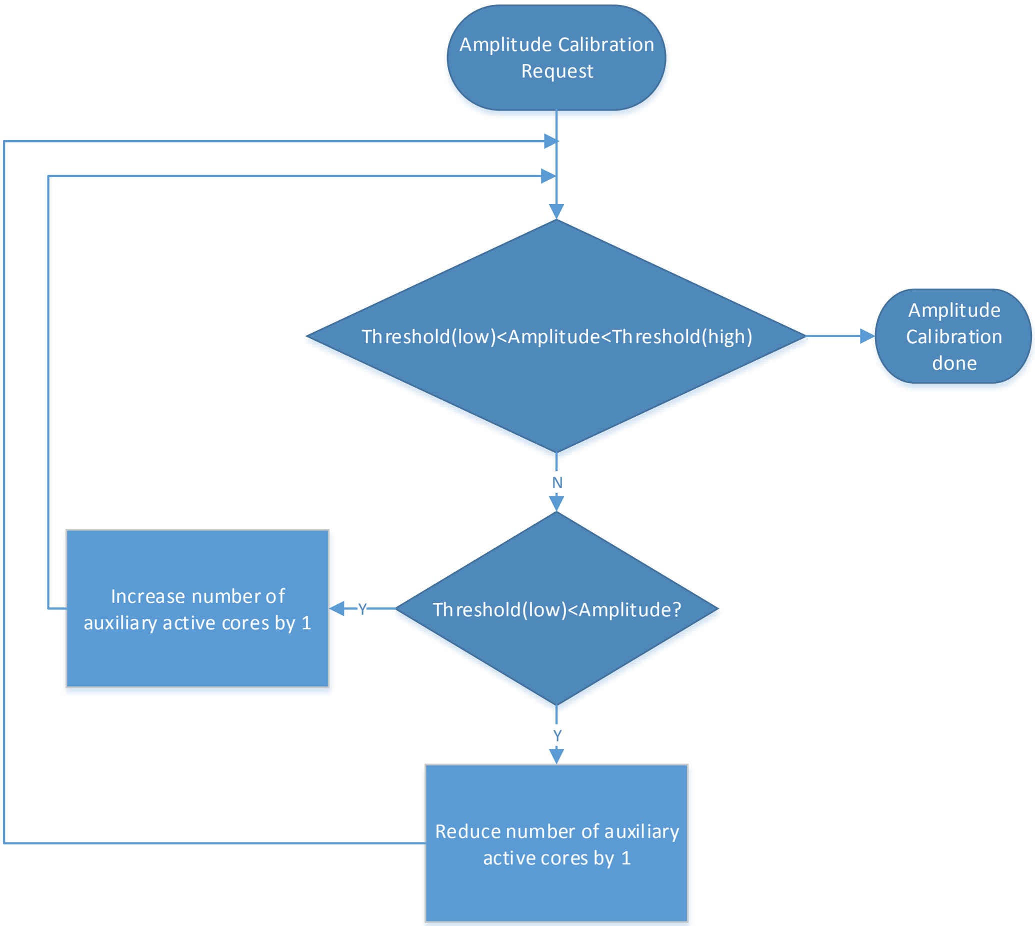

Amplitude Calibration Request
Threshold(low)<Amplitude<Threshold(high)
Amplitude Calibration done
N
Threshold(low)<Amplitude?
Y
Increase number of auxiliary active cores by 1
Y
Reduce number of auxiliary active cores by 1

Figure 7-24: Flow chart for the amplitude calibration

enough. This overflow brings the loop into the amplitude calibration phase, where the number of active cores is reduced until the amplitude stays within 395–400 mV, which completes at t = 11 µs. The number of enabled active cores stays at zero after the amplitude calibration phase, which meets with the expectation since the real part of the LC tank impedance, R_p, is largest at the highest frequency.

After the completion of amplitude calibration, the 2^{nd} coarse frequency tuning begins. Since some auxiliary active cores were turned off during the amplitude calibration phase, parasitic capacitance loading the tank is also smaller. The coarse bank control word settles to 6 at t = 20 µs.

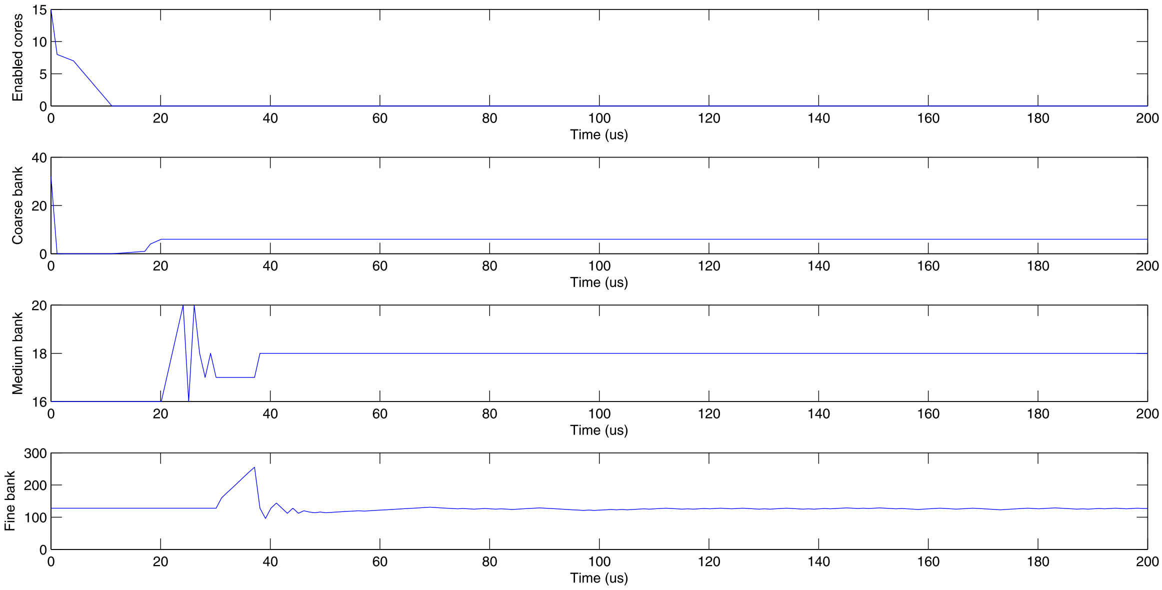


Figure 7-25: Settling process of the DC-ADPLL

Medium frequency tuning comes after the 2^{nd} coarse frequency tuning and settles to 17 at t = 30 µs. Fine tuning follows that. Probably due to the settling of the DTC gain estimation at the same time of the fine frequency acquisition, there is an overflow of the fine bank at t = 37 µs. The overflow bit is carried to the medium bank and then the code of the fine bank is returned to half of its entire scope. Since the fine bank is designed to cover about 2 LSBs of the medium bank, turning on 1 bit of the medium bank while turning off half of the fine bank basically does not change the capacitance in total, although not quite accurately. After the overflow, the fine bank continues acquiring the frequency and reduces its step size during this process. The gain estimation of the DTC becomes better during this time as well, shown in Figure 7-21. The slower settling one between these two dominates the settling time of the entire loop, which is here the gain settling of the fine DTC. Both the frequency control word and the DTC gain settle after 120 cycles.

7-11 Complete System View

The complete system view is shown in Figure 7-26. DTC and other analog blocks such as the peak detector and the comparator are modeled at this moment with behavioral blocks to verify the functionality of the loop. Design of the DTC and other analog blocks is covered in Chapter 8 and Chapter 9. After that, behavioral models are replaced with transistor-level circuits and the simulation results are given in Chapter 10.

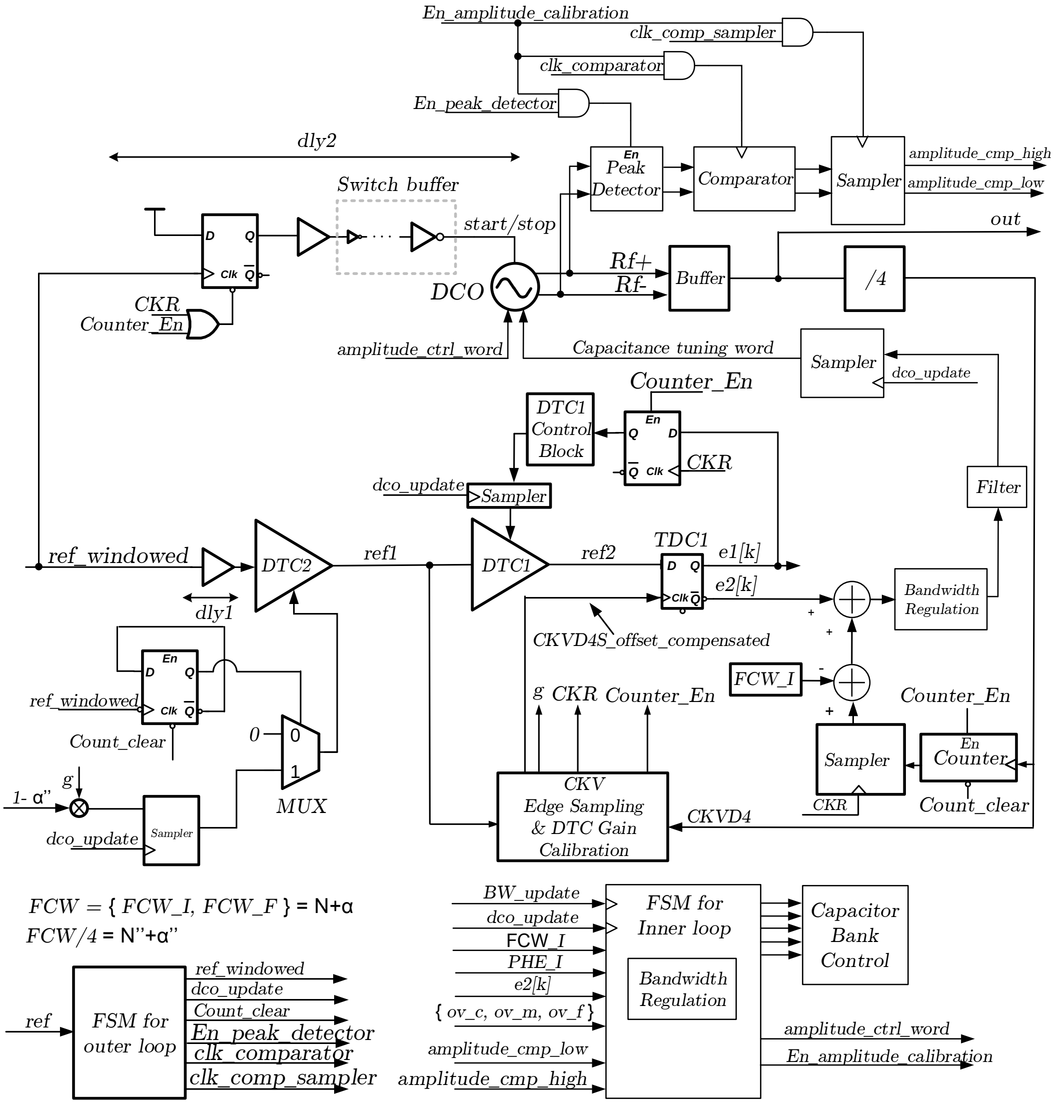


Figure 7-26: System block diagram including all necessary blocks

Chapter 8

Design of the DTC

As is explained in [4], to ensure that the detected timing difference by the Time-to-Digital Converter (TDC) is dominated by thermal noise and get rid of excessive fractional spur, the Digital-to-Time Converter (DTC) resolution needs to be on the order of DCO thermal noise. The DTC structure is based on the structure presented in [34], which is a coarse DTC cascaded with a fine stage. As is mentioned in [69], one of the advantages of using a multi-bit DTC instead of a multi-bit TDC is that a high resolution DTC can be implemented by cascading a fine stage after a coarse one, while for TDC, a fine stage is required for every unit. Several modifications are made to the original structure and the reasons for these modifications will be covered in the following parts. An overview of the coarse-fine DTC is shown in Figure 8-1.

8-1 Coarse DTC

The coarse DTC is effectively implemented by cascading delay units and each delay unit can be enabled or disabled to change the signal path from the input to the output. Structure for the coarse DTC is shown in Figure 8-2.

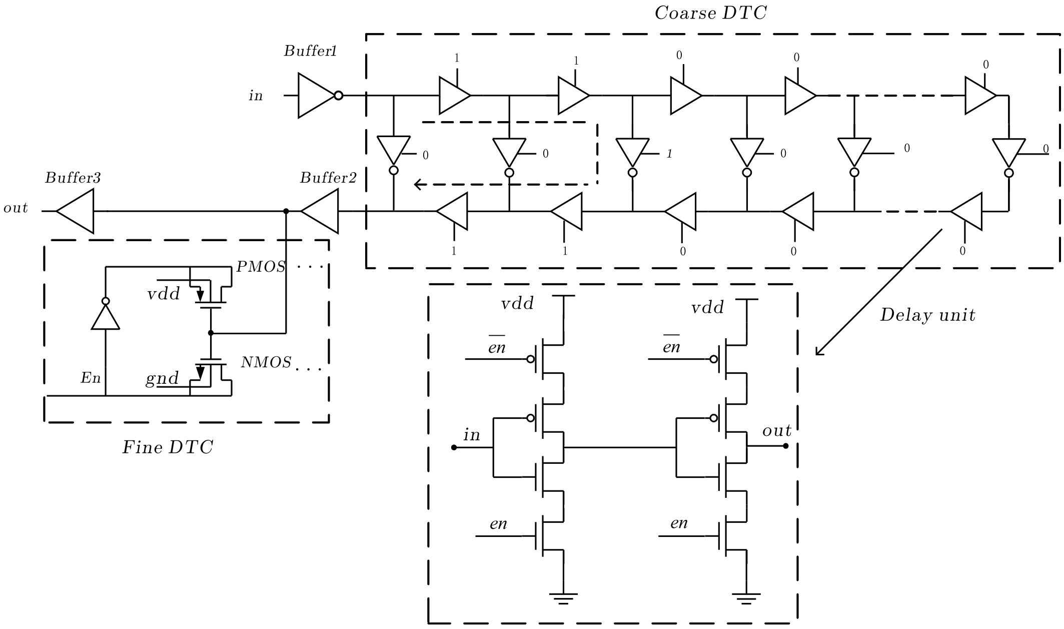


Figure 8-1: The coarse-fine DTC architecture used in this design

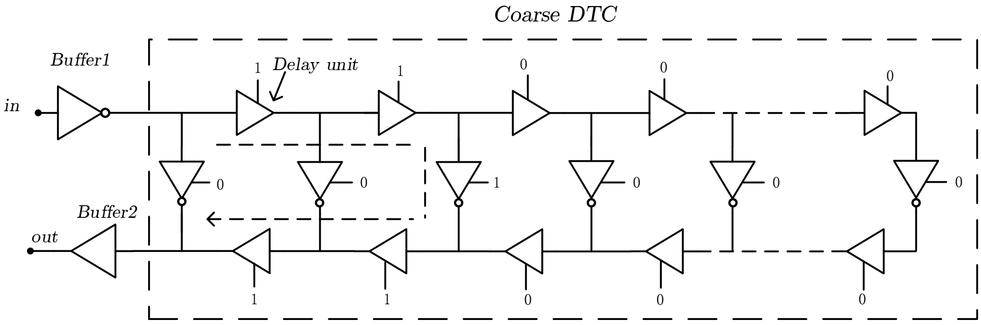


Figure 8-2: Architecture of the coarse DTC [34]

The buffers at the input of the coarse DTC is used to increase the driving force of the input signal of the DTC, which is probably the output from a DFF. Input capacitance of the delay unit in the DTC could significatively alter the slope of the signal waveform of the DFF output if connected directly. The buffer at the output of the coarse DTC is used to match the capacitive loading of the output stage and also to isolate the coarse DTC with the fine DTC.

8-1-1 Design Optimization for Even-odd Code Matching

The delay unit is shown in Figure 8-3. Each delay unit is composed of two tri-state inverters instead of just one. With only one inverter, serious mismatch would occur between even and odd codes, which is illustrated from Figure 8-4 to Figure 8-6.

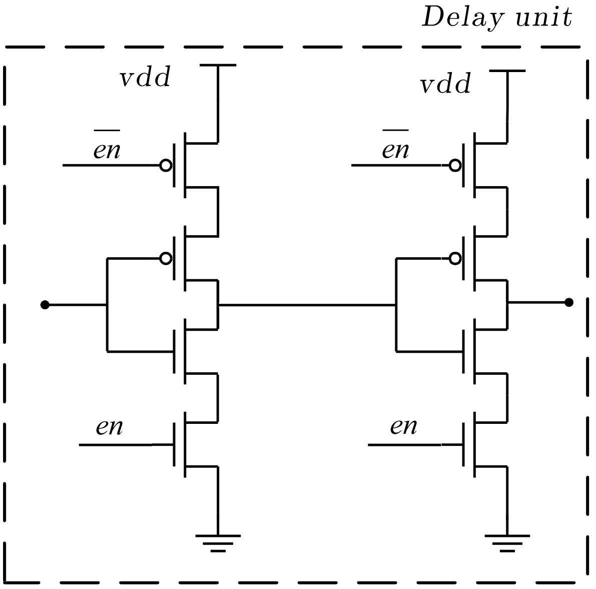


Figure 8-3: Unit delay composed of two tri-inverters

Suppose the analysis starts from an arbitrary code n. The signal path at this code is shown by the solid lines in Figure 8-4. Other irrelevant components are greyed-out. The enabled transistors in Figure 8-4 are the two PMOS at the top, the two NMOS at the center and the two PMOS at the bottom.

When the code increases from n to $(n + 1)$, the signal path from the input to the output is extended by re-arranging the enabled and disabled delay stages, which is shown in Figure 8-5.

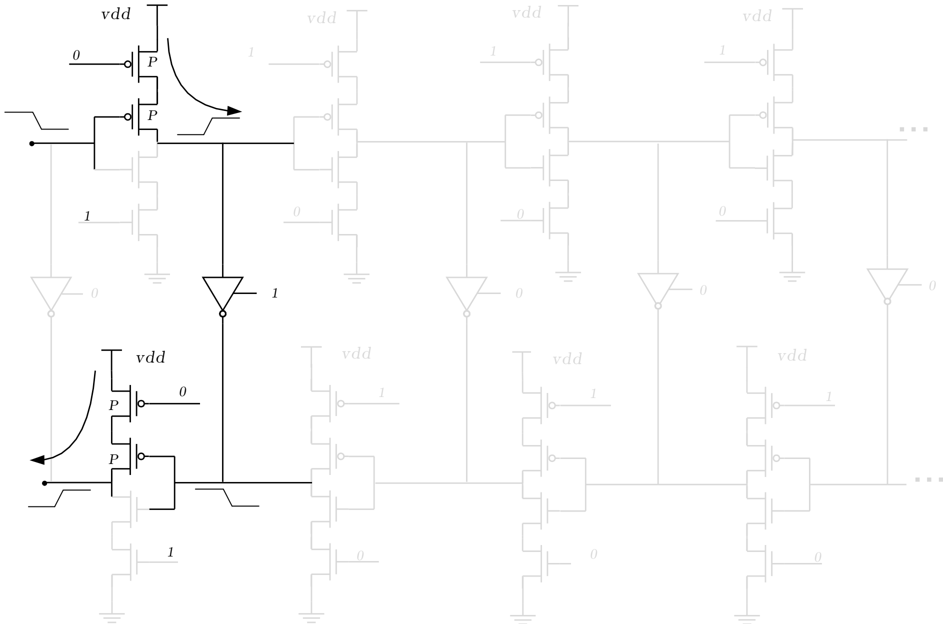


Figure 8-4: Signal flow of a DTC composed of single-inverter delay stages at code n

The added delay, compared to the delay at code n is composed of the discharging time constant specified by the two NMOS at the top, the charging time constant specified by the two PMOS at the center and that of the two NMOS at the bottom.

