

**Analysis and Design of a
Ring-Oscillator-Based Fractional-N
Injection-Locked Digital PLL for IoT
Applications**

Analysis and Design of a Ring-Oscillator-Based Fractional-N Injection-Locked Digital PLL for IoT Applications

Master of Science Thesis

For the degree of Master of Science in Electrical Engineering at Delft University of
Technology

Jiang Gong

November 08, 2017

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS),
Delft University of Technology

Promoter: Prof. Dr. L. C. N. de Vreede

Academic Supervisor: Dr. Masoud Babaie

Industry Supervisors: Ir. Ao Ba
Ir. Yuming He

Committee Members: Prof. Dr. L. C. N. de Vreede
Dr. Fabio Sebastiano
Dr. Masoud Babaie
Ir. Ao Ba
Ir. Yuming He



Contents

1	Introduction	1
1.1	Synthesizer Requirements	1
1.1.1	Frequency Accuracy and Tuning Range	2
1.1.2	Phase Noise	2
1.1.3	Spurious Tones	6
1.1.4	Switching Time	6
1.2	Motivation and Target Specifications	6
1.3	Research Contributions	7
1.4	Thesis Outline	8
	References	9
2	Ring-Oscillator-Based Fractional-N Injection-Locked DPLL	11
2.1	Ring Oscillator or LC Oscillator	11
2.2	Injection-Locked Oscillator	13
2.2.1	Phase Noise Improvement	14
2.2.2	Reference Spurs Degradation	16
2.2.3	Injection Strength Degradation	16
2.3	Injection-Locked DPLL	18
2.3.1	Reference Spurs Degradation	18
2.3.2	Inherent Integer-N Operation	19
2.3.3	Frequency Tracking Effectiveness	19
2.4	Proposed Architecture	21
	References	25
3	System-Level Modeling and Simulation in MATLAB	27
3.1	MATLAB Modeling of DPLL Sub-blocks	27
3.1.1	Reference Phase and Timestamps	27
3.1.2	Coarse-Fine DTC	29
3.1.3	TDC, Loop Filter and Phase Detection	30
3.1.4	Injection-Locked DCO	32
3.1.5	Time Offset Calibration	33
3.2	System-Level Simulation in MATLAB	34
3.2.1	Locking Behavior	34
3.2.2	Time Offset Calibration	35
3.2.3	Phase Noise and Spectrum	36
3.3	S-Domain Phase Noise Model	37
3.4	Summary	37
	References	39

4	Analog/RF Design	41
4.1	Digitally-Controlled Ring Oscillator	41
4.1.1	Oscillator Core	41
4.1.2	Oscillator Frequency Tuning and Injection Locking	44
4.1.3	Post-Layout Simulation Results	45
4.2	Coarse-fine DTC	48
4.2.1	CF-DTC Architecture	48
4.2.2	CF-DTC Design	49
4.2.3	Post-Layout Simulation Results	50
4.3	Coarse-fine DCDL and Fixed Delay	52
4.3.1	CF-DCDL and Fixed Delay Design	52
4.3.2	Post-Layout Simulation Results	53
4.4	Pulse Generator	56
4.4.1	Pulse Generator Design	56
4.4.2	Post-Layout Simulation Results	56
4.5	1b TDC	58
4.5.1	1b TDC Architecture	58
4.5.2	Post-Layout Simulation Results	60
4.6	Other Blocks and Top Layout	62
	References	63
5	RTL Design and Simulation	65
5.1	RTL Design	65
5.1.1	Finite State Machine	67
5.1.2	Phase Detection	68
5.1.3	PVT_Norm and ACQ_Norm	68
5.1.4	TRK_Norm	69
5.1.5	CF-DTC Control	70
5.1.6	Time Offset Calibration and Injection	71
5.2	System Simulation in Verilog	72
6	Measurement Results	73
6.1	IL-DPLL Test Setup	73
6.2	Measurement Results	75
6.2.1	Frequency Tuning	75
6.2.2	Phase Noise	75
6.2.3	Spectral Purity	78
6.2.4	Power Breakdown	79
6.3	Performance Summary and Comparison Table	80
7	Conclusion	81
7.1	Thesis Conclusion	81
7.2	Future Work	82
7.2.1	Tracking Bank Range	82
7.2.2	Area Optimization	83
7.2.3	Reference Spur Mitigation	83

Acknowledgements	85
A Chip Pinout	87
B Phase Noise Measurements Results	91
C MATLAB Time-Domain Code	93

Abstract

Frequency translation is required in any modern wireless communication systems. This is in large part due to the fact that the modulated signal is easy to transmit in radio frequency in the form of an electromagnetic wave, and the demodulated signal is easy to process in the baseband frequency by a powerful digital processor. Frequency synthesizers are required during this frequency translation process. Though the upcoming commercial communication (e.g., 5G) is continuing to propel the semiconductor market, the Internet of things (IoT) aimed at health monitoring, intelligent agriculture and environmental sensing, home automation and security sensing is gaining more and more momentum in recent years. It requires ultra-low computing power and very low-cost hardware, which challenges RF engineers to design low power and small-area wireless transceivers. Frequency synthesizers typically take up considerable silicon area and are one of most power-hungry blocks of these transceivers.

This thesis aims to design a clock generation digital phase-locked loop (DPLL) for the Bluetooth Low Energy (BLE) standard for IoT applications. The DPLL should have a small area and low power consumption. Hence, a ring-oscillator (RO)-based fractional-N DPLL is implemented to generate the desired clock. A phase noise improvement technique is proposed to reduce the in-band phase noise of the DPLL by around 6dB. Furthermore, a fast reference calibration loop is implemented to mitigate the reference spur effectively. A prototype is fabricated in the TSMC LP 40nm CMOS process. Measurements show that the proposed RO-based fractional-N DPLL achieves 1.6ps integrated jitter, -45.8dBc fractional spur, -43.6dBc reference spur and 1.8-2.7GHz tuning range while consuming only 1.33mW power. The resulting figure of merit (FOM) of the implemented DPLL is -234.7dB, which is the best compared with the state-of-the-art RO-based fractional-N PLLs.

1

Introduction

The general merits of a frequency synthesizer are introduced in the first section of this chapter. In the meantime, one of the most important Internet of things (IoT) standards (i.e., Bluetooth Low Energy (BLE)) is discussed and employed to derive the synthesizer specifications. Then, target specifications for the BLE synthesizer are presented in the second section. The research contribution is given in the third section, and the thesis outline is presented in the last section of this chapter.

1.1 Synthesizer Requirements

A frequency synthesizer generates the desired frequency (f_{out}) based on the user-specified frequency command word (FCW) and the reference frequency (f_{ref}). The block diagram of a frequency synthesizer is shown in Figure 1.1. In general, a frequency synthesizer is a frequency multiplier where

$$f_{out} = FCW \cdot f_{ref}. \quad (1.1)$$

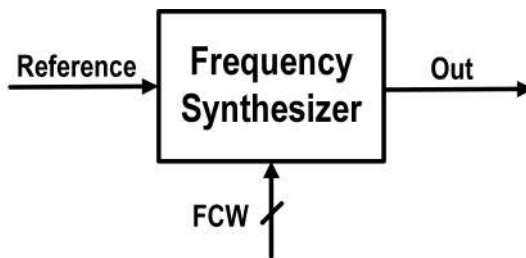


Figure 1.1: Block diagram of a frequency synthesizer.

There are many parameters such as frequency accuracy, tuning range, phase noise, spurious tones, and switching time that should be considered during the

design of frequency synthesizers. The requirements of the BLE standard [1] used to derive the synthesizer specifications are summarized in Table 1.1.

Table 1.1: The requirements of the BLE standard.

Parameters	Requirements
RF Channels	$f = 2402 + 2k$ MHz $k = 0, 1, \dots, 39$
Receiver Sensitivity	-70dBm (in 1MHz bandwidth)
Signal-to-noise Ratio (SNR)	11dB
Carrier-to-interference Ratio (CIR): Channel (2MHz) Channels (≥ 3 MHz)	-17dB -27dB
Spurious Emissions: Adjacent Channel (2MHz) Adjacent Channel (≥ 3 MHz)	-20dBm -30dBm
Carrier Frequency Tolerance	± 150 kHz
Longest Package Length	376 μ s

1.1.1 Frequency Accuracy and Tuning Range

The channel bandwidth for the BLE standard is 2MHz, leading to 40 different channels in the operating frequency band of 2.402 to 2.480GHz (ISM Band). However, the frequency of a free-running ring oscillator can drift up to tens of MHz due to the Process-Voltage-Temperature (PVT) variations and the flicker noise of its transistors. Consequently, a golden stable reference source (with ppm range accuracy) is needed to make the oscillator phase-locked to the reference and achieve the desired frequency with adequate precision. This reference source can be a crystal oscillator, a temperature-controlled crystal oscillator (TCXO) or a digitally-controlled crystal oscillator depending on the application. Furthermore, the synthesizer has to cover the desired frequency range with enough margin to tolerate PVT variations.

1.1.2 Phase Noise

Phase noise by IEEE definition [2] is the power spectral density (PSD) of the phase. To illustrate this definition, the output voltage waveform of a synthesizer is shown in Figure 1.2. The output clock transition timestamps are denoted as $tckv[n]$. Due to the system noise, $tckv[n]$ is a random process, which is collected and put it into a vector as:

$$\{tckv\} = \{tckv[1], tckv[2], \dots, tckv[n]\}. \quad (1.2)$$

The phase noise of the synthesizer then can be calculated as:

$$PN(\Delta f) = PSD(2 \cdot \pi \cdot f_{out} \cdot \{tckv\}). \tag{1.3}$$

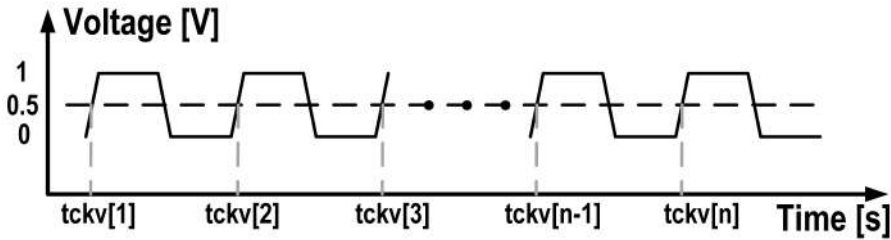


Figure 1.2: Output voltage waveform of a frequency synthesizer.

$PN(\Delta f)$ has a significant impact on the transceiver performance. On the receiver side, reciprocal mixing occurs where a large blocker is mixing with a noisy local oscillator and shows itself at the top of the desired channel, degrading receiver sensitivity. This phenomenon is illustrated in Figure 1.3. Based on Table 1.1, the carrier-to-interference ratio can be as low as -17dB and -27dB at the 2MHz and 3MHz offset frequency from the desired carrier, respectively. The required signal-to-noise

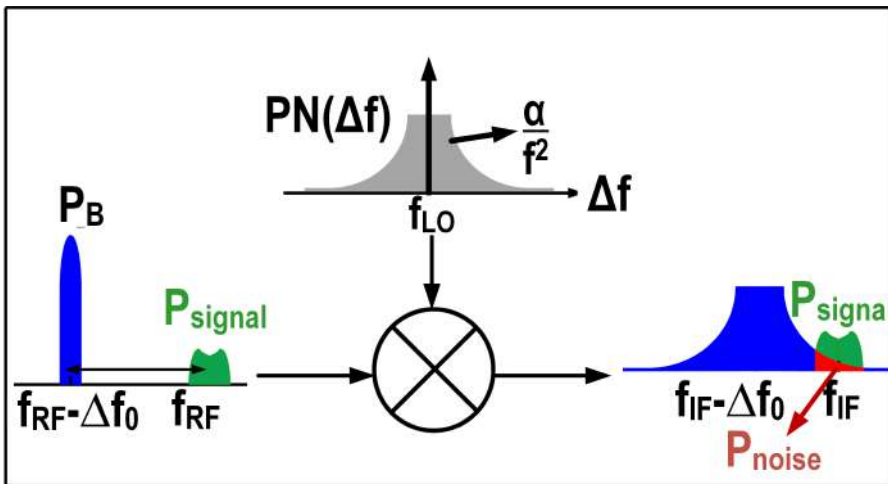


Figure 1.3: Reciprocal mixing due to the noisy LO [3].

ratio (SNR) for a GFSK demodulator with a modulation index $h=0.5$ to achieve the targeted bit-error-rate (BER) 0.1% is 11dB (measured in 1MHz bandwidth). The synthesizer phase noise can be modeled by

$$PN(\Delta f) = \frac{\alpha}{\Delta f^2}, \tag{1.4}$$

where α is a constant for a specific phase noise profile. As a result, the noise power due to the reciprocal mixing for a blocker with a power of P_B located at Δf_0 away from the desired channel can be calculated as:

$$P_{noise} = \int_{\Delta f_0 - 0.5MHz}^{\Delta f_0 + 0.5MHz} P_B \cdot PN(\Delta f) d(\Delta f). \quad (1.5)$$

Substituting Equation (1.4) into Equation (1.5), we get the following noise power expression:

$$P_{noise} = \alpha P_B \left(\frac{1}{\Delta f_0 - 0.5MHz} - \frac{1}{\Delta f_0 + 0.5MHz} \right) \quad (1.6)$$

By employing the SNR definition, we have

$$SNR = \frac{P_{signal}}{P_{noise}}, \quad (1.7)$$

where P_{signal} is the signal power. Substituting Equation (1.6) into Equation (1.7), the phase noise parameter α can be calculated as:

$$\alpha = \frac{P_{signal}}{SNR \cdot P_B \left(\frac{1}{\Delta f_0 - 0.5MHz} - \frac{1}{\Delta f_0 + 0.5MHz} \right)}. \quad (1.8)$$

Given the blocker power level P_B and blocker location (Δf_0 away from the carrier) and the required SNR, the phase noise can be simply calculated by Equation (1.8). When $\Delta f_0 = 2MHz$ and the carrier-to-interference ratio ($\frac{P_{signal}}{P_B}$) is -17dB as shown in the Table 1.1, using Equation (1.8), we have

$$\alpha \approx 5943 \cdot 1Hz. \quad (1.9)$$

The phase noise at the 2MHz offset frequency then can be calculated as:

$$PN(\Delta f)_{\Delta f=2MHz} = \frac{\alpha}{\Delta f^2} = -88dBc/Hz. \quad (1.10)$$

When $\Delta f_0 = 3MHz$ and the carrier-to-interference ratio ($\frac{P_{signal}}{P_B}$) is -27dB, using Equation (1.8), we get

$$\alpha \approx 1387 \cdot 1Hz. \quad (1.11)$$

The phase noise at the 3MHz offset frequency then can be calculated as:

$$PN(\Delta f)_{\Delta f=3MHz} = \frac{\alpha}{\Delta f^2} = -98dBc/Hz. \quad (1.12)$$

Note that Equation (1.11) should be considered as the synthesizer phase noise parameter since it puts more stringent requirement on the phase noise of the synthesizer. The phase noise requirement at the 1MHz offset frequency can be derived as:

$$PN(\Delta f)_{\Delta f=1MHz} = \frac{\alpha}{\Delta f^2} = -88dBc/Hz. \quad (1.13)$$

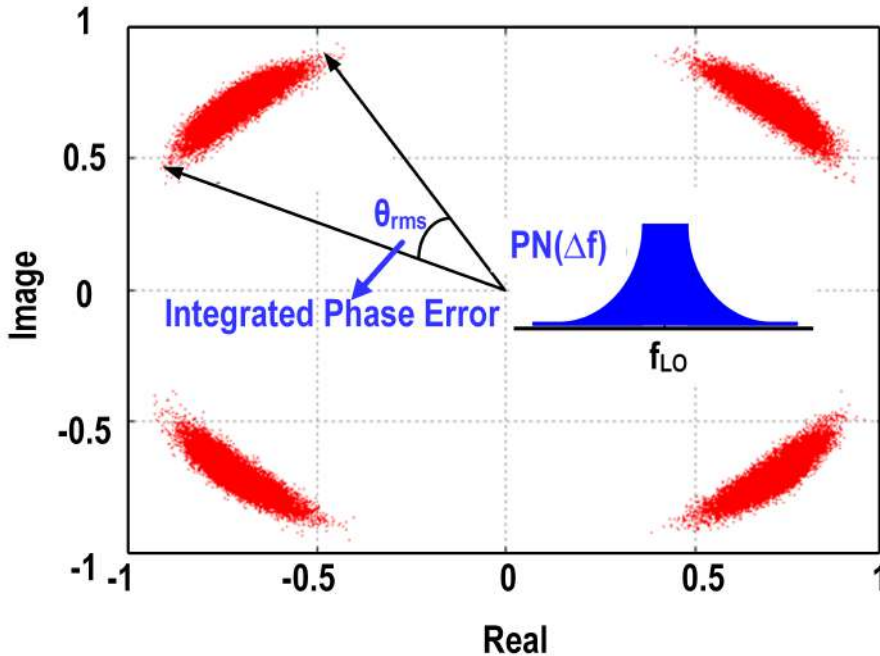


Figure 1.4: Constellation plot with a noisy LO [4].

On both receiver and transmitter side, due to the phase noise of the synthesizer, the constellation of the (de)modulated signal rotates randomly as shown in Figure 1.4. The performance degradation is characterized by Error Vector Magnitude (EVM), which is defined as the ratio of the magnitude of the error vector to the magnitude of the reference vector¹. In short,

$$EVM = 20 \log_{10} \left(\frac{A_{err}}{A_{ref}} \right). \quad (1.14)$$

The root mean square phase error can then be calculated as:

$$\theta_{rms} \approx \tan^{-1} \left(10^{\frac{EVM}{20}} \right) \approx 10^{\frac{EVM}{20}}. \quad (1.15)$$

For the BLE standard, the EVM should be lower than -20dB. It translates to an RMS phase error $\theta_{rms} = 5.7$ degrees (equivalent to 0.1rad and 6.5ps for a 2.4GHz carrier).

The in-band phase noise requirement of the synthesizer can be estimated based on the calculated integrated phase error and jitter. Assuming the synthesizer bandwidth of BW_{PLL} and the in-band phase noise of S_1 , we can calculate the RMS phase

¹Signal vector in the constellation plot where the LO is noise free.

²Here we assume that the EVM degradation is only caused by phase noise and $A_{err} \ll A_{ref}$.

error as:

$$\theta_{rms} = 2\sqrt{BW_{PLL} \cdot S_1^1}. \quad (1.16)$$

If we assume BW_{PLL} is 200kHz, then the in-band phase noise can be calculated as:

$$S_1 = \frac{(0.5\theta_{rms})^2}{BW_1} = -79dBc/Hz. \quad (1.17)$$

Consequently, the in-band phase noise should be below -79dBc/Hz with BW_{PLL} smaller than 200kHz; the phase noise at the 1MHz offset frequency should be below -88dBc/Hz to meet the transceiver EVM and the receiver SNR specifications simultaneously.

1.1.3 Spurious Tones

In general, any periodic control signals which appear at the input of the digitally-controlled oscillator (DCO) result in undesired spurious tones (spurs) in the synthesizer output spectrum. If the periodic control signals occur at the reference (fractional) frequency rate, reference (fractional) spurs appear in the synthesizer output spectrum.

On the receiver side, the carrier-to-interference ratio at the 2MHz and 3MHz offset frequency from the carrier can be as low as -17dB and -27dB respectively as discussed before. To achieve 11dB SNR, the corresponding fractional spurs should be below -28dBc and -38dBc respectively. The reference spurs should be below -38dBc for the far-out specification. On the transmitter side, the maximum transmitted power is 0dBm. The maximum allowed spurious emissions at the 2MHz and 3MHz offset frequency are -20dBm and -30dBm respectively. Consequently, the fractional spurs should be below -20dBc and -30dBc at the 2MHz and 3MHz offset frequency respectively to meet the requirement of spurious emissions. The reference spurs should be below -30dBc for the far-out specification. In this scenario, the spur performance of the synthesizer is set by the receiver side.

1.1.4 Switching Time

The synthesizer must settle down to the desired frequency after triggering the channel switch. During the channel switching time, digital calibration of various mixed-signal blocks is concurrently carried out. Much more power is burned in this phase than in the phase-locked state. It is preferable to make the channel switching time as short as possible.

1.2 Motivation and Target Specifications

Ultra-low power (<10mW) BLE transceivers enable IoT applications [5]. Phase-locked loops (PLL) based frequency synthesizers are one of the power-hungry blocks of these transceivers. The IoT PLLs successfully operate below 1mW [6, 7]. However, LC oscillators are employed in these PLLs. They are not area efficient for IoT applications and have a long time to market. On the other hand, a ring oscillator

¹Assume that the in-band and out-band phase noise contribute the same integrated jitter.

Table 1.2: The minimum and target specifications for the BLE frequency synthesizer.

Parameters	Minimum Requirements	Target Specifications
Frequency Range [GHz]	2.400-2.4835	1.800-2.700
Frequency Accuracy [kHz]	40	1
Phase Noise [dBc/Hz]		
@100kHz	-79	<-85
@1MHz	-93	<-99
@2MHz	-99	<-105
@3MHz	-103	<-109
Spurious Tones		
@2MHz	-28	<-34
≥3MHz	-38	<-44
Integrated Jitter [ps _{rms}]	6.5	<1.5
Power [mW]	-	<1.5
Area [mm ²]	-	<0.15

is more area-efficient and easier to implement, but its phase noise is around 20dB worse than that of the LC counterpart for the same power consumption. Hence, the PLL loop bandwidth should be increased to suppress the excessive phase noise of the ring oscillator. However, the loop bandwidth of the traditional type-II architectures is limited to $0.1f_{ref}$ due to loop stability requirements.

This thesis aims to explore a ring-oscillator-based digital PLL (DPLL) architecture with less area consumption while simultaneously achieving comparable performance as an LC-based DPLL for IoT applications. The minimum requirements of the BLE frequency synthesizer are derived in the first section. The target specifications altogether with minimum requirements of the BLE frequency synthesizer are summarized in Table 1.2. An external 64MHz crystal oscillator is employed to provide the reference for the DPLL.

1.3 Research Contributions

This thesis presents a ring-oscillator (RO) -based fractional-N injection-locked (IL) DPLL for IoT applications. A phase noise improvement technique (Two-path injection) is proposed to reduce the DPLL in-band phase noise by around 6dB with negligible power overhead. Besides, a fast reference spur calibration is proposed to reduce the reference spur calibration time $\sim 2\mu\text{s}$. The fractional-N operation is achieved by employing a coarse-fine DTC (CF-DTC) in the reference path to align the reference injection phase with the oscillator phase. To detect the frequency error,

the phase error of the IL-DPLL is first detected, and then injection locking is applied to reset the oscillator phase. A prototype is fabricated in the TSMC CMOS 40nm process. As a result, the measurements show that the proposed IL-DPLL achieves 1.6ps integrated jitter, -45.8dBc fractional spur and -43.6dBc reference spur while consuming only 1.33mW power and $0.13mm^2$ silicon area. The resulting figure of merit (FOM) is -234.7dB, which is the best for the RO-based fractional-N PLLs.

1.4 Thesis Outline

The thesis outline is in accordance with the project progress. Chapter 2 is dealing with the background theories of the proposed architecture and issues of the prior arts. The functionality of the system is verified and discussed in chapter 3. In chapter 4, various analog and RF building blocks of the system are designed and simulated. The register-transfer level (RTL) design and system verification in Verilog-AMS are presented in chapter 5. The measurement results are shown and discussed in chapter 6. Chapter 7 wraps up this thesis with conclusions and improvements of the system in the future.

References

- [1] J. Masuch, et al., "Ultra Low Power Transceiver for Wireless Body Area Networks," Springer, 2013, pp.109-110.
- [2] J. R. Vig, "IEEE standard definitions of physical quantities for fundamental frequency and time metrology—random instabilities (IEEE standard 1139-1999)," IEEE, New York 1 (1999).
- [3] M. Babaie, "Effects of Phase Noise on RF Transceivers," in *ET437 Digital RF course slides*, 2015, pp.11.
- [4] Krishna Sankar, "EVM with Phase Noise" in <http://www.dsplog.com/2012/07/09/evm-phase-noise/>.
- [5] Y.-H. Liu, et al., "A 3.7 mW-RX 4.4 mW-TX Fully Integrated Bluetooth Low-Energy/IEEE802.15.4/Proprietary SoC with an ADPLL-Based Fast Frequency Offset Compensation in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2015, pp. 236–237.
- [6] V. Chillara, et al., "An 860 μ W 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 *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 172–173.
- [7] Y. He, et al., "A 673 μ W 1.8-to-2.5GHz Dividerless Fractional-N Digital PLL with an Inherent Frequency-Capture Capability and a Phase-Dithering Spur Mitigation for IoT Applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2017, pp.420–421.

2

Ring-Oscillator-Based Fractional-N Injection-Locked DPLL

This chapter aims to introduce a ring-oscillator (RO) -based DPLL architecture to meet the synthesizer specifications listed in Table 1.2. A general comparison between a ring oscillator and an LC oscillator is presented in the first section of this chapter to address the design challenges of the RO-based frequency synthesizers. A noise suppression technique (injection locking) for the ring oscillator is then introduced in the second section. In the third section, the related design issues of injection-locked frequency synthesizers are discussed. The proposed architecture is presented in the last section of this chapter to address the issues discussed in the third section.

2.1 Ring Oscillator or LC Oscillator

As discussed in chapter 1, the occupied area of a ring oscillator is much smaller than that of an LC oscillator. This is the foremost reason to employ ring oscillators in frequency synthesizers even with worse phase noise performance. To get more insight, the occupied area of two recently published LC-based and RO-based PLLs are compared in Figure 2.1. The LC oscillator itself in [1] occupies almost the same amount of area as that of the entire RO-based DPLL in [2]. However, the ring oscillator demonstrates around 20dB worse figure of merit (FOM) than the LC oscillator gathered from Figure 2.2. The FOM of an oscillator defined in [3] is

$$FOM(\Delta f) = PN(\Delta f) - 20\log_{10}\left(\frac{f_{out}}{\Delta f}\right) + 10\log_{10}\left(\frac{P_{DC}}{1mW}\right), \quad (2.1)$$

where $PN(\Delta f)$ is the phase noise at an offset frequency Δf . The poor FOM of a ring

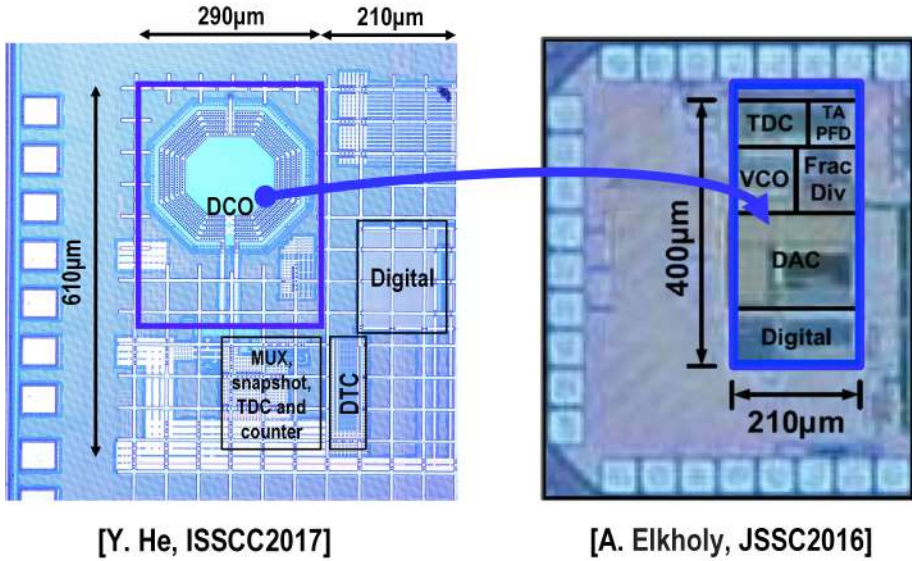


Figure 2.1: Area comparison between an LC-based and an RO-based PLLs [1, 2]

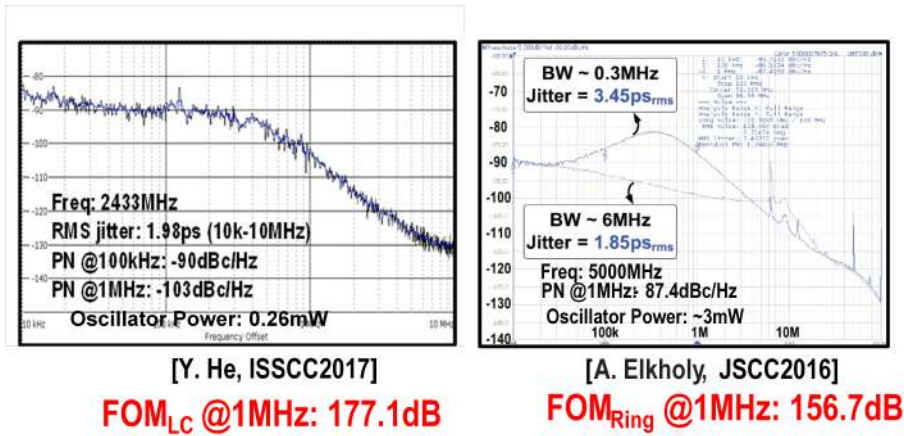


















Figure 2.2: FOM comparison between an LC oscillator and a ring oscillator [1, 2].

oscillator can be traced to its time reference, which is dependent on noisy current charging or discharging a capacitor [4]. On the other hand, the time reference of an LC oscillator is related to its capacitor and inductor size, which is noiseless in the ideal case¹. The extensive comparison is summarized in Table 2.1. Due to the multiple-stage implementation of a ring oscillator, multiple phases are simultaneously available, which gives more freedom to designers in choosing appropriate

¹Lossless.

Table 2.1: General comparison between a ring oscillator and an LC oscillator.

	LC Oscillator	Ring Oscillator
Area		
Power		
Phase Noise		
Multi-phase		
Tuning Range		
Magnetic Coupling		
Frequency Multiplexing		
Design Complexity		

architectures for the receiver and frequency synthesizer. Furthermore, the frequency multiplexing (e.g., frequency doubling) is also possible by manipulating the oscillator multiple phases. The absence of the inductor coil in a ring oscillator also reduces the magnetic coupling between other blocks. Besides, on-chip inductor design is challenging, and it is difficult to make an accurate model of an inductor.

The work in [2] increases the bandwidth of the PLL to suppress the excessive phase noise of the ring oscillator. However, even the phase noise requirements of the synthesizer for IoT applications are less stringent, this work still fails to meet the specifications. Consequently, to meet the BLE specification, the bandwidth and in-band phase noise of the PLL should be further improved. In this work, the main goal is to design a fractional-N RO-based DPLL, which meets the BLE standard while consuming less than 1.5mW power.

2.2 Injection-Locked Oscillator

An oscillator is a perfect phase integrator. Consequently, jitter is accumulated indefinitely over the time in a free-running oscillator. If this oscillator is embedded in a phase-locked loop, its jitter is corrected by the loop every reference cycle, and it can be only corrected within a particular bandwidth (typically $<0.1f_{ref}$), which is set by loop stability requirements. As mentioned before, this bandwidth is not wide enough to suppress the noise of a ring oscillator with sub-mw power consumption. The noise tracking (suppression) bandwidth of the ring oscillator needs a further extension.

If we replace the noisy edge of a ring oscillator with a clean edge of the reference with a rate of f_{ref} , jitter accumulation will be interrupted, and the accumulated jitter will be reset every FCW oscillator clock cycles. As shown in Figure 2.3 (left), the jitter of a three-stage ring oscillator is accumulated over the time indefinitely without any external interruption. However, it is reset to zero by an injection transistor

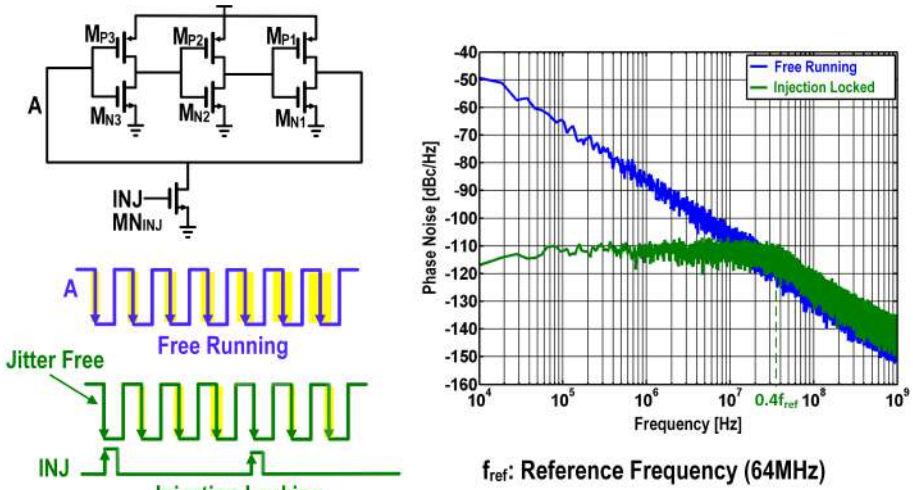


Figure 2.3: Illustration of injection locking and the simulated phase noise.

(injector) MN_{INJ} . Since the falling edge of the oscillator waveform is determined by both MN_{INJ} and MN_1 , the injector MN_{INJ} should be sized large enough to force the oscillator reset its phase. Since the “incorrect phase” of the oscillator is completely corrected by the INJ signal every reference cycle, it is expected that the noise tracking bandwidth becomes wider than traditional DPLLs where the “incorrect phase” is partially corrected due to loop stability requirements.

2.2.1 Phase Noise Improvement

The intuitive description of injection locking phenomenon in the time domain is presented in the previous section, and this section is mathematically dealing with injection locking in the frequency domain. As shown in Figure 2.3 (right)¹, the phase noise of the free-running oscillator decreases by 20dB/dec over the frequency. However, the phase noise improves significantly (from -86dBc/Hz to -112dBc/Hz at the 1MHz offset frequency) after applying the reference injection. For a 64MHz reference, the noise tracking bandwidth can be up to 25MHz, and the phase noise improvement at the 1MHz offset frequency is around 26dB.

The transfer function of the injection locking phenomenon² is high pass as it can be gathered from the simulation (see Figure 2.3). As proven in [6], the phase noise of an injection-locked oscillator (for $FCW \gg 1$) can be expressed by

$$PN_{INJ}(\Delta f) = \frac{2FCW - 1}{FCW} \cdot PN_{free}(\Delta f) \cdot \frac{\left(\frac{\Delta f}{f_{BW}}\right)^2}{1 + \left(\frac{\Delta f}{f_{BW}}\right)^2} + \frac{FCW^2 \cdot PN_{ref}(\Delta f)}{1 + \left(\frac{\Delta f}{f_{ref}}\right)^2}, \quad (2.2)$$

¹Results come from the MATLAB time-domain simulation. Flicker noise of the oscillator is not modeled.

²The ratio of the phase noise of an injection-locked oscillator to the phase noise of a free-running oscillator.

where PN_{free} is the phase noise of the free-running oscillator, PN_{ref} is the phase noise of the reference path, and the noise tracking bandwidth is

$$f_{BW} \approx 0.39f_{ref}. \quad (2.3)$$

This analysis assumes that the oscillator is jitter-free at the injection moment, and the jitter is then accumulated again right after the reference injection¹. The transfer functions (TF) of the reference phase and the oscillator phase to the oscillator output phase after the injection locking are plotted in Figure 2.4 based on Equation (2.2). The transfer function of the oscillator phase is high-pass shaped with a 3dB high-

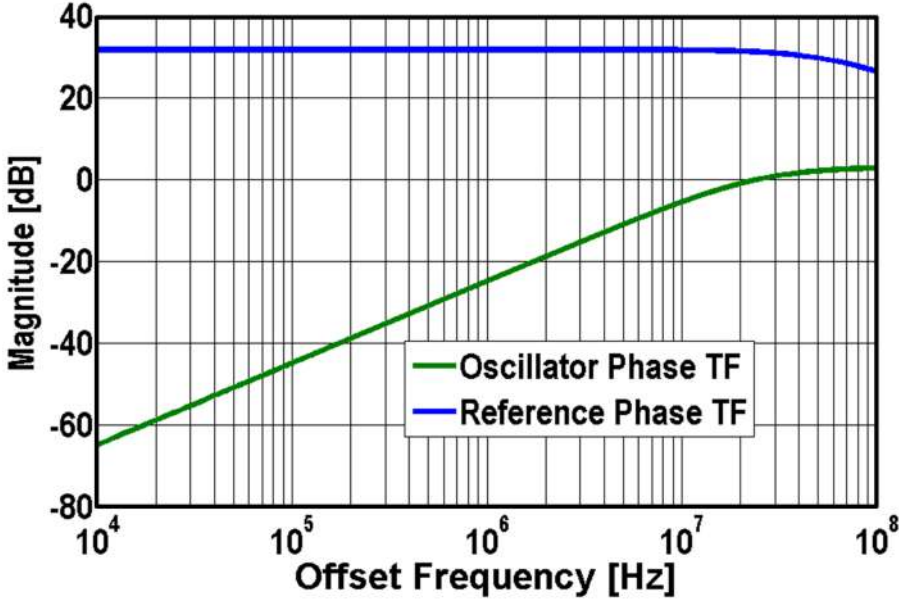


Figure 2.4: Phase noise transfer function of the reference and free-running oscillator in the injection locking system.

frequency gain. Due to the injection operation, the variance of the period jitter is doubled, resulting in 3dB phase noise degradation in high frequency². Since the reference phase noise appears inside the bandwidth of the injection-locked oscillator with a multiplication factor of FCW^2 , a major attention must be paid to minimize the phase noise of the reference path.

It should be noted that high-frequency reference is desired since the phase noise tracking bandwidth is proportional to f_{ref} . Many reported works in recent years achieved superior jitter performance partly due to the use of a large f_{ref} ³. It is not realistic to use that large reference clock in IoT systems considering the associated digital power consumption and costly reference.

¹Add zero mean white noise to the oscillator timestamps.

²Please refer to [6] for the detailed analysis.

³Up to 400MHz.

2.2.2 Reference Spurs Degradation

The random jitter of a ring oscillator is largely suppressed by employing the injection locking technique. However, the deterministic jitter may arise if the center frequency of the ring oscillator drifts as it is shown in Figure 2.5. When the oscillator phase is not aligned with the injection phase due to the frequency error (f_{err}), the injection signal drags the oscillator phase with a large deterministic phase jump. This behavior is extensively discussed in [7]. The reference spur level can be calculated as:

$$Spur_{Ref} \approx 20 \log_{10} \left(\frac{|f_{out} - f_{fr}|}{f_{ref}} \right), \quad (2.4)$$

where f_{fr} is the free-running frequency and f_{out} is the desired output frequency. The deterministic jitter can be expressed as:

$$DJ \approx \frac{|f_{out} - f_{free}|}{f_{out} \cdot f_{ref}}. \quad (2.5)$$

For FCW=38 and f_{ref} =64MHz, 1MHz frequency error generates -36dBc reference

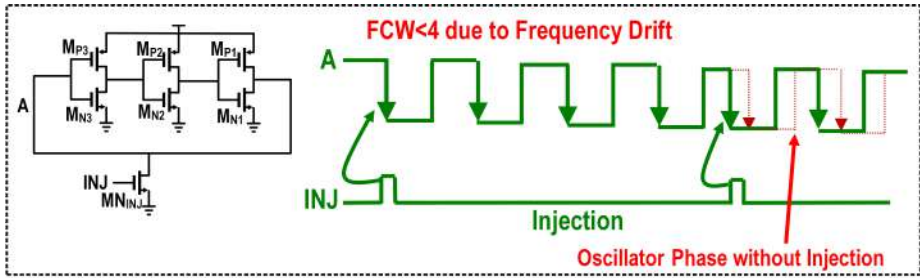


Figure 2.5: Reference spurs generation due to f_{err} .

spur and 6.4ps deterministic jitter. Note that a high-frequency reference clock is again desired for reference spur and deterministic reduction.

Due to the PVT variations and the flicker noise of the transistors, the free-running frequency of a ring oscillator can drift up to tens of MHz. As a consequence, the injection-locked oscillator typically exhibits large spurious tones and deterministic jitter. Therefore, injection-locked oscillators are usually embedded into a frequency tracking system to make the frequency of the free-running oscillator as close as possible to the desired frequency.

2.2.3 Injection Strength Degradation

Up to now, we assumed that the oscillator is jitter-free right after applying the reference injection. However, in reality, even after the injection, some jitter still remains mainly because of a limited current drivability, on-resistance of injection transistors, and the timing uncertainty of the injection signal due to the system noise. As a result, the injection strength [5], the IL-DPLL bandwidth, and close-in phase noise are degraded.

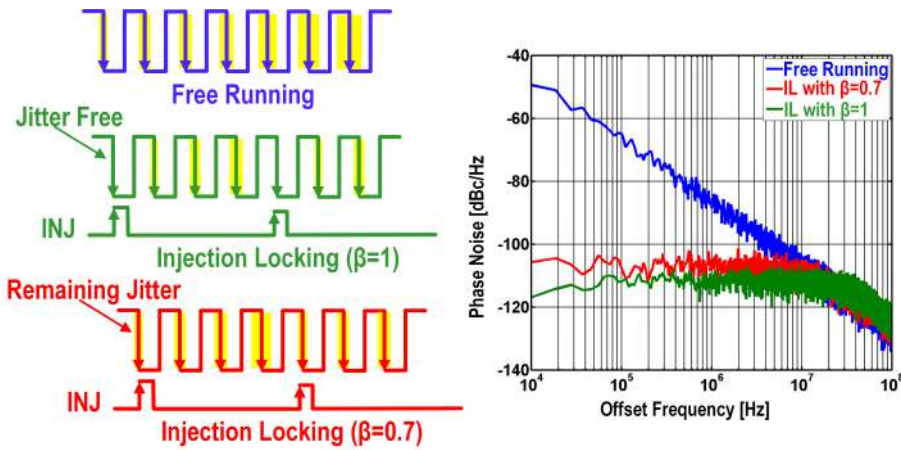


Figure 2.6: Illustration of phase noise degradation.

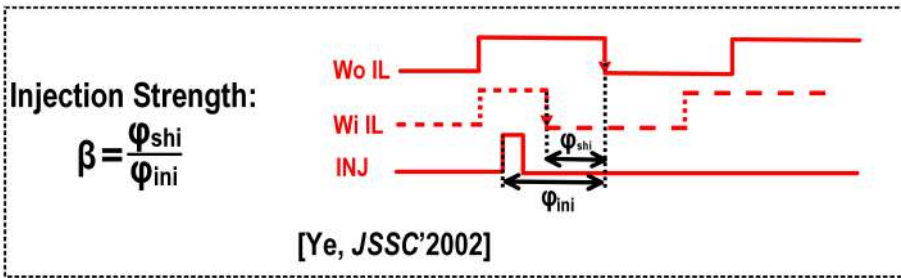


Figure 2.7: Definition of injection strength.

The definition of injection strength (β) is illustrated in Figure 2.7. β is defined as the ratio of the oscillator phase shift (ϕ_{shi}) due to the injection to the instantaneous phase error (ϕ_{ini}) between the injection phase and the oscillator when the reference injection is not applied. It characterizes how much jitter is cleaned by the injection signal. if $\beta=0$, the oscillator phase is not corrected at all, which is corresponding to the free-running case. if $\beta=1$, the oscillator phase is entirely determined by the injection phase at the injection moment. As an example, $\beta=0.7$ is shown in Figure 2.6. Due to the finite injection strength, the phase noise tracking bandwidth is smaller than $0.39f_{ref}$, degrading the oscillator in-band phase noise by 4dB.

It is worthwhile to discuss the factors that make β smaller than 1. First, the insufficient strength of the injection transistor (injector) itself makes the jitter of the free-running oscillator partly corrected. The injector is often sized a few (e.g., 5) times larger than the oscillator delay cell [5]. Second, when the injection phase occurs not exactly at the transition of the oscillator waveform, the reference injection

is less effective¹ or creates a large phase jump in the oscillator output², resulting in phase noise degradation or large reference spurs as discussed before [7]. This phase shift can also happen due to PVT variations and interference coupling to the ring oscillator through supply and substrate. Again, we should make the injection-locked oscillator embedded into a frequency tracking loop to adjust the frequency of the free-running oscillator to the desired frequency.

2.3 Injection-Locked DPLL

As discussed in the above section, a frequency tracking loop is typically needed when the injection-locking technique is employed. However, there are still some issues related to the IL-DPLL operation. The reference spur degradation effect still exists if any static time offset (t_{os}) exists between the rising edge of the oscillator and injection pulse. The second issue is that the IL-DPLL is inherently limited to the integer-N operation and cannot be directly used for the fractional-N frequency synthesis. However, the fractional-N operation is required for IoT applications. Furthermore, due to the reference injection, the accumulated phase error is almost reset at every reference cycle and thus it becomes difficult for the frequency tracking loop to detect any frequency/phase error. These issues are addressed in this section.

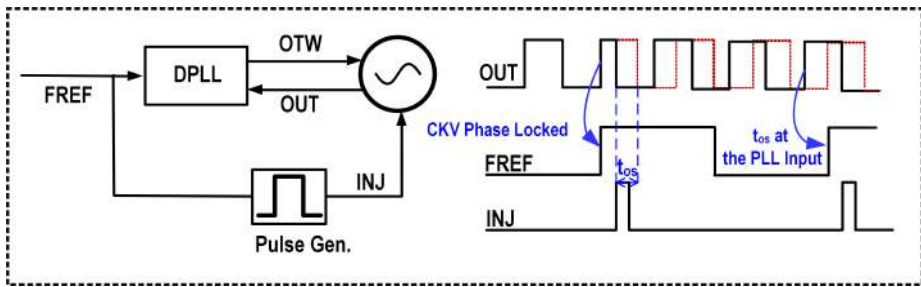


Figure 2.8: Reference spur degradation due to t_{os} .

2.3.1 Reference Spurs Degradation

As it is shown in Figure 2.8, the DPLL works as an FTL to correct the frequency error of the free-running oscillator. It also locks the oscillator phase to the reference phase. On the other hand, the injection path generates a short pulse through the pulse generator to reset the noisy oscillator phase. The oscillator waveform is distorted after the reference injection when the static time offset (t_{os}) between the injection phase and the oscillator phase is not equal to zero, resulting in duty cycle error. This behavior is well studied in [7, 8], and the reference spur level is expressed as:

$$Spur_{Ref} \approx 20 \log_{10}(t_{os} f_{out}). \quad (2.6)$$

¹The injection phase lags behind the oscillator phase.

²The injection phase leads the oscillator phase.

To reach a -50dBc reference spur level at $f_{out}=2.4\text{GHz}$, t_{os} should be smaller than 1ps. Consequently, a mechanism should be added to the PLL to reduce t_{os} within 1ps. The t_{os} calibration can be done by directly injecting a pulse into the oscillator, then measuring the phase error appearing at the input of the phase detector (PD) and finally adjusting the delay of the injection pulse [8, 9]. However, the DPLL needs to relock again right after the injection moment due to the large uncalibrated t_{os} (e.g., $t_{os}=200\text{ps}$) [9]. The locking time (i.e., 10-15 μs) of the DPLL is inversely proportional to the loop bandwidth. Due to the noise of the system¹, the time difference between the injection pulse and the oscillator is changing randomly. Therefore, the output code of the PD after each injection should be averaged and a multi-step calibration is required to compensate t_{os} effectively. It means the DPLL has to relock multiple times (e.g., 10-step calibration requires 100-150 μs). The calibration time is typically too long for IoT applications (e.g., BLE has longest packet length 376 μs).

2.3.2 Inherent Integer-N Operation

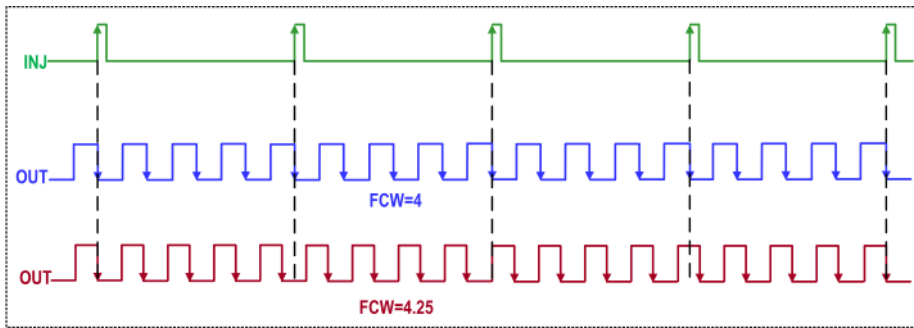


Figure 2.9: Inherent integer-N operation for the IL-DPLL.

The second issue of the IL-DPLL is that it is limited to the inherent integer-N operation. To illustrate this limitation, consider the waveforms shown in Figure 2.9, where the free-running frequency of the oscillator is tuned close to the target frequency f_{out} with $FCW = 4.25$. The phase difference between the injection pulse and the oscillator output is cycle-slipping. It is 0, $\frac{0.25}{f_{out}}$, $\frac{0.5}{f_{out}}$, $\frac{0.75}{f_{out}}$ in four consecutive reference clock cycles. Consequently, the oscillator cannot be injection locked. Otherwise large spurious tones will occur in the DPLL output spectrum.

2.3.3 Frequency Tracking Effectiveness

The final issue of the IL-DPLL is that its performance (reference spur and jitter) is very sensitive to the frequency error (f_{err}), namely the frequency difference between the target frequency (f_{out}) and the oscillator free-running frequency (f_{fr}). Hence, a frequency-tracking loop (FTL) is needed to continuously tune f_{fr} to f_{out} [10]. However, due to the reference injection, the accumulated phase error is

¹The system phase noise is too large especially when a low-power ring oscillator is used.

almost reset at every reference cycle, so it is difficult for a FTL to precisely detect any f_{err} .

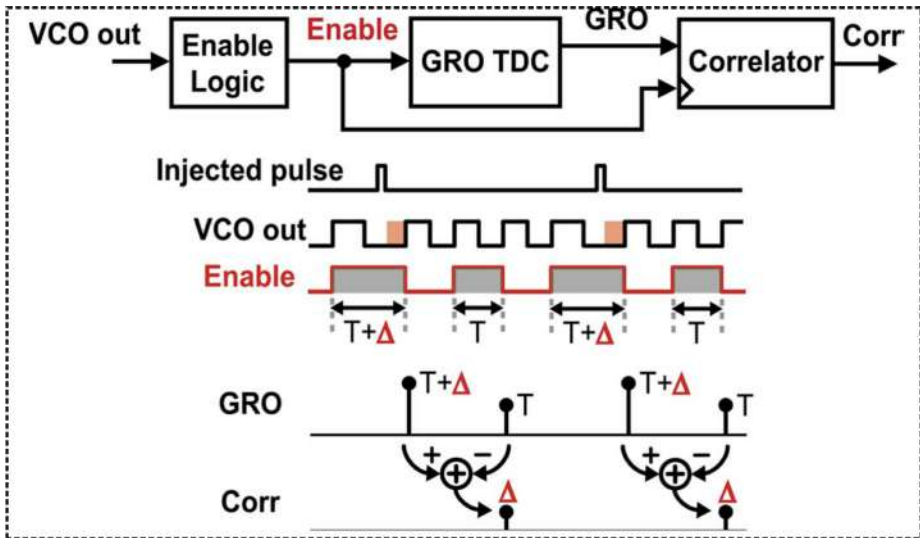


Figure 2.10: GRO TDC for frequency error detection [7, 11].

The gated ring oscillator time-to-digital converter (GRO TDC) is used to detect the frequency error in [7, 11] as shown in Figure 2.10. If there is any f_{err} when the reference injection is applied, the duty cycle of the oscillator voltage waveform will deviate its nominal value by some amount of Δ . The GRO TDC is then used to detect this duty cycle error Δ . Its output is subsequently used to tune the oscillator frequency such that the detected Δ is zero. Hence, the frequency error is eliminated in the steady state. This method does not rely on the direct phase comparison between the injection phase and RF phase, hence the frequency error can be detected. However, the GRO TDC is not power-efficient¹.

Figure 2.11 shows the second f_{err} tracking technique using a replica digitally-controlled oscillator (DCO) [12–15]. Since the replica DCO is not injection-locked, its phase is not disrupted and it can be employed to capture any frequency drift. The main DCO could have the same free-running frequency as the replica DCO since they share the same oscillator tuning word. This technique decouples the trade-off between the injection locking and frequency error detection. However, the replica DCO consumes the same power as the main DCO and the mismatch between the delay cells limits the frequency tracking accuracy.

Figure 2.12 shows another frequency tracking technique [16–18]. Pulse gating is used where the reference is not injected every 4th reference cycle as shown in Figure 2.12. The oscillator is free-running when the reference is not injected. As a consequence, the f_{err} introduced phase error can be detected and corrected. However, fractional spurs may occur due to the gated pulse injection [10].

¹5.3mW in [11].

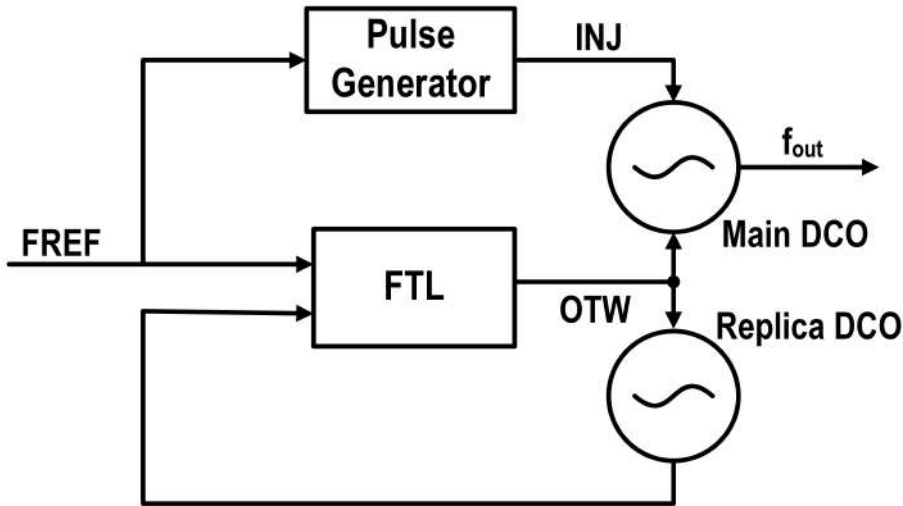


Figure 2.11: Replica VCO for frequency error detection [12–15].

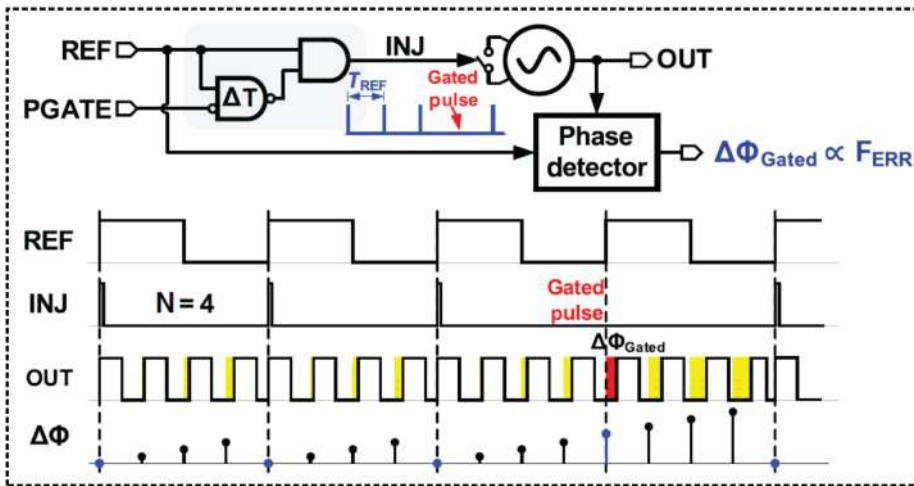


Figure 2.12: Pulse gating for frequency error detection [16–18].

2.4 Proposed Architecture

There are mainly four issues for the IL-DPLL operation, namely the inherent integer-N operation, limited frequency error tracking ability, injection strength degradation, and reference spur as mentioned before. These problems are addressed in the proposed architecture.

To enable the fractional-N operation of the IL-DPLL, the injection phase (INJ) can be shifted by a digital-to-time converter (DTC) according to the accumulated FCW fractional part. As shown in Figure 2.13, for $FCW = 4.25$, the phase difference

between INJ and the oscillator output (OUT) is 0 , $0.25t_{out}$, $0.5t_{out}$ and $0.75t_{out}$ in four consecutive reference cycles, where t_{out} is the output period of the IL-DPLL. If INJ is correspondingly shifted by t_{out} , $0.75t_{out}$, $0.5t_{out}$ and $0.25t_{out}$ by a DTC, a new injection signal INJ' is obtained, which is always aligned with OUT. Consequently, INJ' can be applied to reset the oscillator phase for the injection locking operation.

2

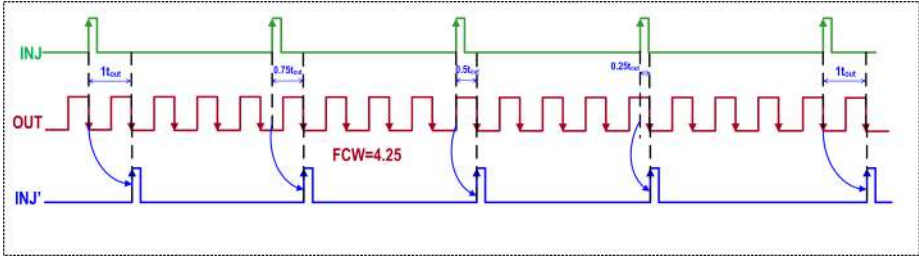


Figure 2.13: Illustration of the fractional-N operation.

The frequency error (f_{err}) of the oscillator is reset each time the reference injection is applied. In this design, f_{err} is first detected by the DPLL and the reference injection is then applied at the next RF cycle as shown in Figure 2.14 [9]. Since the DPLL can detect the undisrupted oscillator phase, it can be used to correct any frequency drift of the oscillator. There is no need to employ a power-hungry GRO TDC or replica DCO for f_{err} detection.