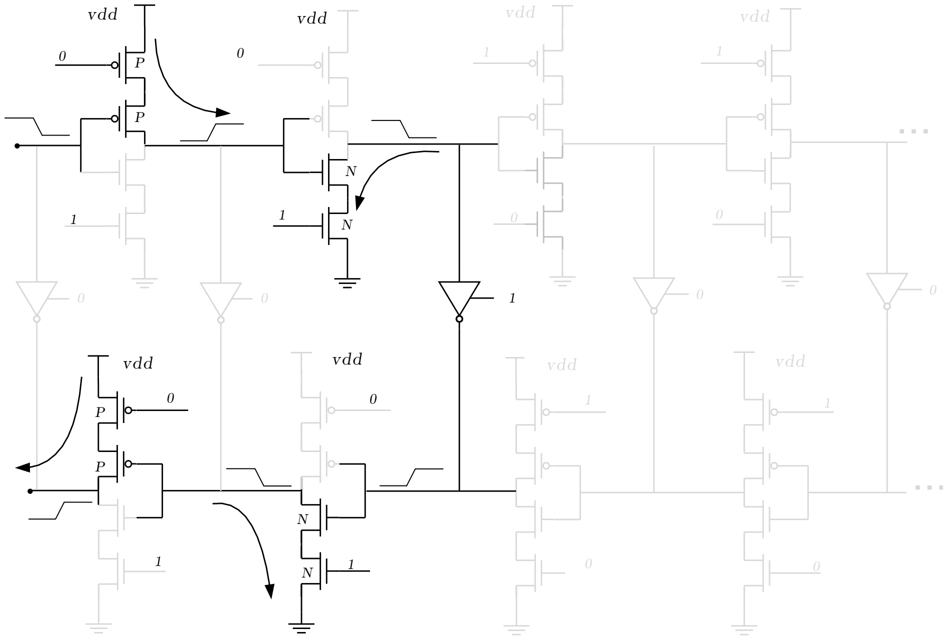


Figure 8-5: Signal flow of a DTC composed of single-inverter delay stages at code $(n + 1)$

When the code increases to $(n + 2)$, the signal path is extended even further, shown in Figure 8-6. This time, the added delay, compared to that at code $(n + 1)$, is the two PMOS at the top, the two NMOS at the center and the two PMOS at the bottom. It is clear now that the increase in delay between code (n) and code $(n + 1)$ is specified by the NMOS at the top and bottom and the PMOS at the center, while the increase in delay between $(n + 1)$ and $(n + 2)$ is specified by the PMOS at the top and bottom and the NMOS at the center. While the driving force of PMOS and NMOS can be designed relatively close by simply sizing the PMOS twice as wide as the NMOS, the residual mismatch is still large enough affect the performance, especially under different process corners.

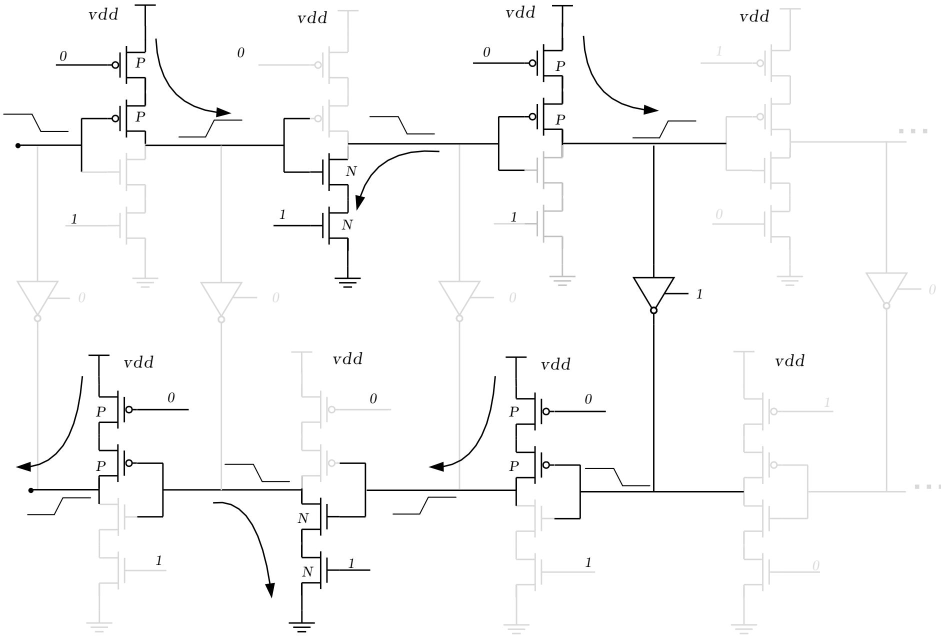


Figure 8-6: Signal flow of a DTC composed of single-inverter delay stages at code $(n + 2)$

The step size versus the code for a coarse DTC composed of single-inverter delay stages is shown in Figure 8-7. The step mismatch between even and odd codes is quite obvious.

To alleviate the even-odd mismatch issue, a delay unit composed of two tri-inverters is employed. By using two tri-inverters instead of just one, the increase in delay is specified by the same kind of MOS transistors and thus the even odd mismatch problem is alleviated. A similar analysis showing the signal paths at code n, $(n + 1)$ and $(n + 2)$ are plotted from Figure 8-8 to Figure 8-10. The coarse DTC step size vs. code after using two inverters is plotted in Figure 8-11.

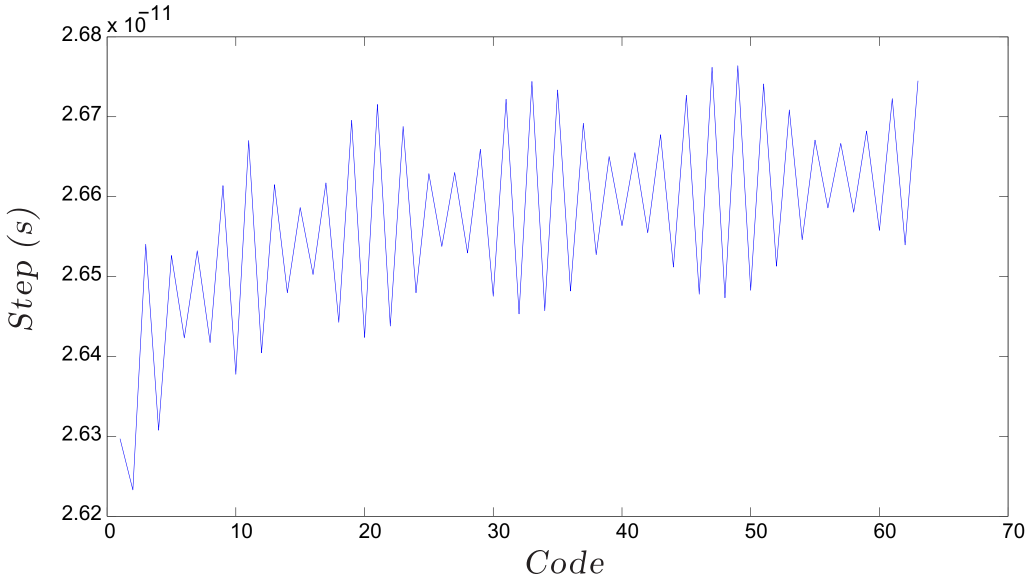


Figure 8-7: Step size vs. code for a DTC composed of single-inverter delay stages

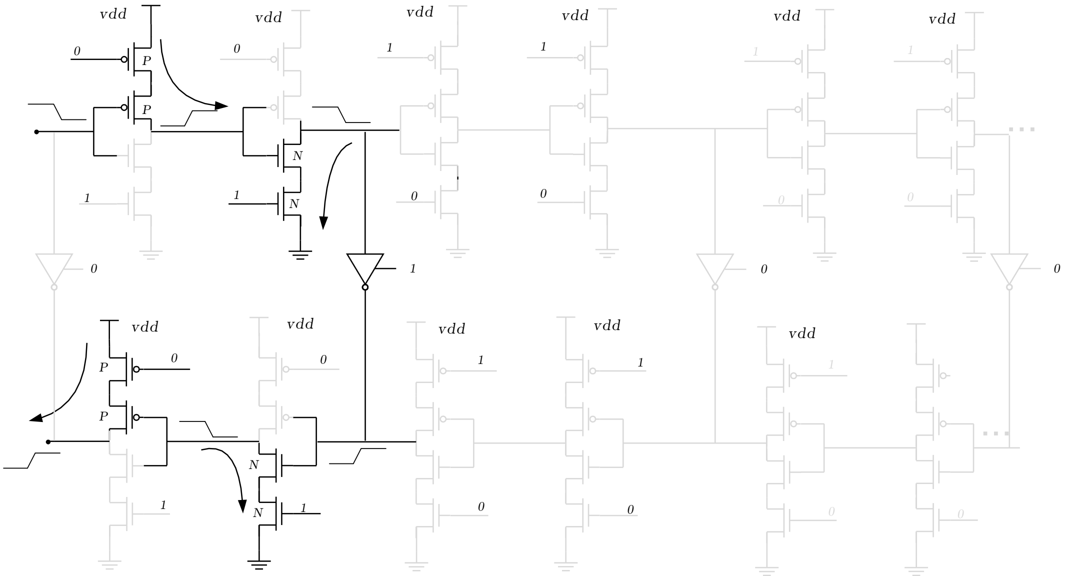


Figure 8-8: Signal flow of a DTC composed of double-inverter delay stages at code n

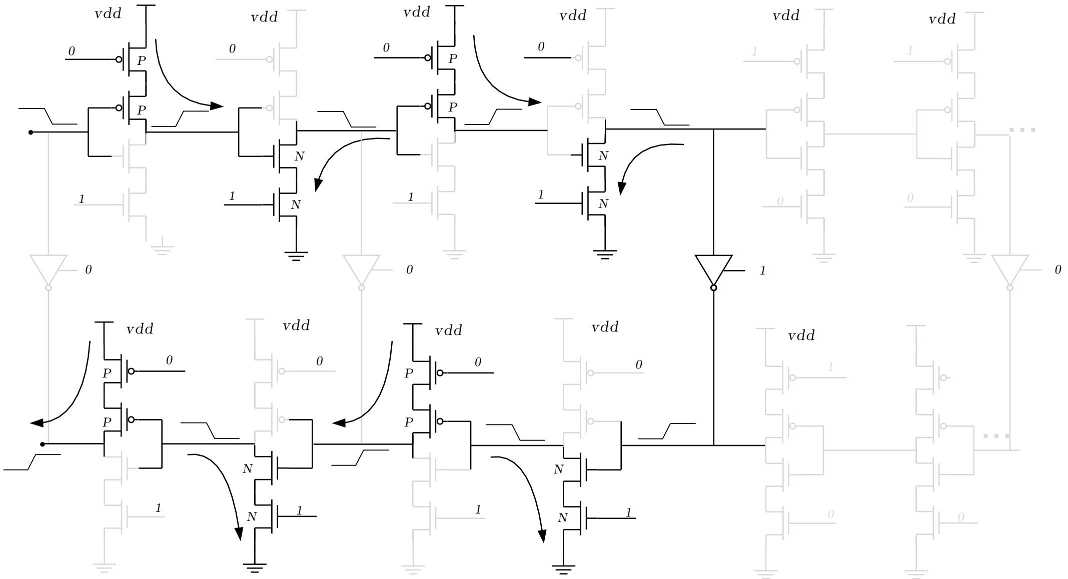


Figure 8-9: Signal flow of a DTC composed of double-inverter delay stages at code $(n+1)$

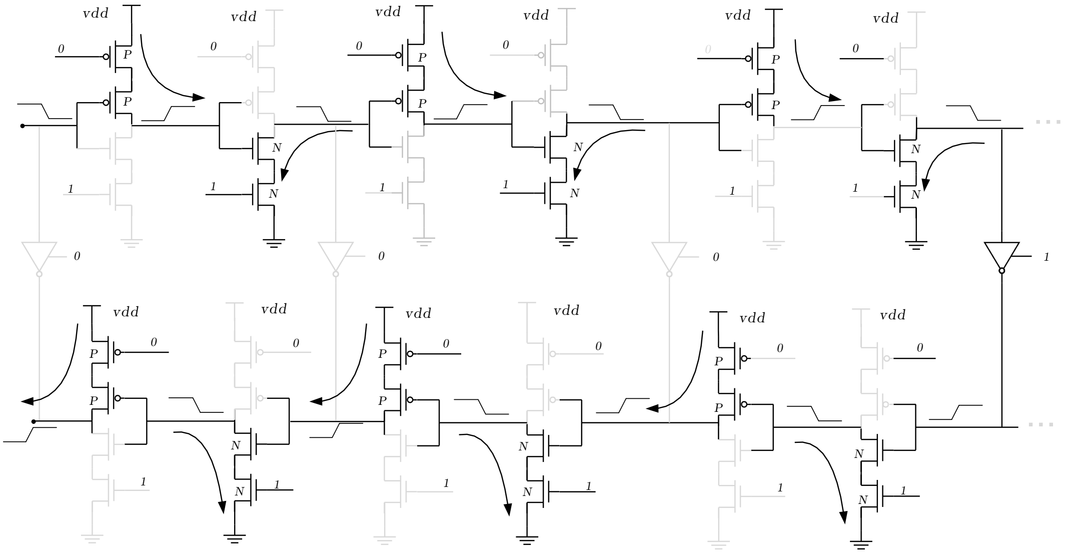


Figure 8-10: Signal flow of a DTC composed of double-inverter delay stages at code $(n+2)$

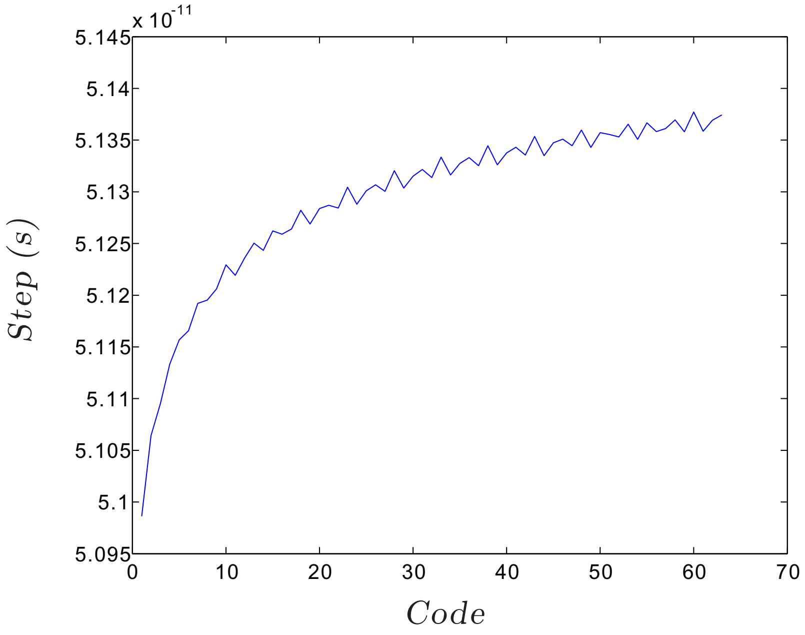


Figure 8-11: Default step size vs. code of a double-inverter DTC

8-1-2 Design Optimization for Reducing the Coarse Step

While using two inverters instead of one alleviates the even-odd mismatch problem, it is also noted during the comparison between Figure 8-11 and Figure 8-7 that the length of unit delay is increased by more than 50%. This will require the fine DTC to have one more bit if the resolution of the fine DTC is to be kept the same. To reduce the delay of unit delay stage, skewed and unsymmetrical tri-inverters are used. Basically, the transistors closer to the foot are sized larger than transistors closer to the center. Also, by exploring the fact that the DTC is only interested in providing the defined delay at either the rising or falling edge, the enabled transistors in a tri-inverter at that edge can be determined. Because of this, the transistors forming the signal path at that edge can be sized larger than transistors that are disabled at that edge. The transistor sizes after optimization are shown in Figure 8-12.

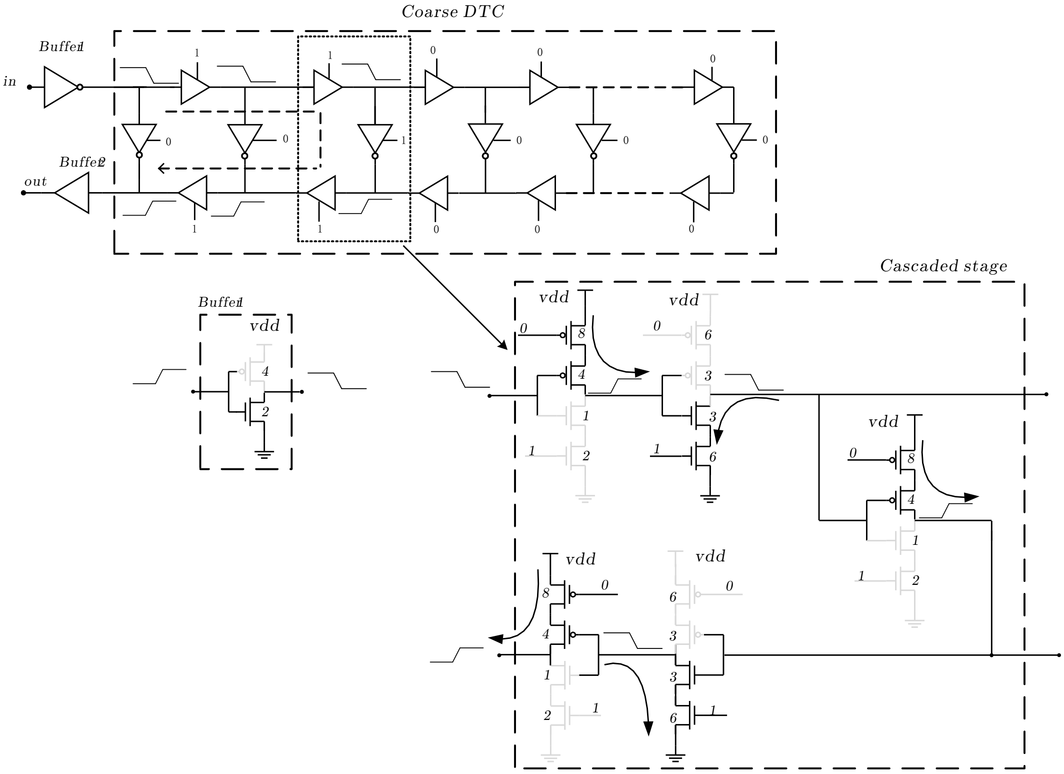


Figure 8-12: Unit cascaded stage with transistor sizes optimized for less delay

By combining the techniques of unsymmetrical and skewed gates, the optimized step size of the double-inverter coarse DTC at the critical signal edge is reduced to the same level as that of a single-inverter coarse DTC. The step size versus code is plotted in Figure 8-13. Although there is a little increase in the mismatch between codes, the amplitude is much smaller than the mismatch shown in Figure 8-7 and it is not caused by even-odd mismatch.