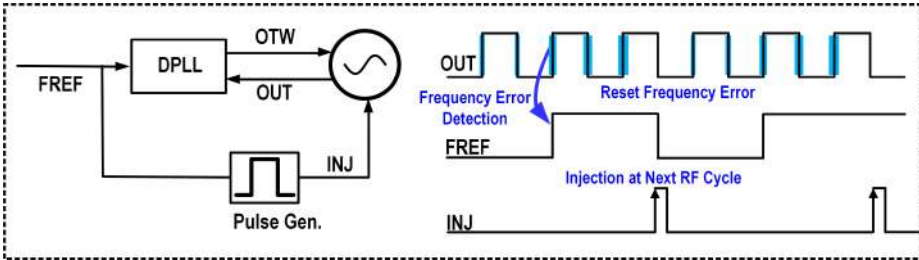


Figure 2.14: Illustration of frequency error detection.

Since the IoT packet is typically short, a fast foreground time offset (t_{os}) calibration is proposed to avoid relocking the DPLL multiple times, minimizing the energy consumption overhead. During the t_{os} calibration, the reference injection is disabled and the DPLL is operating, hence the DPLL is always in phase-locked state. A 1b TDC¹ is first used to detect the phase error between the injection phase and the oscillator phase. Its output is then used to adjust the delay of the pulse such that t_{os} is zero in the steady state. Consequently, the proposed technique separates the phase locking and time offset calibration.

¹Another phase detector.

As shown in Figure 2.15, the reference injection signal (INJ) cannot reset the oscillator phase entirely, and the unleared jitter will be accumulated again, resulting in-band phase noise degradation of the IL-DPLL. To improve the phase noise of the IL-DPLL, an auxiliary injection signal (INJ1) is used by delaying INJ a few RF cycles to replace the noisy edge of the oscillator. Hence, the remaining jitter due to the limited strength of INJ is suppressed.

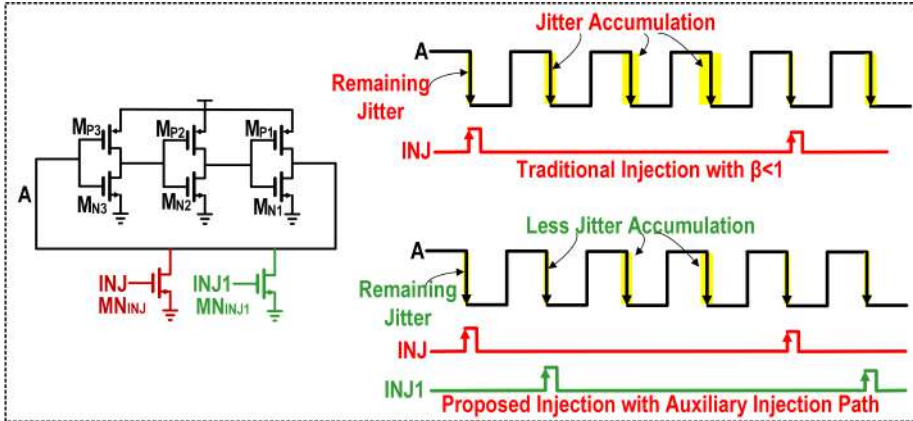


Figure 2.15: Auxiliary injection path for phase noise improvement.

Figure 2.16 shows the block diagram of the implemented fractional-N IL-DPLL by employing the above-discussed techniques. It is composed of a coarse-fine DTC (CF-DTC), a digitally-controlled ring oscillator (DCO), two delay-locked loops (DLLs) and a phase locking block. The CF-DTC is used to enable the fractional-N operation of the IL-DPLL. Besides, the F-DTC is employed to cancel the quantization error (QE) of the C-DTC, improving in-band phase noise and spurious tones due to QE. The DLL is used to adjust the reference injection phase for low-spur operation and frequency error detection. The IL-DPLL has three operation phases, including frequency/phase locking, time offset (t_{os}) calibration, and injection locking.

At the beginning (Phase-I), the phase locking block locks the oscillator phase to the delayed reference phase ($FREF_{dly1}$). Note that by employing the CF-DTC, the $FREF_{dly1}$ is brought close to the DCO phase in the phase-locked state. Therefore, reference injection could be now applied to the oscillator even in a fractional channel. However, the time offset (t_{os}) between the injection phase REF_{INJ1} and the oscillator phase (RF_{INJ1}) should be calibrated to reduce the reference spur.

In Phase-II, the DLLs are turned on but both injection paths are disabled. The 1b TDC compares the time difference between REF_{INJ1} and RF_{INJ1} . It then drives an accumulator (ACC) block which generates the control code to adjust the delay of the coarse-fine digitally-controlled delay line (CF-DCDL). The accumulator keeps changing the delay of the CF-DCDL until the 1b TDC output toggles between 0 and 1. Due to the employment of a low-power ring oscillator, the jitter of DPLL is relatively large (i.e., ~ 4 ps) and affects calibration accuracy. An average block (AVG) is then used to reduce the system noise impact on the calibration accuracy.

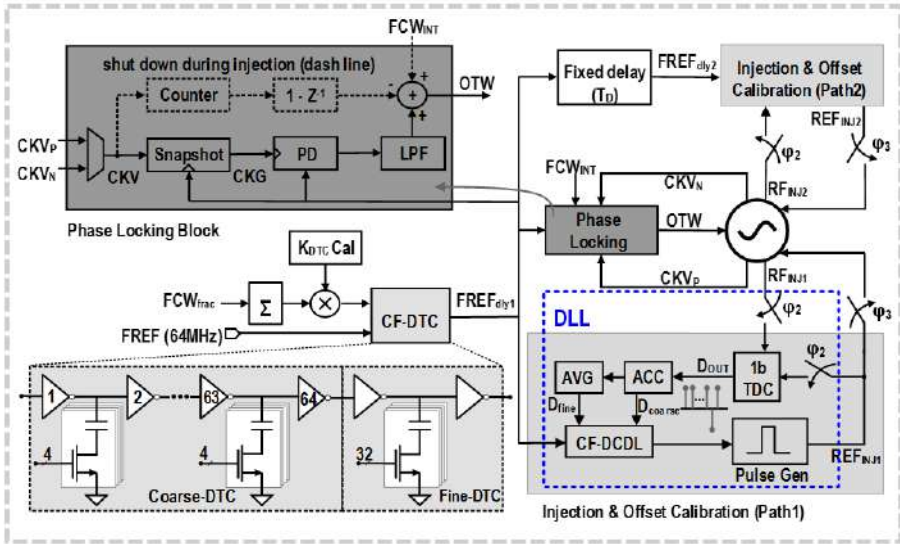


Figure 2.16: The proposed architecture.

Once the reference spur calibration is finished, a short pulse is generated by a pulse generator to reset the oscillator phase for the IL-DPLL operation (Phase-III). In this phase, the phase locking block is still operating to capture the frequency error of the DCO continuously. However, the frequency error detected by the phase detector (PD) is cleared right after the injection. To detect the frequency error effectively, the phase difference between the $FREF_{dly1}$ and the DCO phase $CKV_{N/P}$ is first measured by the PD, and then the injection is applied to reset the oscillator phase error. Due to the delay of the CF-DCDL and the pulse generator, the injection occurs always after the phase detection. Consequently, the free-running phase error of the DCO is undisrupted; hence it can be detected to tune the DCO frequency to the desired frequency.

Due to the finite injection strength of the Path1, the in-band phase noise of the IL-DPLL is degraded due to the remaining jitter after each injection. To resolve this issue, a second phase $FREF_{dly2}$ is generated by delaying the $FREF_{dly1}$ 10 DCO cycles (T_D); and an auxiliary injection path (Path2) is added, which generates a pulse RF_{INJ2} to reset the oscillator phase again. By injecting the RF_{INJ2} into the oscillator, the remaining jitter due to the finite injection strength of Path1 is cleaned.

References

- [1] Y. He, et al., "A 673 μ W 1.8-to-2.5GHz Dividerless Fractional-N Digital PLL with an Inherent Frequency-Capture Capability and a Phase-Dithering Spur Mitigation for IoT Applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2017, pp.420–421.
- [2] A. Elkholy, et al., "A 2.0-5.5 GHz Wide Bandwidth Ring-Based Digital Fractional-N PLL with Extended Range Multi-Modulus Divider," *IEEE J. Solid-State Circuits*, vol. 51, pp. 1771-1784, Aug. 2016.
- [3] P. Kinget, *Integrated GHz Voltage Controlled Oscillators*, in *Analog Circuit Design*, Kluwer, Springer, 1999, pp. 353–381.
- [4] R. Navid, et al., "Minimum Achievable Phase Noise of RC Oscillators," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 630–637, Mar. 2005.
- [5] S. Ye, et al., "A Multiple-Crystal Interface PLL with VCO Realignment to Reduce Phase Noise," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 1795–1803, Dec. 2002.
- [6] N. Da Dalt, "An Analysis of Phase Noise in Realigned VCOs," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 61, no. 3, pp. 143–147, Mar. 2014.
- [7] B. Helal, et al., "A Low Jitter Programmable Clock Multiplier Based on a Pulse Injection-Locked Oscillator With a Highly-Digital Tuning Loop," *IEEE J. Solid-State Circuits*, vol. 44, no. 5, pp. 1391–1400, May. 2009.
- [8] S. Levantino, et al., "A 1.7 GHz Fractional-N Frequency Synthesizer Based on a Multiplying Delay-Locked Loop," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2678–2691, Nov. 2015.
- [9] Y. C. Huang and S. I. Liu, "A 2.4-GHz Subharmonically Injection-Locked PLL with Self-Calibrated Injection Timing," *IEEE J. Solid-State Circuits*, vol. 48, no. 2, pp. 417–428, Feb. 2013.
- [10] S. Choi, et al., "A PVT-Robust and Low-Jitter Ring-VCO-Based Injection-Locked Clock Multiplier With a Continuous Frequency-Tracking Loop Using a Replica-Delay Cell and a Dual-Edge Phase Detector," *IEEE J. Solid-State Circuits*, vol. 51, no. 8, pp. 1878–1889, Aug. 2016.
- [11] B. Helal, et al., "A Highly Digital MDLL Based Clock Multiplier that Leverages a Self-Scrambling Time-to-Digital Converter to Achieve Subpicosecond Jitter Performance," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 855–863, Apr. 2008.
- [12] W. Deng, et al., "A 0.022 mm² 970 μ W Dual-Loop Injection-Locked PLL with –243 dB FOM Using Synthesizable All-Digital PVT Calibration Circuits," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2013, pp. 248–249.

- [13] A. Musa, et al., "A Compact, Low-Power and Low-Jitter Dual-Loop Injection Locked PLL Using All-Digital PVT Calibration," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 50–60, Jan. 2014.
- [14] W. Deng, et al., "A 0.0066 mm 780 μ W Fully Synthesizable PLL with a Current-Output DAC and an Interpolative Phase-Coupled Oscillator Using Edge-Injection Technique," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 266–267.
- [15] W. Deng, et al., "A Fully Synthesizable All-Digital PLL with Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 68–80, Jan. 2015.
- [16] A. Elkholy, et al., "Design and Analysis of Low-Power High-Frequency Robust Sub-Harmonic Injection-Locked Clock Multipliers," *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 3160–3174, Dec. 2015.
- [17] A. Elkholy, et al., "A 6.75-to-8.25 GHz 2.25 mW 190 fsrms Integrated Jitter PVT-Insensitive Injection-Locked Clock Multiplier Using All-Digital Continuous Frequency-Tracking Loop in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2015, pp. 188–189.
- [18] A. Elkholy, et al., "A 6.75-to-8.25 GHz, 250 fsrms-Integrated-Jitter 3.25 mW Rapid on/off PVT-Insensitive Fractional-N Injection-Locked Clock Multiplier in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2016, pp. 192–193.

3

System-Level Modeling and Simulation in MATLAB

The behavioral modeling and simulation of the proposed IL-DPLL in MATLAB are presented in this chapter. The modeling of various sub-blocks of the DPLL is given in the first section of this chapter. The second section shows the time-domain simulation results in terms of settling behavior, phase noise and spectrum of the IL-DPLL. The s-domain phase noise model is presented in the third section, and the last section summarizes this chapter.

3.1 MATLAB Modeling of DPLL Sub-blocks

The timestamps of various sub-blocks are sufficient to characterize the behavior of a DPLL. Non-idealities such as jitter, quantization error and non-linearity can be added to model the system imperfections. The time-domain modeling of the DPLL sub-blocks is presented in this section.

3.1.1 Reference Phase and Timestamps

The DPLL phase detection mechanism is based on cycles accumulation [1]. In one reference clock cycle, there are FCW oscillator clock cycles. During the modeling, the reference phase is referred to the oscillator phase (i.e., the reference phase accumulates FCW in one reference cycle and the oscillator phase accumulates one in one RF cycle.). The accumulated reference phase $FCW_{acc}[k]$ can be expressed as:

$$FCW_{acc}[k] = FCW_{acc}[k - 1] + FCW, \quad (3.1)$$

where k is the reference clock domain time index. $FCW_{acc}[k]$ can be further split into an integer part ($FCW_{acc_int}[k]$) and a fractional part ($FCW_{acc_frac}[k]$) as:

$$FCW_{acc}[k] = FCW_{acc_int}[k] + FCW_{acc_frac}[k]. \quad (3.2)$$

On the other hand, the reference timestamps, $tR[k]$ can be simply expressed as:

$$tR[k] = (k - 1) \cdot \frac{1}{f_{ref}}. \quad (3.3)$$

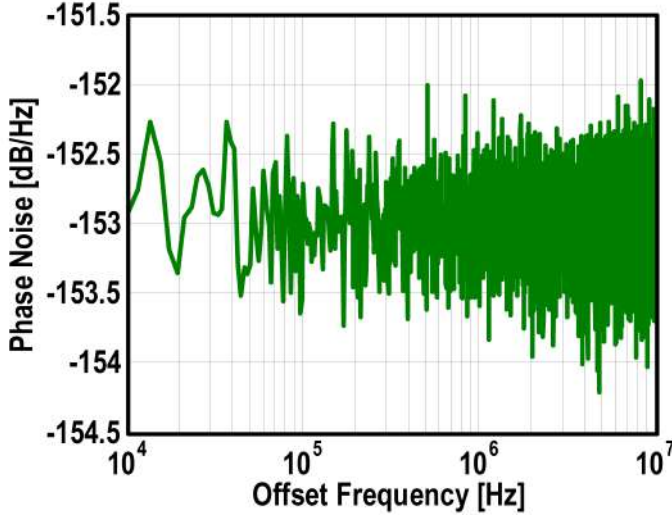


Figure 3.1: Simulated phase Noise of the clock buffer.

Each time the reference clock goes through the clock buffer, the devices' noise adds uncertainty to $tR[k]$. In general, the noise is composed of thermal noise and flicker noise¹. The thermal noise can be modeled by a zero-mean Gaussian distribution function. If the variance of the delay uncertainty added to $tR[k]$ is J_{ref} , $tR[k]$ can be expressed as:

$$tR[k] = (k - 1) \cdot \frac{1}{f_{ref}} + \text{normrnd}(0, J_{ref})^2. \quad (3.4)$$

It is helpful to relate the jitter variance J_{ref} to the phase noise of the clock buffer. We can simply convert J_{ref} to phase noise using Equation (1.3) and get the following expression:

$$J_{ref} = \frac{\sqrt{10^{\frac{PN_{ref}}{10}} f_{ref}}}{2\pi f_{ref}}, \quad (3.5)$$

where PN_{ref} is the phase noise of the clock buffer. When $PN_{ref} = -153 \text{ dBc/Hz}$ and $f_{ref} = 64 \text{ MHz}$, J_{ref} is 0.445 ps from Equation (3.5). To verify this equation,

¹Flicker noise is not modeled in this chapter.

² $\text{normrnd}(0, x)$ is a normal distribution function with zero mean and x variance.

$J_{ref}=0.445\text{ps}$ is added to Equation (3.4). The simulated phase noise is shown in Figure 3.1. It nearly exhibits a flat region with phase noise -153dBc/Hz , which matches well with Equation (3.5).

3.1.2 Coarse-Fine DTC

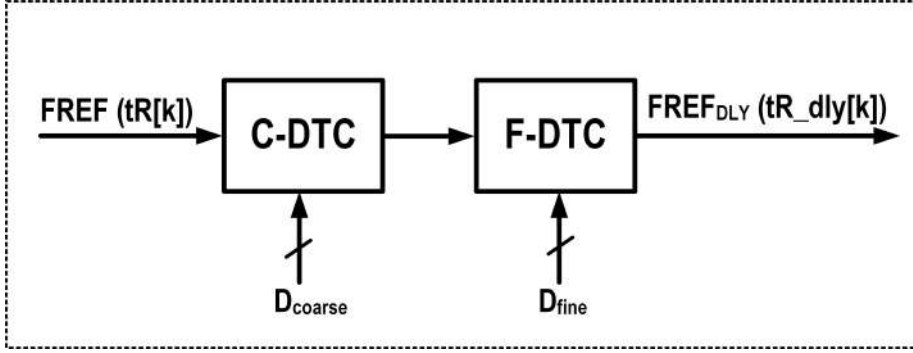


Figure 3.2: Block diagram of the CF-DTC.

Figure 3.2 shows the block diagram of the coarse-fine DTC (CF-DTC). The CF-DTC is used to make the IL-DPLL work in the fractional-N domain. It delays the reference phase ($FREF$) with timestamps $tR[k]$ to generate the delayed reference phase ($FREF_{DLY}$) with timestamps $tR_{dly}[k]$. The desired delay $t_{dly}[k]$ of the CF-DTC as discussed in chapter 2 is

$$t_{dly}[k] = (1 - FCW_{acc_frac}[k]) \cdot \frac{1}{f_{out}}; \quad (3.6)$$

and the timestamps of the CF-DTC can be expressed as:

$$tR_{dly}[k] = tR[k] + t_{dly}[k]. \quad (3.7)$$

Equation (3.6) gives us the ideal delay when the noise, quantization error and non-linearity of the CF-DTC are not considered. These factors should be taken into account for the accurate modeling. For a phase noise of PN_{dte} for the CF-DTC, the jitter variance J_{dte} can be calculated as:

$$J_{dte} = \frac{\sqrt{10^{\frac{PN_{dte}}{10}} \cdot f_{ref}}}{2 \cdot \pi \cdot f_{ref}} \quad (3.8)$$

Consequently, Equation (3.5) can be rewritten as:

$$tR_{dly}[k] = tR[k] + t_{dly}[k] + \text{normrnd}(0, J_{dte}). \quad (3.9)$$

Now, the quantization error of the CF-DTC should be considered as well. The

control code of the coarse-DTC (C-DTC) can be calculated as:

$$D_{coarse}[k] = \text{fix}\left(\frac{t_{dly}[k]}{K_{C-DTC_cal}}\right)^1, \quad (3.10)$$

where the K_{C-DTC_cal} is calibrated gain of the C-DTC. The resulting quantization error of the C-DTC is

$$t_{quant}[k] = t_{dly}[k] - D_{coarse}[k] \cdot K_{C-DTC_cal}. \quad (3.11)$$

The quantization error of the C-DTC is then applied to a F-DTC to achieve a higher time resolution. The control code of the F-DTC can be expressed as:

$$D_{fine}[k] = \text{fix}\left(\frac{t_{quant}[k]}{K_{F-DTC_cal}}\right), \quad (3.12)$$

where the K_{F-DTC_cal} is the calibrated gain of the F-DTC. By considering the mismatch of the delay cells in a Cadence simulation, the transfer function of the CF-DTC can be saved in a look-up table and then used to calculate nonideal $t_{dly}[k]$. For example, if the simulated delay of the CF-DTC is $T_D(D1, D2)$, where $D1$ is the C-DTC control code and $D2$ is the F-DTC control code, the CF-DTC delay is

$$t_{dly}[k] = T_D(D_{coarse}[k], D_{fine}[k]). \quad (3.13)$$

In this case, the non-linearity of the CF-DTC directly appears in the delay $t_{dly}[k]$.

It should be noted that any error in the delay of CF-DTC (deviating from Equation (3.6)) will inject phase error into the DPLL loop. As a result, additional phase noise or/and spurious tones will be observed in the DPLL output. This phase error can come from the gain error, quantization error, non-linearity and even thermal noise of the CF-DTC.

3.1.3 TDC, Loop Filter and Phase Detection

Figure 3.3 shows the block diagram of the TDC, loop filter, phase detector (PD) and timing diagram of the DCO and CF-DTC outputs. The TDC calculates the fractional phase error between the delayed reference timestamps ($tR_{dly}[k]$) and the DCO timestamps $tckv[n]$. The phase error detected by the TDC is

$$\Delta t[k] = tR_{dly}[k] - tckv[n - 1]. \quad (3.14)$$

This phase error is not disrupted by the injection operation as it is shown in Figure 3.3, where injection occurs 1 DCO cycle later after the DPLL phase detection. The fractional phase error should be normalized to the oscillator period [2], and can be expressed as:

$$tdc_{out}[k] = \text{fix}\left(\frac{\Delta t[k]}{K_{tdc}}\right) \cdot f_{out} \cdot K_{tdc}, \quad (3.15)$$

¹fix(x) is a function which rounds x to the nearest integer.

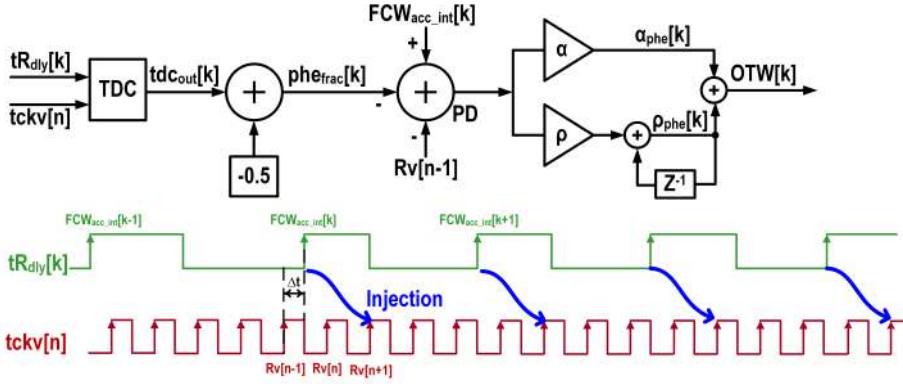


Figure 3.3: Phase detection mechanism.

where the K_{tdc} is the TDC resolution¹. The linearity of the TDC is not modeled here since the TDC works close as a bang-bang phase detector due to the help from the CF-DTC in the phase-locked state. In principle, a single flip-flop could be used as the fractional phase error detector. However, the associated locking time is typically long due to the highly nonlinear loop. On the other hand, a full range TDC can detect the fractional phase error up to 1 DCO period. Hence, the DPLL can observe a linearly quantized phase error, leading to a fast locking time. To avoid the metastability problem, we can make the TDC only toggle up and down around the middle code in the digital domain, which means that $tckv[n]$ leads $tR_{dly}[k]$ around half DCO period in the locked state. The new fractional phase error is expressed as:

$$phe_{frac}[k] = tdc_{out}[k] - 0.5^2. \quad (3.16)$$

Consequently, the total phase error sent to loop filter is

$$phe[k] = FCW_{acc_int}[k] - Rv[n - 1] - phe_{frac}[k]. \quad (3.17)$$

The loop filter is composed of a fast proportional path and a slow integral path. The phase error filtered by the proportional path is

$$\alpha_{phe}[k] = \alpha \cdot phe[k], \quad (3.18)$$

where α is the proportional gain. The phase error filtered by the integral path is

$$\rho_{phe}[k] = \rho \cdot phe[k] + \rho_{phe}[k - 1], \quad (3.19)$$

where ρ is the integral gain. Consequently, the loop filter output denoted as normalized tuning word $NTW[k]$ can be expressed as:

$$NTW[k] = \alpha_{phe}[k] + \rho_{phe}[k]. \quad (3.20)$$

$NTW[k]$ is then used to tune the oscillator frequency.

¹Assume that the calibrated resolution is equal to the real one.

²Assume that the TDC dynamic range is one DCO period.

3.1.4 Injection-Locked DCO

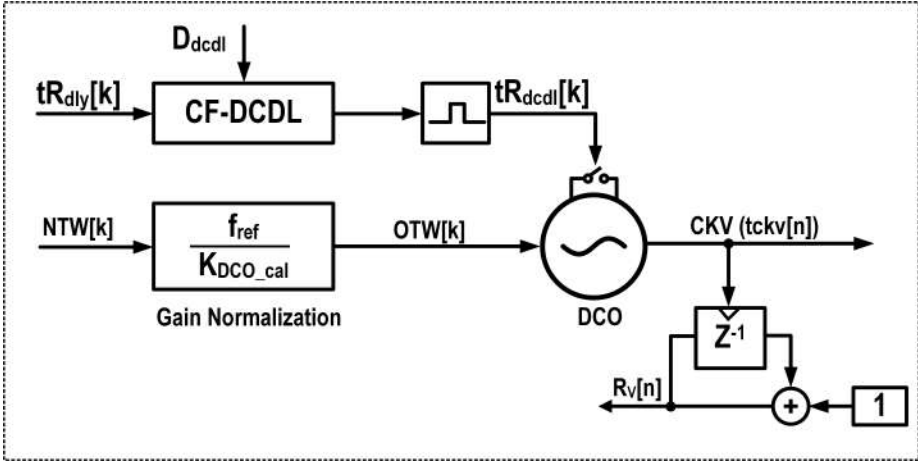


Figure 3.4: Block diagram of the normalized DCO.

Figure 3.4 shows the simplified block diagram of the injection-locked DCO with gain normalization. The integer part of the DCO phase is denoted as $R_v[n]$ and can be expressed as:

$$R_v[n] = R_v[n - 1] + 1, \quad (3.21)$$

where n is the oscillator clock index. The normalized tuning word $NTW[k]$ changes the DCO frequency at the reference frequency rate. To generate the oscillator tuning word $OTW[k]$, the DCO gain should be calibrated and normalized. Namely, the DCO frequency will change by f_{ref} if the integer part of $NTW[k]$ changes by 1 LSB¹. Consequently, the oscillator tuning word $OTW[k]$ can be expressed as:

$$OTW[k] = \frac{f_{ref}}{K_{DCO_cal}} NTW[k], \quad (3.22)$$

where K_{DCO_cal} is the calibrated gain of the DCO [3]. After applying the DCO tuning word, the corresponding frequency variation can be calculated by:

$$\Delta f_v[n] = K_{DCO} \cdot fix(OTW[k]). \quad (3.23)$$

where K_{DCO} is the real gain of the DCO. It is more fruitful to convert the frequency variation to phase variation since the DPLL operates in phase domain. Assume that the free-running frequency of the oscillator is f_{free} ; then the period can be expressed as:

$$t_{free} = \frac{1}{f_{free}}. \quad (3.24)$$

¹Assume that K_{DCO_cal} is equal to the real gain.

The oscillator phase variation ($\Delta t_v[n]$) right after the frequency tuning can be calculated as:

$$\Delta t_v[n] = \frac{\Delta f_v[n]}{f_{free}(f_{free} + \Delta f_v[n])}. \quad (3.25)$$

After the frequency tuning, the oscillator goes into the free-running mode. We can model the oscillator timestamps as:

$$t_{ckv}[n] = t_{ckv}[n-1] + t_{free} - (\Delta t_v[n]); \quad (3.26)$$

Now, the oscillator jitter should be added to Equation (3.26). Assuming the oscillation frequency of f_{out} and the phase noise of PN_{dco} at an offset frequency Δf , the jitter variance can be calculated as:

$$J_{dco} = \frac{\Delta f \sqrt{10 \frac{PN_{dco}}{f_{out}}}}{f_{out}}. \quad (3.27)$$

Consequently, Equation (3.26) can be rewritten as:

$$t_{ckv}[n] = t_{ckv}[n-1] + t_{free} - \Delta t_v[n] + \text{normrnd}(0, J_{dco}). \quad (3.28)$$

The injection occurs right after detecting the phase error. At the injection moment, the oscillator phase is partially determined by the phase of the pulse generator. The oscillator phase right at the injection moment can be expressed as:

$$t_{ckv}[n] = t_{ckv}[n] + \beta(t_{Rdy}[k] + t_{dcdl}[k] - t_{ckv}[n]), \quad (3.29)$$

where $t_{dcdl}[k]$ is the delay of the CF-DCDL and pulse generator, and β is the injection strength². After the injection, the oscillator is free-running again and its timestamps are governed by Equation (3.28).

3.1.5 Time Offset Calibration

Figure 3.5 shows the block diagram of the time offset calibration. The 1b TDC outputs 1 (-1) when the oscillator phase leads (lags) the injection phase. An accumulator is then used to accumulate the 1b TDC output $D_{out}[k]$. The accumulator output is simply expressed as:

$$D_{acc}[k] = D_{acc}[k-1] + D_{out}[k]. \quad (3.30)$$

Finally, this code is averaged and then used to control the delay of the CF-DCDL. We get the following CF-DCDL control code:

$$D_{dcdl}[k] = \text{mean}(D_{out}[k])^3. \quad (3.31)$$

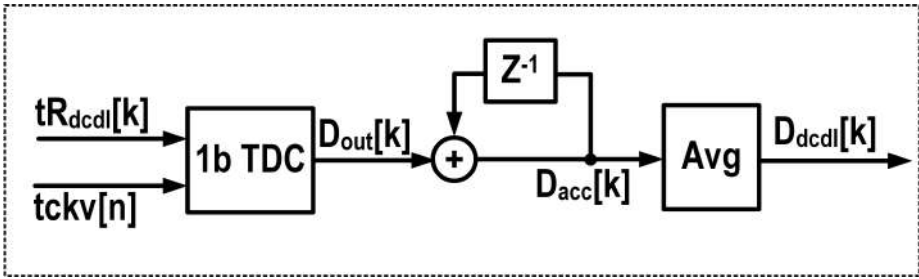


Figure 3.5: Block Diagram of the Time Offset Calibration.