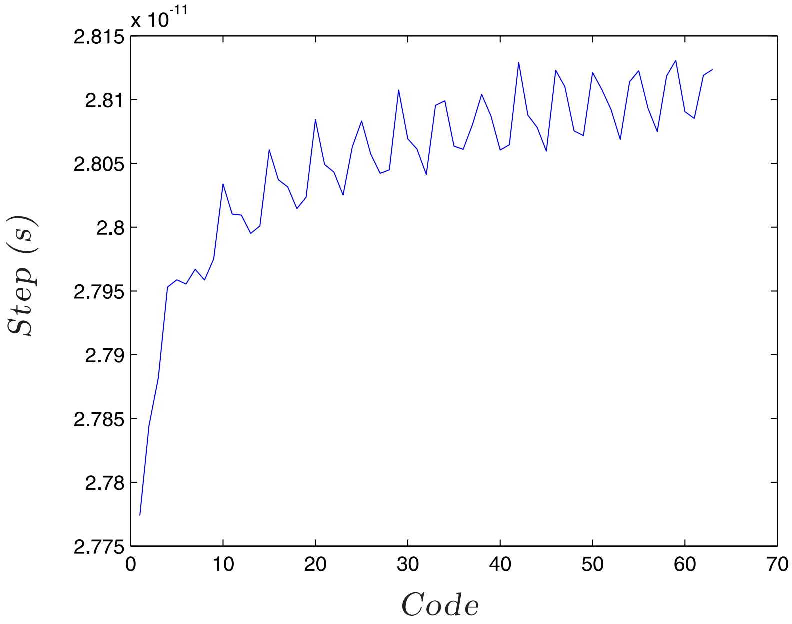


Figure 8-13: Reduced delay with skewed unsymmetrical inverters

8-1-3 Design Summary

Finally, DNL and INL for the 6-bit coarse DTC is shown in Figure 8-14. It shows that the schematic level design for the coarse DTC can meet the requirement for linearity.

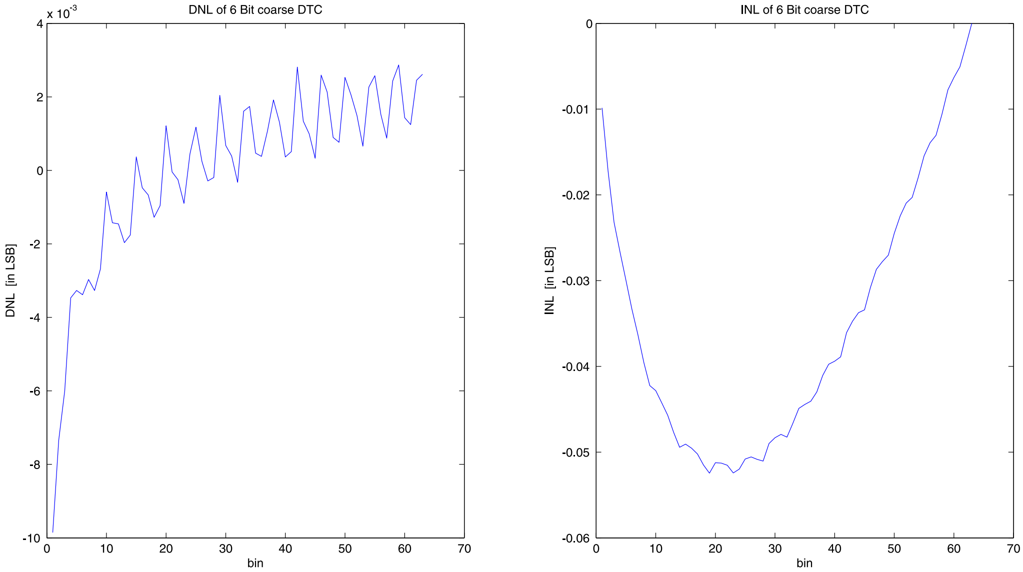


Figure 8-14: DNL & INL for the coarse DTC

8-2 Fine DTC

Fine DTC is implemented by loading MOS capacitors at the output of the coarse DTC. The block diagram for the fine DTC, including the sizes of related buffers, are shown in Figure 8-15. The right inverter in Burrer2 is to create a matched load for the coarse DTC so that the step size for the coarse DTC would not vary considerably at the first few codes. The second inverter in Buffer2 is used to tune the effective increase in delay when increasing the capacitance of the fine DTC. This inverter makes it possible to isolate the system requirement for the step of the fine DTC with the design of the capacitor for fine DTC. Buffer3 is used to make sure that the slope of the output curve of the entire DTC is isolated from the slope change at the node where the fine DTC is loaded.

8-2-1 Principle

Basic theory of translating a capacitance increase to an increase in delay is shown in Figure 8-16. The RC charging time constant is approximately linear with the capacitance at that node and an increase in capacitance at that node would decrease the slope of the waveform. The decrease in the slope will result in a delay of the waveform passing the threshold of its sub-

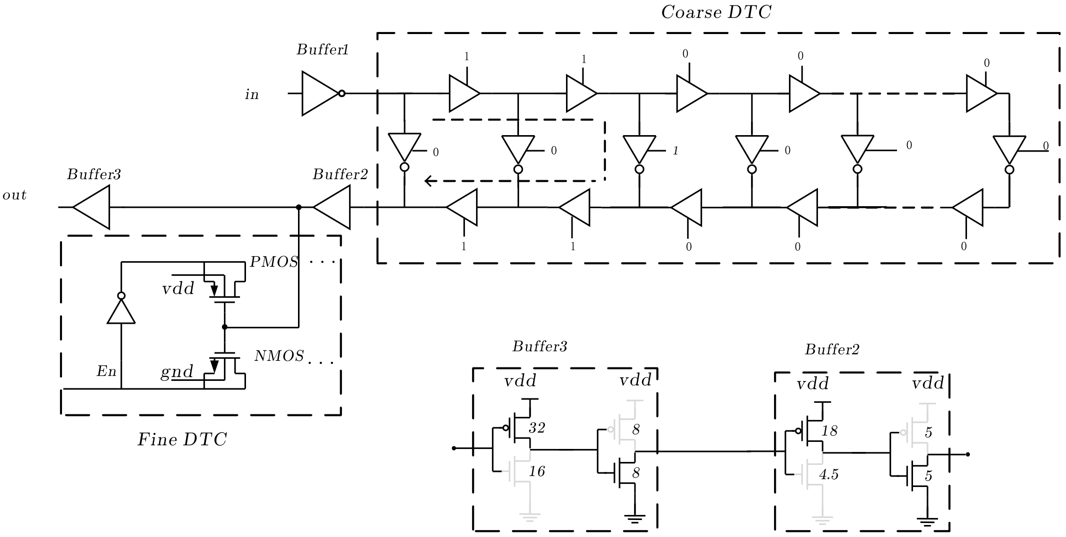


Figure 8-15: Block diagram of fine DTC including buffer size

sequent buffer. At the output of the buffer, the signal is effectively delayed by an additional amount, while the slope of the output signal is about the same.

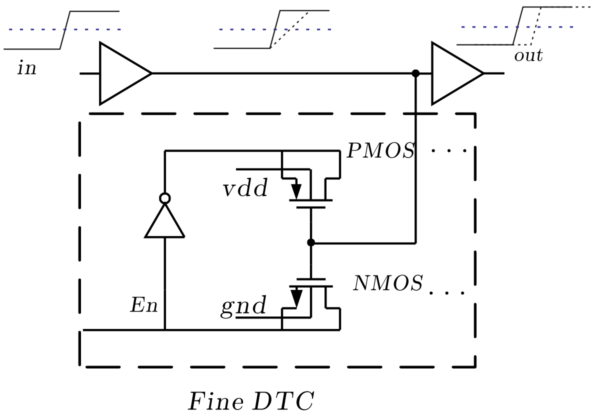


Figure 8-16: Illustration for the theory of the fine DTC

8-2-2 Capacitor

In order to achieve a linear curve of delay, it is first necessary to achieve a linear curve of capacitance. Regarding the use of switchable MOS capacitors, there is a choice to make as whether to use the MOS capacitors in accumulation mode or inversion mode. Inversion mode is chosen in this design because it can ensure almost monotonic curves when the voltage on top of their gate makes almost rail-to-rail transitions [76], which is the case in DTCs. Furthermore, when both PMOS capacitor and NMOS capacitor are needed, it is easier to connect the bulk of the NMOS capacitors to ground, which is the case in inversion mode, than to connect bulk, drain and source of the NMOS together, which is the case in accumulation mode.

Although inversion-mode MOS capacitors can ensure an almost monotonic curve in terms of the biasing points, the capacitance still varies considerably when the gate voltage changes from Gnd to Vdd. Four types of inversion-mode MOS capacitors are shown in Figure 8-17. Their capacitance change in terms of gate biasing points are plotted in Figure 8-18. The $W \cdot L$ for the thin-oxide MOS is 120×120 nm^2, while the size for the thick-oxide MOS is 320×150 nm^2, which is the default minimum size.

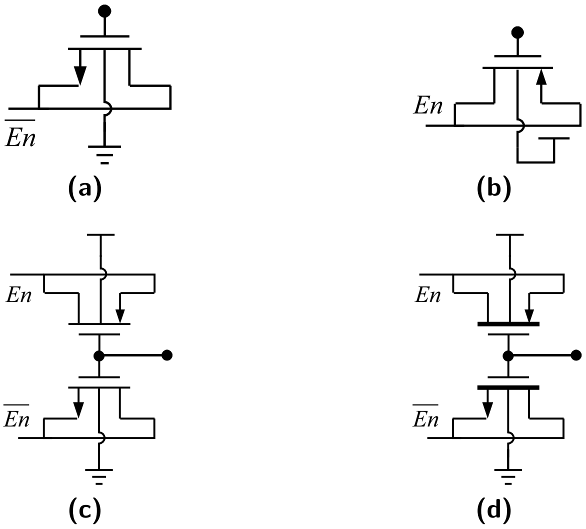


Figure 8-17: Inversion-mode MOS capacitors: (a) Thin-oxide NCAP (b) Thin-oxide PCAP (c) Thin-oxide PNCAP (d) Thick-oxide PNCAP

From Figure 8-18, it can be seen that when only PMOS or only NMOS capacitor is used, the capacitance varies considerably when the gate voltage makes a rail-to-rail transition, which means the capacitance is highly dependent on the biasing point. It is also noted from the principle of the fine DTC shown in Figure 8-16 that the slope of the transition curve at the gate node of the MOS capacitor becomes gentler when the capacitance increases. Thus, every time an additional MOS capacitor is switched in, the effective capacitance of all previous capacitors also varies in an nonlinear way because now the amount of time that the gate voltage stays within a certain region also changes. Judging by the fact that the fine DTC needs to cover two LSBs of the coarse DTC, the slope change at the gate node is not negligible.

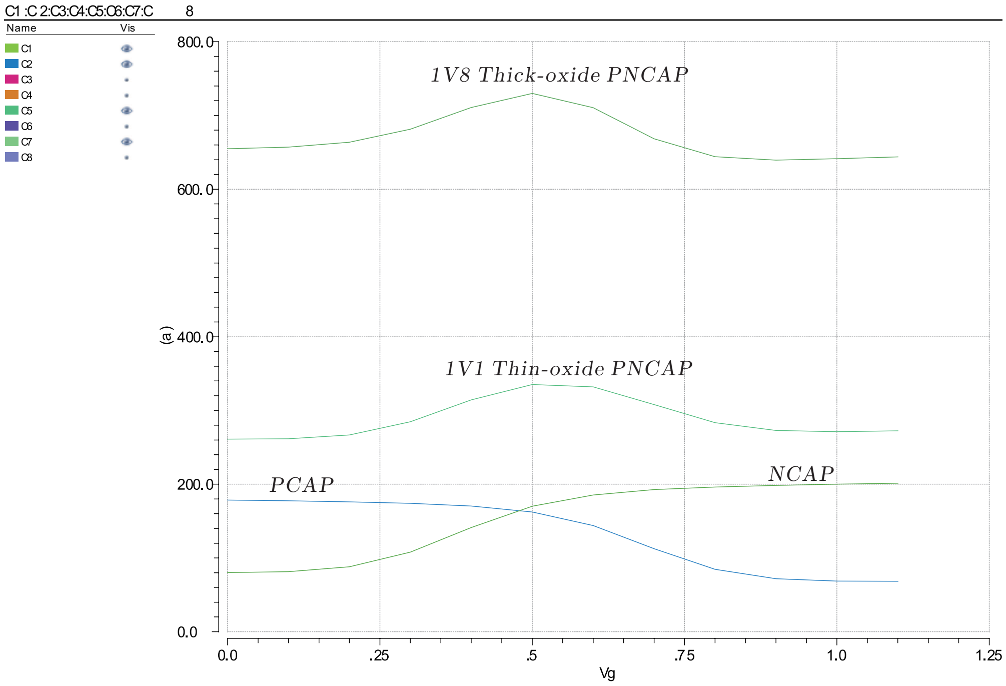
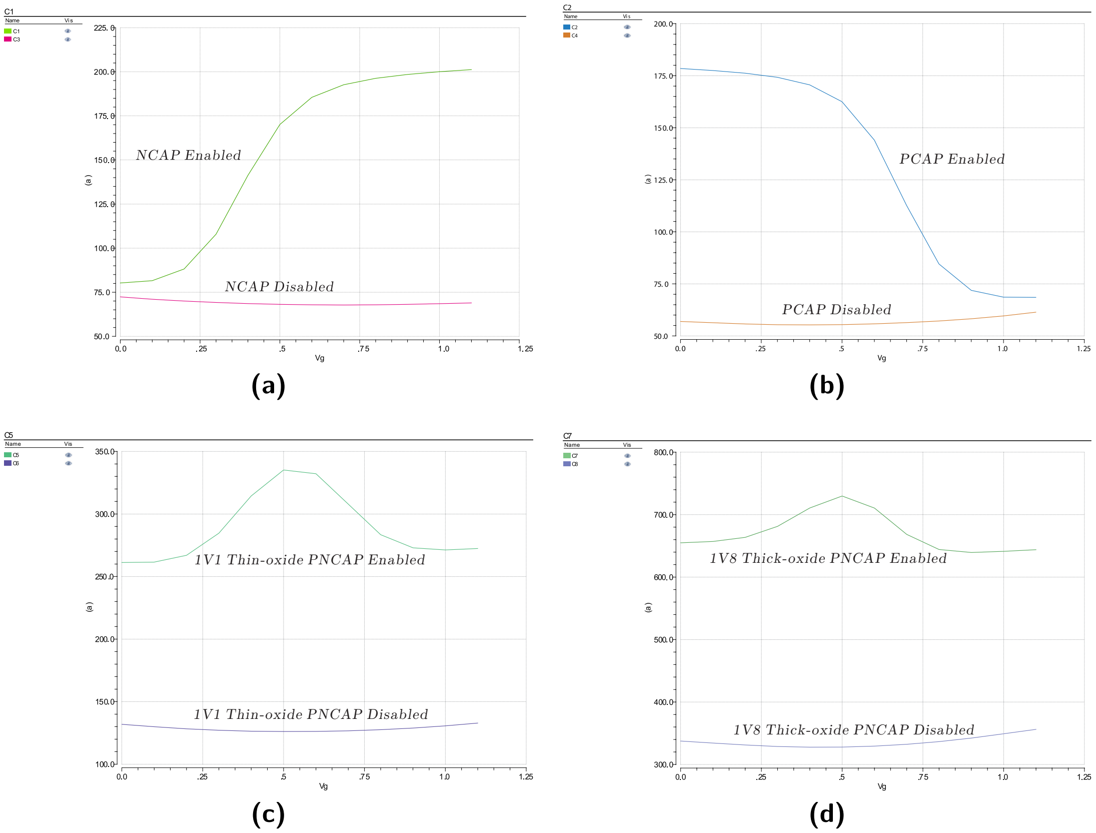


Figure 8-18: Capacitance at enabled state vs. gate biasing point for inversion-mode MOS capacitors

Because of this, a combination of PMOS and NMOS capacitors are used. Their capacitance dependence on gate voltage is opposite to each other and using a combination of them would largely cancel the dependence on gate voltage. The effect can be observed from the 1.1 V thin-oxide PNCAP and the 1.8 V thick-oxide PNCAP curve in Figure 8-18. The variation in capacitance when using a combination of PMOS and NMOS is much smaller than using either PMOS or NMOS alone. Furthermore, the variation is even smaller when using 1.8 V thick-oxide MOS than using 1.1 V thin-oxide MOS. Although, there is a small increase in capacitance around half the supply voltage, that is as far as we can get now.

Finally, the enabled and disabled capacitance versus gate voltage for these four cases are shown in Figure 8-19. It can be seen that the use of a combination of PMOS and NMOS does not sacrifice the tuning range. Thus, 1.8 V PNCAP is chosen as the unit capacitor for the fine DTC.



(a) (b) (c) (d)

Figure 8-19: Enabled vs. disabled capacitance for inversion-mode MOS capacitors: (a) Thin-oxide NCAP (b) Thin-oxide PCAP (c) Thin-oxide PNCAP (d) Thick-oxide PNCAP

8-2-3 Design summary

The step size versus code of the fine DTC is shown in Figure 8-20. It is noted that there is a continuous decrease in the step size along with the increase of the code. This is not due to the inaccuracy from the switched-in capacitance. It is probably due the limitation of the operation principle of the fine DTC.

While the rising time of an edge is approximately linear to the RC time constant at that node, it is not that exact that it will translate into a linear increase in the time points where the curve passes through the threshold of the subsequent buffer. Furthermore, the continuous decrease in the slope of the curve at the node where fine DTC is loaded also has nonlinear impact on the operation of its subsequent buffer. Either of these two factors could be a dominant one in the decrease in the step size shown in Figure 8-20. Finally, the INL and

DNL for the fine DTC is plotted in Figure 8-21. While there is a maximum of 2 LSBs in INL, the DNL is fairly good.

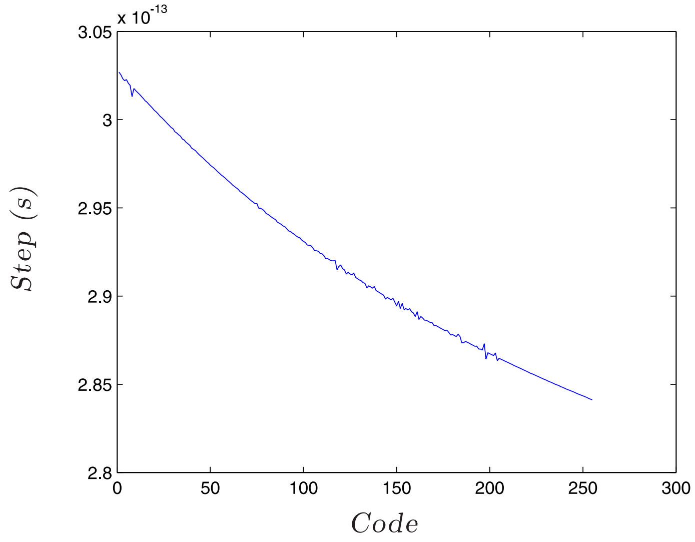


Figure 8-20: Step size vs. code of the fine DTC

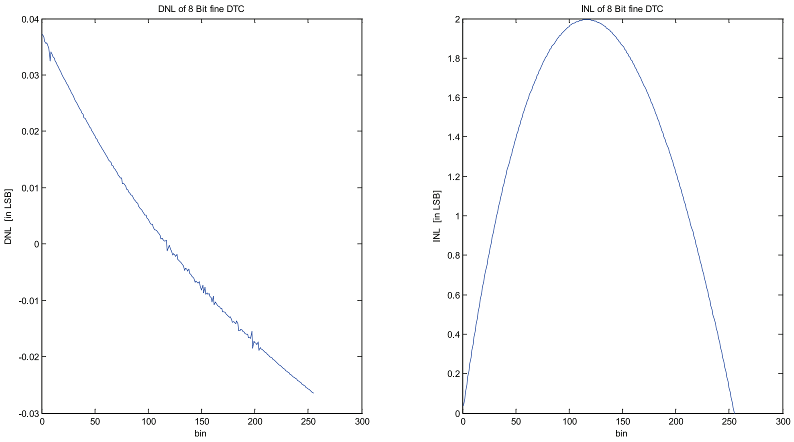


Figure 8-21: DNL & INL of the fine DTC

Chapter 9

Design of Other Analog Blocks

9-1 Design of the Peak Detector and Comparator

9-1-1 Design of the Peak Detector

In order to set the amplitude of the oscillator within in a small region from the differential voltage across the tank before start up, a fore-ground amplitude calibration is used in this design. The first step in amplitude calibration is to detect the amplitude of the DCO. Because of confidential reasons, only a black-box representation of the amplitude detector is shown in Figure 9-1. The peak detector detects the amplitude and delivers two output voltages, V_{o1} and V_{o2}, to its subsequent comparator. The truth table for V_{o1} and V_{o2} in terms of amplitude is shown in Table 9-1.

Table 9-1: Truth table for V_{o1} and V_{o2} in terms of amplitude

Amplitude:	<threshold	>threshold
V_{o1}-V_{o2}:	<0	>0

One of the deign constraints for the peak detector is to settle within half of ref period, which is 20 ns. Since peak detector is only active during the amplitude calibration phase before

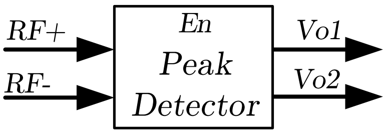


Figure 9-1: Black-box representation for the peak detector

locking and within a small window specified by the outer-loop FSM, its power budget is pretty relaxed. The settling curve for the peak detector output is shown in Figure 9-2. To ensure the settling of the output nodes within specified time, a total of 58 µA is used for the peak detector. It can be seen indeed from Figure 9-2 that the peak detector settles within the deigned 20 ns.

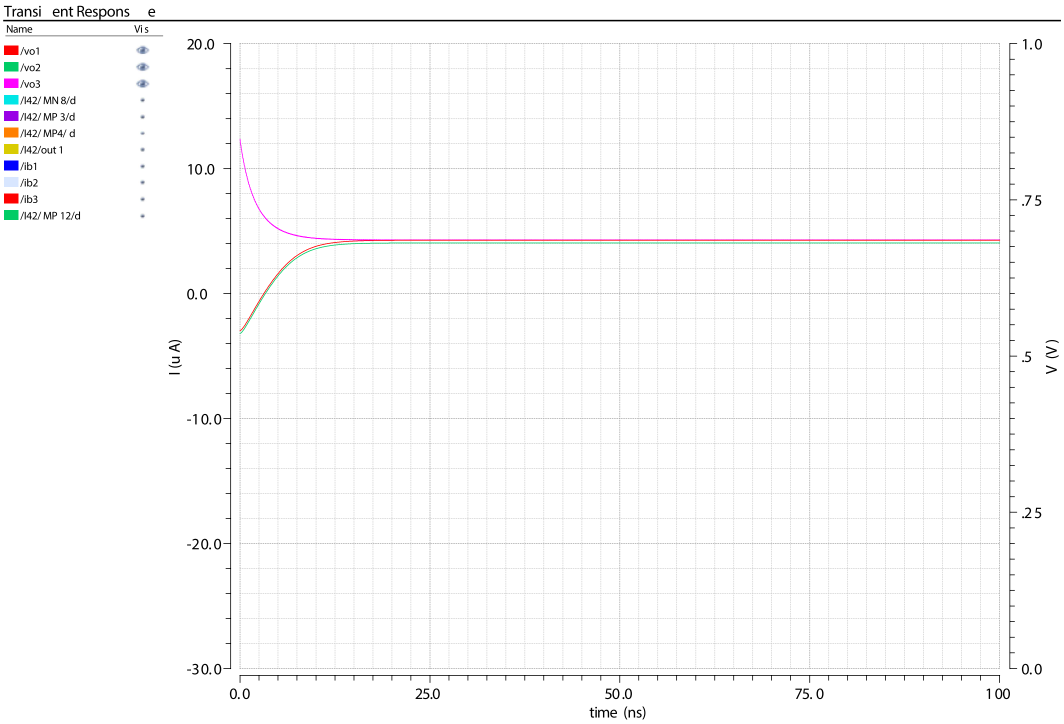


Figure 9-2: Settling behavior for peak detector

9-1-2 Design of the Comparator

The comparator used in this design is a common clocked comparator, the schematic of which is shown in Figure 9-3. Similar to the peak detector, it is only active during the amplitude calibration phase and within a small specified window from the outer-loop FSM. The design of the comparator is simple here because the specification for accuracy is not stringent. Furthermore, the static part of the offset from the comparator can be tuned out when used together with the peak detector. Another design constraint for the comparator is that it also needs to settle within 20 ns.

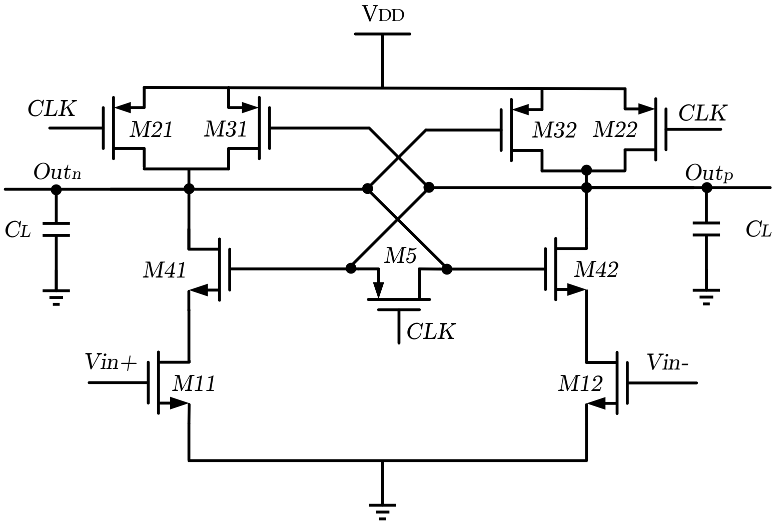


Figure 9-3: Schematic for the clocked comparator used in this design

9-1-3 Verification of the Peak detector and Comparator

The block diagram for the peak detector & comparator block is shown in Figure 9-4. Controls from the outer-loop FSM and the inner-loop FSM are included as well. The block compares the DCO amplitude with two thresholds: 395 mV and 400 mV. It sends two comparison results to the inner-loop FSM. The DCO amplitude will be corrected until the amplitude stays within this region. The two comparison results, when the amplitude varies from 394 mV to 400 mV, are shown in Figure 9-5.

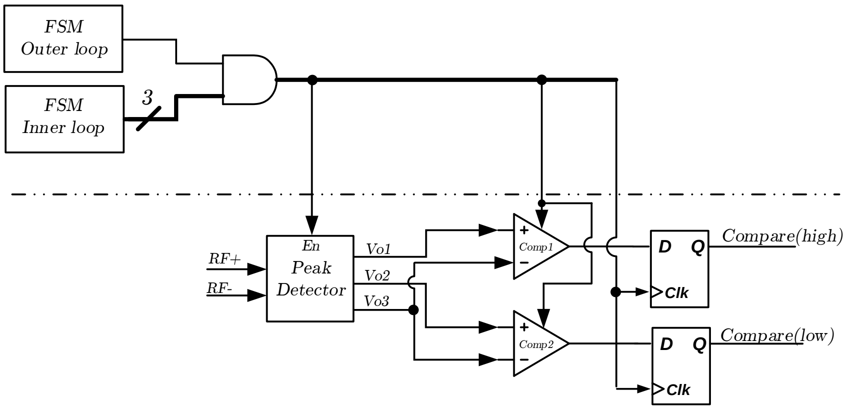


Figure 9-4: Block diagram of the peak detector & comparator

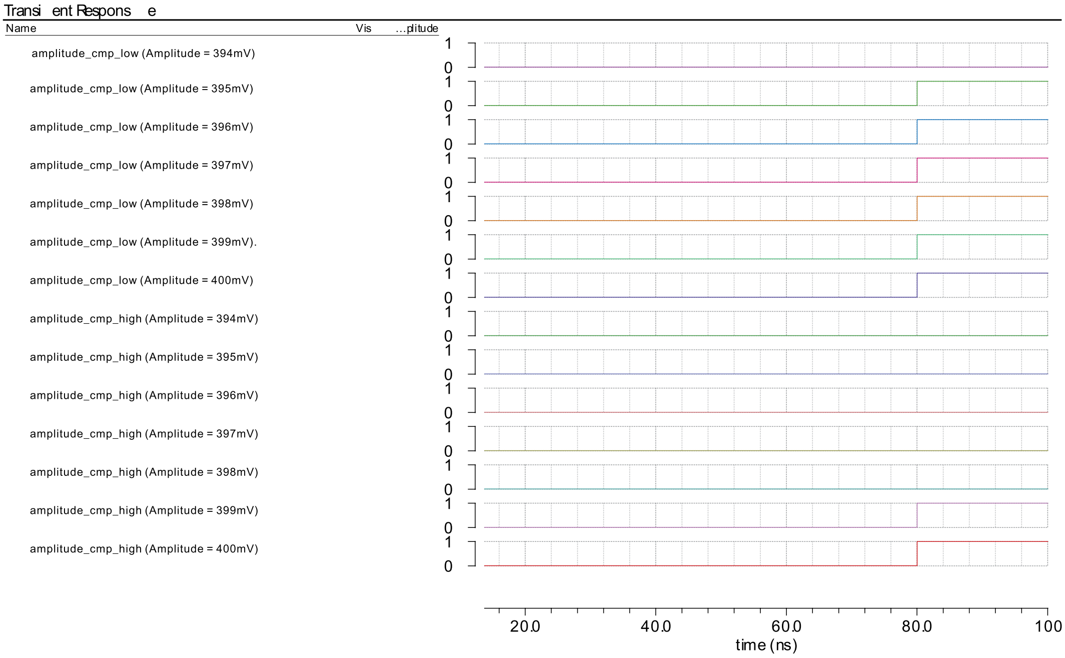


Figure 9-5: Function verification of peak detector and comparator

9-2 Design of the Buffer

Due to the special start-up behavior, it is better to use a DC-coupled buffer instead of an AC-coupled one. The buffer uses the same structure as is presented in [61]. It should be emphasized that inverter buffer is not the optimum choice for a high frequency oscillator running at 7.0 GHz. However, due to limitation of time, a simple inverter is used here. For future works, it is better to carefully design a CML buffer, which would provide a much better CMRR than simple inverters. Schematic of the buffer is shown in Figure 9-6, together with the transistor sizes.

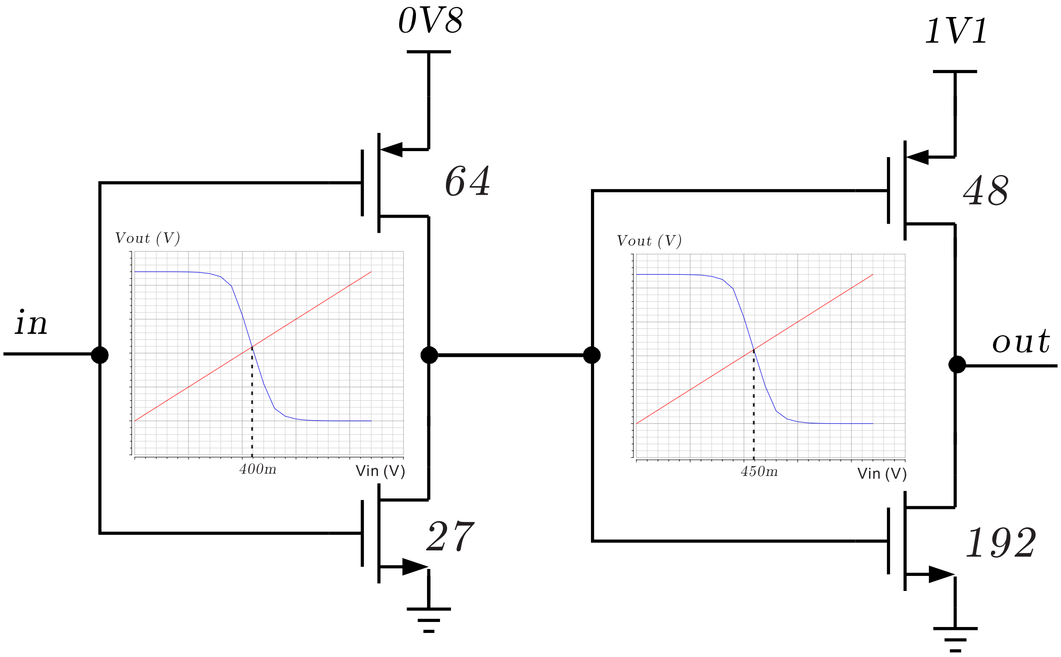


Figure 9-6: Schematic of the buffer used in this design

The 1^{st} stage of the buffer is power by 0.8 V supply, since the output from the DCO also makes approximately 0 to 800 mV transitions. The threshold value, where input of the inverter equals its output, is tuned a bit to be at 400 mV. The 2^{nd} stage is powered by 1.1 V supply so that its output can drive its subsequent divider, which further drives digital circuitry. Since the default threshold of an inverter power by 1.1 V would be about 550 mV, there is a gap between the threshold of the output of the 1^{st} stage and that of the input of the 2^{nd} stage by default. Thus, the threshold of the 2^{nd} stage is made lower by over-sizing the NMOS transistor over the PMOS transistor. The resulting threshold of the 2^{nd} inverter is 450 mV. Over-sizing the NMOS further brings limited decrease in threshold and would start to affect proper operation of the buffer.

9-3 Design of the Divider

Since an ordinary counter cannot possibly handle a signal frequency up to 7.0 GHz, the RF output is divided by 4 before being sent to the counter. A TSPC divider structure is chosen in this design. The dynamic logic of TSPC is advantageous in power consumption at high frequency. The schematic of the divider-by-2 unit is shown in Figure 9-7. Since the divider needs to be reset after each burst, extra transistors for resetting purpose are added. During continuous operation, the internal states of the divider toggle between 1, 2, 3 and 4. Thus, the divider is reset at state 3 before start-up so that the 1^{st} rising edge from the DCO will directly make a transition at divider output.

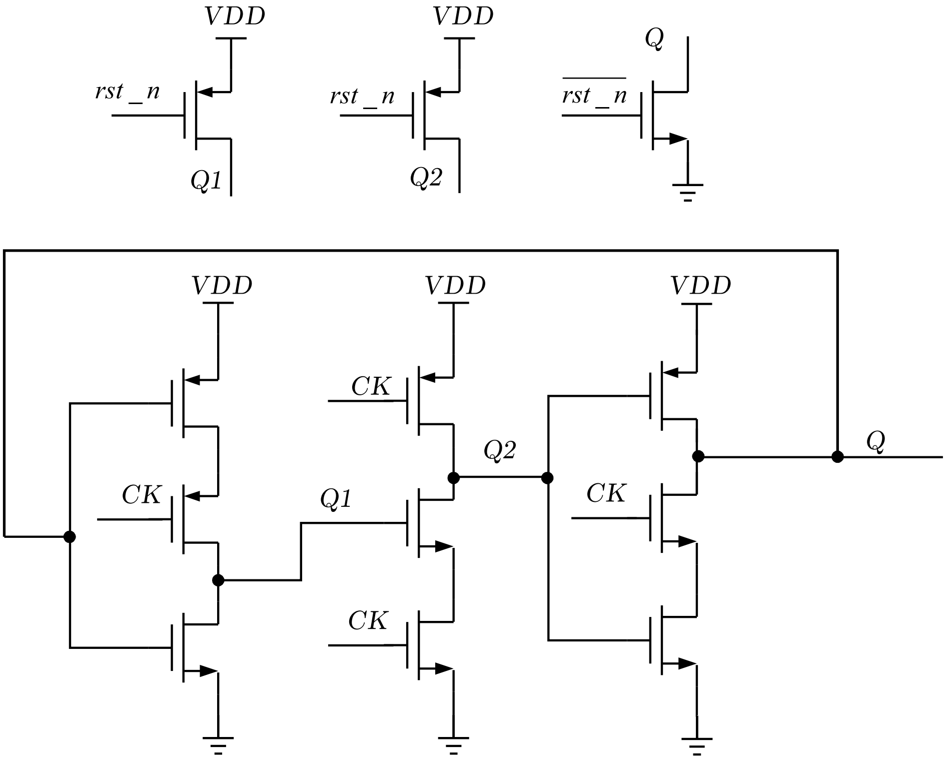


Figure 9-7: Schematic for the TSPC divide-by-2 unit used in this design

Verification of the divider can be found in Figure 9-8. It can be seen that the divider internal nodes are set to the designed value when the input clock is inactive and the reset signal is low.

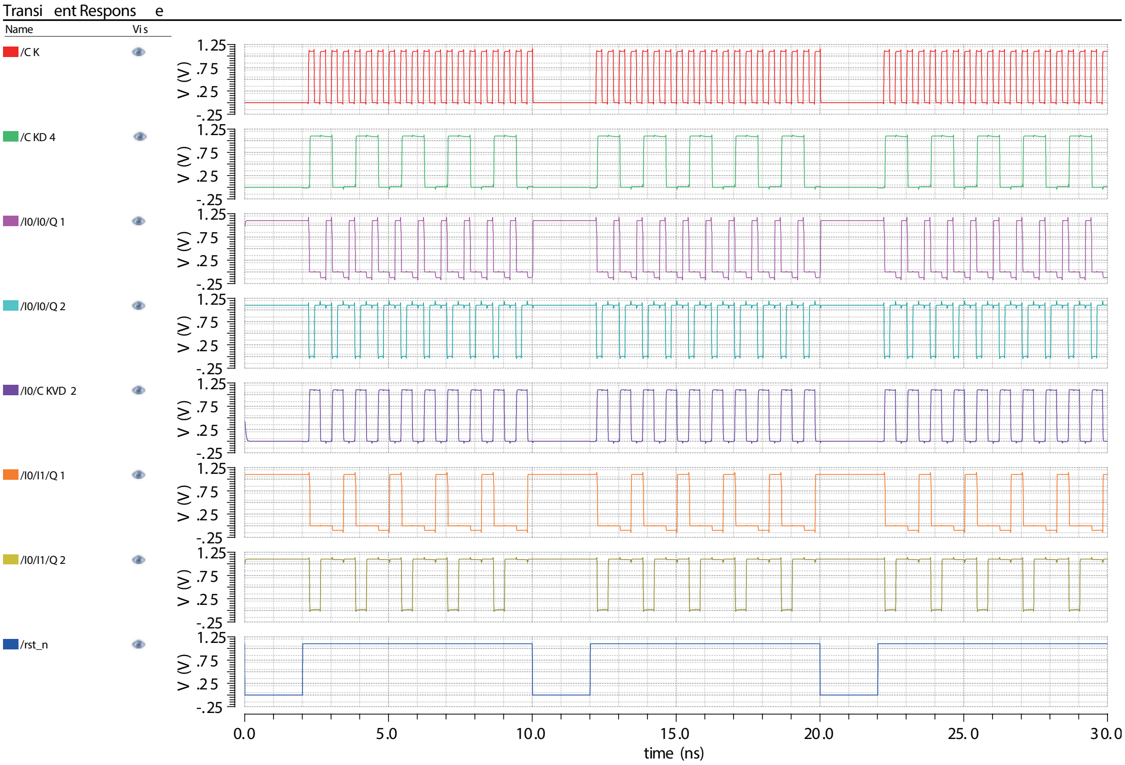


Figure 9-8: Function verification for the divide-by-4 block

Chapter 10

Simulation Results of the DC-ADPLL

In additional to the settling of the DTC gain and the capacitor tuning word shown in Figure 7-21 and Figure 7-25, more aspects regarding the performance of the designed Duty-Cycled All-Digital PLL (DC-ADPLL) will be shown in this chapter. All performances are achieved from simulations, not on-chip measuring. Readers need to pay attention to this when comparing the performance of this work with previous works. However, since this work achieves improvement in performance through change of architecture, and the improvement in noise compared to all previous Duty-Cycled PLLs (DCPLLs) is several orders of magnitude, simulation results are enough to verify the large improvement of this work.