Table 3.1: IL-DPLL simulation parameters.

Injection-Locked DPLL Configuration	
Reference Frequency f_{ref}	64MHz
FCW	$38 + \frac{1}{2^{12}}$
f_{out}	2432MHz + 15.625kHz
f_{free}	2332MHz
Proportional Gain α	2^{-2}
Integral Gain ρ	2^{-5}
TDC Resolution K_{tdc}	12ps
C-DTC Resolution K_{C-DTC_cal}	4.5ps
F-DTC Resolution K_{F-DTC_cal}	0.5ps
F-DCDL Resolution K_{F-DCDL}	0.5ps
DCO PN @1MHz	-86dBc/Hz
DTC and DCDL PN @1MHz	-143dBc/Hz

3.2 System-Level Simulation in MATLAB

In this section, the simulated results of the IL-DPLL are presented. The main system parameters used for simulation are summarized in Table 3.1. The reference frequency is 64MHz and the IL-DPLL needs to generate a fractional frequency 2.432015625GHz. There is a ~ 100 MHz frequency difference between the initial frequency (f_{free}) and the target frequency (f_{out}).

3.2.1 Locking Behavior

Figure 3.6 shows the simulated locking behavior of the IL-DPLL. The initial frequency of the DPLL is ~ 2.332 GHz and the frequency in the locked state is

¹Using Equation (1.3).

² $t_{ckv}[n]$ appearing at the right side of the equation is governed by Equation (3.28); only one injection path is modeled.

³ $\text{mean}(x)$ is a function which calculates the mean value of x .

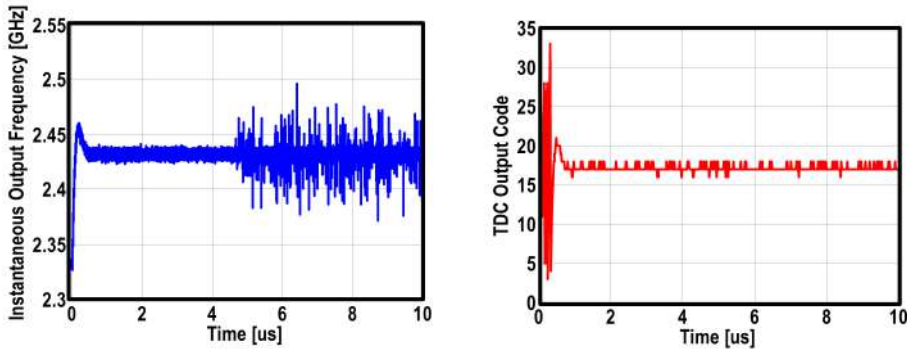


Figure 3.6: The simulated settling behavior.

2.432015625GHz. The DPLL has an estimated locking time $1\mu\text{s}$ using the loop filter parameters listed in Table 3.1. The injection is applied at $5\mu\text{s}$. In the phase-locked state, thanks to the help from the CF-DTC, the output of the TDC toggles between 16 and 18. Hence, the dynamic range of the TDC could be reduced.

3.2.2 Time Offset Calibration

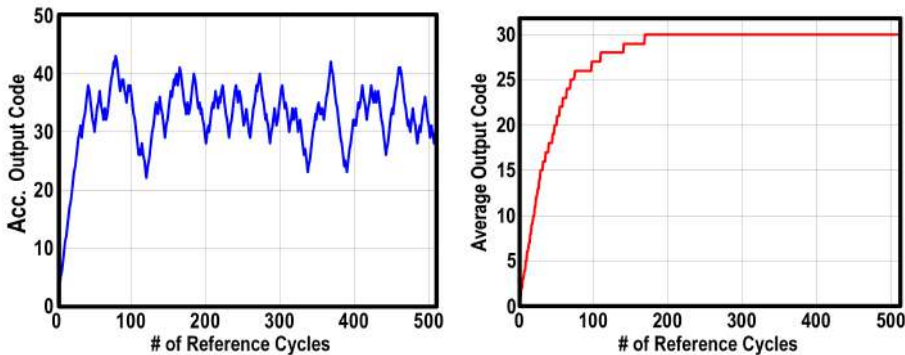


Figure 3.7: The simulated accumulator and average outputs.

Figure 3.7 shows the simulated accumulator output and its average value over 128 samples. Initially, due to the systematic time offset, the 1b TDC output is 1. Hence, the accumulator output keeps increasing to change the delay of the F-DCTL. Then, the 1b TDC output jumps between 1 and -1 due to the system random noise. Therefore, the accumulator also exhibits an erratic behavior. The right part of Figure 3.7 shows the simulated output of the averaging block. Finally, the averaged code converges to 30. The averaging cycle is chosen as the power of 2 to facilitate hardware implementation. To reduce the overall settling of the DPLL, the averaging cycle is 128 ($2\mu\text{s}$) in this design.

3.2.3 Phase Noise and Spectrum

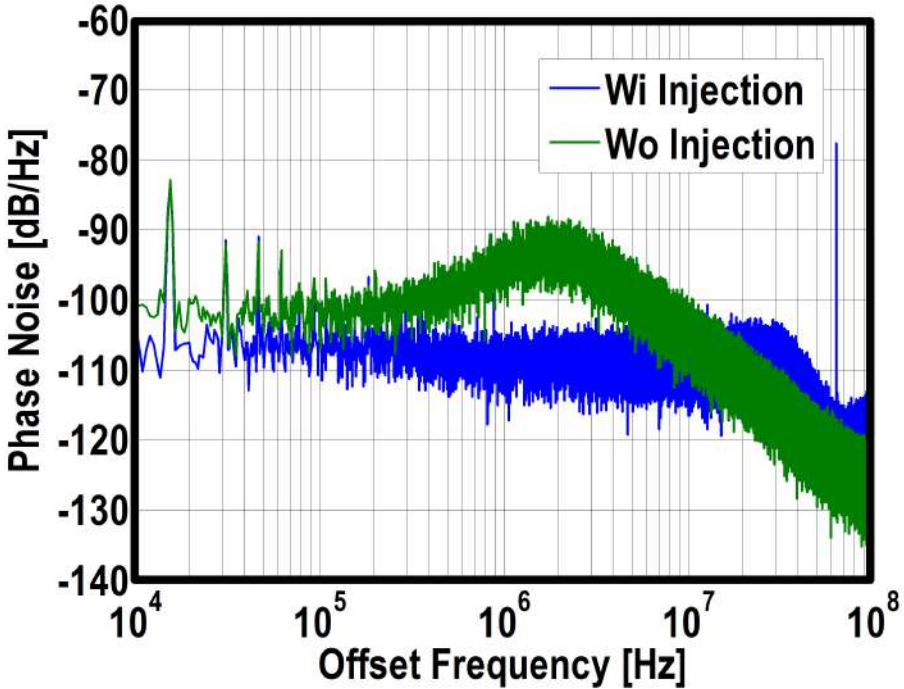


Figure 3.8: Simulated IL-DPLL phase noise.

Figure 3.8 shows the simulated phase noise. The DPLL is simulated over 200000 reference ($\sim 3\text{ms}$) cycles. To get the phase noise plot, the PSD function with 22² windowed data is used. The phase noise at 1MHz without injection locking is $\sim -95\text{dBc/Hz}$; and it is improved to $\sim -108\text{dBc/Hz}$ after the injection. The integrated jitter from 10kHz to 10MHz without injection locking is 4ps; and it is significantly improved to 1.2ps after the injection. Due to the injection-locking operation, 3dB high-frequency gain and reference spur are also observed.

Figure 3.9 shows the simulated spectrum. There is no reference spur for the DPLL itself. The reference spur after injection locking is -49dBc , which translates to $\sim 1\text{ps}$ time offset between the DCO and injection edges. From the spectrum plot, the in-band noise reduction and 3dB out-band noise degradation are also observed for the IL-DPLL. The close-in fractional spur at 15.625kHz is -47dBc . This fractional spur comes from the non-linearity of the coarse-DTC. From [4], the spur level is given as:

$$Frac_{spur} = 10 * \log_{10}\left(\left(\frac{\pi^2}{4}\right) \cdot \left(\frac{\Delta_{INL}}{t_{out}}\right)^2\right), \quad (3.32)$$

where Δ_{INL} is the peak-to-peak INL of the C-DTC. The resulted Δ_{INL} is $\sim 1.2\text{ps}$.

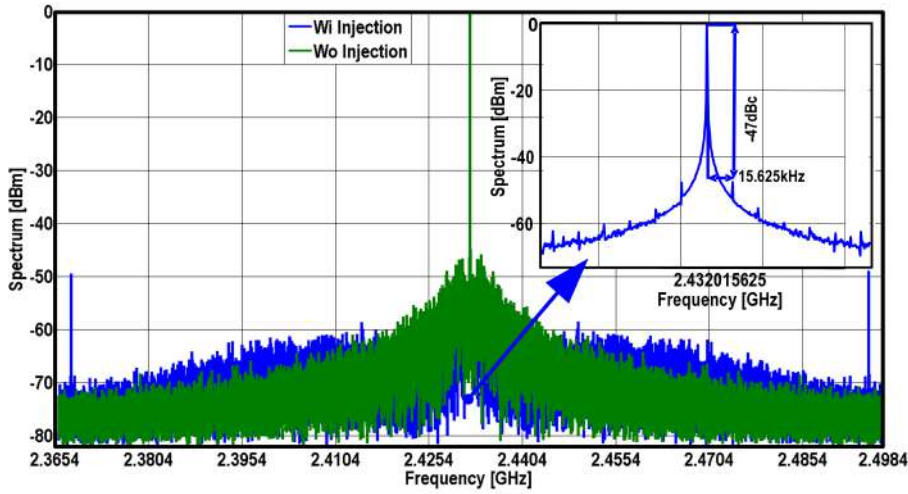


Figure 3.9: Simulated IL-DPLL spectrum.

3.3 S-Domain Phase Noise Model

The detailed s-domain modeling of an IL-PLL is presented in [5], and the s-domain modeling of a DPLL is presented in [6]. Figure 3.10 shows the phase noise model of an IL-DPLL. Due to the injection operation, there are two mechanisms (i.e., traditional phase-locked loop and the injection) that change the oscillator phase. $H_{ri}(s)$ is used to represent the injection effect on the oscillator phase [5]. It is expressed as:

$$H_{ri}(s) = 1 - \frac{\beta}{1 + (\beta - 1)e^{-st_{ref}}} e^{-st_{ref}/2} \frac{\sin(\omega t_{ref}/2)}{\omega t_{ref}/2}. \quad (3.33)$$

Since the oscillator phase is replaced by the reference phase at the injection moment, $H_{up}(s)$ should be used to represent up-conversion of the reference noise to the DCO output [5]. It is

$$H_{ri}(s) = \frac{FCW\beta}{1 + (\beta - 1)e^{-st_{ref}}} e^{-st_{ref}/2} \frac{\sin(\omega t_{ref}/2)}{\omega t_{ref}/2}, \quad (3.34)$$

where t_{ref} is the reference period and ω is the angular frequency. Due to the injection operation, the DTC and reference noise (φ_{DTC} and φ_{ref}) appear directly at the in-band of the IL-DPLL. However, the TDC noise (φ_{TDC}) is suppressed by the injection operation.

3.4 Summary

The behavioral level modeling and simulation of the proposed IL-DPLL are carried out in this section. The time-domain simulations proved the functionality of the

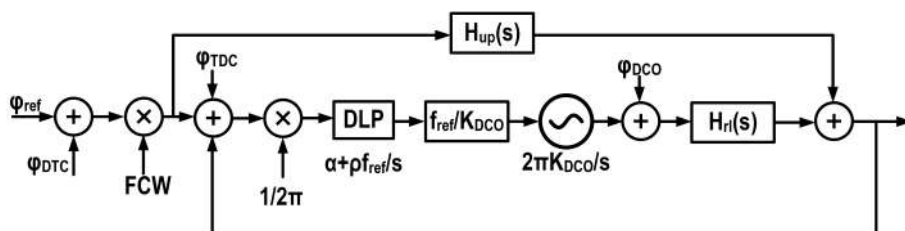


Figure 3.10: Phase noise model of an IL-DPLL.

3

IL-DPLL. The s-domain phase noise model is also presented. The implementation of the various analog blocks is presented in next chapter.

References

- [1] I. L. Syllaios and R. B. Staszewski, "Time-Domain Modeling of a Phase-Domain All-Digital Phase-Locked Loop for RF Applications," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2007, pp. 861–864.
- [2] I. L. Syllaios, et al., "Time-Domain Modeling of an RF All-Digital PLL," in *IEEE Trans. Circuits Syst. II*, 2008, pp. 601–604.
- [3] R. B. Staszewski, et al., "Event-Driven Simulation and Modeling of Phase Noise of an RF Oscillator," in *IEEE Trans. Circuits Syst. I*, 2005, pp. 723–733.
- [4] S. Levantino and C. Samori, "Nonlinearity Cancellation in Digital PLLs," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2013, pp. 1–8.
- [5] S. Ye, et al., "A Multiple-Crystal Interface PLL with VCO Realignment to Reduce Phase Noise," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 1795–1803, Dec. 2002.
- [6] R. B. Staszewski, *Digital Deep-Submicron CMOS Frequency Synthesis for RF Wireless Applications*. Phd thesis, University of Texas, Dallas, 2002.

4

Analog/RF Design

The transistor-level design (schematic and physical layout) of analog building blocks of the IL-DPLL is presented in this chapter. These include digitally-controlled ring oscillator (DC-RO), coarse-fine DTC (CF-DTC), coarse-fine DCDL (CF-DCDL), fixed delay, pulse generator and 1b TDC.

4.1 Digitally-Controlled Ring Oscillator

The digitally-controlled ring oscillator (DC-RO) is the frequency generation block of the IL-DPLL. It takes the oscillator tuning word (OTW) as the input and generates an RF clock with a frequency change proportional to the OTW. Considerable design effort should be performed on this section since the DPLL performance is largely dependent upon the DC-RO. The analysis and design of the DC-RO regarding phase noise, power consumption, supply and interference sensitivity are presented in this section.

4.1.1 Oscillator Core

There are mainly three different (single-ended, fully-differential and pseudo-differential) topologies for ring oscillators [1]. Their performance is first compared, and a suitable architecture for the IL-DPLL is then chosen. Figure 4.1 top-left shows an N-stage single-ended ring oscillator. Each delay cell is composed of an inverter with a device channel length L and effective channel width of W_{eff} ¹. Its free-running frequency can be approximately calculated as:

$$f_{osc} \approx \frac{\mu_{eff} W_{eff} C_{ox} (\frac{V_{DD}}{2} - V_T)^2}{8\eta N L V_{DD} C_{node}}, \quad (4.1)$$

where μ_{eff} is the effective mobility of electron and hole, η is a constant (close to 1), C_{ox} is the gate-oxide capacitance per unit area, and C_{node} is the node capacitance

¹Sum of NMOS width and PMOS width.

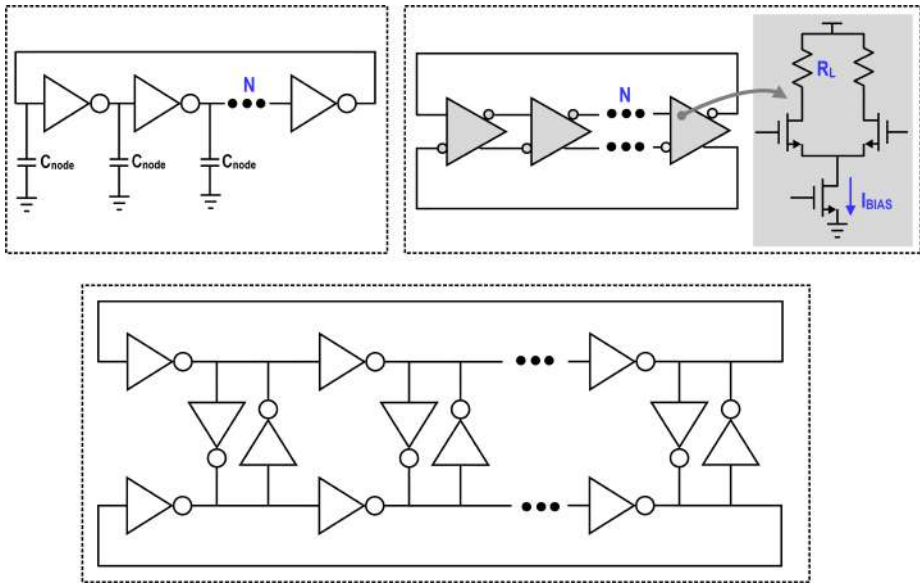


Figure 4.1: Three RO topologies.

[2]. Consequently, the design parameters for the oscillation frequency of an SE-RO are the device ratio, supply voltage, node capacitance and the number of stages. For the frequency tuning, one can change the supply voltage and node capacitance of the RO. The channel length L and number of delay stages N are usually designed to set the center frequency of the RO. Larger L and N reduce the oscillation frequency. It should be noted that N should be an odd number to guarantee an SE-RO oscillates.

The phase noise¹ of an SE-RO with an oscillation frequency f_{osc} and a DC power consumption P at an offset frequency Δf can be approximately expressed as:

$$PN_{free_single}(\Delta f) \approx \frac{16\gamma}{3\eta} \cdot \frac{KT}{P} \cdot \left(\frac{f_{osc}}{\Delta f}\right)^2, \quad (4.2)$$

where γ is the transistor noise factor ($\frac{2}{3}$ for a long channel device) and KT is the thermal energy [2]. For a free-running RO operating at a given frequency f_{osc} , the only way to improve its phase noise is to increase the DC power consumption. RO's phase noise improves 3dB by 2x increase in its power consumption.

For an N -stage fully-differential RO (FD-RO) as shown in the top right of Figure 4.1, its free-running frequency can be calculated as:

$$f_{osc} = \frac{1}{2NT_D}, \quad (4.3)$$

where T_D is the delay of its each delay cell. The delay of each cell is mainly determined by the load resistance and capacitance. Hence, the oscillation frequency is

¹Only the thermal noise is considered.

less sensitive to supply voltage. One can change its oscillation frequency by tuning the node capacitors or resistors. The phase noise of an FD-RO can be approximately expressed as:

$$PN_{free_diff}(\Delta f) \approx \frac{8}{3\eta} N \frac{KT}{P} \left(\frac{V_{DD}\gamma}{V_{GS} - V_T} + \frac{V_{DD}}{R_L I_{tail}} \right) \cdot \left(\frac{f_{osc}}{\Delta f} \right)^2, \quad (4.4)$$

where I_{bias} is the bias current and R_L is the load resistance [2]. Again, the phase noise of an FD-RO is directly related to its power consumption. The FD-RO is rarely used in a low-power DPLL because its phase noise is much worse than that of an SE-RO. For example, if we assume $N = 3$, $\gamma = 2$, $V_{GS} - V_T = 0.2V$, $V_{DD} = 1V$ and $R_L I_{tail} = 1V$, the phase noise of an FD-RO is about 8dB worse than that of an SE-RO even without considering the up-converted noise of the bias current source. An FD-RO should consume $\sim 8x$ more power to achieve the same performance as an SE-RO.

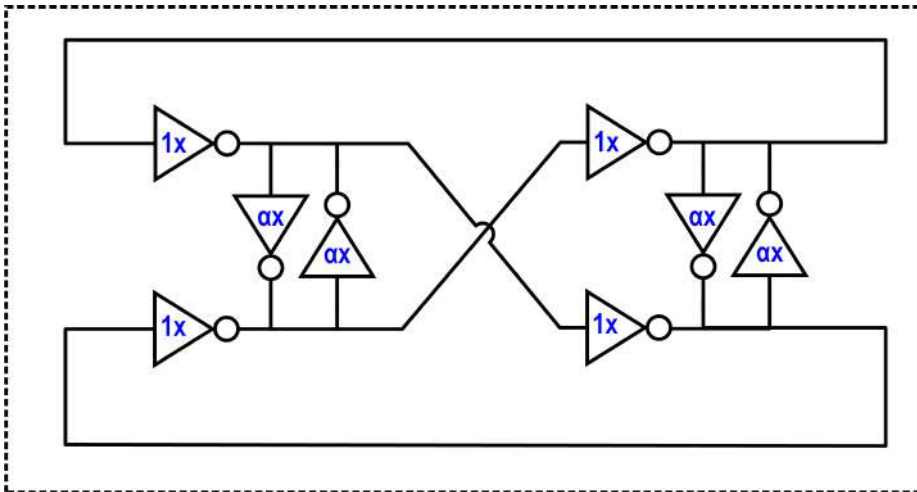


Figure 4.2: Schematic of the RO core.

In summary, the oscillation frequency of an FD-RO is independent of the supply voltage while an SE-RO exhibits a large supply pushing. However, an SE-RO has a better trade-off between the phase noise and power consumption. On the other hand, if there is any parasitic coupling to the internal node of an SE-RO, its frequency and phase will be disturbed, degrading the jitter performance. Due to this reason, an SE-RO is also not popular to be employed in a DPLL. To suppress the parasitic coupling, a pseudo-differential RO (PD-RO) is often used as shown in the bottom of Figure 4.1. Due to the existence of the cross-coupled inverters, the internal nodes of a PD-RO have a differential swing. Hence the common-mode interference from other blocks can be rejected. It should be noted that this architecture still suffers from supply pushing.

The number of the delay stages (N) should be carefully chosen. If N is too large, the oscillation frequency will be too low, failing to cover the desired frequency range.

On the other hand, the layout will be more complicated with larger N , increasing parasitics and reducing oscillation frequency. In this design, a two-stage PD-RO is implemented to make the routing wires shorter and parasitic capacitance smaller [3] as shown in Figure 4.2.

Careful attention must be paid to pick the device ratio (α) between the cross-coupled inverter pairs and the main inverters. The PD-RO is simplified to a four-stage SE-RO when $\alpha = 0$, resulting in the oscillation failure. Consequently, we should make $\alpha > 0$ to make sure that the circuit can sustain oscillation. The value of α , in reality, determines the possibility of the oscillation start-up, and a larger α facilitates oscillation start-up. On the other hand, larger α makes the cross-coupled inverters add more loading to the main inverters, degrading power consumption. $\alpha = 0.7$ is chosen in this design by considering both oscillator start-up margin and its power consumption.

4

4.1.2 Oscillator Frequency Tuning and Injection Locking

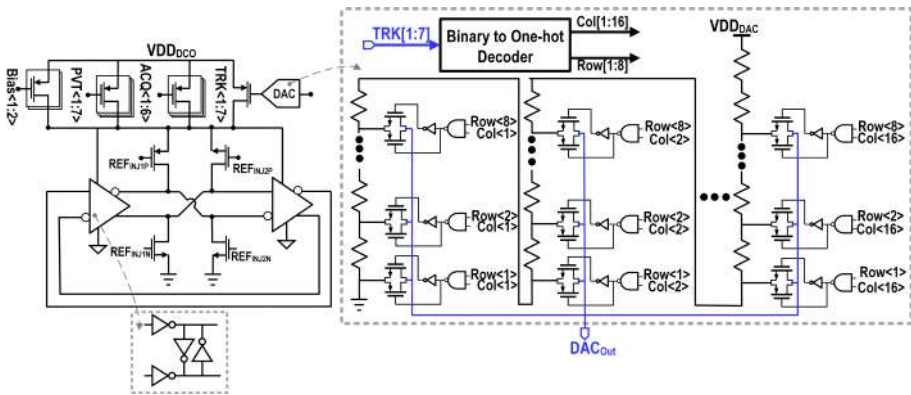


Figure 4.3: Schematic of the IL-DCO.

As discussed before, one can change the supply voltage or capacitive load of the PD-RO to tune the frequency. The capacitance tuning is not used because it limits the maximum oscillation frequency of the circuit. The off-state and routing parasitic capacitances degrade the oscillation frequency. Another consideration is that the fine-frequency tuning through the capacitance is usually difficult. The frequency tuning in this design is realized by changing the oscillator supply voltage through PMOS current sources.

The schematic of the RO with frequency tuning banks is shown in Figure 4.3. To cover the entire desired frequency range, two extra PMOS current sources are implemented. The PVT bank is realized by a 7b binary-weighted PMOS current DAC. It is used to coarsely tune the RO frequency and to cover the tuning range of the frequency synthesizer. For the acquisition bank, a 6b binary-weighted PMOS current DAC is used for the medium frequency tuning. For the fine frequency tuning in the tracking bank, a 128b resistive flash DAC and a PMOS transistor are used to tune the oscillator supply. To expedite the layout of the DAC, the row-column decoding

is employed. The 7 binary control bits of the tracking bank are segmented to 3 MSBs for the row control and 4 LSBs for the column control. Two binary to one-hot decoders are implemented to control the DAC voltage unarily.

The injection locking is realized by pulling and pushing the differential nodes of the RO down to the ground and up to the oscillator supply respectively. The transistors of the injector pair (REF_{INJ1N}) are sized 5 times larger than the main delay cells of the RO to maximize the injection strength. Even after the injection, some jitter still remains mainly because of the timing uncertainty of the injection signal due to the system noise. As a result, the in-band phase noise of the IL-DPLL is degraded. To resolve this issue, a second injection pair (REF_{INJ2N}) is implemented to replace the DCO edge after 10 RF cycles from the first injection as shown in Figure 4.3. Hence, the remaining jitter due to the limited strength of the first injector pair REF_{INJ1N} is suppressed.

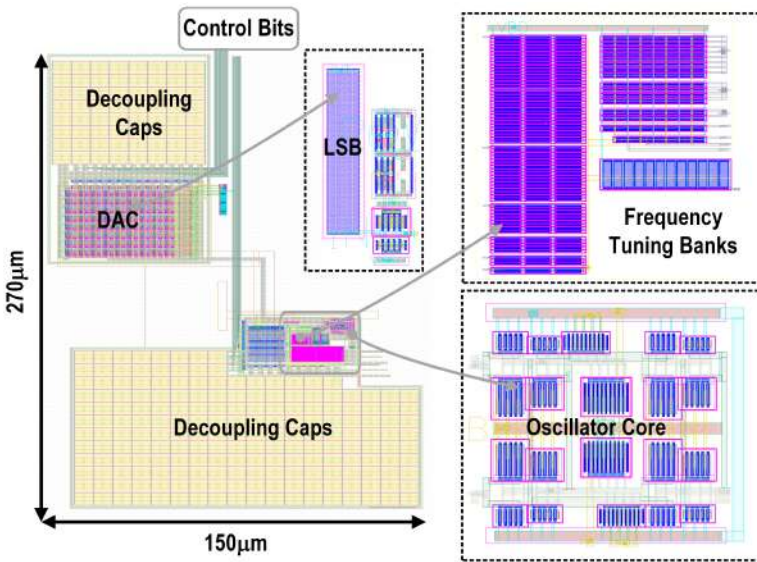


Figure 4.4: Layout of the IL-DCO.

The layout of the implemented DCO is shown in Figure 4.4, where the layout of the oscillator core, frequency tuning banks, and DAC LSB is also shown. The area of the oscillator core and frequency tracking banks is very small. To reduce the coupling from other blocks and filter the supply noise, $\sim 40\text{pF}$ decoupling capacitors are added to the supplies of the oscillator core and DAC. Hence decoupling capacitors dominate the DCO area. The DAC and the oscillator core are placed $\sim 50\mu\text{m}$ away in the layout to reduce the coupling of the DAC to the oscillator core.

4.1.3 Post-Layout Simulation Results

This section shows the post-layout simulation results of the implemented DCO. A 3nH inductor is connected to the 1V supply of the DCO to model the bond wire

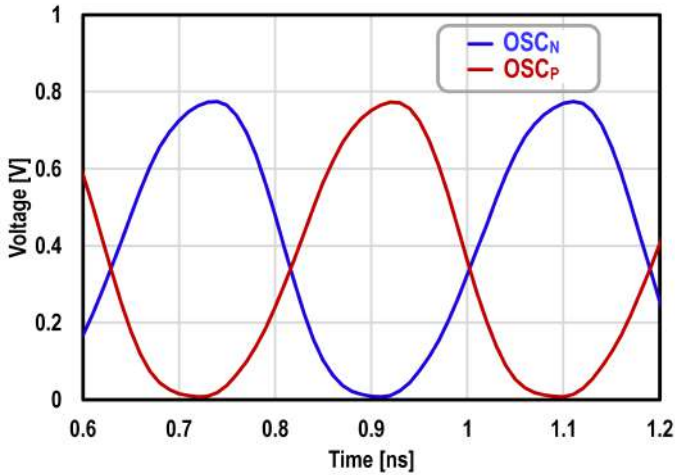


Figure 4.5: Simulated voltage waveform of the free-running DCO.

effects. Besides, the supply noise from an off-chip LDO is also modeled by the thermal noise generated from a 60M ohm resistor. Two 30fF capacitors are connected to the oscillator outputs to model the loading of the DCO. The parasitics from the layout of the DCO are extracted using Quantus QRC. The post-layout simulation is under a typical corner at 25°C.