10-1 Frequency Within a Burst

Figure 10-1 shows the frequency of RF output within a burst when FCW_I = 280 and FCW_F = $14.325 \times 10^3/2^{16} = 0.2186$. The top graph in Figure 10-1 shows a far-out view while the bottom one zooms in the vertical frequency axis. The first few RF cycles are not measured because there is excessive frequency error in those cycles, which are the gray points at the beginning of the lines in Figure 10-1. The DCO is turned on for a few more cycles after the 2^{nd} comparison, which are shown as the gray points at the end of the curves.

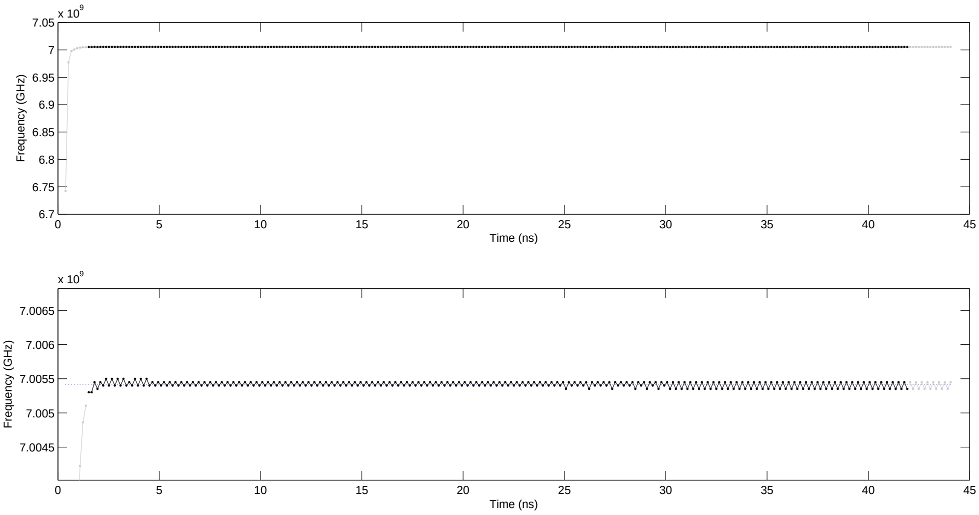


Figure 10-1: frequency within a burst after settling

With the programmed FCW and f_{ref} = 25 MHz, $f_{CKV,ideal}$ = 7.005464 GHz. The valid points, shown as dark points in Figure 10-1, have an average frequency of 7.005396 GHz. The remaining error of -68 kHz in this burst are due to the following several sources:

1. Accumulated jitter during this burst: From the simulation of the DCO open-loop phase noise, the accumulated jitter at 7.005 GHz over 40 ns is 0.151 ps, which is equal to an accumulated timing error caused by a 26 kHz frequency offset, which means the average frequency of each burst has a variance of 26 kHz at 7.005 GHz.

2. Capacitor bank quantization: The fine bank step of 2.99 aF translates to a frequency step of 40 kHz at 7.005 GHz. The quantization error is 20 kHz.

3. DTC2 quantization error: The minimum step for DTC2 is 300 fs, which means a 150 fs quantization error at the 2^{nd} ref rising edge. Since the fractional cycle set by DTC2 is open loop, this error will translate to a frequency error without being suppressed by the loop, which is 26 kHz.

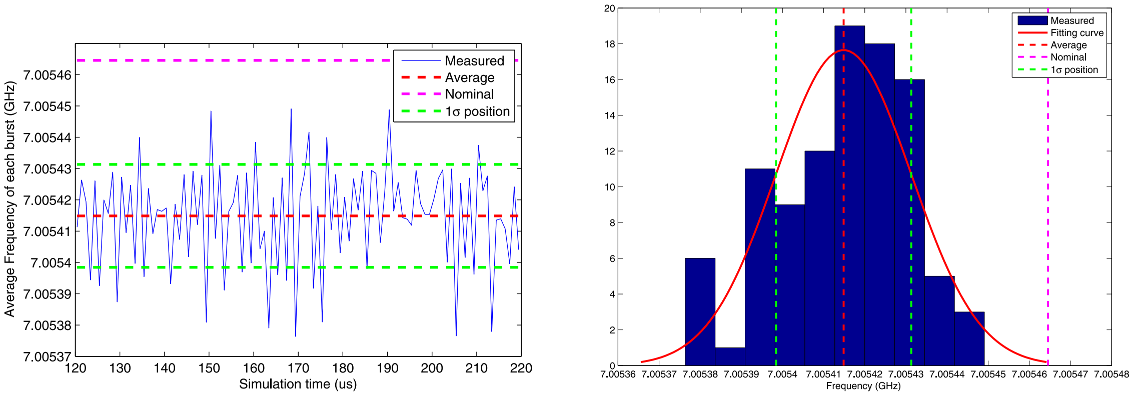
4. DTC1 quantization error: The minimum step for DTC1 is also 300 fs, which means a 150 fs quantization error at the 1^{st} ref rising edge. It has an similar effect to the quantization error from DTC2, since the comparison is based on the interval between the two delayed ref rising edges.

5. DTC gain error: The control word to DTC2 is a multiplication of the fractional part of the control with the extracted DTC gain. DTC nonlinearity or layout-level mismatch will result in an deviation between the extracted gain and the exact gain of DTC2. At the specified FCW, DTC gain is multiplied with a fractional part that equals 0.945, which is close to unity. Any gain error will be highly amplified at this point. In the simulation, the combinational error of DTC2 quantization error and the gain error result in a delay that is 700 fs larger than the nominal 539.781 ps. This error translates into a negative bias of 123 kHz in frequency, which dominates over other random error sources and explains the total negative frequency error.

6. Accuracy of Verilog AMS simulation: The error from simulator mainly comes from the comparison during edges, where digital edges are translated into analog edges by the simulator and further passed through the DTC to be compared at the TDC. Since the start-up frequency settling of the DCO is important and at this moment the author has not found a way to model it accurately in a behavioral way, the DCO has to be simulated in schematic view. During the simulation, the step size of simulation time increment is not fixed so that the simulator can use a larger time step during the vacancy between bursts. A high frequency dummy oscillator is turned on during the existence of a burst, suggesting the simulator to use a smaller time step. The simulation time step during the burst is 100–200 fs, dynamically determined by the simulator. The simulation time for 200 µs is 48 hours. Simulator error can still be observed while it is difficult to increase simulator accuracy.

10-2 Frequency over Multiple Bursts

While Figure 10-1 shows the frequency within a single burst, the frequency stability over 100 cycles in locking state is shown in Figure 10-2a, whose corresponding histogram is shown in Figure 10-2b. The nominal frequency is 7.005464 GHz while the averaged frequency over 100 cycles is 7.005415 GHz. The 1σ deviation is 16 kHz. The -50 kHz deviation between the average value and the nominal value is biased by a larger-than-nominal delay from DTC2.



(a) **(b)**

Figure 10-2: Frequency distribution over 100 bursts: (a) Burst frequency over 100 cycles (b) Histogram of burst frequency over 100 bursts

The fractional cycles set by DTC2 over these 100 cycles are shown in Figure 10-3a while their histogram is shown in Figure 10-3b. The nominal value is 539.781 ps while the average value is 540.498 ps. The 1σ deviation is 246 fs. The excessive 716 fs creates a bias of -125 kHz in frequency, which explains the frequency deviation on the negative side in Figure 10-2.

10-3 Summary of the DC-ADPLL Performance

Performance of DC-ADPLL performance is summarized in Table 10-1 and compared with previous works. Although the metrics in this design are based on Verilog AMS simulation while those in literature are silicon-proved, the improvement compared to [11] is tremendous. Accumulated jitter is 2 orders of magnitude lower. The offset in [11] due to the delay of turning-on moment is cancelled with DTC1 with residual quantization error. The deviation

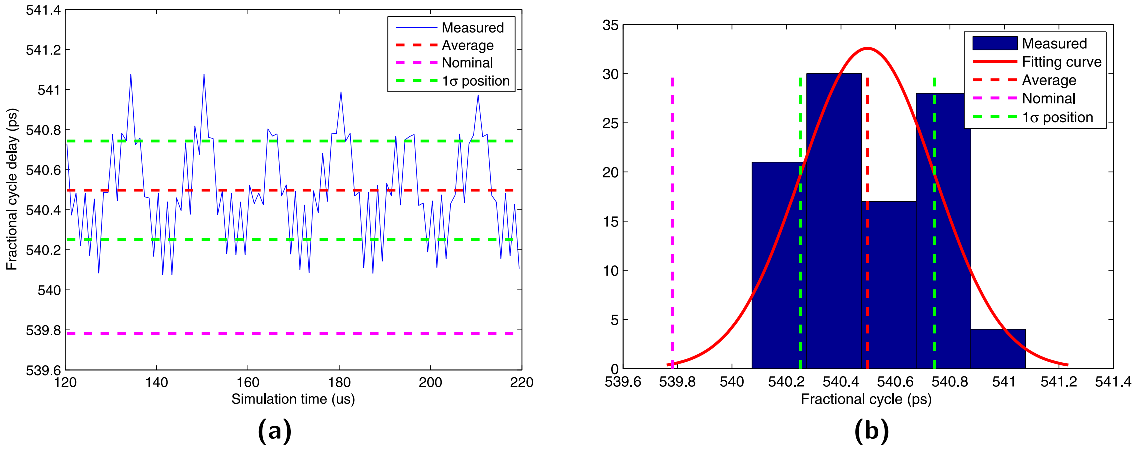


Figure 10-3: Fractional cycle set by DTC2 over 100 bursts:
(a) Fractional-cycle delay over 100 bursts
(b) Histogram of Fractional-cycle delay over 100 bursts

between the average and the nominal frequency in this design is dominated by the fractional-cycle delay error set by DTC2. Yet, this error of 716 fs in Figure 10-3, when divided by the burst length of 40 ns, translates into a 0.002% error.

Table 10-1: DC-ADPLL performance and comparison with previous works

	This work		[11]	[67]
Technique	DC-ADPLL		DCPLL	ADPLL
Frequency (GHz)	5.05	7.01	1.005	0.825
DCO power (mW)	1.2	0.65	0.1	21.6
Resolution (Hz)	15 k	40 k	20 M	15
Accuracy				
Accumulated jitter @40 ns	0.131 ps	0.151 ps	26.8 ps	0.045 ps
Frequency offset	None*		0.2%	None

10-4 Extendability

As the first-ever DC-ADPLL, this design includes only the most basic features of a DCPLL, such as the ones in [11]. However, the architecture can allow a lot more additional features according to the specific application it is used in.

10-4-1 Arbitrary Start-up Phase

The start-up phase of the RF output in this specific design is made to be aligned to the reference clock with a constant phase difference between them. For applications where the start-up phase difference between CKV and ref needs to maintain the phase between CKV and ref at the end of last burst, presumably an integer multiple of fractional cycle, another DTC, whose delay is an accumulation of fractional cycle, can be inserted at the shared ref path. The modified block diagram, as apposed to Figure 7-15, is shown in Figure 10-4. It should be noted that although the phase at the end of previous burst is maintained, accumulated jitter at the end of previous burst is still cleared.

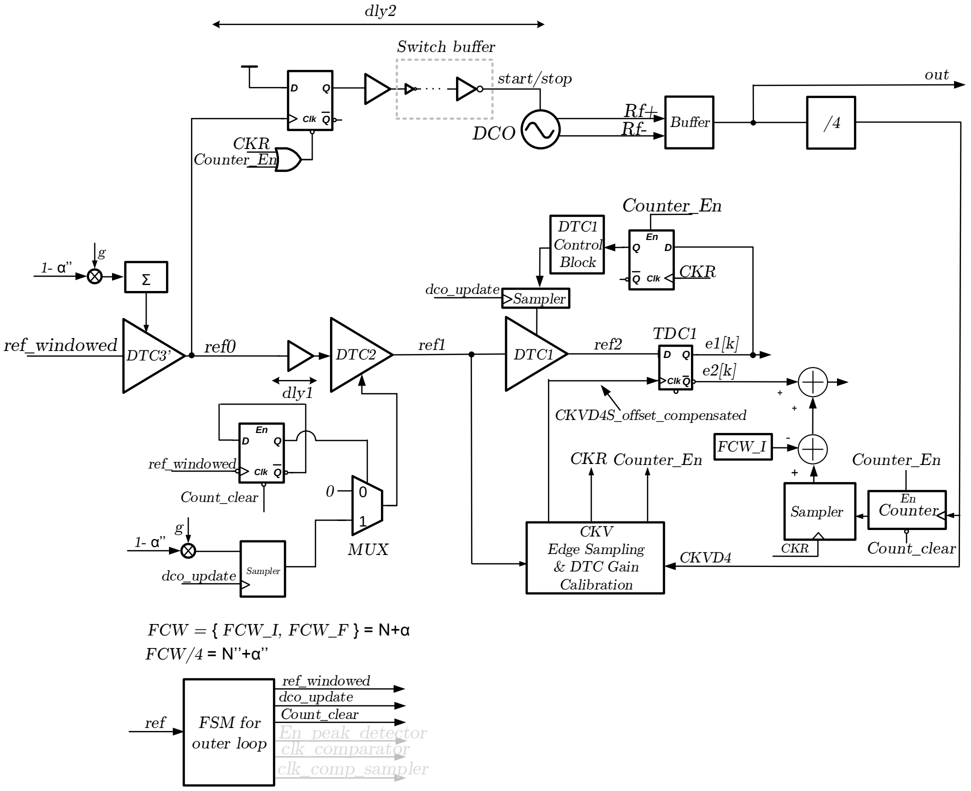


Figure 10-4: Modified block diagram to maintain fractional phase between bursts

The timing diagram for the block diagram in Figure 10-4 is shown in Figure 10-5 and Figure 10-6. Because of the phase-maintaining feature, the timing diagram changes with each burst, opposed to the one in Figure 7-14. The relative phase between $ref\_windowed[r](1)$

and $CKVD4\_offset\_compenstated[r](1)$ is a periodic process for this particular phase-maintaining requirement. Without loss of generality, the timing diagram of the n_{th} burst within this period is plotted in Figure 10-5, where *end_phase(n)* - *start_phase(n)* = *Fractional cycle*. The timing diagram for the $(n+1)^{th}$ burst is shown in Figure 10-6. An additional delay that equals *(Fractional cycle)* is added to DTC3' so that *start_phase(n+1)* = *end_phase(n)*.

While an additional DTC, DTC3', is added to maintain the phase between bursts, since its normalized control word traverses from 0 to unity, its gain can be self-calibrated in the same way as in [61] and further used by DTC2. Thus, DTC3 in the present design can be eliminated and the total number of DTCs needed is kept at 3.

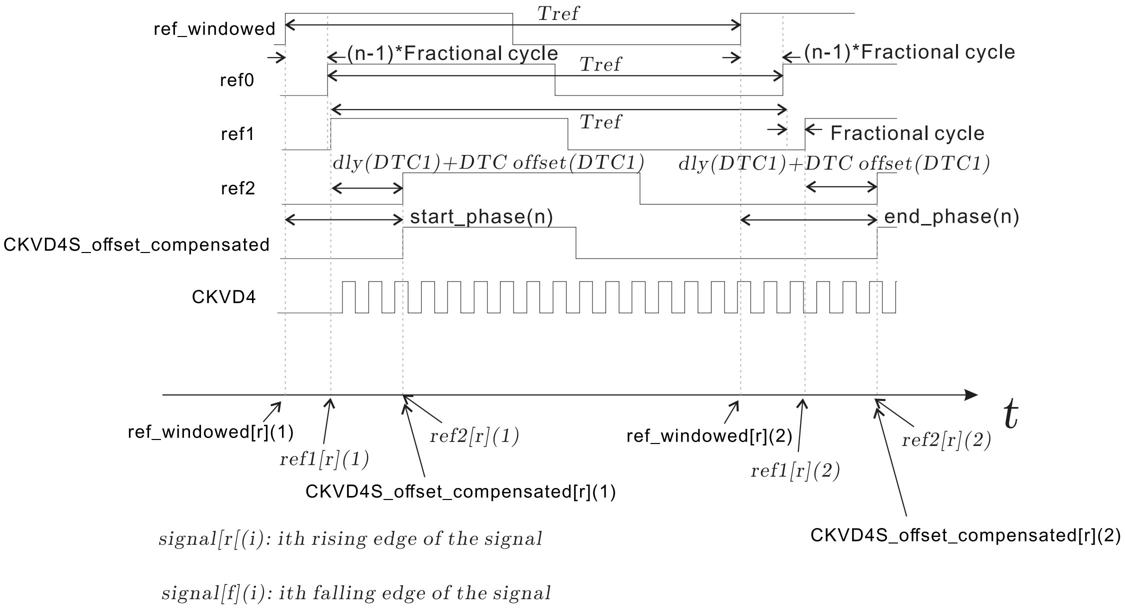

ref_windowed
Tref
(n-1)*Fractional cycle
Tref
(n-1)*Fractional cycle
ref0
Tref
ref1
Fractional cycle
dly(DTC1)+DTC offset(DTC1)
dly(DTC1)+DTC offset(DTC1)
ref2
start_phase(n)
end_phase(n)
CKVD4S_offset_compensated
CKVD4
t
ref_windowed[r](1)
ref1[r](1)
ref2[r](1)
CKVD4S_offset_compensated[r](1)
ref_windowed[r](2)
ref1[r](2)
ref2[r](2)
CKVD4S_offset_compensated[r](2)
signal[r[(i): ith rising edge of the signal
signal[f](i): ith falling edge of the signal

Figure 10-5: Timing diagram at n^{th} burst

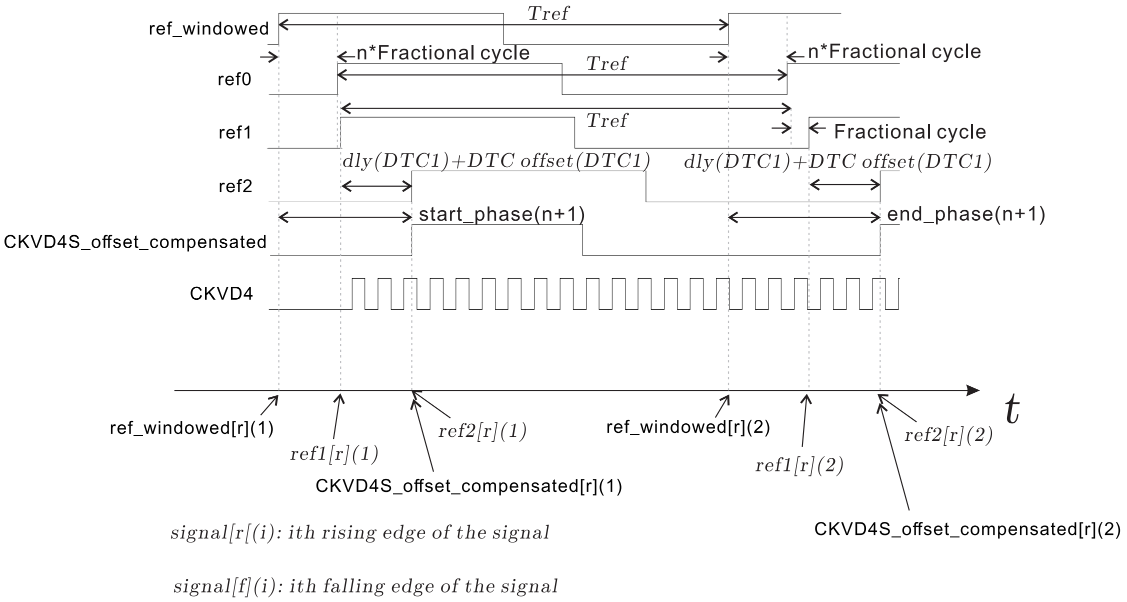

ref_windowed
Tref
ref0
n*Fractional cycle
Tref
n*Fractional cycle
ref1
Tref
Fractional cycle
dly(DTC1)+DTC offset(DTC1)
dly(DTC1)+DTC offset(DTC1)
ref2
start_phase(n+1)
end_phase(n+1)
CKVD4S_offset_compensated
CKVD4
t
ref_windowed[r](1)
ref1[r](1)
ref2[r](1)
CKVD4S_offset_compensated[r](1)
ref_windowed[r](2)
ref1[r](2)
ref2[r](2)
CKVD4S_offset_compensated[r](2)
signal[r[(i): ith rising edge of the signal
signal[f](i): ith falling edge of the signal

Figure 10-6: Timing diagram at $(n+1)^{th}$ burst

10-4-2 Shorter Locking Time

The DC-ADPLL in this design locks after 120 updating cycles. The locking time is dominated by the settling of DTC fine gain, which is released from a 20%-difference position, enough to model process variations. Both the capacitor bank and the DTC gain calibration rely on the comparison result of the 1b-TDC and the searching is done in a binary way to keep the number of required cycles at minimum.