Figure 4.5 shows the simulated differential waveforms of the DCO operating at 2.432GHz from the transient simulation. The peak-to-peak swing is $\sim 0.8\text{V}$ and the oscillator consumes $520\mu\text{W}$ DC power at this frequency.

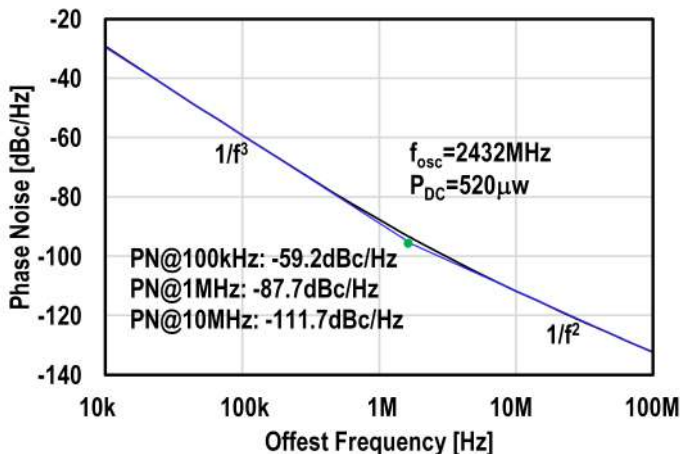


Figure 4.6: Simulated phase noise of the free-running DCO.

To simulate the phase noise of the DCO, a period-steady-state (PSS) is first run

to find the periodic operating point and then Pnoise is run to calculate the phase noise. Figure 4.6 shows the simulated free-running phase noise of the implemented DCO. The simulated phase noise at the 1MHz offset frequency is -87.7dBc/Hz, which translates to an oscillator FOM of -157.6dB, where the FOM [4] is defined as:

$$FOM(\Delta f) = PN(\Delta f) + 20 * \log_{10}\left(\frac{\Delta f}{f_{osc}}\right) + 10 * \log_{10}\left(\frac{P_{DC}}{1mW}\right). \quad (4.5)$$

This FOM is ~30dB worse than that of LC oscillators. It also should be pointed out that the corner frequency of the flicker noise is a few MHz, which is ~10x larger than that of the state-of-the-art LC oscillator.

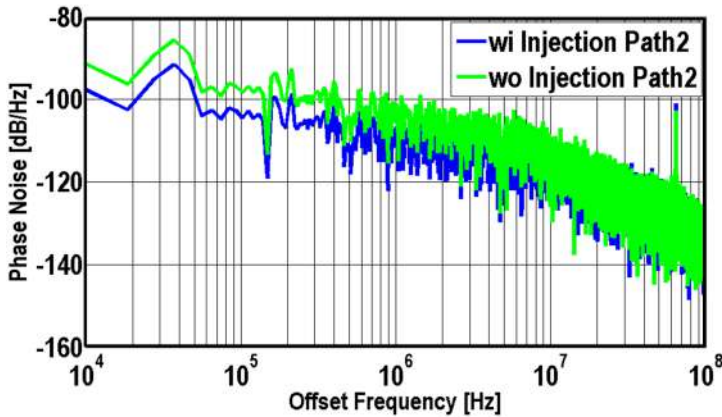


Figure 4.7: Simulated phase noise of the IL-DCO.

Figure 4.7 shows the simulated phase noise of the DCO after injection locking with and without the second injection path. The second injection phase is generated by delaying the first injection phase 10 DCO cycles. Transient noise simulation in SpectreRF is used to calculate the phase noise. The "Noise Fmax" is chosen as 20GHz and the "Noise Fmin" is chosen as 1kHz to capture the high-frequency noise and flicker noise. The "Noise Update" is chosen as "step" with 1ps "Noise Tmin" for an accurate noise generation. Then, the 0.4V-crossing point of the oscillator waveform is sampled and its corresponding timestamp is recorded. Finally, the timestamp is used for the phase noise plot. The transient noise simulation time is 0.2ms. Figure 4.7 shows that the phase noise improves from -103dBc/Hz to -109dBc/Hz at the 1MHz offset frequency when the second injection path is on. Besides, the integrated jitter between 10kHz-10MHz is improved from 1.1ps to 0.6ps. Consequently, the proposed technique can reduce the in-band phase noise of the IL-DPLL effectively.

4.2 Coarse-fine DTC

The coarse-fine DTC (CF-DTC) enables the fractional-N operation of the DPLL even when the oscillator is injection-locked. It delays the rising edge of the reference clock $FREF$ and generates the delayed reference clock $FREF_{DLY}$. Its delay is proportional to the control code plus a delay offset. Since the reference phase noise appears inside the bandwidth of the injection-locked oscillator with a multiplication factor of FCW^2 , a major attention must be paid to minimize the phase noise of the CF-DTC. This section deals with the analysis and design of the CF-DTC regarding resolution, linearity, phase noise, and power consumption.

4.2.1 CF-DTC Architecture

There are mainly two popular DTC topologies used in DPLLs [5, 6] as shown in Figure 4.8. The switched-buffer-based architecture has a resolution about two

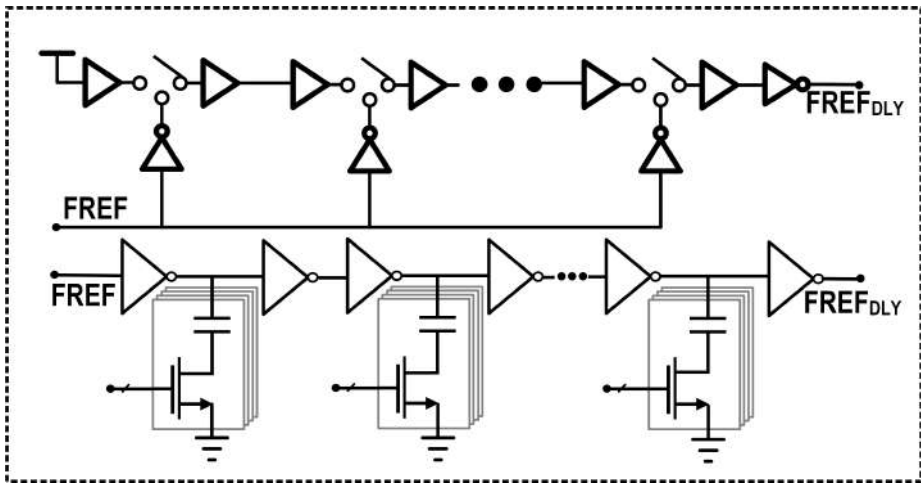


Figure 4.8: Two popular DTC architectures.

inverters' delay (i.e., $\sim 15\text{ps}$ in this process). Due to the mismatch of the inverters and parasitic off-capacitances of the delay cells, the integral nonlinearity (INL) of this architecture is limited to a few ps [5]. On the other hand, superior phase noise and power consumption can be obtained since the dynamic range is approximately equal to the maximum absolute delay of the DTC. However, due to the linearity and resolution issues, the switched-capacitor (SC) -based architecture is used in this design.

The resolution of an SC-based architecture can be approximately calculated as:

$$t_{res_{dtc}} \approx \frac{C_{res} V_{DD}}{2I_N}, \quad (4.6)$$

where I_N is the device's saturation current and C_{res} is the resolution of switched

¹Assume that the devices' current is constant during switching.

capacitance. A sub-ps resolution can be achieved by this architecture, since two design parameters C_{res} and I_N are available. For example, 1ps resolution can be realized if $C_{res} = 1fF$ and $I_N = 1mA$ are picked. The drawback of this architecture is that its dynamic range is only a small portion of the total absolute delay. Due to the extra delay in the signal path, excessive phase noise is introduced. Hence, in general, this architecture has worse power and phase noise trade-off. However, excellent linearity performance can be achieved since the dominant source of the nonlinearity is the off-state capacitance of the switches, which can be very small compared with the switched unit capacitance.

4.2.2 CF-DTC Design

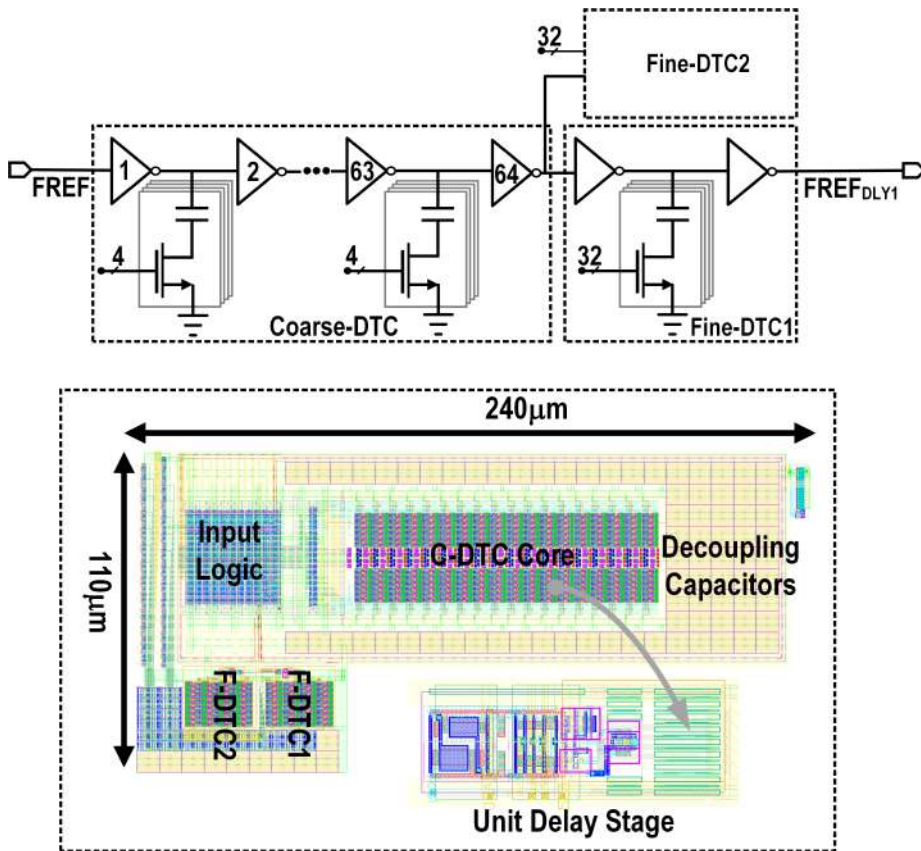


Figure 4.9: Schematic and layout of the CF-DTC.

Figure 4.9 shows the schematic and layout of the implemented CF-DTC. The C-DTC has to cover 1 DCO period ($\sim 400ps$) under the worst PVT corners. It has 32 stages, where each stage consists of two inverters and a 2b MOM capacitor bank with a resolution of 1.2fF [6]. The selection of the segmentation (i.e., the number

of switched-capacitors in each delay stage and the number of delay stages) of the C-DTC is based on a careful trade-off between the INL and phase noise.

The F-DTC needs to cover 1 LSB of the C-DTC under the worst PVT corners. Only one delay stage with 32 switched-MOM capacitors and capacitance resolution of 1.2fF is used to minimize the phase noise of F-DTC. To achieve a fine resolution, a large delay cell is used. It draws a large peak current and can heavily modulate the supply of the C-DTC, degrading its linearity. To reduce this supply variation, an identical F-DTC (F-DTC2) with complementary control code is added to the design. Now, the F-DTCs together draw constant current from the supply during switching of the F-DTC code. Hence, the C-DTC observes the same supply variation even when the control code of F-DTC1 changes.

At the falling edge of the reference clock($FREF$), the CF-DTC control code is retimed to avoid any potential racing or glitches. This guarantees the control code of the CF-DTC settles down when the next rising edge of $FREF$ comes.

4.2.3 Post-Layout Simulation Results

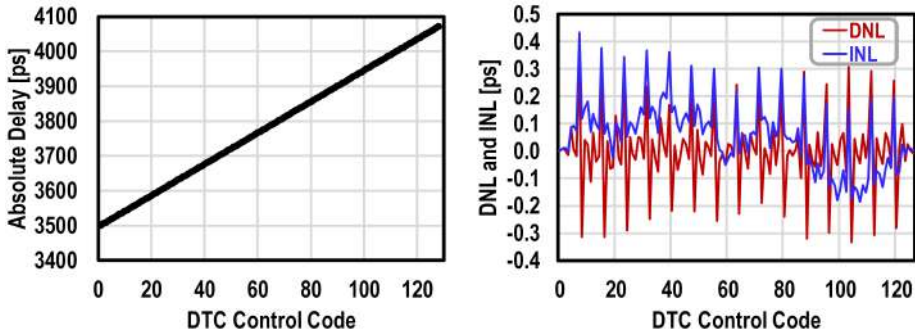


Figure 4.10: Simulated delay and linearity of the C-DTC.

The post-layout simulation results of the implemented CF-DTC are presented in this section. The parasitics from the layout of the CF-DTC are extracted using Quantus QRC, and the post-layout simulation is under a typical corner with 1V supply at 25°C. The simulated power consumption of the CF-DTC is 72 μ W. Figure 4.10 shows the simulated delay and linearity performance of the C-DTC. The delay difference between the maximum and minimum control code is \sim 570ps, which translates to a C-DTC gain of 4.49ps/LSB. This dynamic range sets the lower limit of the synthesized fractional frequency (e.g., 1.75GHz). The C-DTC has an absolute delay of 3.5ns even when the control code is 0. The right part of Figure 4.10 shows the simulated integral nonlinearity (INL) and differential nonlinearity (DNL) of the C-DTC. For the linearity simulation, only the systematic mismatch of the devices are considered. The peak-to-peak INL is smaller than 0.7ps and the peak-to-peak DNL is smaller than 0.6ps. The nonlinearity of the C-DTC will introduce fractional spurs in the PLL output spectrum.

Figure 4.11 shows the simulated delay and linearity of the F-DTC. The dynamic

range of the F-DTC is $\sim 15\text{ps}$, which translates to an F-DTC gain of 0.5ps/LSB . The peak-to-peak INL is $\sim 150\text{fs}$ and the peak-to-peak DNL is $\sim 50\text{fs}$, which does not degrade the linearity of the CF-DTC significantly.

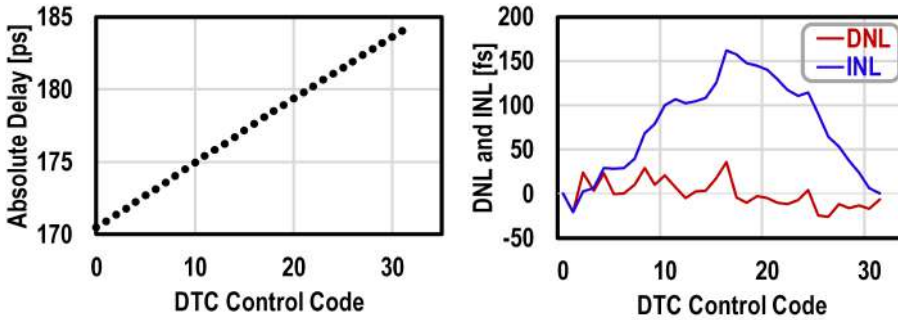


Figure 4.11: Simulated delay and linearity of the F-DTC.

Figure 4.12 shows the simulated phase noise of the CF-DTC with a 64MHz reference. PSS and Pnoise are used to simulate the phase noise. The control code of the CF-DTC is set to the maximum in the simulation. The phase noise is referred to the oscillator output (2432MHz carrier). The CF-DTC exhibits a phase noise floor of -117.2dBc/Hz and a phase noise of -116.7dBc/Hz at the 1MHz offset frequency. The integrated jitter of the CF-DTC between 10kHz-10MHz is 440fs , which is $\sim 0.7\times$ smaller than that of the IL-DCO.

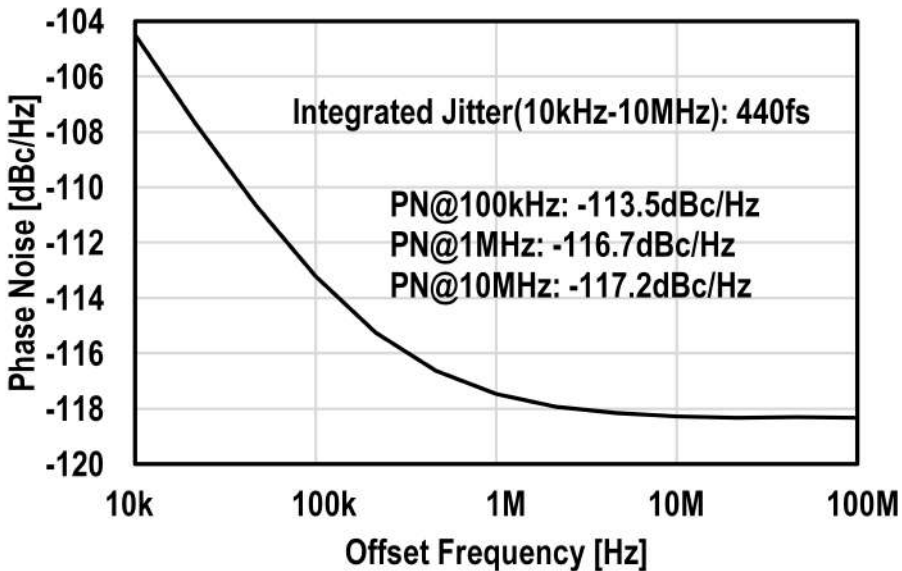


Figure 4.12: Simulated phase noise of the CF-DTC.

4.3 Coarse-fine DCDL and Fixed Delay

The coarse-fine DCDL (CF-DCDL) is used to adjust the timing of the injection pulse for the reference spur mitigation. Besides, it is also employed to delay the injection pulse to avoid the race condition between the PLL and injection path. Similar to a CF-DTC, the CF-DCDL also delays the rising edge of the input clock with a delay proportional to the control code plus a delay offset. Since the IoT packet length is relatively short (i.e., $400\mu\text{s}$), the control code of the CF-DCDL is fixed during the operation. Hence the linearity issue is relaxed. The C-DCDL needs to cover 1 DCO period, and the F-DCDL needs to cover 1 LSB of the C-DCDL under worst PVT corners. In the second injection path, an additional fixed delay is also required to shift $FREF_{DLY1}$ for about 10DCO cycle. Again, the phase noise of the CF-DCDL and fixed delay should be optimized since they are on the reference path.

4

4.3.1 CF-DCDL and Fixed Delay Design

Figure 4.13 shows the schematic of the CF-DCDL and fixed delay¹. The CF-DCDL has a similar architecture as the CF-DTC. The 5b C-DCDL has 4 delay stages, where each stage consists of two inverters and a 3b switched-MOM capacitor bank with a resolution of 6.8fF . In principle, one delay stage could be used for a better noise and power trade-off due to the minimum absolute delay. However, the phase noise will degrade if the transition of the voltage waveform is not sharp especially when all of the capacitors are switched on. Furthermore, it is desired to have a minimum delay larger than 1 DCO period to detect any frequency error. Therefore, 4 delay stages are used for the C-DCDL. The F-DCDL has only one delay stage with 64 switched-MOM capacitors with a resolution of 1.2fF to minimize the jitter. For the F-DCDL, the rise/fall time of the waveform remains sharp since the capacitors' size is small. The fixed delay is realized by inverter chains to achieve the required delay with negligible phase noise degradation.

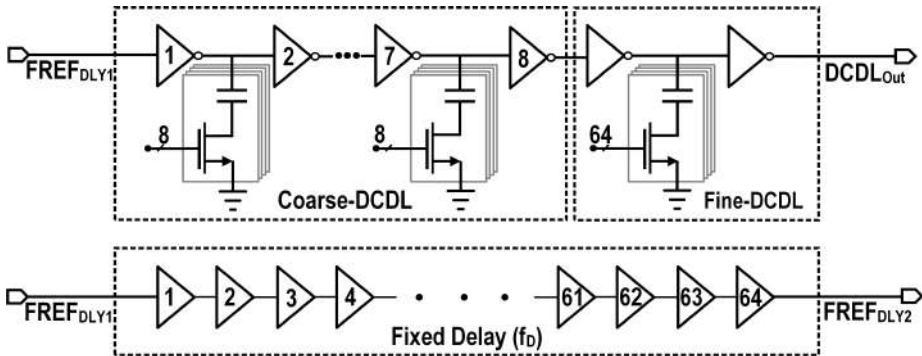


Figure 4.13: Schematic of the CF-DCDL and fixed delay block.

¹The layout will be shown in the later section.

4.3.2 Post-Layout Simulation Results

The parasitics from the layout are extracted using Quantus QRC, and the post-layout simulation is performed to estimate the power consumption, delay, and phase noise of the CF-DCDL and fixed delay. The simulation is under a typical corner with 1V supply at 25°C. A 64MHz clock is used as an input signal for the CF-DCDL and fixed delay. The maximum power consumption of the CF-DCDL from the PSS simulation is 42 μ W.

Figure 4.14 shows the delay of the CF-DCDL from the transient simulation. The simulated dynamic range of the C-DCDL is \sim 700ps, which translates to a C-DCDL gain of \sim 22ps/LSB. The minimum absolute delay of the C-DCDL delay is \sim 900ps, which is large enough to make sure that the injection always occurs after the phase detection. The F-DCDL has a dynamic range of \sim 50ps, which covers more than 2 LSBs of the C-DCDL.

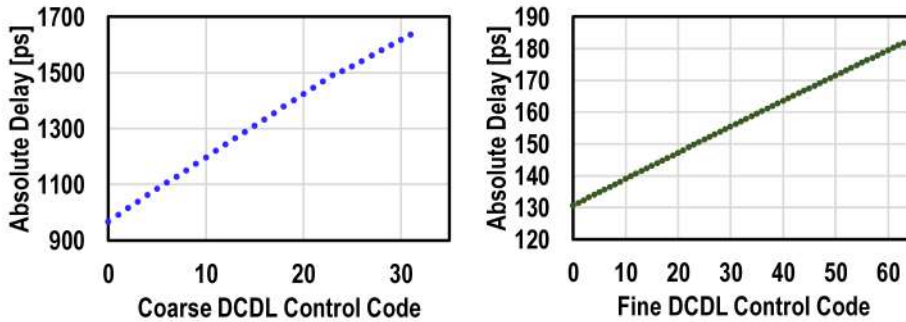


Figure 4.14: Simulated delay of the CF-DCDL.

Figure 4.15 shows the simulated phase noise of the CF-DCDL and fixed delay. The maximum code of the CF-DCDL is used, and the phase noise is referred to the oscillator output (2432MHz). The CF-DCDL has a phase noise floor of -123dBc/Hz and a phase noise of -120dBc/Hz at the 1MHz offset frequency. The integrated jitter from 10kHz to 10MHz is 270fs, which is small enough compared with that of the IL-DCO. The fixed delay (FD) exhibits a phase noise floor of -116dBc/Hz shown in Figure 4.15. The integrated jitter (10kHz-10MHz) of the fixed delay is 441fs. Figure 4.15 also shows the simulated phase of the CF-DCDL and FD together. The integrated jitter (10kHz-10MHz) of is 515fs, which is comparable as that of the IL-DCO. It could be improved by consuming more power. Figure 4.16 shows the simulated phase noise of the injection path1 and injection path2 when the CF-DTC is included. The integrated jitter (10kHz-10MHz) of the injection path1 and injection path2 is 516fs and 677fs respectively. Figure 4.17 shows the time-domain voltage waveform of the fixed delay. The fixed delay has an absolute delay of \sim 4ns with 50 μ W DC power consumption.

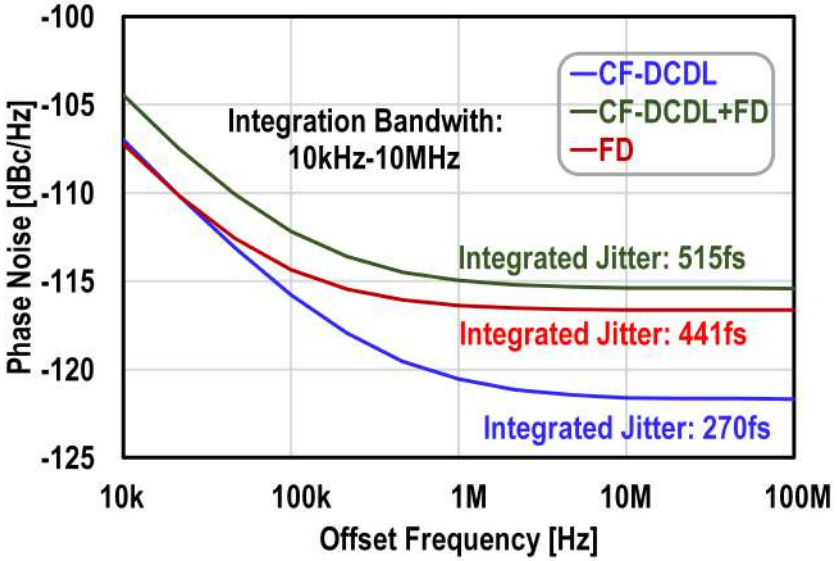


Figure 4.15: Simulated phase noise of the CF-DCDL and fixed delay.

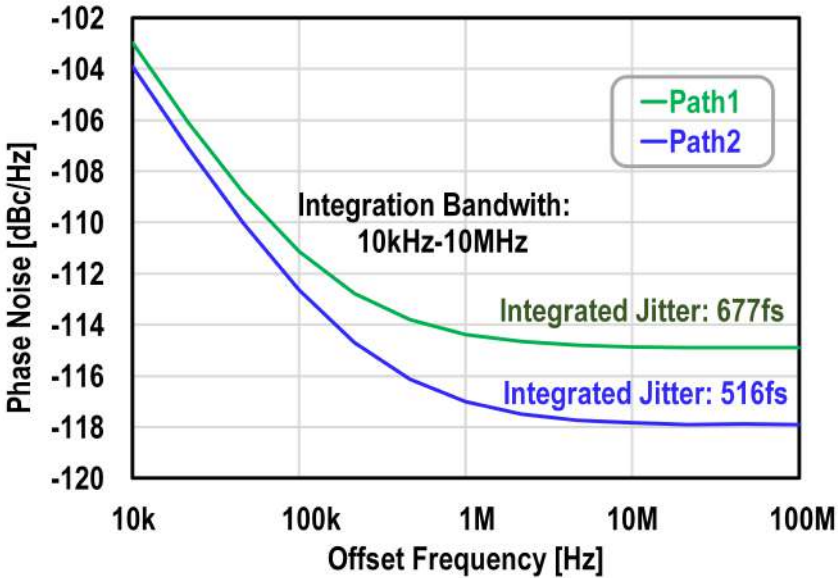


Figure 4.16: Simulated phase noise of the injection path1 and path2.

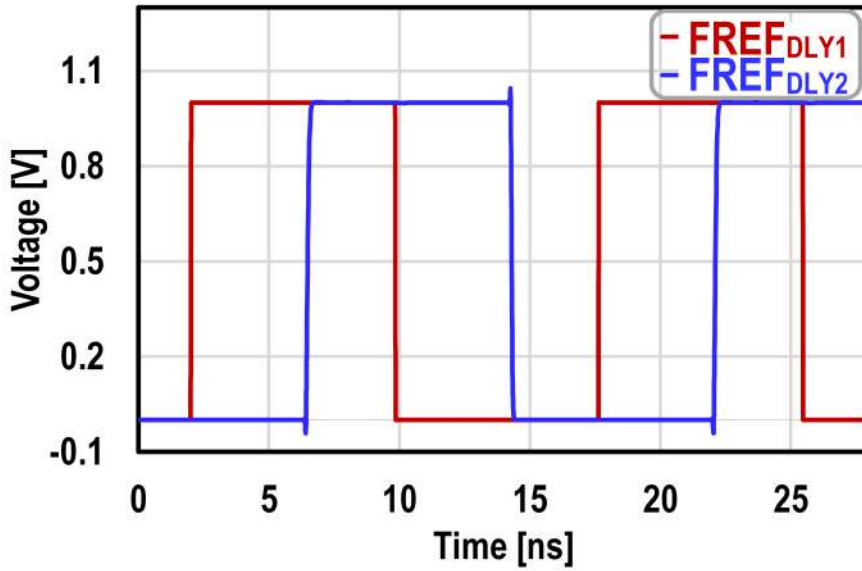


Figure 4.17: Simulated voltage waveform of the fixed delay.

4.4 Pulse Generator

The pulse generator creates a 64MHz differential pulse train using the positive edge of the reference clock. The pulse is injected into the DCO to improve the in-band phase noise of the DPLL. The main design parameters of the pulse generator are the pulse width, phase noise, and power consumption.

4.4.1 Pulse Generator Design

Figure 4.18 shows the schematic and layout of the pulse generator. The pulse width is carefully chosen as $\sim 100\text{ps}$ since a wider pulse could distort the amplitude of DCO waveforms thus degrading reference spur, whereas a narrower pulse is not reliable over PVT variations.