While the total number of cycles required to lock is relatively low, an updating interval of 1 µs by default makes the absolute time not so appealing. Regarding this, the DC-ADPLL can be designed to have a different updating interval before locking. The absolute locking time directly scales with this. In the ultimate case, the DC-ADPLL can be designed to operate in continuous mode at first and switch to duty-cycled mode after the searching has been narrowed down to a much smaller range.

Chapter 11

Conclusions and Future Work

11-1 Conclusions

The contribution of this thesis project is summarized as following:

1. The phase noise spectrum for Duty-Cycled PLLs (DCPLLs) is derived, providing a quantitative analysis tool for DCPLL design.
2. A split inductor for instantaneous start-up circuitry is designed and modeled.
3. A novel instantaneous start-up circuitry for LC Digitally Controlled Oscillators (DCOs) is proposed by the author and his NXP supervisor, making LC oscillators a feasible choice as duty-cycled oscillators.
4. A novel instantaneous start-up LC DCO is designed and its traditional metrics as well as instantaneous start-up property are verified in simulations.
5. A first-ever LC oscillator based Duty-Cycled All-Digital PLL (DC-ADPLL) is proposed and designed. Compared with previous DCPLLs, the DC-ADPLL based on the LC DCO is much lower in noise, cancels initial offset and provides fine fractional resolution. The

functionality of the loop is verified by Verilog-AMS simulations. From the simulations, the accumulated jitter is 2 orders of magnitude lower than all previous DCPLLs.

11-2 Future work

As a raw model, the DC-ADPLL provides a first-ever prototype in this category. It means at the same time that there is much room for optimization. For example, a resistor-biased capacitor bank structure can be explored. The DCO can be designed more carefully to achieve a lower noise. The fine DTC can be optimized to increase in linearity. The output buffer can be replaced with a carefully designed DC-coupled Current Mode Logic (CML) buffer to increase Common-Mode Rejection Ratio (CMRR).

Regarding features, extra features such as auxiliary start-up phase and shorter locking time can be implemented according to the requirements of specific applications.

Appendix A

MATLAB Listing

A-1 MATLAB Listing for Chapter 2

Listing A.1: MATLAB listing for Figure 2-10

```
clear;
format long;
PN = -110;
f_offset = 1e6;
fosc = 5e9;
Tosc = 1/fosc;
t=linspace(0,2e-7,1000);
f=linspace(-10e6,10e6,1000);
omega = 2*pi.*f;
N0 = 2*pi*(10^(PN/10))*(f_offset)^2/(fosc^2);
[t,omega] = meshgrid(t,omega);
W = N0/pi*((sinc(omega.*t./pi)).^2).*t.^2;
W_phase_domain = W/Tosc ^2*(2*pi)^2;
mesh(t,omega,W_phase_domain);
xlabel('t (s)'),ylabel('\omega (rad/s)'),zlabel('W (normalized to phase
   domain)');
view(30,30);
```

Listing A.2: MATLAB listing for Figure 2-11

```
clear
format long;
PN = -110;
f_offset = 1e6;
fosc = 5e9;
Tosc = 1/fosc;
t=2e-7;
f=linspace(-10e6,10e6,1000);
omega = 2*pi.*f;
```

```
N0 = (2*pi)*(10^(PN/10))*(f_offset)^2/(fosc^2);
W = N0/pi*((sinc(omega.*t./pi)).^2).*t.^2;
loglog(f(501:end),W(501:end),'b');
xlabel('f'),ylabel('W');
```

Listing A.3: MATLAB listing for Figure 2-12a

```
clear;
close all;
format long;
PN = -110;
f_offset = 1e6;
fosc = 5e9;
Tosc = 1/fosc;
t=logspace(-9,-4,1000);
f=logspace(4,7,1000);
omega = 2*pi.*f;
N0 = (2*pi)*(10^(PN/10))*(f_offset)^2/(fosc^2);
[t,omega] = meshgrid(t,omega);
S_t_time_domain = N0/pi./(omega.^3)./t.*(omega.*t/2-0.25*sin(2.*omega.*t)
    );
S_t_phase_domain = S_t_time_domain/(Tosc^2)*(2*pi)^2;
mesh(log10(t),log10(f),10*log10(S_t_phase_domain));
set(gca, 'XDir', 'reverse');
hold on;
x_label=xlabel('$$log_{10}\ t\ [s]$$','interpreter','latex','fontsize'
    ,14);
y_label=ylabel('$$log_{10}\ (f)\ [Hz]$$','interpreter','latex','fontsize'
    ,14);
z_label=zlabel('$$S_{t}(f)_{\Delta \Phi}\ [dBc/Hz]$$','interpreter','
    latex','rot',360,'fontsize',14);
set(z_label, 'Units', 'Normalized', 'Position', [0.1, 0.95, 0.6]);
set(y_label, 'Units', 'Normalized', 'Position', [0.3, 0, 0]);
set(x_label, 'Units', 'Normalized', 'Position', [0.8, 0.01, 0]);
view(140,20);
hold off;
```

Listing A.4: MATLAB listing for Figure 2-13

```
clear
format long;
PN = -110;
f_offset = 1e6;
fosc = 5e9;
Tosc = 1/fosc;
N0 = (2*pi)*(10^(PN/10))*(f_offset)^2/(fosc^2);
syms t omega;
W =N0/pi*((sinc(omega.*t./pi)).^2).*t.^2;
t_accum_length = [40e-9 10e-6 20e-6 100e-6];
f=logspace(3,8,1000);
omega = 2*pi.*f;
W_int = zeros(4,length(f));
color = ['m','r','g','b'];
for i =1:4
```

```
W_syms = int(W,t,0,t_accum_length(i))/t_accum_length(i);
W_int(i,:) = eval(W_syms);%power of jitter in 's^2'
W_int_phase(i,:) = W_int(i,:)/(Tosc^2)*((2*pi)^2);% power of phase in '(
    rad/s)^2'
end
figure(1);
for i =2:4
semilogx(f, 10*log10(W_int_phase(i,:)),color(i));
hold all;
end
xlabel('f (Hz)');
ylabel('S(f)_{\Delta \Phi} (dBc/Hz) ');
title('DCPLL phase noise when T_{1} = \{10 \mus, 20 \mus, 100 \mus\}');
legend('T_{1} = 10 \mus', 'T_{1} = 20 \mus', 'T_{1} = 100 \mus');
hold off;

figure(2);
semilogx(f, 10*log10(W_int_phase(1,:)),color(1));
hold on;
xlabel('f (Hz)');
ylabel('S(f)_{\Delta \Phi} (dBc/Hz) ');
title('DCPLL phase noise when T_{1} = 40 ns');
legend('T_{1} = 40 ns')
hold off;
```

Appendix B

Design Hierarchy

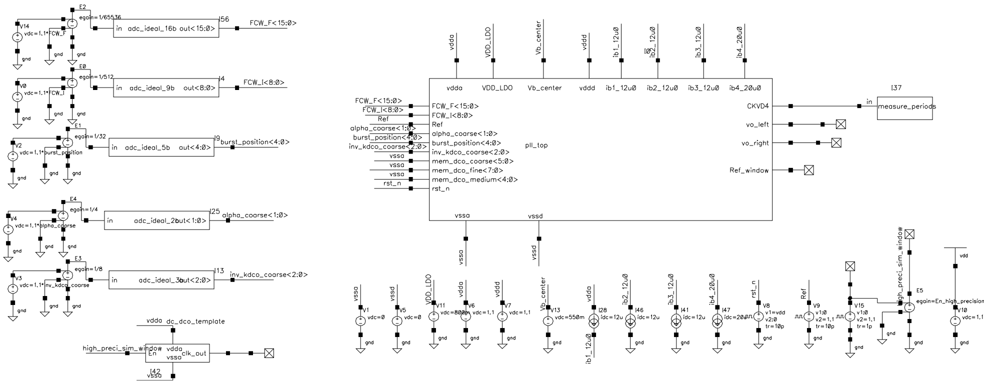


Figure B-1: Testbench connections for the DC-ADPLL

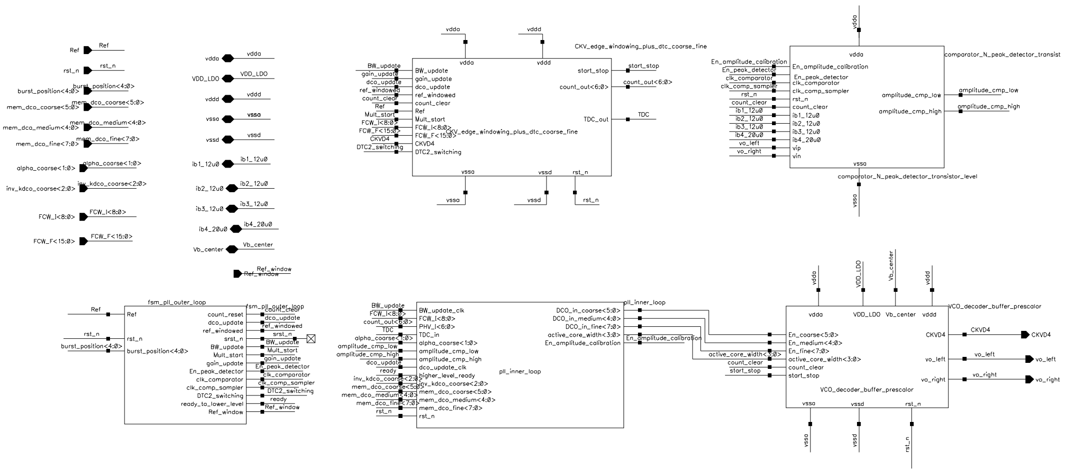


Figure B-2: Schematic of the 'pll_top' block in Figure B-1

B-1 CKV Edge-sampling and DTC Gain Calibration Block

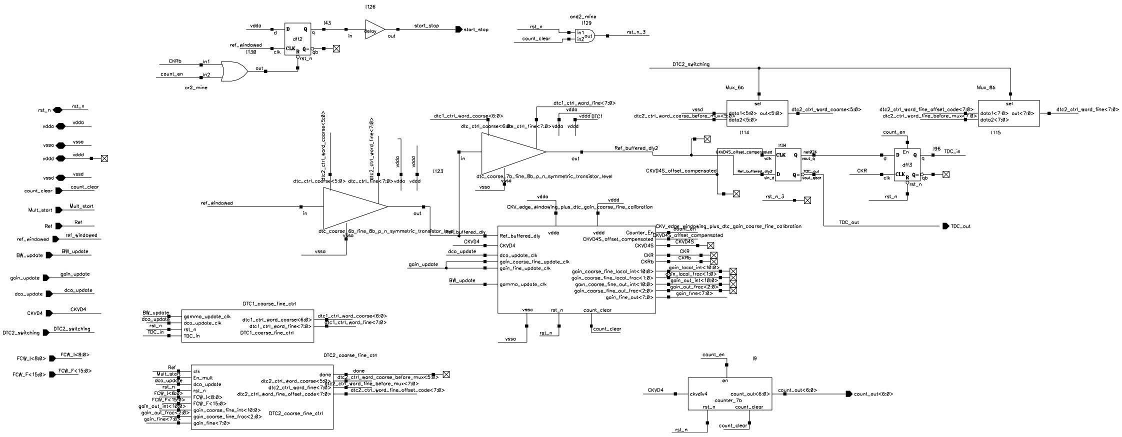


Figure B-3: Schematic of the 'CKV_edge_windowing_plus_dtc_coarse_fine' in Figure B-2

B-1-1 CKV Edge-sampling Circuitry and DTC3

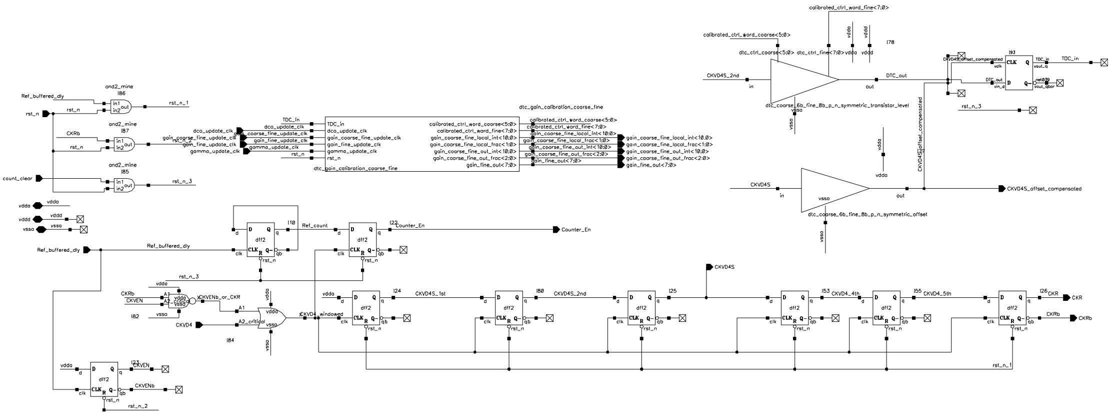

calibrated_ctrl_word_fine<7:0>
calibrated_ctrl_word_coarse<5:0>
I78
CKVD4S_2nd
DTC_out
I83
TDC_in
CKVD4S_offset_compensated
rst_n_3
dtc_coarse_6b_fine_8b_p_n_symmetric_transistor_level
dtc_coarse_6b_fine_8b_p_n_symmetric_offset
CKVD4S
Ref_buffered_dly
and2_mine
I86
I87
I85
rst_n_1
rst_n_3
rst_n
CKRb
count_clear
dtc_gain_calibration_coarse_fine
TDC_in
dco_update_clk
gain_coarse_fine_update_clk
gain_fine_update_clk
gamma_update_clk
gain_coarse_fine_local_int<10:0>
gain_coarse_fine_local_frac<1:0>
gain_coarse_fine_out_int<10:0>
gain_coarse_fine_out_frac<2:0>
gain_fine_out<7:0>
vdda
vddd
vssa
I10
Ref_count
I22
Counter_En
dff2
CKVENb_or_CKR
I82
I84
CKVD4_windowed
I24
CKVD4S_1st
I80
CKVD4S_2nd
I25
I53
CKVD4_4th
I55
CKVD4_5th
I26
CKR
CKRb
I23
CKVEN
CKVENb
rst_n_2
CKVD4

Figure B-4: Schematic of the 'CKV_edge_windowing_plus_dtc_gain_coarse_fine_calibration' block in Figure B-3

DTC Gain Calibration Control Block

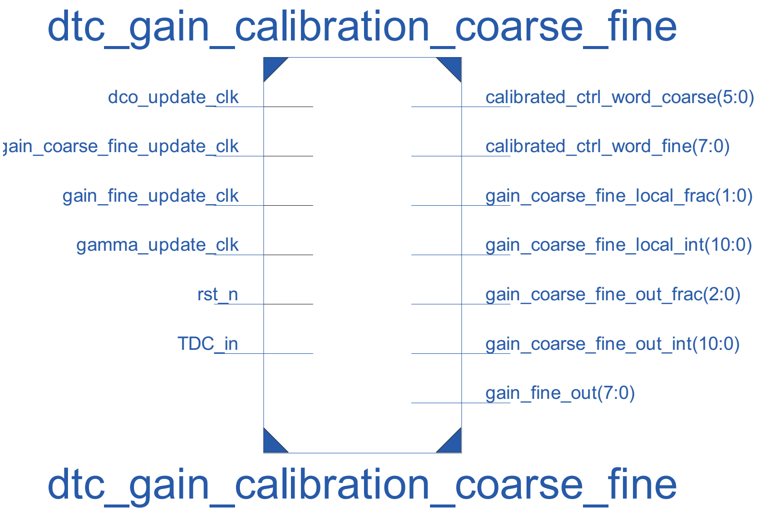

dtc_gain_calibration_coarse_fine
dco_update_clk
gain_coarse_fine_update_clk
gain_fine_update_clk
gamma_update_clk
rst_n
TDC_in
calibrated_ctrl_word_coarse(5:0)
calibrated_ctrl_word_fine(7:0)
gain_coarse_fine_local_frac(1:0)
gain_coarse_fine_local_int(10:0)
gain_coarse_fine_out_frac(2:0)
gain_coarse_fine_out_int(10:0)
gain_fine_out(7:0)
dtc_gain_calibration_coarse_fine

Figure B-5: Schematic of the 'dtc_gain_calibration_coarse_fine' block in Figure B-4

Table B-1: Port list for the 'dtc_gain_calibration_coarse_fine' block

Port name	I/O	Comment
dco_update_clk	I	Clock for the register of updating auxiliary DTC control word
gain_coarse_fine_update_clk	I	Clock for updating the gain of auxiliary coarse DTC
gain_fine_update_clk	I	Clock for updating the gain of auxiliary fine DTC
gamma_update_clk	I	Clock for updating the step size of correlator
rst_n	I	Asynchronous globla reset signal from off-chip
TDC_in	I	Single-bit result from TDC
calibrated_ctrl_cord_coarse[5:0]	O	Calibrated control word of auxiliary coarse DTC
calibrated_ctrl_word_fine[7:0]	O	Calibrated control word of auxiliary fine DTC
gain_coarse_fine_local_frac[1:0]	O	Fractional part of calibrated auxiliary gain of coarse DTC
gain_coarse_fine_local_int[10:0]	O	Integer part of calibrated gain of auxiliary coarse DTC
gain_coarse_fine_out_frac[2:0]	O	Fractional part of exported coarse DTC gain
gain_coarse_fine_out_int[10:0]	O	Integer part of exported coarse DTC gain
gain_fine_out[7:0]	O	Fractinal and integer part of exported fine DTC gain

B-2 DCO, Buffer and Divider Block

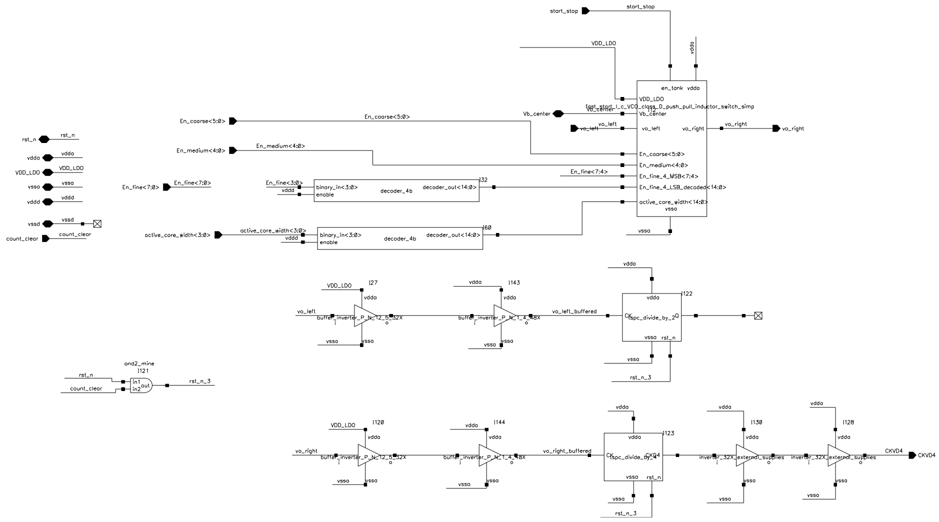


Figure B-6: Schematic of the ‘VCO_decoder_buffer_prescalor’ block in Figure B-2

B-2-1 DCO

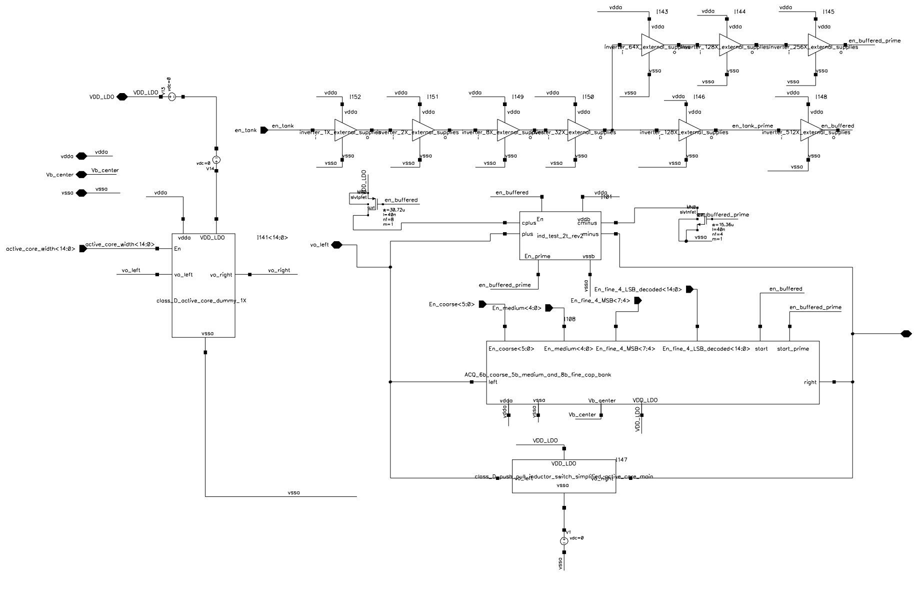

VDD_LDO
V13
vdc=0
V14
vdda
Vb_center
vssa
en_tank
I152
I151
I149
I150
I143
I144
I145
I146
I148
inverter_1X_external_supplies
inverter_2X_external_supplies
inverter_8X_external_supplies
inverter_32X_external_supplies
inverter_64X_external_supplies
inverter_128X_external_supplies
inverter_256X_external_supplies
inverter_512X_external_supplies
en_tank_prime
en_buffered
en_buffered_prime
MP0
slvtpfet
w=30.72u
l=40n
nf=8
m=1
MN0
slvtnfet
w=15.36u
l=40n
nf=4
m=1
I101
cplus
plus
En
vddb
cminus
minus
ind_test_2t_rev2
En_prime
vssb
active_core_width<14:0>
I141<14:0>
vo_left
vo_right
class_D_active_core_dummy_1X
En_coarse<5:0>
En_medium<4:0>
En_fine_4_MSB<7:4>
En_fine_4_LSB_decoded<14:0>
start
start_prime
I108
ACQ_6b_coarse_5b_medium_and_8b_fine_cap_bank
left
right
I147
class_D_push_pull_inductor_switch_simplified_active_core_main
V1
vdc=0

Figure B-7: Schematic of the ‘fast_start_l_c_class_D_push-pull_inductor_switch_simp’ [1]block in Figure B-6

Main active core

[1] In the early design phase, the voltage-biased Class B oscillator structure was incorrectly categorized into Class D. It was until later design stage that this mistake was discovered. To avoid naming conflicts with other modules in the library, the symbol for voltage-biased Class B oscillator is still kept its original name.

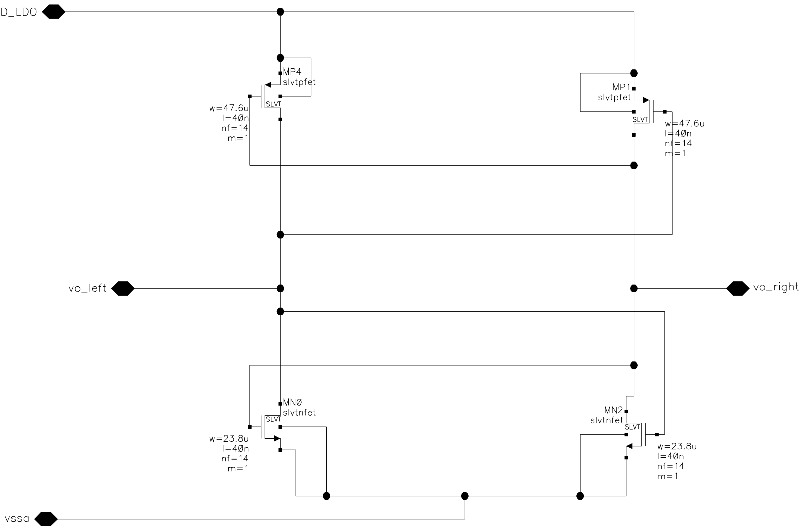


Figure B-8: Schematic of the main active core

Auxiliary active core

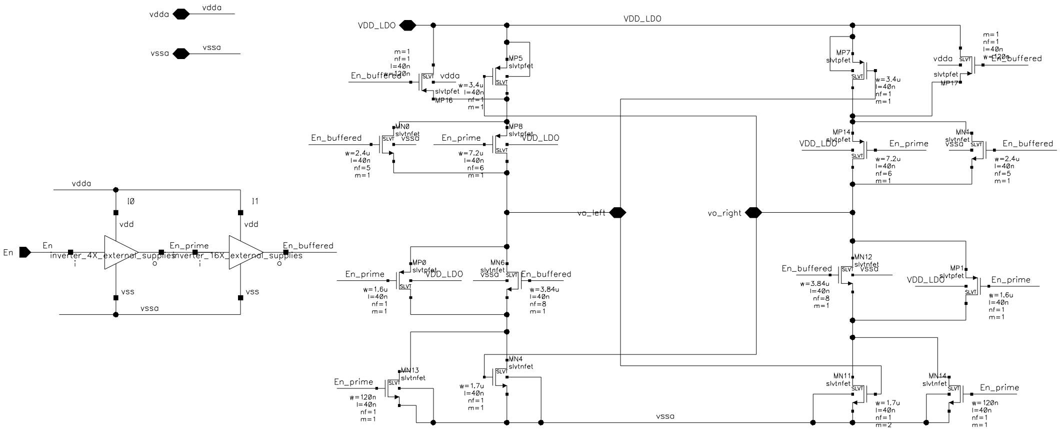


Figure B-9: Schematic of auxiliary active cores

B-3 Peak Detector and Comparator

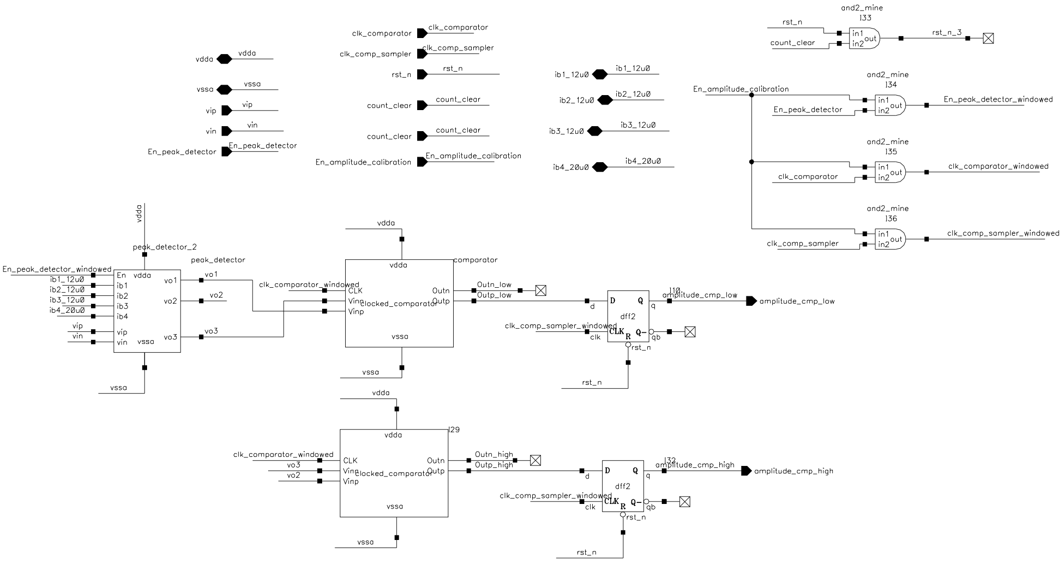


Figure B-10: Schematic of the ‘comparator_N_peak_detector_transistor_level’ block in Figure B-2

B-4 FSM for Outer-loop of the DC-ADPLL

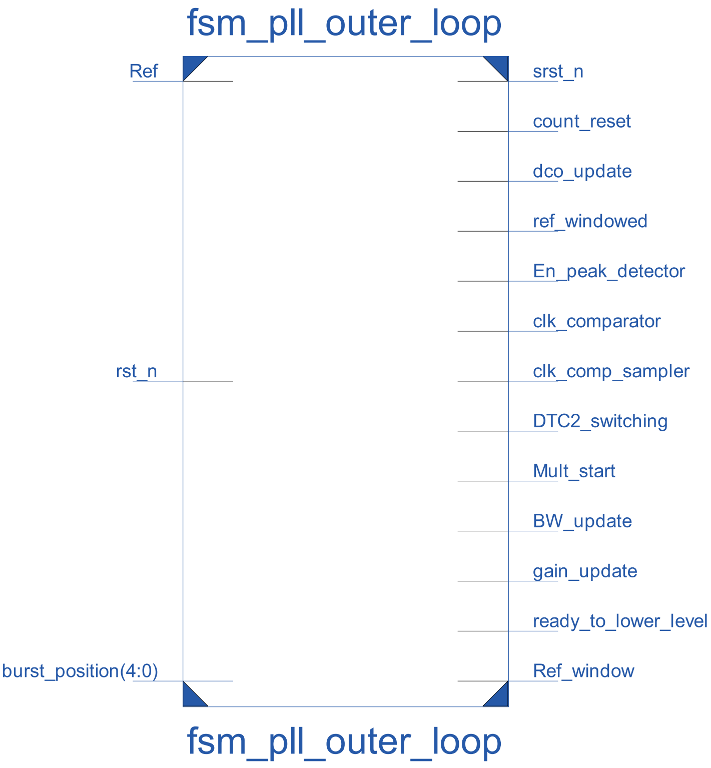


Figure B-11: Symbol of the ‘fsm_pll_outer_loop’ block in Figure B-2

Table B-2: Port list for the 'fsm_pll_outer_loop' block

Port name	I/O	Comment
Ref	I	Reference clock for FSM
rst_n	I	Asynchronous globla reset signal from off-chip
burst_position[4:0]	I	Programmable control word for burst position
srst_n	O	Synchronous reset to lower-level digital blocks
count_reset	O	Reset signal for RF counter
dco_update	O	Update clock for updating capacitor bank and DTC
ref_windowed	O	Reference clock windowed at burst position
En_peak_detector	O	Enable signal for peak detector
clk_comparator	O	Clock for comparator
clk_comp_sampler	O	Clock for the sampler of comparator output
DTC2_switching	O	Signal for switching the control word of DTC2
Mult_start	O	Signal to start the sequential multiplier
BW_update	O	Clock for updating the bandwidth of locking transient
gain_update	O	Clock for update DTC gain
ready_to_lower_level	O	Signal to lower-level blocks indicating higher-level ready
Ref_window	O	Signal to window the input reference clock

B-5 Inner-loop Control for the DC-ADPLL

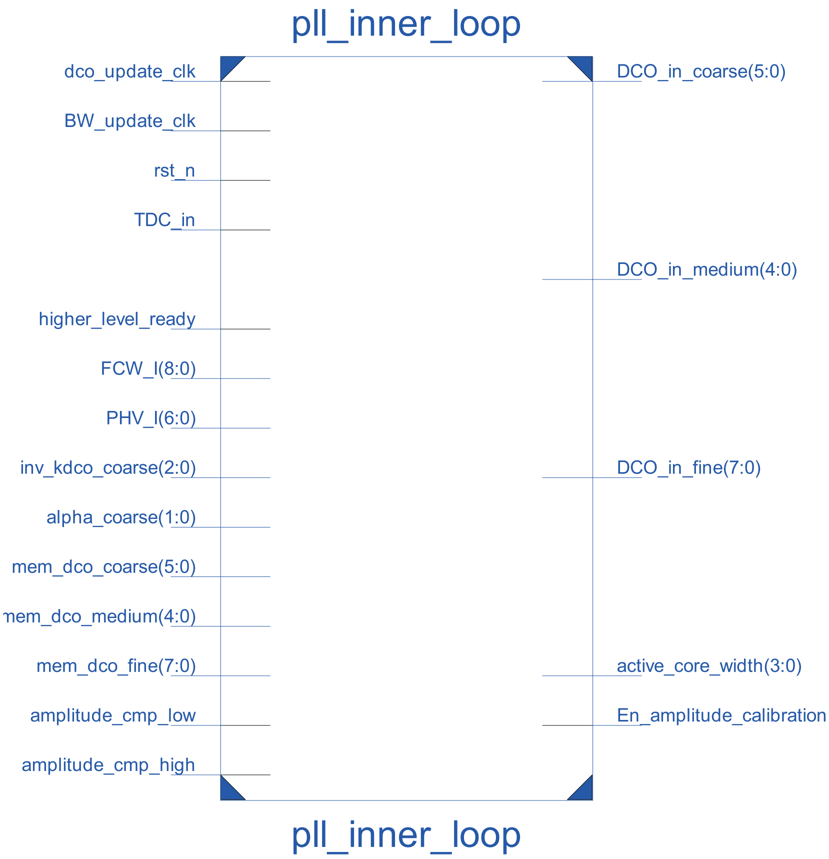


Figure B-12: Symbol of the ‘pll_iner_loop’ block in Figure B-2

Table B-3: Port list for the 'pll_iner_loop' block

Port name	I/O	Comment
dco_update_clk	I	Clock for updating control words of capacitor banks
BW_update_clk	I	Clock for updating the bandwidth of locking transient
rst_n	I	Asynchronous globla reset signal from off-chip
TDC_in	I	Single-bit result from TDC
higher_level_ready	I	Ready signal from higher level
FCW_I[8:0]	I	Integer part of Frequency Control Word
PHV_I[8:0]	I	Integer part of RF clock counter
inv_kdco_coarse[2:0]	I	Gain normalization factor for coarse capacitor bank
alpha_coarse[1:0]	I	Coefficient of 1st-order loop filter for coarse bank
mem_dco_coarse[5:0]	I	Programmable control word to set initial bias of coarse capacitor bank
mem_dco_medium[4:0]	I	Programmable control word to set initial bias of medium capacitor bank
mem_dco_fine[7:0]	I	Programmable control word to set initial bias of fine capacitor bank
amplitude_cmp_low	I	Quantized result of comparing DCO amplitude with the lower threshold
amplitude_cmp_high	I	Quantized result of comparing DCO amplitude with the higher threshold
DCO_in_coarse[5:0]	O	Control word for coarse capacitor bank
DCO_in_coarse[4:0]	O	Control word for medium capacitor bank
DCO_in_fine[7:0]	O	Control word for fine capacitor bank
active_core_width[3:0]	O	Control word to set the number of enabled active core branch
En_amplitude_calibration	O	Enable signal for amplitude calibration

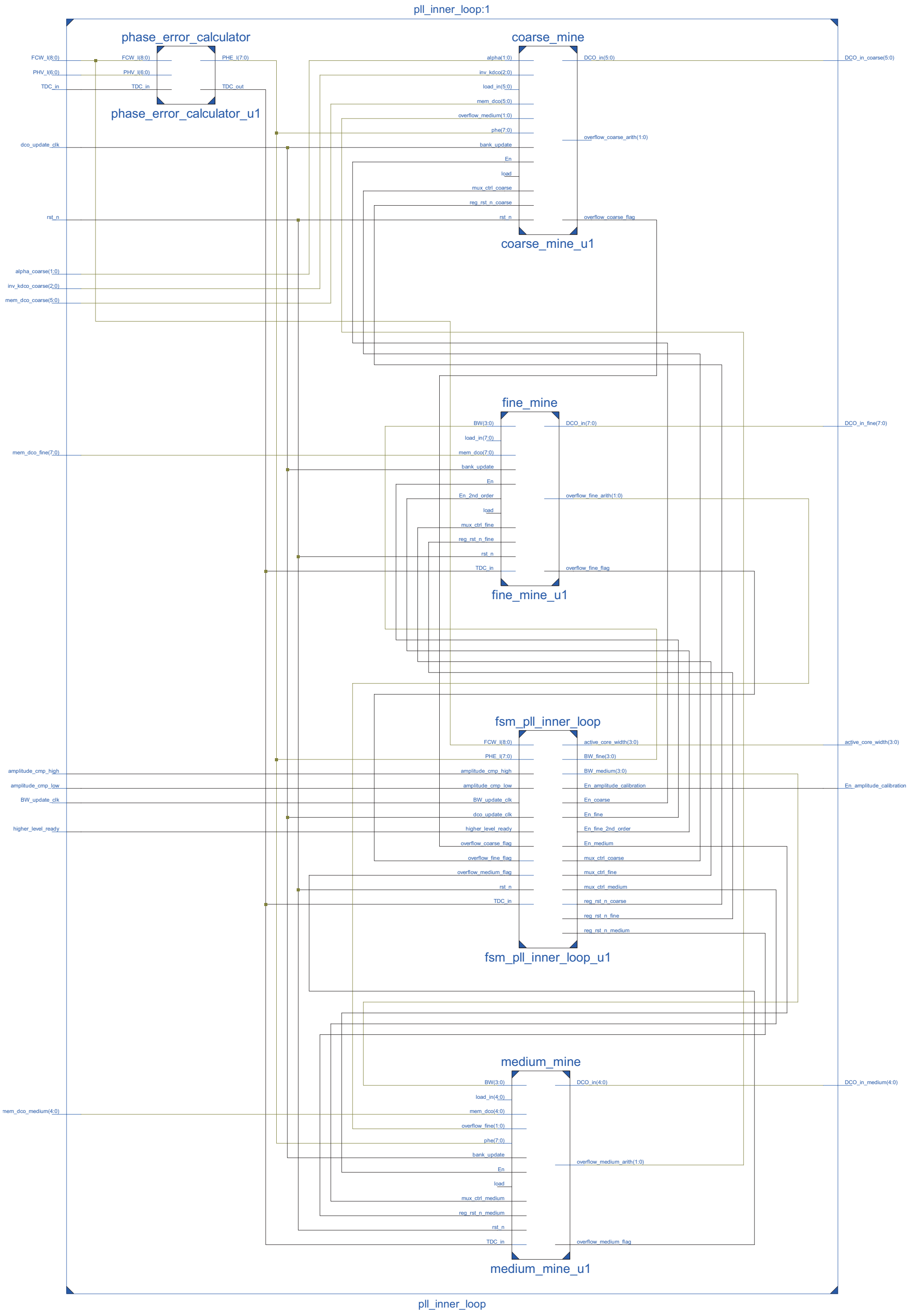


Figure B-13: Scematic of the ‘pll_inner_loop’ block in Figure B-2

Bibliography

[1] M Welborn, “Frequency Hoppers and FCC UWB Rules, IEEE P802. 15 Wireless Personal Area Networks,” *IEEE P802. 15-03/27lr2*, 2003.

[2] I. D. O’Donnell, “A baseband, impulse ultra-wideband transceiver front-end for low power applications,” PhD thesis, Dept. Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, 2006.