4

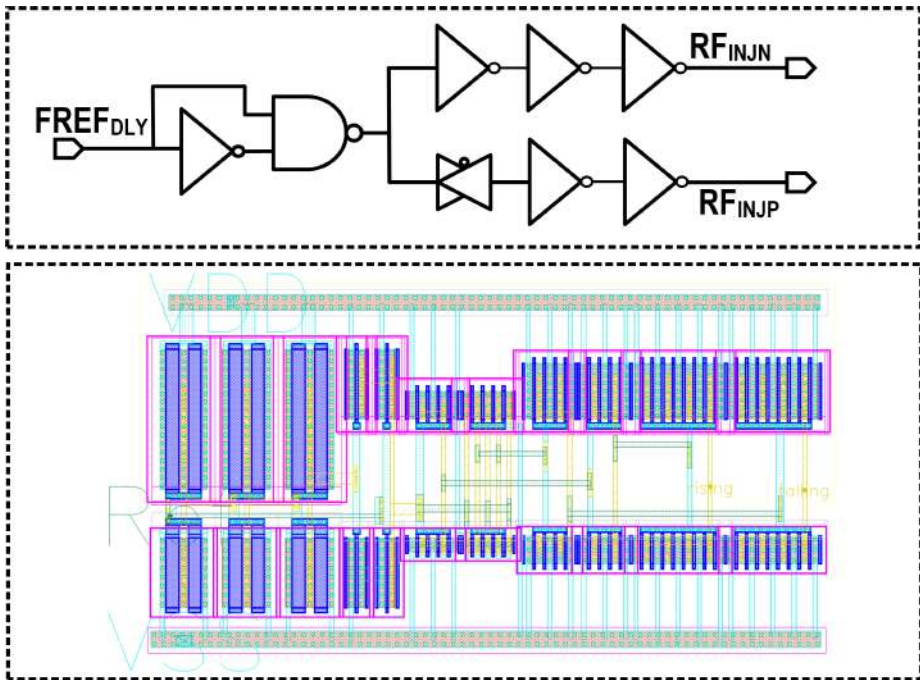


Figure 4.18: Schematic and layout of the pulse generator.

4.4.2 Post-Layout Simulation Results

The simulation results of the pulse generator are presented in this section. The parasitics from the layout are extracted using Quantus QRC, and the post-layout simulation is performed to estimate the power consumption, delay, and phase noise of the pulse generator. The simulation is under a typical corner with 1V supply at 25°C. Figure 4.19 (top) shows the simulated transient waveform of the pulse generator with a pulse width $\sim 100\text{ps}$. The pulse generator consumes $11\mu\text{W}$ DC power. The bottom of Figure 4.19 shows the simulated phase noise and integrated

jitter of 182fs. Compared with the IL-DCO, CF-DCDL, CF-DTC and fixed delay, the jitter of the pulse generator is relatively small.

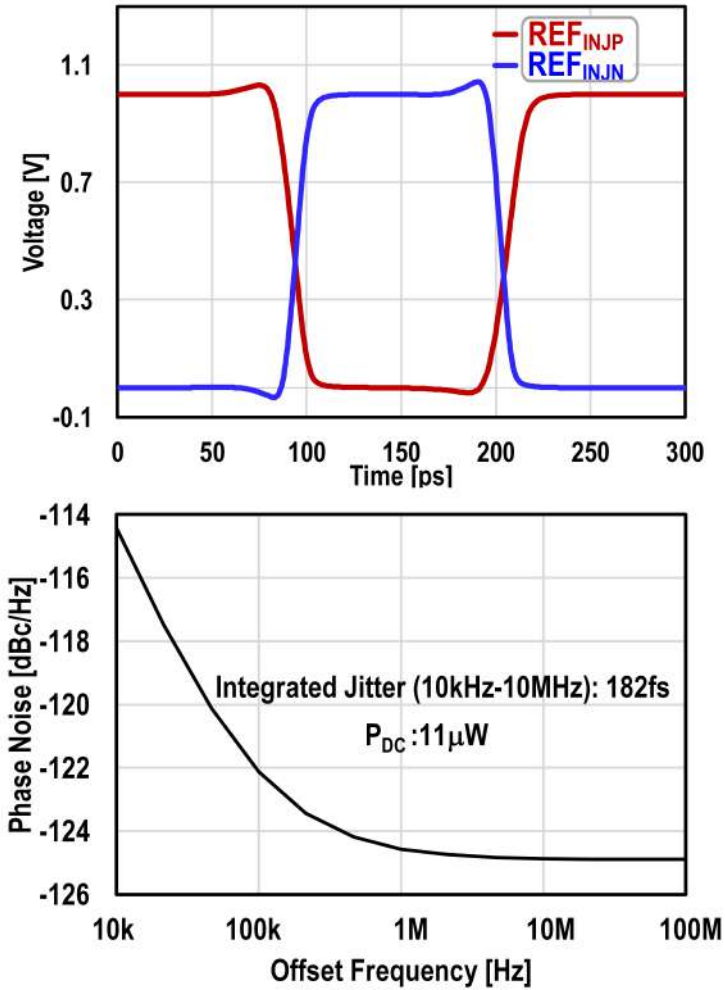


Figure 4.19: Simulated voltage waveform and phase noise of the pulse generator.

4.5 1b TDC

The 1b TDC is employed to detect the time offset between the injection path and the oscillator path for reference spur mitigation. To achieve a -50dBc reference spur, the time offset of the 1b TDC itself and the F-DCDL resolution should be within 1ps.

4.5.1 1b TDC Architecture

A sense-amplifier-based flip-flop is widely used in phase detector TDCs, where the fractional phase error of DPLLs is detected. However, this architecture typically has a large systematic time offset (~ 10 ps in [7]) due to its asymmetry. Zero-offset 1b TDC is proposed in [8] as shown in Figure 4.20, where CLK is a reference clock with 50% duty cycle and DATA is an RF clock. An SR-latch is used for zero-offset phase detection, and a D flip-flop is used to store the detected value.

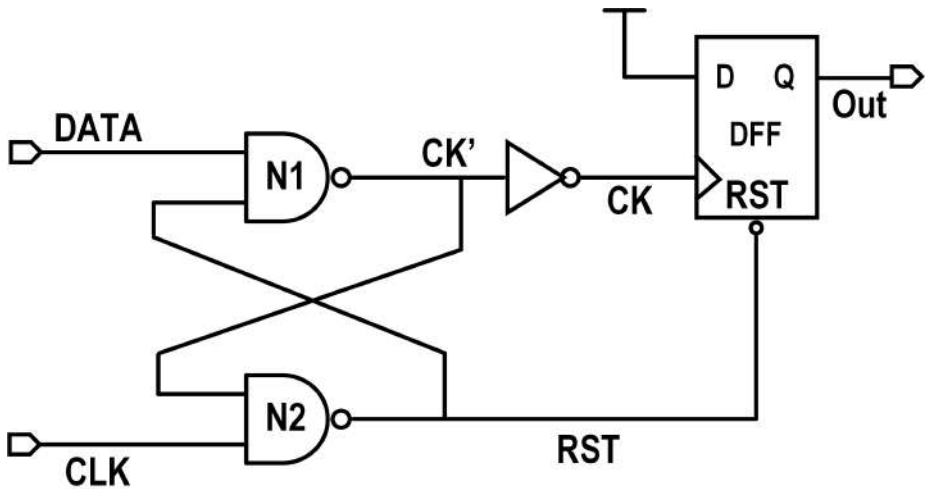


Figure 4.20: Schematic of the conventional 1b TDC.

When both DATA and CLK are low, nodes labeled RST and CK are respectively high and low in the steady state. Hence, the 1b TDC keeps the previous value.

If the rising edge of DATA comes first as shown in Figure 4.21 (left), the node labeled CK' will start to discharge since RST is still high. Consequently, CK will transit from low to high and the flip-flop (DFF) output will be high since node D is high. If the rising edges of DATA and CLK are very close, the NAND gate N2 will tend to pull down RST node. However, CK' will settle down to low since CK' has a lower voltage compared with node RST at the moment CLK goes to one¹. On the other hand, if RST node has a smaller capacitance, RST could settle to zero in the steady state, resulting in setup time violation.

¹Assume their node capacitances are the same.

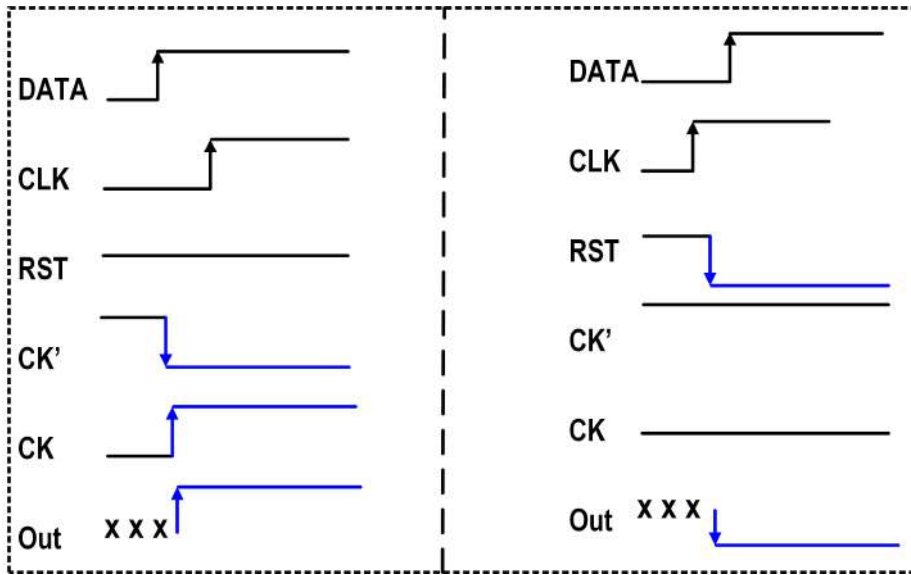


Figure 4.21: Timing of the conventional 1b TDC.

If the rising edge of CLK comes first as shown in Figure 4.21 (right), RST will start to discharge since CK' is still high. Hence DFF will be reset to low when the RST goes to low. If the rising edges of the DATA and CLK are very close, the NAND gate N1 will tend to pull down CK'. However, RST will settle down to low since RST has a lower voltage compared with that of CLK at the moment DATA goes to high¹. If CK' has a smaller node capacitance, CK' can be low and CK can transit from low to high in the steady state, resulting in hold time violation.

Only the rising edges of the DATA and CLK clock are taken into account in the above discussion. However, CLK has a pulse width only 100ps in this design. Hence, the falling edge of the CLK may affect the output of 1b TDC. If the rising edge of CLK comes first, DFF output will be low. RST and CK' are respectively low and high if both CLK and DATA are high. However, when CLK goes to low, RST will go to high since CK' is still high. Once RST goes to high, CK' will become low since DATA is still high. Consequently, CK will go low to high and DFF output will be high. This is not desirable since DFF should output low when CLK leads DATA. To solve this issue, the output of DFF is resampled by a second flip-flop (DFF1), which is clocked by the delayed CLK (CLK1) as shown in Figure 4.22. Before DFF transits from low to high due to the falling edge of CLK, DFF1 is used to sample the low output of the first flip-flop (DFF).

The time offset of the improved 1b TDC mainly depends on devices' mismatch and loading mismatch of the inputs of NAND gates. To reduce the time offset, the dimensions of the input transistors of NAND gates dimensions are increased. Figure 4.23 shows the layout of the 1b TDC.

¹Again, assume their node capacitances are the same.

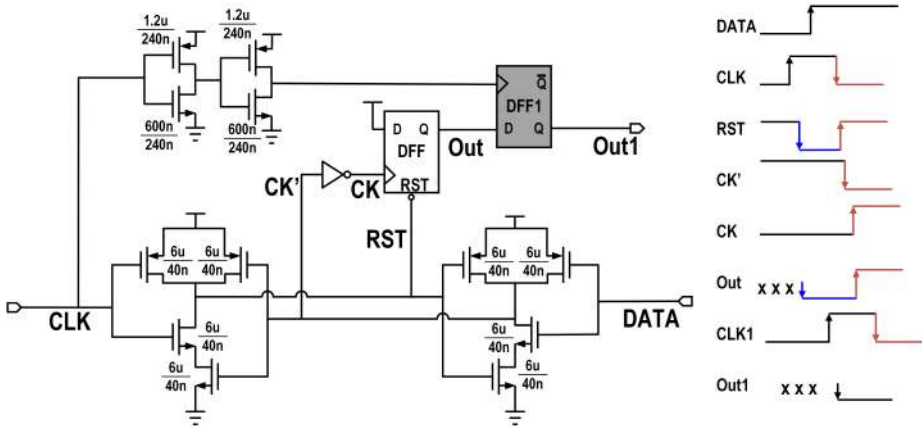


Figure 4.22: Schematic and timing of the improved 1b TDC.

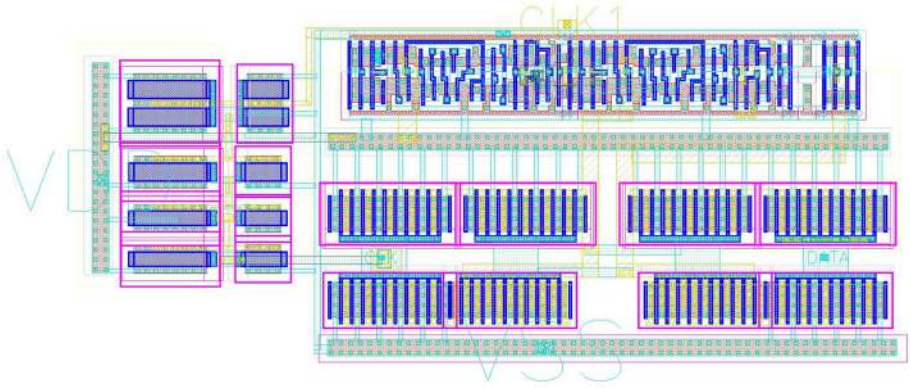


Figure 4.23: Layout of the improved 1b TDC.

4.5.2 Post-Layout Simulation Results

The parasitics from the layout are extracted using Quantus QRC, and the post-layout simulation is run to evaluate the time offset. The simulated DC power consumption of the 1b TDC is $100\mu\text{W}$. The Monte-Carlo mismatch simulation is used to calculate the time offset of the 1b TDC due to the devices mismatch. The delay difference between CLK and DATA is swept from -1ps to 1ps with a 0.1ps step in a single run¹. When CLK leads DATA, the 1b TDC output should be 0 in the ideal case. However, due to the devices' mismatch, 1b TDC output could be 1. In the Monte-Carlo simulation, the fraction times that the output is 1 is calculated. The simulated cumulative distribution function with 200 runs is plotted in Figure 4.24. The simulated RMS time offset of the 1b TDC is 0.18ps .

¹Note that the delay difference is ladder-shaped in the time domain to capture the hysteresis [9].

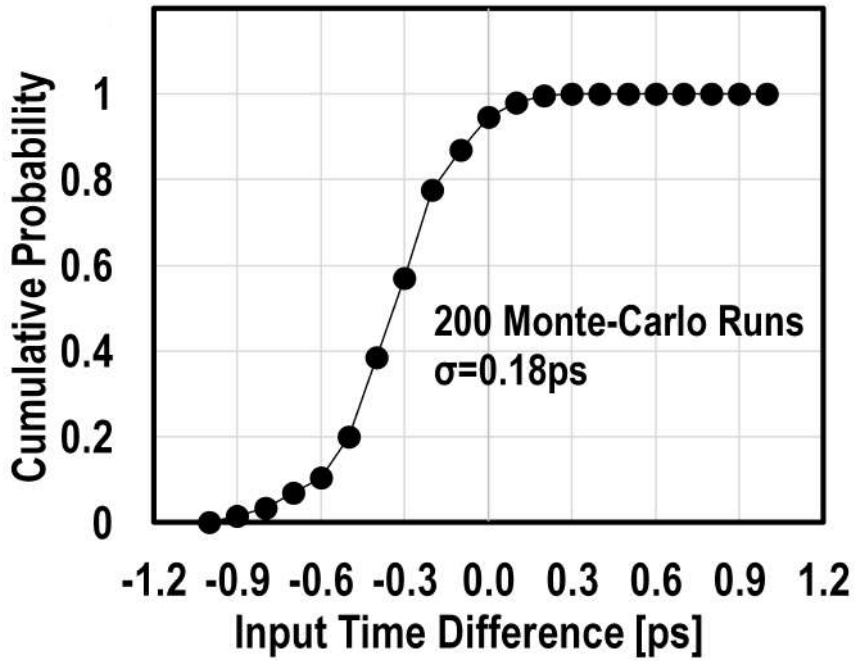


Figure 4.24: Simulated cumulative distribution function.

4.6 Other Blocks and Top Layout

Some analog blocks such as DCO buffer, integer counter, and Snapshot TDC were previously designed by Imec-NL and are exploited in this design in the interest of saving time. The DCO buffer is used to boost the swing of the DCO and to drive the large Pad capacitance. The integer counter and snapshot TDC are used for frequency and phase locking respectively. The counter with 1mW DC power consumption is shut down once the PLL gets frequency-locked.

Figure 4.25 shows the top-level layout of the analog blocks, where the phase quantizer includes the integer counter and snapshot TDC. The drawing area is $350\mu\text{m} \times 250\mu\text{m}$, which is dominated by the decoupling capacitors.

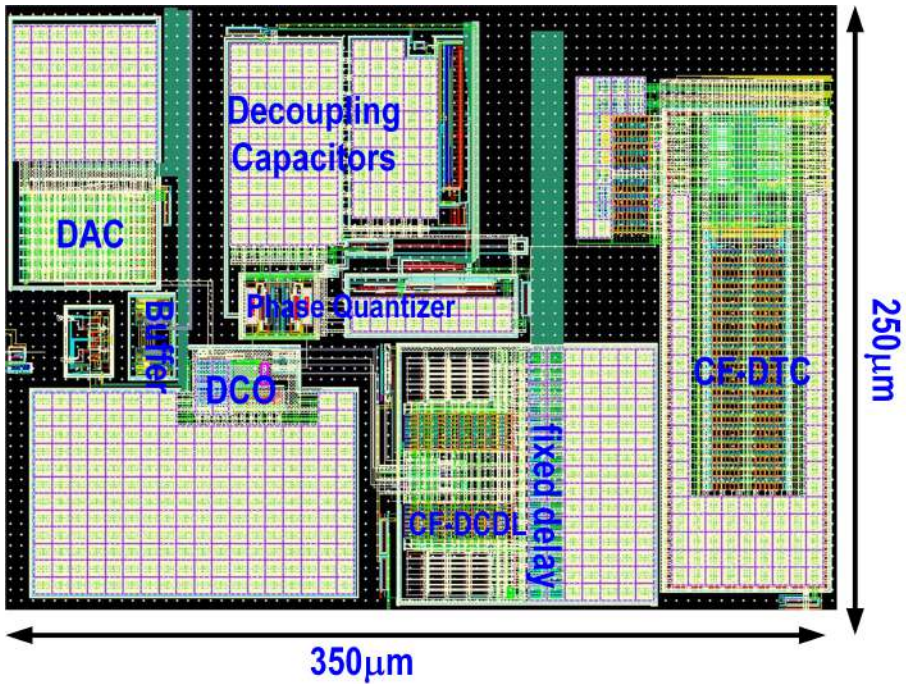


Figure 4.25: Layout of the analog top blocks of the proposed IL-DPLL.

References

- [1] J. A. McNeill and D. Ricketts, *The Designer's Guide to Jitter in Ring Oscillators*. New York: Springer, 2009, pp. 13-34.
- [2] A. Hajimiri, "Jitter and Phase Noise in Ring Oscillators," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1893–1816, June. 1999.
- [3] W. Bae et al., "A 7.6 mW, 414 fs RMS-Jitter 10 GHz Phase-Locked Loop for a 40 Gb/s Serial Link Transmitter Based on a Two-Stage Ring Oscillator in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 51, no. 10, pp. 2357–2367, June. 2016.
- [4] P. Kinget, *Integrated GHz voltage controlled oscillators*, in *Analog Circuit Design*, Kluwer, Springer, 1999, pp. 353–381.
- [5] V. Chillara, et al., "An 860 μ W 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 *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 172–173.
- [6] Y. He, et al., "A 673 μ W 1.8-to-2.5GHz dividerless fractional-N digital PLL with an inherent frequency-capture capability and a phase-dithering spur mitigation for IoT applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2017, pp.420–421.
- [7] S. Levantino, et al., "A 1.7 GHz fractional-N frequency synthesizer based on a multiplying delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2678–2691, Nov. 2015.
- [8] S. Kundu, et al., "A 0.2-to-1.45GHz Subsampling Fractional-N All-Digital MDLL with Zero-Offset Aperture PD-Based Spur Cancellation and In-Situ Timing Mismatch Detection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2016, pp.326–327.
- [9] A. Graupner, "A Methodology for Offset Simulation of Comparators," *The Designer Guide Community*, Oct. 2006.

5

RTL Design and Simulation

The RTL design of the low-speed digital logic in Verilog HDL is presented in this chapter. The control words for the digitally-controlled ring oscillator (DC-RO), coarse-fine digital-to-time converter (CF-DTC) and coarse-fine digitally-controlled delay lines (CF-DCDLs) are generated from this synthesizable digital logic. The inputs of the digital logic can be characterized into two categories, namely the outputs of the analog blocks (i.e., Snapshot TDC, integer counter and 1b TDCs) and the serial peripheral interface (SPI).

5.1 RTL Design

Figure 5.1 shows the high-level block diagram of the low-speed digital logic. There are seven sub-blocks of the digital logic, including FSM, Phase Detection, CF-DTC Control, Time Offset Calibration and Injection, PVT_norm, ACQ_Norm, and TRK_Norm. All of these sub-blocks are clocked by the retimed reference clock (CKR), which is generated by the Snapshot TDC. The finite state machine (FSM) controls the locking sequence of the digital (DPLL) and generates reset signals for other blocks¹. The integer phase error of the DPLL is calculated by the Phase Detection block, whose output is normalized by two loop filters in PVT_Norm and ACQ_Norm to control the oscillator frequency coarsely. The output of the Snapshot TDC (TDC_{OUT}) is sent to another loop filter in TRK_Norm for phase locking. Two other blocks (CF-DTC Control and Time Offset Calibration and Injection) are employed to respectively generate the control words for the CF-DCDLs and CF-DTC. The Time Offset Calibration and Injection block also generates the enable signal for the injection locking after the time offset calibration. The implementation details of these digital blocks are described in the following sections.

¹The reset signal is not shown in other blocks for the sake of simplicity.

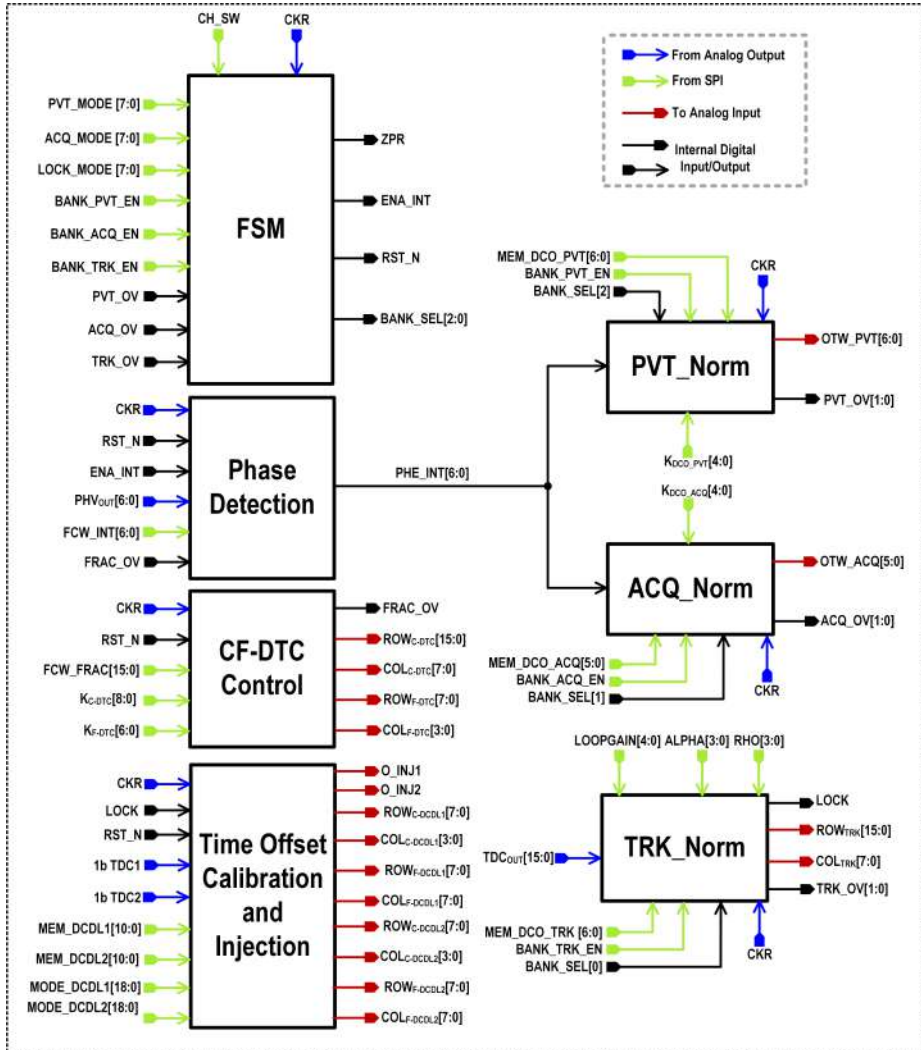


Figure 5.1: Top view of the digital logic.

5.1.1 Finite State Machine

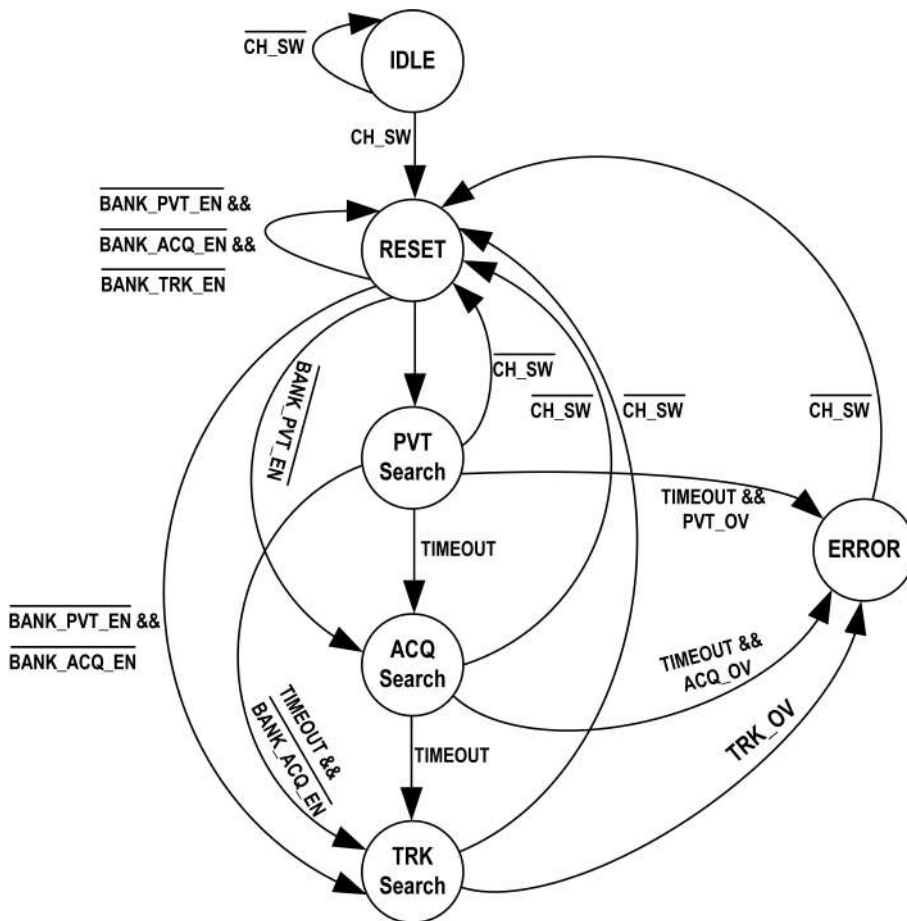


Figure 5.2: FSM state diagram.

Figure 5.2 shows the state-flow chart of the FSM, which controls the frequency locking sequence of the DPLL. There are six states for the FSM, including IDLE, RESET, PVT Search, ACQ Search, TRK Search, and ERROR. Initially, the FSM stays in the IDLE state. It will enter into the RESET state on a rising edge of the CH_SW to switch the frequency of the synthesizer and then trigger the DPLL for phase locking. Depending on the settings of BANK_PVT_EN, BANK_ACQ_EN and BANK_TRK_EN, the PVT Search, ACQ Search and TRK Search are skipped. It means that the DPLL will not change these banks if their enable signals are set to 0. The duration of PVT search and ACQ search are predefined by the PVT_MODE and ACQ_MODE settings respectively. The FSM enables the PVT (ACQ/TRK) bank frequency calibration by generating a BANK_SEL[2] (BANK_SEL[1]/BANK_SEL[0]) signal in the PVT (ACQ/TRK) Search state. Once the PVT (ACQ) frequency calibration is finished, their

corresponding frequency tuning words are frozen, and the DPLL will stay in the tracking bank until the CH_SW signal triggers the DPLL relocking. The control signal ENA_INT is generated to disable or enable the Phase Detection block depending on the setting of TRK_MODE in the tracking bank. If there is an overflow or underflow of the frequency tuning words, the FSM will enter into the ERROR state. Once the CH_SW goes to low again, the FSM will enter into the RESET state irrespective of the current state to reset all registers of other blocks. The ZPR (Zero Phase Reset) signal is applied to the related blocks at the end of the PVT SEARCH and ACQ SEARCH. It is connected to the phase accumulator (see Figure 5.3) to ensure that the phase error is reset before the DPLL entering into the successive frequency bank.

5.1.2 Phase Detection

The phase detection block is shown in Figure 5.3. It takes the integer counter output (PHV_{OUT}) from the analog part as its input and generates the integer phase error of the DPLL for the type-I frequency tuning through the PVT and acquisition banks. It could be disabled by the signal (ENA_INT) generated from the FSM when the DPLL enters into the tracking bank.

PHV_{OUT} is first stored in a register at the falling edge of CKR to avoid timing violation. A differentiator is then used to calculate the difference between two successive samples of the retimed PHV_{OUT} . This difference is then compared with the FCW integer part and the overflowed FCW fractional part to obtain the frequency error, which is then accumulated to produce the integer phase error (PHE_INT) for the oscillator frequency tuning.

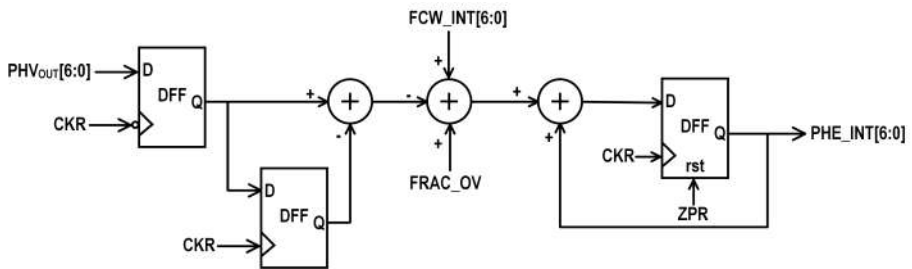


Figure 5.3: Block diagram of the Phase Detection block.

5.1.3 PVT_Norm and ACQ_Norm

Figure 5.4 shows the block diagram of the PVT_Norm and ACQ_Norm, which take the integer phase error (PHE_INT) generated by Phase Detection block as the input and generate the frequency tuning words for the PVT and acquisition banks. A multiplier is first used to multiply PHE_INT and the gain of the PVT bank (K_{DCO_PVT}), which is defined as the ratio of the reference frequency and the frequency step of PVT bank. Its output is then truncated to generate the oscillator tuning word for the type-I operation.

The PVT bank is activated only when `BANK_PVT_EN` is set to 1 from the external control and `BAN_SEL[2]` is 1 from the FSM. The initial control code of the PVT bank is set by `MEM_DCO_PVT`. When `BANK_PVT_EN` is set to 0, the DPLL loop will not change the tuning word of the PVT bank. Consequently, the DCO frequency tuning curve can be obtained by sweeping the value of `MEM_DCO_PVT` externally via SPI. Once the FSM enters into the acquisition bank, the control signal `BAN_SEL[2]` becomes 0, freezing the PVT bank. If there is an overflow or an underflow in the PVT control code, `PVT_OV` will be nonzero and the FSM will enter the ERROR state.

The `ACQ_Norm` works similarly as the `ACQ_Norm`. The detailed analysis of the `ACQ_Norm` is not presented here.

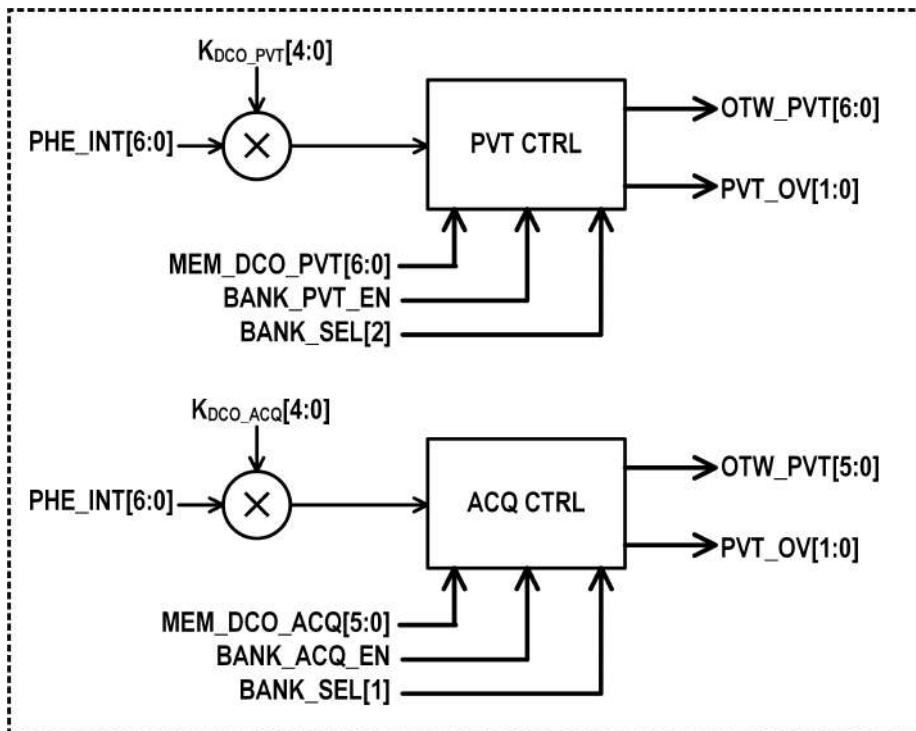


Figure 5.4: Block diagram of PVT and acquisition Normalization blocks.

5.1.4 TRK_Norm

The `TRK_Norm` block takes the Snapshot TDC output (`TDCOUT`) as its input and generates the tuning word for the oscillator tracking bank for the type-II operation. Figure 5.5 shows the block diagram of the `TRK_Norm`. A thermometer-to-binary decoder is first used to convert `TDCOUT` to its binary form, which is then multiplied by a loop gain (`LOOPGAIN`) factor before sending it into the loop filter. The loop

gain is defined as:

$$LOOPGAIN = \frac{f_{ref} K_{TDC}}{t_{out} K_{DCO_TRK}}, \quad (5.1)$$

where K_{TDC} is the TDC resolution and K_{DCO_TRK} is the frequency resolution of the tracking bank. The low-pass digital loop filter is designed as a PI system (i.e.,

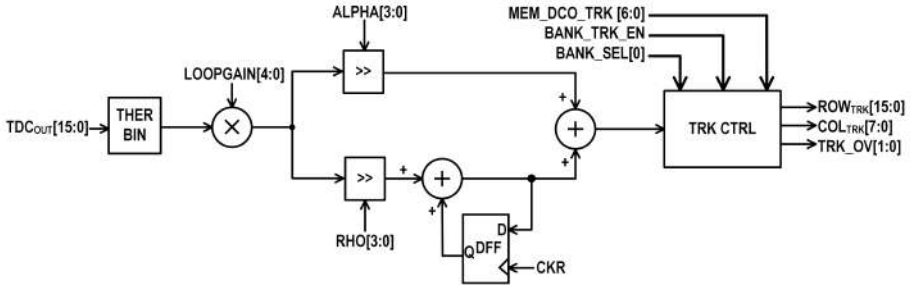


Figure 5.5: Block diagram of the TRK_Norm block.

proportional-integrational system). The loop filter parameters are implemented in an efficient way as right-bit-shift operations. The outputs of the proportional and integral path are summed to obtain the control word for the tracking bank. The TRK CTRL block is similar to the ACQ_Norm and PVT_Norm except that there are two extra binary-to-thermometer decoders which are used to generate the row and column control for the DAC.

5.1.5 CF-DTC Control

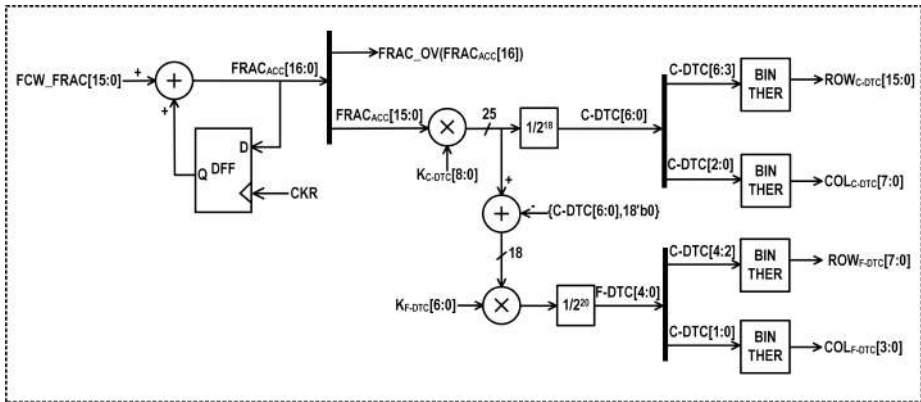


Figure 5.6: Block diagram of the CF-DTC Control block.

Figure 5.6 shows the block diagram of the CF-DTC control, which takes the FCW fractional part (FCW_{FRAC}) as the input and generates the control words for the CF-DTC. The desired delay of the CF-DTC is equal to the product of the accumulated

FCW fractional part ($FRAC_{ACC}$) and the output period (t_{out})¹.

As shown in Figure 5.6, an accumulator is first used to generate $FRAC_{ACC}$ and an overflow signal ($FRAC_{OV}$). A multiplier is then used to multiply $FRAC_{ACC}$ and the C-DTC gain (K_{C_DTC}), which is defined as the ratio of t_{out} to the C-DTC resolution. Its output is then truncated to generate the binary control code of the C-DTC. The row and column control codes of the C-DTC are generated by two binary-to-thermometer decoders. A second multiplier is used to multiply the truncated residue error and F-DTC gain (K_{F_DTC}), which is defined as the ratio of the C-DTC resolution to the F-DTC resolution. Its output is then truncated and decoded to generate the F-DTC column and row control.

5.1.6 Time Offset Calibration and Injection

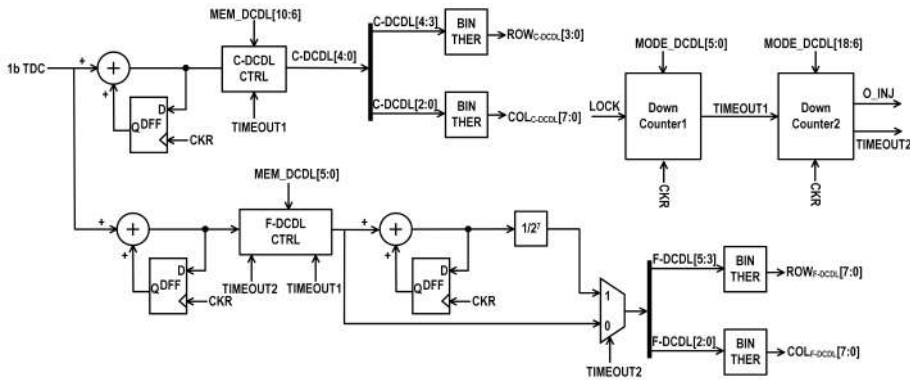


Figure 5.7: Block diagram of Time Offset Calibration and Injection block.

Figure 5.7 shows the internal details of the Time Offset Calibration and Injection block² of Figure 5.1. The Time Offset Calibration and Injection is used to generate control code for the CF-DCDL, cancel the time offset between the injection phase and the oscillator phase, and enable injection locking. The calibration is activated when the DPLL gets phase-locked (LOCK signal goes to high). Two down counters are employed to control the calibration sequence. While the first down counter (Down Counter1) is counting, its output (TIMEOUT1) is 0. In the meantime, the 1b TDC output is accumulated to generate the control code for the coarse DCDL and the fine DCDL control is frozen by the F-DCDL CTRL block. Once Down Counter1 finishes counting, TIMEOUT1 becomes 1 and Down Counter2 is then activating for 128 reference cycles. The 1b TDC output is accumulated by another accumulator to generate the control code for the F-DCDL. In the meantime, an average block is used to calculate the mean value of the accumulator output while the Down Counter2 is counting. The averaged control code of the accumulator is frozen for the F-DCDL control once the Down Counter2 finishes counting. Finally, the injection circuit is enabled by the signal O_INJ.

¹See Equation (3.6).

²Only one calibration path is shown.

The initial control code for the CF-DCDL is externally set by the signal MEM_DCDL and the counting cycles of the down counters are controlled by the signal MODE_DCDL.

5.2 System Simulation in Verilog

The analog/RF blocks including DCO, CF-DTC, CF-DCDLs, Snapshot TDC, 1b TDC and integer counter are modeled behaviorally in System Verilog. Imperfections such as noise, nonlinearity, and quantization error are also added to these blocks. These modeled blocks are used together with the RTL code of the low-speed logic to perform the system-level simulation in Verilog.

Figure 5.8 shows important signals from a Verilog simulation of the IL-DPLL. The channel switch signal CH_SW triggers the DPLL for phase locking. The PVT and acquisition banks are activating subsequently to adjust the DPLL frequency. Their control codes are frozen when DPLL enters into the tracking bank. In the tracking bank, the output code of the Snapshot TDC toggles in the middle when the DPLL gets phase-locked. Once the time offset calibration is finished, an injection signal is generated to reset the oscillator phase for the IL-DPLL operation.

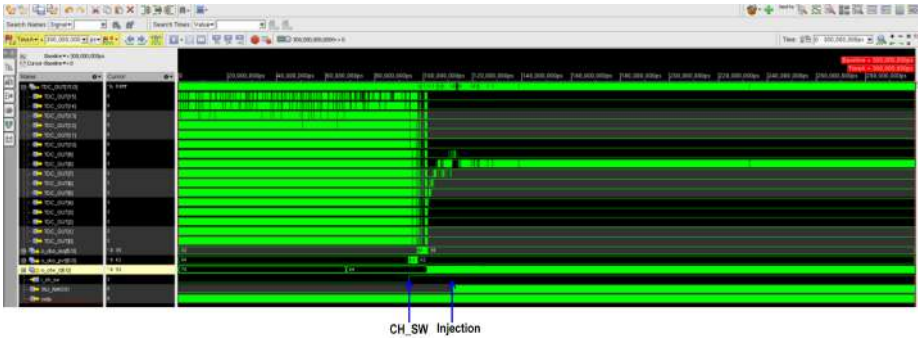


Figure 5.8: System-level simulation of the IL-DPLL.

6

Measurement Results

In this chapter, the test setup and measurement results are presented in the first two sections. The comparison with the state-of-the-art is given in the last section of this chapter.

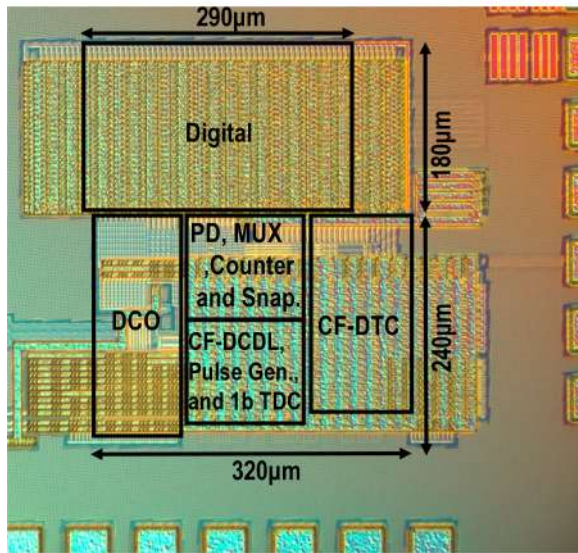


Figure 6.1: Chip micrograph of the proposed IL-DPLL.

6.1 IL-DPLL Test Setup

The proposed ring-oscillator-based fractional-N injection-locked digital PLL (IL-DPLL) is fabricated in the TSMC LP 40nm CMOS process. The chip micrograph of

the implemented IL-DPLL is shown in Figure 6.1. This chip occupies an active core area of 0.13mm^2 (including decoupling capacitors). It has separate power supplies for the DCO, DAC, digital circuit, coarse-fine DTC, phase quantizer (including a TDC, mux, integer counter, and snapshot circuit), and injection circuit (including two 1b TDCs and coarse-fine DCDLs, and pulse generators). Excluding the noisy IO ground, all of the grounds on the chip are connected together, and all of the ground pads are down-bonding to the PCB ground to reduce the inductance.

The test setup of the IL-DPLL is shown in Figure 6.2. The PCB is composed of a microcontroller, level shifters, LDO regulators, a crystal oscillator and an SMA port. The IL-DPLL chip is directly bonded to the PCB for the sake of saving time. The microcontroller is used to read and write on-chip registers for the IL-DPLL control. The microcontroller is further programmed by a PC. All of the supplies on the chip are provided by LDO regulators. The 64MHz crystal oscillator is used to provide the clock for the IL-DPLL. For the phase noise and spectrum measurements, the SMA port is used to connect the RF port to the spectrum analyzer.

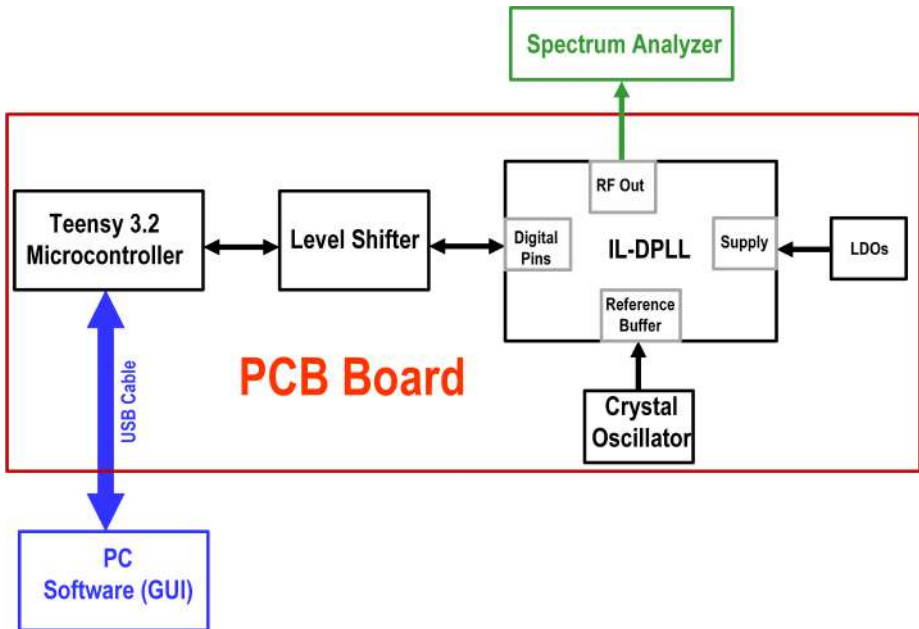


Figure 6.2: Test setup of the proposed IL-DPLL.

6.2 Measurement Results

The measurement results of the proposed IL-DPLL are presented in this section. The R&S®FSV4/FSV7 spectrum analyzer is employed to measure the phase noise and spectrum of the IL-DPLL.

6.2.1 Frequency Tuning

Table 6.1: Measured DCO tuning range.

DCO_BIAS_STD_EN	PVT [MHz]	ACQ [MHz]	TRK [MHz]
00	128-1876	-	-
01	1687-2342	-	-
10	2131-2549	2423-2432	2131-2131.5
11	2486-2740	-	-

Table 6.1 shows the measured frequency range of the DCO. An extra control signal DCO_BIAS_STD_EN is used to make sure that the DCO can cover the desired frequency range. Automatic frequency calibration can be implemented in the future design. The DCO can cover a wide frequency range of 128MHz-2.74GHz. When the control signal DCO_BIAS_STD_EN is set to 10, the PVT bank has a frequency range of 418MHz, which can cover all of the BLE channels. The acquisition bank has a frequency range of 9MHz, resulting in a 140kHz step. The frequency range of the tracking bank is around 500kHz, which can cover more than 3x of the acquisition step. It is reduced compared with the post-layout simulation. This could be due to the underestimated loading of the DCO in the simulation.

6.2.2 Phase Noise

Figure 6.3 shows the measured phase noise in an integer channel with FCW=38 and $f_{out}=2.432\text{GHz}$ ¹. The integrated jitter between 10kHz-10MHz is 2.37ps when the injection Path2 is off. It is significantly improved to 1.34ps when the injection Path2 is on. The averaged in-band phase noise improvement is ~6dB. Besides, the noise improvement bandwidth is larger than 10MHz, and the phase noise the high offset frequency is degraded when turning on the second injection path. This proves the effectiveness of the proposed two-path injection technique for the phase noise improvement.

Figure 6.4 shows the measured phase noise of the implemented IL-DPLL in a fractional channel with FCW=38.00024414. The phase noise at the 1MHz offset frequency is -108.4dBc/Hz. The phase noise performance meets the BLE specifi-

¹Due to the inaccuracy (1kHz lower) of the crystal oscillator, the synthesized frequency is 34kHz lower.

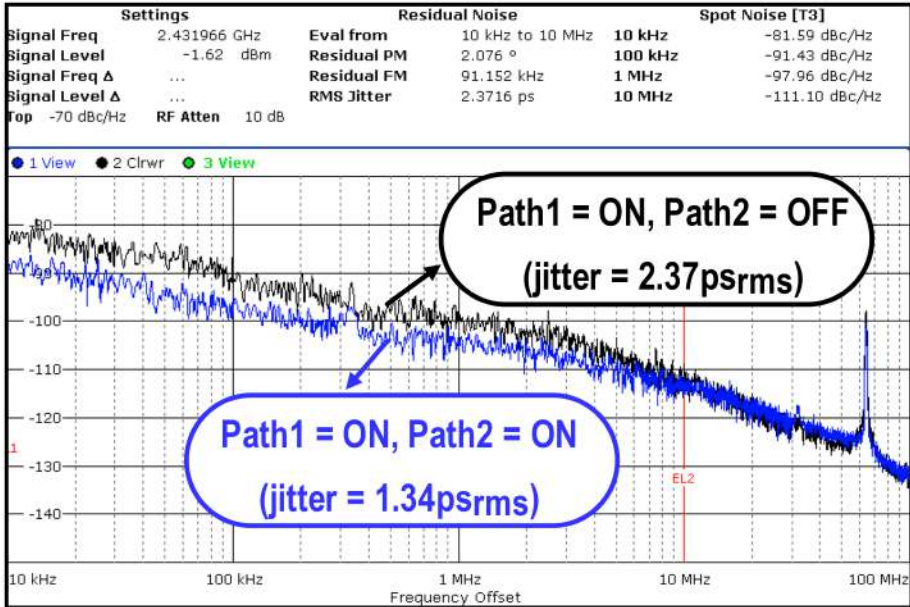


Figure 6.3: Measured phase noise with and without the injection Path2.

6

cation with enough margin. The integrated jitter between 10kHz-10MHz is 1.6ps, which is much smaller than the BLE specification (6.5ps).

Figure 6.5 (left) shows the measured phase noise of the implemented IL-DPLL in fractional channels with an FCW integer part equal 38. The integrated jitter is within the range of 1.3-1.6ps when the injection Path2 is on, sweeping the FCW fractional part from $\frac{1}{2^{12}}$ to $\frac{1}{2}$. It varies from 2.5ps to 3ps when the injection Path2 is off. The integrated jitter of the near-in integer channels is degraded due to fractional spurs. Figure 6.5 (right) shows the measured phase noise of the IL-DPLL in integer channels with and without the second injection path. FCW is swept from 28 to 42, and the synthesized frequency of the IL-DPLL varies from 1.8GHz to 2.7GHz. The integrated jitter changes from 2.1ps to 2.5ps when the injection Path2 is off, and it is 1.5ps to 1.2ps when the injection Path2 is on.

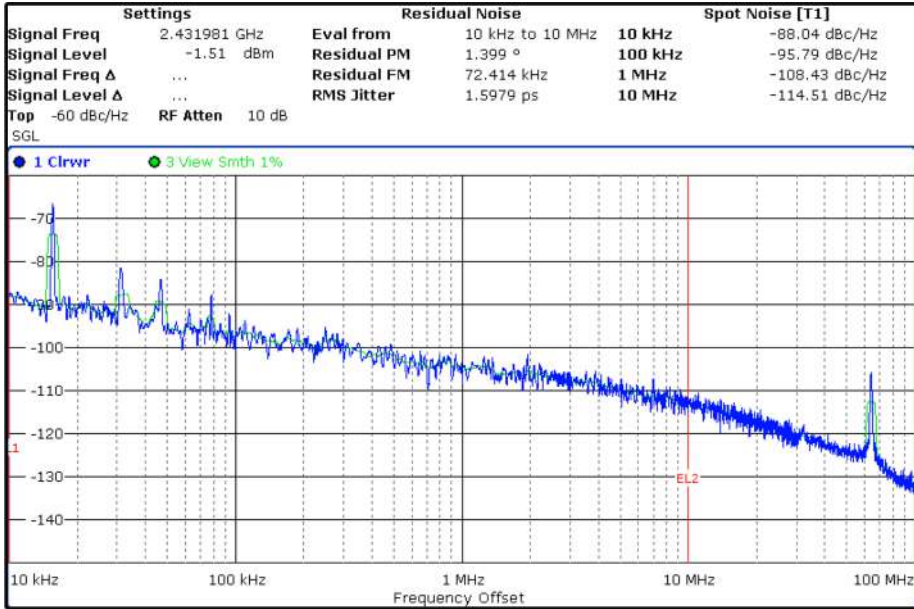


Figure 6.4: Measured phase noise in a fractional channel.

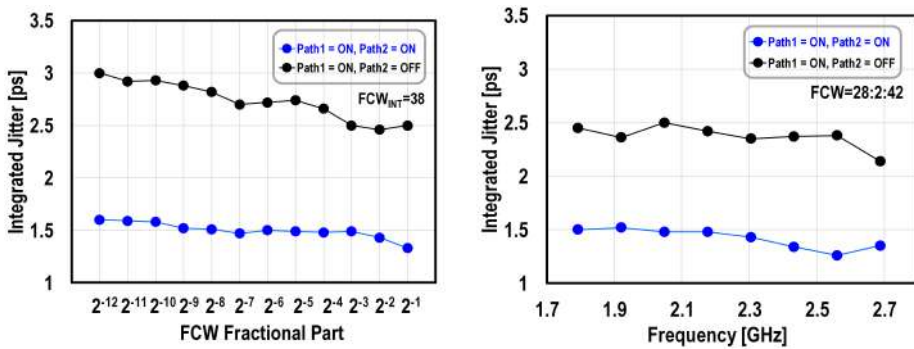


Figure 6.5: Measured phase noise in fractional and integer channels.

6.2.3 Spectral Purity

Figure 6.6 shows the measured spectrum with FCW=38.00024414. The highest close-in fractional spur (at 15.625kHz offset frequency) is -45.8dBc, which is ~12dB better than the target specification. Without the time offset calibration loop, the measured reference spur is -30dBc, which fails to meet the BLE specification. It is improved to -43.6dBc when the time offset calibration is enabled, which meets the BLE specification and is 0.4dB worse than the target specification (-44dBc). The degraded reference spur could be due to the digital coupling since ~-45dB reference spur is observed when the oscillator is free-running.

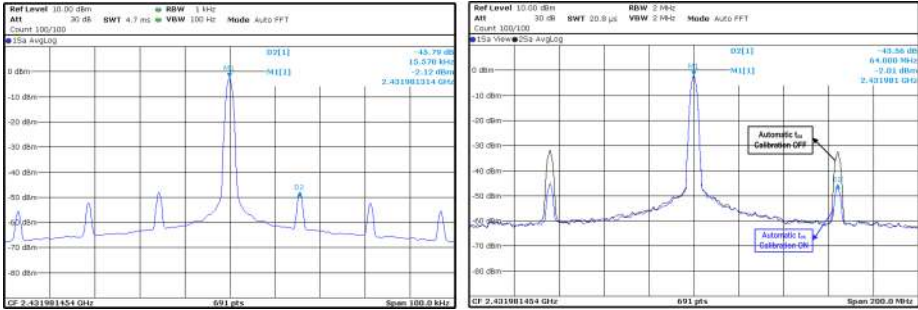


Figure 6.6: Measured spectrum of the proposed IL-DPLL.

Figure 6.7 shows the measured reference spur when the F-DCDL control code is manually tuned. The measured reference spur varies from -27dBc to -49dBc. Within 2μs time offset calibration, the measured worst-case reference spur is -43.6dBc.

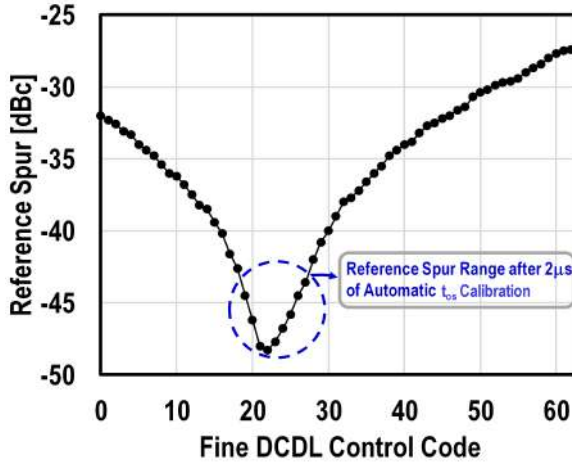


Figure 6.7: Measured reference spur by manually tuning the F-DCDL control code.