[3] E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, “A 700-kHz bandwidth $\Sigma\Delta$ fractional synthesizer with spurs compensation and linearization techniques for WCDMA applications,” *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 9, pp. 1446–1454, Sep. 2004, ISSN: 0018-9200. DOI: 10.1109/JSSC.2004.831598.

[4] “Digital Fractional-N Frequency Synthesis,” in *Advances in Analog and RF IC Design for Wireless Communication Systems*, L. Salvatore and S. Carlo, Eds., Academic Press, May 13, 2013, ISBN: 9780123983329.

[5] A. Kral, F. Behbahani, and A. Abidi, “RF-CMOS oscillators with switched tuning,” in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, May 1998, pp. 555–558. DOI: 10.1109/CICC.1998.695039.

[6] M. Zanuso, S. Levantino, C. Samori, and A. Lacaita, “A Wideband 3.6 GHz Digital $\Delta\Sigma$ Fractional-N PLL With Phase Interpolation Divider and Digital Spur Cancellation,” *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 3, pp. 627–638, Mar. 2011, ISSN: 0018-9200. DOI: 10.1109/JSSC.2010.2104270.

[7] R. Staszewski, C.-M. Hung, K. Maggio, J. Wallberg, D. Leipold, and P. Balsara, “All-digital phase-domain TX frequency synthesizer for Bluetooth radios in 0.13 mu;m CMOS,” in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, 2004, 272–527 Vol.1. DOI: 10.1109/ISSCC.2004.1332699.

[8] J. Phillips and K. Kundert, “An introduction to cyclostationary noise,” The Designer’s Guide Community, Tech. Rep., 2007.

[9] J. Ryckaert, G. Van der Plas, V. De Heyn, C. Desset, B. Van Poucke, and J. Craninckx, "A 0.65-to-1.4 nJ/Burst 3-to-10 GHz UWB All-Digital TX in 90 nm CMOS for IEEE 802.15.4a," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 12, pp. 2860–2869, Dec. 2007, ISSN: 0018-9200. DOI: 10.1109/JSSC.2007.908665.

[10] X. Wang, B. Busze, J. Romme, R. M. Vinella, C. Zhou, K. Philips, and H. de Groot, "A multi-GHz 130ppm accuracy FLL for duty-cycled systems," in *2011 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, Jun. 2011, pp. 1–4. DOI: 10.1109/RFIC.2011.5946283.

[11] S. Drago, D. Leenaerts, B. Nauta, F. Sebastiano, K. Makinwa, and L. Breems, "A 200 μA duty-cycled PLL for wireless sensor nodes in 65 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 7, pp. 1305–1315, Jul. 2010, ISSN: 0018-9200. DOI: 10.1109/JSSC.2010.2049458.

[12] B. Kim, T. Weigandt, and P. Gray, "PLL/DLL system noise analysis for low jitter clock synthesizer design," in *, 1994 IEEE International Symposium on Circuits and Systems, 1994. ISCAS '94*, vol. 4, May 1994, 31–34 vol.4. DOI: 10.1109/ISCAS.1994.409189.

[13] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*. Boston: Houghton Mifflin Company, 1972, 203 pp., ISBN: 9780395120767.

[14] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th edition. Boston: McGraw-Hill Europe, 2002, 852 pp., ISBN: 9780071226615.

[15] N. Kasdin, "Discrete simulation of colored noise and stochastic processes and 1/f $^{\alpha}$ power law noise generation," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 802–827, May 1995, ISSN: 0018-9219. DOI: 10.1109/5.381848.

[16] A. V. Oppenheim, A. S. Willsky, and w. S. Hamid, *Signals and Systems*, 2 edition. Upper Saddle River, N.J: Prentice Hall, 1996, 957 pp., ISBN: 9780138147570.

[17] Wikipedia, *Normal distribution — wikipedia, the free encyclopedia*, [Online; accessed 6-December-2014], 2014.

[18] A. D. Berny, R. G. Meyer, and A. Niknejad, "Analysis and design of wideband LC VCOs," PhD thesis, University of California, Berkeley, 2006.

[19] B. Razavi, *RF Microelectronics*, 2 edition. Upper Saddle River, NJ: Prentice Hall, 2011, 960 pp., ISBN: 9780137134731.

[20] R. Staszewski, C. Fernando, and P. Balsara, "Event-driven Simulation and modeling of phase noise of an RF oscillator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 723–733, Apr. 2005, ISSN: 1549-8328. DOI: 10.1109/TCSI.2005.844236.

[21] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: a unifying theory and numerical methods for characterization," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 655–674, May 2000, ISSN: 1057-7122. DOI: 10.1109/81.847872.

[22] A. Hajimiri, S. Limotyrakis, and T. Lee, "Jitter and phase noise in ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 6, pp. 790–804, Jun. 1999, ISSN: 0018-9200. DOI: 10.1109/4.766813.

[23] A. Zanchi, A. Bonfanti, S. Levantino, and C. Samori, "General SSCR vs. cycle-to-cycle jitter relationship with application to the phase noise in PLL," in *2001 Southwest Symposium on Mixed-Signal Design, 2001. SSMSD*, 2001, pp. 32–37. DOI: 10.1109/SSMSD.2001.914933.

[24] R. Poore, "Phase noise and jitter," *Agilent EEs of EDA*, 2001.

[25] H. Schmid, "Offset, flicker noise, and ways to deal with them," *Power*, vol. 10, p. 3, 2008.

[26] A. Hajimiri, "Noise in phase-locked loops [invited]," *Proceedings of the SSMSD*, vol. 1, pp. 1–6, 2001.

[27] J. McNeill, "Jitter in ring oscillators," *Solid-State Circuits, IEEE Journal of*, vol. 32, no. 6, pp. 870–879, Jun. 1997, ISSN: 0018-9200. DOI: 10.1109/4.585289.

[28] C. H. Page, "Instantaneous Power Spectra," *Journal of Applied Physics*, vol. 23, no. 1, pp. 103–106, Jan. 1952, ISSN: 0021-8979. DOI: 10.1063/1.1701949.

[29] Wikipedia, *Parseval's theorem — wikipedia, the free encyclopedia*, [Online; accessed 6-December-2014], 2014.

[30] S. Marshall, "Nonstationary stochastic differential equations," in *Advances in nonlinear signal and image processing*, G. Lorenzo and C. Leon, Eds., New York, NY: Hindawi Publishing Corporation, 2006, p. 361, ISBN: 9789775945372.

[31] Wikipedia, *Wigner distribution function — wikipedia, the free encyclopedia*, [Online; accessed 7-December-2014], 2014.

[32] W. D. Mark, "Spectral analysis of the convolution and filtering of non-stationary stochastic processes," *Journal of Sound and Vibration*, vol. 11, no. 1, pp. 19–63, Jan. 1970, ISSN: 0022-460X. DOI: 10.1016/S0022-460X(70)80106-7.

[33] Wikipedia, *Squeeze theorem — wikipedia, the free encyclopedia*, [Online; accessed 7-December-2014], 2014.

[34] G. Marucci, A. Fenaroli, G. Marzin, S. Levantino, C. Samori, and A. Lacaita, "A 1.7GHz MDLL-based fractional-N frequency synthesizer with 1.4ps RMS integrated jitter and 3mW power using a 1b TDC," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb. 2014, pp. 360–361. DOI: 10.1109/ISSCC.2014.6757469.

[35] P. Finnstam and M. Söderlund, "Design of CMOS ring oscillator," Master's thesis, Department of Signals and Systems, Chalmers University of Technology, Götebort, Swedan, 2005.

[36] M. T. Hsieh and G. E. Sobelman, "Comparison of LC and Ring VCOs for PLLs in a 90 nm Digital CMOS Process," in *Proceedings, international SoC design conference*, 2006, pp. 19–22.

[37] D. Barras, F. Ellinger, H. Jackel, and W. Hirt, "Low-power ultra-wideband wavelets generator with fast start-up circuit," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 5, pp. 2138–2145, May 2006, ISSN: 0018-9480. DOI: 10.1109/TMTT.2006.873631.

[38] J. Kim and M. Green, "Fast startup of LC VCOs using circuit asymmetries," in *2011 20th European Conference on Circuit Theory and Design (ECCTD)*, 2011, pp. 69–72. DOI: 10.1109/ECCTD.2011.6043611.

[39] M. Taghivand, K. Aggarwal, and A. Poon, “A 3.24-to-8.45GHz low-phase-noise mode-switching oscillator,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb. 2014, pp. 368–369. DOI: 10.1109/ISSCC.2014.6757473.

[40] L. Fanori, T. Mattsson, and P. Andreani, “A 2.4-to-5.3GHz dual-core CMOS VCO with concentric 8-shaped coils,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb. 2014, pp. 370–371. DOI: 10.1109/ISSCC.2014.6757474.

[41] B. Sadhu and R. Harjani, “Capacitor bank design for wide tuning range LC VCOs: 850MHz-7.1GHz (157%),” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2010, pp. 1975–1978. DOI: 10.1109/ISCAS.2010.5537040.

[42] S. Diao, Y. Zheng, and C.-H. Heng, “A CMOS Ultra Low-Power and Highly Efficient UWB-IR Transmitter for WPAN Applications,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 3, pp. 200–204, Mar. 2009, ISSN: 1549-7747. DOI: 10.1109/TCSII.2009.2015369.

[43] V. De Heyn, G. Van der Plas, J. Ryckaert, and J. Craninckx, “A Fast Start-up 3GHz-10GHz Digitally Controlled Oscillator for UWB impulse radio in 90nm CMOS,” in *Solid State Circuits Conference, 2007. ESSCIRC 2007. 33rd European*, 2007, pp. 484–487. DOI: 10.1109/ESSCIRC.2007.4430347.

[44] E. Ragonese, A. Scuderi, T. Biondi, and G. Palmisano, *Integrated Inductors and Transformers: Characterization, Design and Modeling for RF and MM-Wave Applications*, 1 edition. Boca Raton, FL: Auerbach Publications, 2010, 169 pp., ISBN: 9781420088441.

[45] Z. Li and K. O, “A 900-MHz 1.5-V CMOS voltage-controlled oscillator using switched resonators with a wide tuning range,” *IEEE Microwave and Wireless Components Letters*, vol. 13, no. 4, pp. 137–139, Apr. 2003, ISSN: 1531-1309. DOI: 10.1109/LMWC.2003.811054.

[46] O. Meike, A. Kruth, S. Sappok, S. Joeres, G. Ordu, R. Wunderlich, and S. Heinen, “Design of a reconfigurable, differentially driven symmetric inductor,” in *Proceedings ProRISC*, 2004.

[47] Z. Li and K. O, “A low-phase-noise and low-power multiband CMOS voltage-controlled oscillator,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1296–1302, Jun. 2005, ISSN: 0018-9200. DOI: 10.1109/JSSC.2005.848031.

[48] B. Catli and M. M. Hella, “A 1.94 to 2.55 GHz, 3.6 to 4.77 GHz Tunable CMOS VCO Based on Double-Tuned, Double-Driven Coupled Resonators,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 9, pp. 2463–2477, Sep. 2009, ISSN: 0018-9200. DOI: 10.1109/JSSC.2009.2023155.

[49] M. Tiebout, “A CMOS fully integrated 1 GHz and 2 GHz dual band VCO with a voltage controlled inductor,” in *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, Sep. 2002, pp. 799–802.

[50] S.-M. Yim et al., “Switched resonators and their applications in a dual-band monolithic CMOS LC-tuned VCO,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 1, pp. 74–81, Jan. 2006, ISSN: 0018-9480. DOI: 10.1109/TMTT.2005.856102.

[51] A. Mazzanti and P. Andreani, "Class-C Harmonic CMOS VCOs, With a General Result on Phase Noise," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2716–2729, Dec. 2008, ISSN: 0018-9200. DOI: 10.1109/JSSC.2008.2004867.

[52] L. Fanori and P. Andreani, "Highly Efficient Class-C CMOS VCOs, Including a Comparison With Class-B VCOs," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 7, pp. 1730–1740, Jul. 2013, ISSN: 0018-9200. DOI: 10.1109/JSSC.2013.2253402.

[53] A. Mazzanti and P. Andreani, "A Push–Pull Class-C CMOS VCO," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 724–732, Mar. 2013, ISSN: 0018-9200. DOI: 10.1109/JSSC.2012.2230542.

[54] L. Fanori and P. Andreani, "Class-D CMOS Oscillators," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 12, pp. 3105–3119, Dec. 2013, ISSN: 0018-9200. DOI: 10.1109/JSSC.2013.2271531.

[55] J. Chen, F. Jonsson, M. Carlsson, C. Hedenas, and L.-R. Zheng, "A Low Power, Startup Ensured and Constant Amplitude Class-C VCO in 0.18 CMOS," *IEEE Microwave and Wireless Components Letters*, vol. 21, no. 8, pp. 427–429, Aug. 2011, ISSN: 1531-1309. DOI: 10.1109/LMWC.2011.2160620.

[56] W. Deng, K. Okada, and A. Matsuzawa, "A feedback class-C VCO with robust startup condition over PVT variations and enhanced oscillation swing," in *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, Sep. 2011, pp. 499–502. DOI: 10.1109/ESSCIRC.2011.6044931.

[57] S. Xu, W. He, C. Shi, R. Zhang, C. Ma, and Z. Lai, "Design of a wideband and low phase noise LC VCO," in *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Nov. 2010, pp. 650–652. DOI: 10.1109/ICSICT.2010.5667296.

[58] D. Hauspie, E.-C. Park, and J. Craninckx, "Wideband VCO With Simultaneous Switching of Frequency Band, Active Core, and Varactor Size," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 7, pp. 1472–1480, Jul. 2007, ISSN: 0018-9200. DOI: 10.1109/JSSC.2007.899105.

[59] A. Momtaz, J. Cao, M. Caresosa, A. Hairapetian, D. Chung, K. Vakilian, M. Green, W.-G. Tan, K.-C. Jen, I. Fujimori, and Y. Cai, "A fully integrated SONET OC-48 transceiver in standard CMOS," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 1964–1973, Dec. 2001, ISSN: 0018-9200. DOI: 10.1109/4.972147.

[60] S. Levantino, C. Samori, A. Bonfanti, S. Gierkink, A. Lacaita, and V. Boccuzzi, "Frequency dependence on bias current in 5 GHz CMOS VCOs: impact on tuning range and flicker noise upconversion," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 8, pp. 1003–1011, Aug. 2002, ISSN: 0018-9200. DOI: 10.1109/JSSC.2002.800969.

[61] V. Chillara, Y.-H. Liu, B. Wang, A. Ba, M. Vidojkovic, K. Philips, H. de Groot, and R. Staszewski, "An 860μW 2.1-to-2.7GHz all-digital PLL-based frequency modulator with a DTC-assisted snapshot TDC for WPAN (Bluetooth Smart and ZigBee) applications," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb. 2014, pp. 172–173. DOI: 10.1109/ISSCC.2014.6757387.

[62] P. Andreani, K. Kozmin, P. Sandrup, M. Nilsson, and T. Mattsson, "A TX VCO for WCDMA/EDGE in 90 nm RF CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 7, pp. 1618–1626, Jul. 2011, ISSN: 0018-9200. DOI: 10.1109/JSSC.2011.2144030.

[63] Y. Chen, V. Neubauer, Y. Liu, U. Vollenbruch, C. Wicpalek, T. Mayer, B. Neurauter, L. Maurer, and Z. Boos, “A 9 GHz dual-mode digitally controlled oscillator for GSM/UMTS transceivers in 65 nm CMOS,” in *Solid-State Circuits Conference, 2007. ASSCC ’07. IEEE Asian*, Nov. 2007, pp. 432–435. DOI: 10.1109/ASSCC.2007.4425723.

[64] L. Fanori, T. Mattsson, and P. Andreani, “A Class-D CMOS DCO with an on-chip LDO,” in *European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014 - 40th*, Sep. 2014, pp. 335–338. DOI: 10.1109/ESSCIRC.2014.6942090.

[65] M. Babaie and R. Staszewski, “A Class-F CMOS Oscillator,” *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 12, pp. 3120–3133, Dec. 2013, ISSN: 0018-9200. DOI: 10.1109/JSSC.2013.2273823.

[66] C.-M. Hsu, M. Straayer, and M. Perrott, “A Low-Noise Wide-BW 3.6-GHz Digital Fractional-N Frequency Synthesizer With a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec. 2008, ISSN: 0018-9200. DOI: 10.1109/JSSC.2008.2005704.

[67] R. Staszewski, J. Wallberg, S. Rezeq, C.-M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, “All-digital PLL and transmitter for mobile phones,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, 2005, ISSN: 0018-9200. DOI: 10.1109/JSSC.2005.857417.

[68] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. Lacaita, “A 2.9–4.0-GHz Fractional-N Digital PLL With Bang-Bang Phase Detector and 560-Integrated Jitter at 4.5-mW Power,” *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec. 2011, ISSN: 0018-9200. DOI: 10.1109/JSSC.2011.2162917.

[69] N. Pavlovic and J. Bergervoet, “A 5.3GHz digital-to-time-converter-based fractional-N all-digital PLL,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, Feb. 2011, pp. 54–56. DOI: 10.1109/ISSCC.2011.5746216.

[70] J. Zhuang and R. Staszewski, “A low-power all-digital PLL architecture based on phase prediction,” in *2012 19th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Dec. 2012, pp. 797–800. DOI: 10.1109/ICECS.2012.6463539.

[71] R. Staszewski, J. Wallberg, S. Rezeq, C.-M. Hung, O. Eliezer, S. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, “All-digital PLL and GSM/EDGE transmitter in 90nm CMOS,” in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, Feb. 2005, 316–600 Vol. 1. DOI: 10.1109/ISSCC.2005.1493996.

[72] F.-C. Liu, Y.-X. Zhang, and Y.-J. Wang, “A variable step size LMS adaptive filtering algorithm based on the number of computing mechanisms,” in *2009 International Conference on Machine Learning and Cybernetics*, vol. 4, Jul. 2009, pp. 1904–1908. DOI: 10.1109/ICMLC.2009.5212187.

[73] R. Harris, D. Chabries, and F. Bishop, “A variable step (VS) adaptive filter algorithm,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 2, pp. 309–316, Apr. 1986, ISSN: 0096-3518. DOI: 10.1109/TASSP.1986.1164814.

[74] J. E. Stine, *Digital Computer Arithmetic Datapath Design Using Verilog HDL*, 2004 edition. Boston: Springer, 2003, 181 pp., ISBN: 9781402077104.

[75] T. Storey. (2013). Booth Radix-4 Multiplier for Low Density PLD Applications (Verilog), [Online]. Available: https://eewiki.net/pages/viewpage.action?pageId=13599773 (visited on 12/12/2014).

[76] P. Andreani and S. Mattisson, "On the use of MOS varactors in RF VCOs," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 6, pp. 905–910, Jun. 2000, ISSN: 0018-9200. DOI: 10.1109/4.845194.

Glossary

List of Acronyms

ADC	Analog-to-Digital Converter
ADPLL	All-Digital PLL
ADS	Advanced Design System
AMS	Analog-Mixed-Signal
CML	Current Mode Logic
CMRR	Common-Mode Rejection Ratio
CPLL	Continuous-operation PLL
DAC	Digital-to-Analog Converter
DC-ADPLL	Duty-Cycled All-Digital PLL
DCO	Digitally Controlled Oscillator
DCPLL	Duty-Cycled PLL
DFF	Digital Flip Flop
DTC	Digital-to-Time Converter
FCC	Federal Communications Commission
FSM	Finite State Machine
IR-UWB	Impulse Radio Ultra Wideband
MOS	Metal-Oxide-Semiconductor
PLL	Phase Locked Loop
PRF	Pulse Repetition Period

PSS	Periodic Steady State Analysis
RHP	Right-Half Plane
SRF	Self-Resonant Frequency
TDC	Time-to-Digital Converter
TSPC	True Single-Phase Clocking
UWB	Ultra-Wideband
VCO	Voltage-Controlled Oscillator