6.2.4 Power Breakdown

The measured power breakdown of the IL-DPLL is shown in Figure 6.8. The total power consumption is 1.33mW, which is lower than the target specification (1.5mW). The power consumption of the oscillator is $\sim 90\mu\text{W}$ higher than that the simulated value. This is due to the fact that the oscillator buffer and the oscillator share the same supply; hence it is difficult to measure the oscillator power consumption alone. The oscillator consumes around half of the total power. The combined power consumption of the CF-DTC and injection circuit is as low as $210\mu\text{W}$. Consequently, the power consumption of the DCO can be further reduced to make the IL-DPLL work in the sub-mW region. In the meantime, the IL-DPLL can still meet the phase noise and integrated jitter specifications of the BLE. The proposed two-path injection technique improves the in-band phase noise $\sim 6\text{dB}$ while its power consumption overhead is smaller than 0.1mW. To achieve the same amount of improvement, one could double the reference frequency at the price of a significant increase in the power consumption of a frequency doubled DTC and digital blocks (e.g., $>0.5\text{mW}$ overhead in this design).

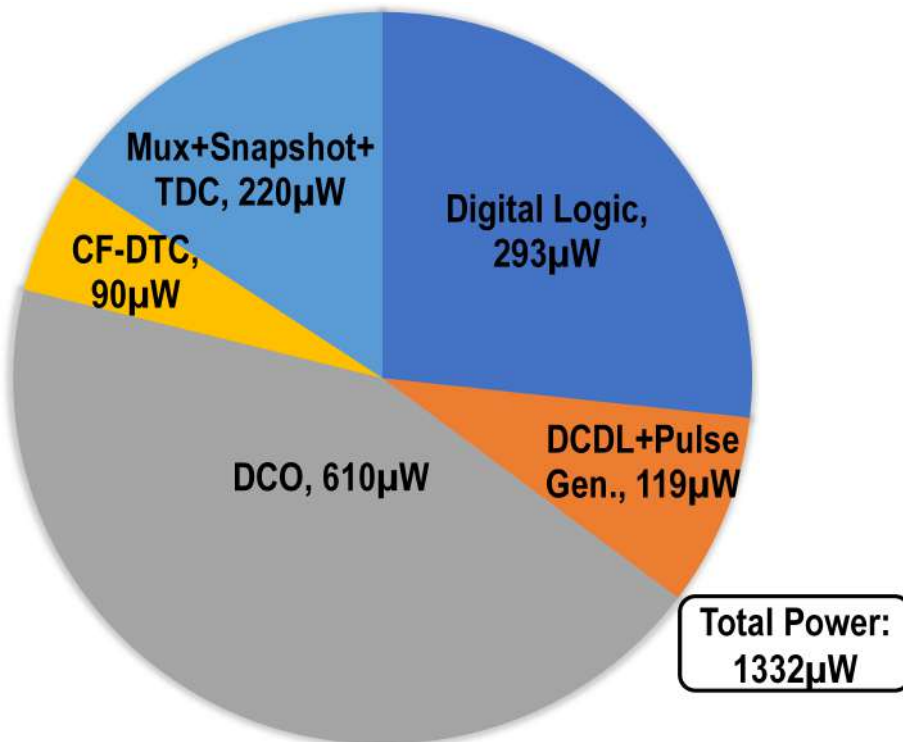


Figure 6.8: Measured power breakdown of the proposed IL-DPLL.

6.3 Performance Summary and Comparison Table

Table 6.2 shows the performance summary of the proposed IL-DPLL and the comparison of the state-of-the-art PLLs in the literature. The total power consumption of the IL-DPLL is 1.33mW operating at 2.4GHz with integrated jitter of 1.6ps. This work achieves the lowest power consumption and best FOM (-234.7dB) compared with other fractional-N ring-oscillator-based PLLs published in the literature. Compared with the LC-based PLL, this work achieves similar FOM with ~35% area reduction. The close-in fractional spur is -45.8dBc, and the reference spur is -43.6dBc, which is good enough for the BLE application.

Table 6.2: Comparison with the state-of-the-art.

	This Work	[3] ISSCC'14	[5] ISSCC'16	Park ISSCC'12	[1] ISSCC'17
Oscillator Type	Ring	Ring	Ring	Ring	LC
Injection Locking	Yes	Yes	Yes	Yes	No
Ref. Frequency (MHz)	64	50	87.5	32	32
Freq. Resolution (kHz)	1	0.19	17500	1000	0.5
Freq. Range (GHz)	1.8-2.7	1.6-1.9	0.2-1.45	0.58-0.61	1.8-2.5
Fractional Spur (dBc)	-45.8	-47	NA	NA	-56
Reference Spur (dBc)	-43.6	-47	-45	NA	-62
Ref. Spur Cal. Time (μ s)	2	~30	NA	NA	NA
RMS Jitter (ps)	1.6 [10k-10MHz]	1.4 [30k-30MHz]	2.8 [10k-10M]	4.23 [100-40MHz]	1.98 [10k-10MHz]
Power (mW)	1.33	3	8	10.5	0.67
FOM* (dB)	-234.7	-232.3	-222	-217.3	-235.8
Area (mm^2)	0.13	0.4	0.054	0.158	0.18
Technology (nm)	40	65	65	65	40
*FOM= $20 \cdot \log_{10}(\text{Jitter}/1\text{s}) + 10 \cdot \log_{10}(\text{Power}/1\text{mW})$					

7

Conclusion

7.1 Thesis Conclusion

Analysis, design, and validation of a ring-oscillator (RO) -based fractional-N injection-locked digital phase-locked loop (IL-DPLL) for the frequency generation are presented in this thesis. The specifications of the IL-DPLL for the BLE application are derived in chapter 1. Based on the derived specifications, prior arts are studied, and an IL-DPLL architecture is presented in chapter 2. To verify the functionality of the proposed architecture, the MATLAB time-domain modeling and simulation are presented in chapter 3. The transistor-level design of various analog/RF blocks is then given in chapter 4. In chapter 5, RTL design is presented, and the behavioral simulation in analog/mixed-signal environment is further used to verify the functionality of the proposed system. A prototype is implemented in the TSMC LP 40 nm CMOS process. Finally, measurement results are shown in chapter 6 to prove the ideas proposed in chapter 2.

A coarse-fine digital-to-time converter (CF-DTC) is employed to make the IL-DPLL work in the fractional-N domain, and the measured fractional spur is -45.8dBc at the 15.625kHz offset frequency. For the frequency error detection, the phase error is first measured, and injection is applied after a few DCO cycles. A fast reference spur calibration loop is implemented to reduce the time offset to 3ps (measured -43.6dBc reference spur) within $\sim 2\mu s$. As it is shown in the measurement, the phase noise of the IL-DPLL improves 6dB, and the integrated jitter reduces 1.8x when the proposed two-path injection is enabled while the power overhead is smaller than 0.1mW. As a result, a $0.13mm^2$, 2.4GHz, RO-based fractional-N injection-locked digital PLL is demonstrated with 1.6ps RMS jitter, -43.6dBc reference spur and -45.8dBc fractional spur while consuming only 1.33mW DC power.

Figure 7.1 shows the Jitter²-power of the state-of-the-art RO-based fractional-N PLLs. Thanks to the proposed two-path injection technique, this work achieves the best figure of merit (FOM) of -234.7dB while consuming the lowest power (1.33mW). Compared with the work [Elkholy, JSSC'16], this work consumes the

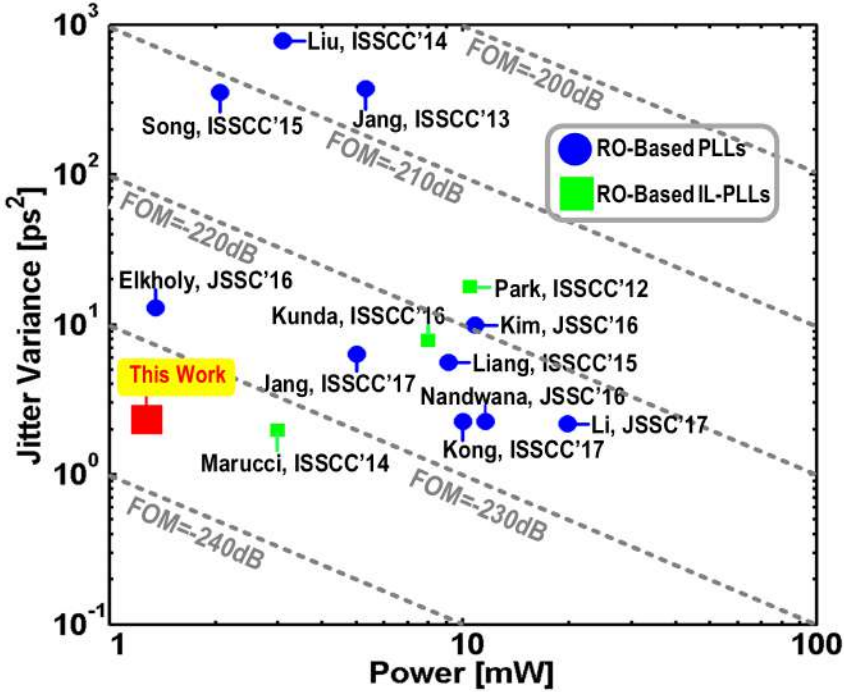


Figure 7.1: Jitter²-power of the state-of-the-art RO-based fractional-N PLLs.

similar power, but it achieves 7dB better FOM. On the other hand, this work has the similar integrated jitter as the work [Marucci, ISSCC'14], but it consumes 0.5x less power. For a fair comparison, the area of the RO-based frequency synthesizers should be taken into account. Figure 7.2 shows the FOM-area of the state-of-the-art RO-based fractional-N PLLs. It can be observed that this work achieves best FOM/area.

7.2 Future Work

7.2.1 Tracking Bank Range

In the measurement, the range of tracking bank is reduced significantly due to the parasitic capacitance. The oscillator frequency drops, and a larger PVT tuning code should be used to compensate this frequency drop. Hence, the oscillator enters non-linear frequency tuning region, reducing tracking bank range. The potential problem is that large reference spur is introduced when the oscillator exhibits frequency drift, and the tracking bank is not wide enough to correct the frequency drift. One potential solution is to make the DPLL enter acquisition bank again for frequency locking. The second solution is to use a larger PMOS transistor.

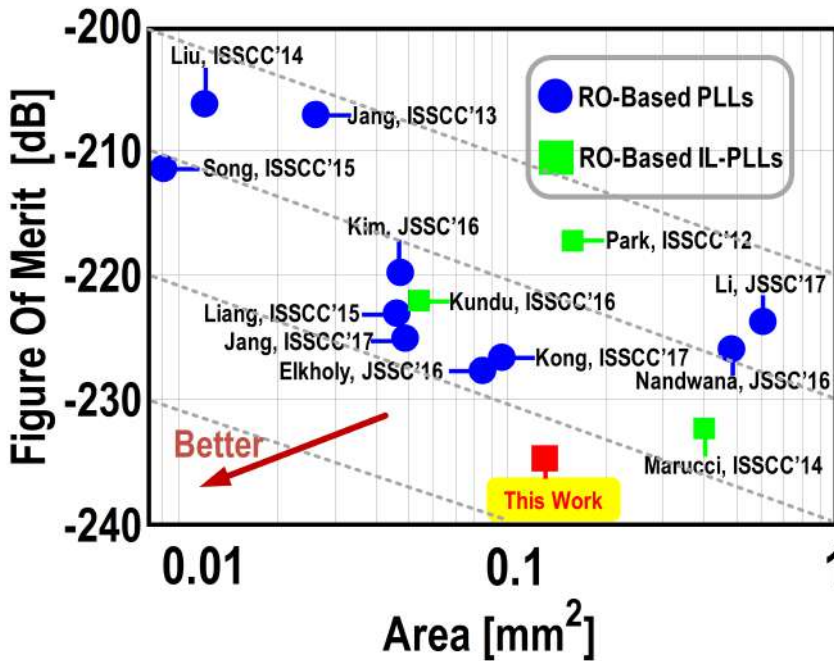


Figure 7.2: FOM-area of the state-of-the-art RO-based fractional-N PLLs.

7.2.2 Area Optimization

The digital area is quite large (~2x larger) compared with the Imec-NL's previous design. It can be easily optimized for the future design. Beyond that, more than 0.5x of the total area is taken up by decoupling capacitors. Due to these reasons, this chip is quite large compared with the state-of-the-art. The analog area can be optimized by reducing the number of decoupling capacitors.

7.2.3 Reference Spur Mitigation

Although the calibrated reference spur can meet the BLE standard, it is still quite high. During the measurement, the reference spur level is as high as -45dBc even when the oscillator is free-running. It could be introduced by the coupling from the digital part since it was found that there was no deep N-well for the digital layout. Better isolation between the oscillator and digital part should be designed. On the other hand, the PVT variations of the CF-DCDL can affect the injection timing, degrading the reference spur when the packet is long. The CF-DCDL control code could be dynamically adjusted based on the calibrated gain of the CF-DTC to compensate the PVT variations in a background manner.

Acknowledgements

First of all, I would like to express my most sincere gratitude and appreciation to my academic supervisor, Dr. Masoud Babaie. I want to thank him for all his supervision, encouragement, understanding and support. His attitude toward high-quality work has inspired me to continuously challenge myself to reach new levels. Except for the research project, he also spent numerous time helping me improve my writing skills.

I want to extend my appreciations to Dr. Fabio Sebastiano and Prof. Leo de Vreede for reading my thesis and serving as my committee. Prof. Leo de Vreede is also the promoter of this project.

I want to deeply thank my industry supervisors Ao Ba and Yuming He for their invaluable guidance. They were always there for any advice and helped me in every detail of this project (system architecture, schematic, layout, PCB, and measurement). Ao meticulously taught me how to design the layout. Yuming is an expert of DPLL and I learned a lot from him. Sometimes, he helped me debug the chip until midnight and measure the chip on weekends. Beyond the project part, they also helped me improve my skills of making slides and presentation.

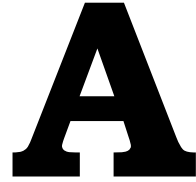
My special gratitude goes to my great colleagues at Imec-NL: Johan Heuvel, Stefano Traferro, Yao-Hong Liu, Paul Mateman, Jialue Wang and Zhihao Zhou. Give many thanks to Johan, who helped me do the chip tiling and design the top layout of the chip. He also helped me solve any CAD tools related problems. Thanks go to Stefano for helping me synthesize the RTL code on weekends when approaching the tape-out date. Special thanks go to Yao-Hong for the technical discussions and helping me do the measurements on weekends. Sincere thanks go to Paul. Paul is a very experienced PLL designer and I benefited a lot from discussing with him. I also want to thank JIanglue and Zhihao. The life at Holst Centre was much more enjoyable with them.

I want to express my appreciations to my Chinese friends at TU Delft. Thanks go to Hao Fan for the wonderful cooperation on the course projects. Thanks go to my roommates Xiao Wang and Xianran Li for the wonderful life in Veldhoven. Thanks go to Kangli Huang, Xin Zhan, Yang Liu and Xiangwei Zeng for the beautiful times in Delft. Their help and friendship have made my study abroad a lot easier and more enjoyable.

Finally, my most special thanks go to my beloved family. I am deeply grateful to my sister for being so supportive and encouraging throughout the years. I am greatly in debt to my parents and grandparents for their unconditional love and support. Their willingness to do anything for me is beyond remarkable.

Jiang Gong

Eindhoven, 2017



Chip Pinout

Figure [A.1](#) shows the bonding diagram of the implemented IL-DPLL prototype¹. The chip is directly bonded to the PCB. Ground pins (excluding the noisy IO pin) are down bonding to the PCB ground to minimize the inductance of the bondwires. Table [A.1](#) shows the pinout of the IL-DPLL prototype.

¹Two designs are on the same die; the highlighted pins are the IL-DPLL pins.

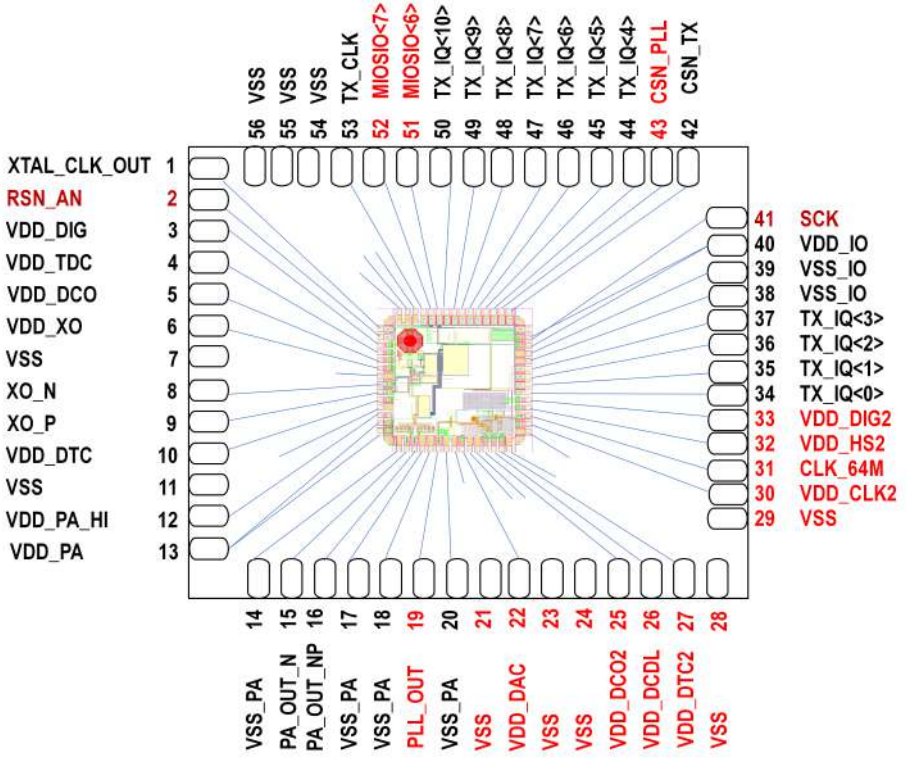


Figure A.1: Bonding diagram of the implemented IL-DPLL.

Table A.1: Pinout of the implemented IL-DPLL.

Pin Number	Name	Description
2	RSN_AN	Reset
19	PLL_OUT	RF Out
21	VSS	Ground
22	VDD_DAC	DAC Supply
23	VSS	Ground
24	VSS	Ground
25	VDD_DCO2	DCO Supply
26	VDD_DCDL	Injection Supply
27	VDD_DTC2	CF-DTC Supply
28	VSS	Ground
29	VSS	Ground
30	VDD_CLK2	Clock Buffer Supply
31	CLK_64M	Clock In
32	VDD_HS2	Snapshot TDC and Counter Supply
33	VDD_DIG2	Digital Supply
41	SCK	SPI Clock
43	CSN_PLL	SPI Select
51	MISIO<6>	SPI Control
52	MISIO<7>	SPI Control

B

Phase Noise Measurements Results

Figure B.1 shows the measured phase noise of two fractional and two integer channels.

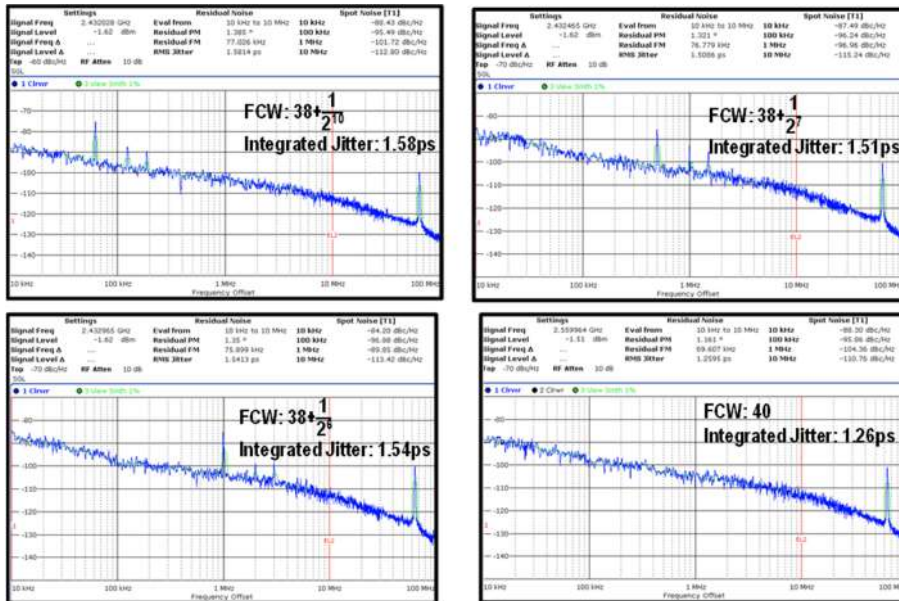
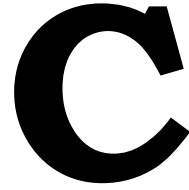


Figure B.1: Measured phase noise.



MATLAB Time-Domain Code

```
clear
close all;

load paro3.dat;
data=paro3/1e12;
t_res_dtc_c=(max(data)-min(data))/(length(data)-1)*1.00;

% % % % % % % % modify loop parameter here % % % % %
en_fout_jump=0;    % enable or disable the frequency jump of DCO
en_fine_dtc=1;    % enable or disable fine DTC
en_cali=1;        % enable or disable offset calibration
en_inj=1;        % enable or disable injection
beta=1;          % injection strength
f_drift=0;        % oscillator frequency drift
alpha=2^-2;      % loop filter parameters
rho=2^-5;
nr_points=10e3;   % number of reference cycles for simulation
fref=64e6;        % reference frequency
f_DCO_free=2332e6; % free-running frequency of DCO
f_desired=f_DCO_free+100*1e6+1*15.625e3;
                 % desired frequency
t_detection=0.9e3; % injection after t_detection reference cycles
cal_cycle=128;    % number of time offset calibration cycles

% % % % % % % % % % Time offset % % % % % % % % % % % %
delta_tos=10e-12; % offset between injection path and RF phase
tdc_os=0.5/f_desired;
                 % add tdc offset
% % % % % % % % % % TDC/DTC DCO gain % % % % % % % % %
```

```

t_res_tdc=12e-12; % TDC resolution
t_res_dtc=t_res_dtc_c;
% DTC resolution
t_res_dtc_f=5e-13; % Fine DTC resolution
t_res_dcdl=5e-13; % DC DL resolution
KDCO_es=50e3; % estimated DCO gain
K_DCO =50e3; % Real DCO gain

% % % % % % % % % % phase noise from cadence simulation % % % % % % %
pn_tdc=-153; % TDC thermal noise
pn_dtc=-143; % DTC thermal noise
pn_fref=-153; % reference phase noise
pn_DCO=-86; % DCO phase noise @1MHz
delta_f=1e6; % 1MHz offset frequency

% % % % % % % % % phase noise to jitter conversion % % % % % % %
sigma_square=delta_f/f_desired*
sqrt(1/f_desired)*sqrt(10^(pn_DCO/10)); % DCO 1MHz PN

sigma_ref=1/fref/2/pi*sqrt(10^(pn_fref/10)*fref); % ref thermal PN
sigma_tdc=1/fref/2/pi*sqrt(10^(pn_tdc/10)*fref); % tdc thermal PN
sigma_dtc=1/fref/2/pi*sqrt(10^(pn_dtc/10)*fref); % dtc thermal PN

% % % % % % % % % % FCW integer and fractional part % % % % % % %
FCW=f_desired/fref; % calculate FCW
FCW_int=fix(FCW); % calculate FCW integer part
FCW_frac=FCW-FCW_int; % calculate FCW fractional part

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% signal definition and initialization
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% % % % % % % % % % Reference-rate signals % % % % % % % % % % % % % % % % %
tR=zeros(1,nr_points); % reference clock timestamps
tR_dly=tR; % C-DTC delayed reference clock timestamps
tR_dly_f=tR; % F-DTC delayed reference clock timestamps
t_quan_err=tR; % C-DTC quantization error
f_dtc_ctl=tR; % F-DTC control code
tdc_out=tR; % fractional error
frac=tR; % accumulated FCW fractional part
FCW_acc_int=tR; % accumulated FCW integer part
ph_e=tR; % total phase error sent to loop filter
int_path=tR; % integer phase error
d_tdc=tR; % TDC control code

```



```

overflow=tR;           % FCW fractional part overflow
Acc_out=tR;           % instantaneous Accumulator output
Acc=0;                % initialize accumulator output
t_off=tR;             % time difference between ref path and RF
pd_out=tR;            % instantaneous one bit TDC output
% IIR filter y4=tR; y1=tR; y2=tR; y3=tR;
rho_path=tR;          % Integer path integrator
tR_dcdl=tR;           % DCDL timestamps

%% %% %% %% %% RF-rate signals %% %% %% %% %%
NTW=zeros(1,nr_points*fix(FCW-1));
% normalized tuning word
OTW=NTW;              % oscillator tuning word
Rv=NTW;               % ckv phase
delta_fv=NTW;         % DCO frequency change due to OTW
delta_tv=NTW;         % DCO period change due to OTW
%TDEV=NTW;            % accumulated phase error of DCO
tckv=NTW;             % DCO timestamps
fout=NTW;             % instantaneous output frequency
period=NTW;           % instantaneous output period
%Acc=NTW;              % DCO accumulated thermal noise jitter
fout(1)=f_DCO_free;   % initialize output frequency and period
fout(2)=f_DCO_free;
period(1)=1/f_DCO_free;
n=2;                  % initialize DCO time index

%% %% %% %% %% PLL loop implementation
% %% %% %% %%
for k=2:nr_points, % reference rate loop

% %% %% %% %% Acc FCW and reference timestamps %% %%
frac(k)=FCW_frac+frac(k-1); % acc FCW frac part
FCW_acc_int(k)=FCW_int+FCW_acc_int(k-1); % acc FCW int part
overflow(k)=fix(frac(k)); % get overflow
FCW_acc_int(k)=FCW_acc_int(k)+fix(frac(k)); % add overflow
frac(k)=frac(k)-fix(frac(k)); % get new acc FCW frac
tR(k)=(k-1)*1/fref+normrnd(0,sigma_ref); % ref timestamps

% %% %% %% %% DTC delay reference %% %% %% %% %%
tR_dly(k)=tR(k)+data(fix((1-frac(k))/fout(n)/t_res_dtc)+1)
-data(1)+normrnd(0,sigma_dtc);
% delayed ref timestamps

t_quan_err(k)=(1-frac(k))/fout(n)-

```

```

fix((1-frac(k))/fout(n)/t_res_dtc)*t_res_dtc;
% C-DTC quan. error
f_dtc_ctl(k)=fix(t_quan_err(k)/t_res_dtc_f);% F-DTC control code
tR_dly_f(k)=tR_dly(k)+en_fine_dtc*f_dtc_ctl(k)*t_res_dtc_f;
% delayed ref timestamps
tR_dcdl(k)=tR_dly_f(k)+1/f_desired+tdc_os+
delta_tos-en_cali*Acc*t_res_dcdl;
% DCDL for tos cal.
if k>=t_detection
tR_dcdl(k)=tR_dly_f(k)+1/f_desired+tdc_os+
delta_tos-en_cali*fix(mean(Acc_out(5e2-cal_cycle:5e2)))*t_res_dcdl;
end
% DCDL for injection
% avg acc output

%%% RF rate loop phase %%%
while tckv(n)<tR_dly_f(k)
n=n+1;
delta_fv(n)=K_DCO*fix(OTW(k-1)); % DCO frequency change

if n>24320
delta_fv(n)=delta_fv(n)+(n-24320)*f_drift;
% frequency drift
end

delta_tv(n)=delta_fv(n)/f_DCO_free/(f_DCO_free+delta_fv(n));
% DCO phase change
if tckv(n-2)<tR_dly_f(k-1) && tckv(n-1)>tR_dly_f(k-1) && en_inj==1
tckv(n)=tckv(n-1)+period(n-1)+beta*(tR_dcdl(k-1)-
(tckv(n-1)+period(n-1)));
if tckv(n-1)+period(n-1)>tR_dcdl(k-1)
Acc=Acc-1; % offset detection
else Acc=Acc+1;
end
if k<t_detection
t_off(k)=tckv(n-1)+period(n-1)-tR_dcdl(k-1);
% offset
Acc_out(k)=Acc; % accumulator output
pd_out(k)=Acc_out(k)-Acc_out(k-1);% 1b TDC output
end
else tckv(n)=tckv(n-1)+1/f_DCO_free-(delta_tv(n))
+normrnd(0,sigma_square);
% DCO timestamps
end
if k<t_detection % disable injection when k<t_detection
tckv(n)=tckv(n-1)+1/f_DCO_free-(delta_tv(n))+normrnd(0,sigma_square);

```

```

end
Rv(n)=Rv(n-1)+1;           % accumulate ckv phase
period(n)=(tckv(n)-tckv(n-1));% calculate period
fout(n)=1/period(n);      % output frequency
end

% % % % % % % % % % Phase detection % % % % % % % % % %
% TDC calculates fractional error, normalized to tckv period
d_tdc(k)=fix((1/fout(n)-(tckv(n)-tR_dly_f(k))+
normrnd(0,sigma_tdc))/t_res_tdc);
tdc_out(k)=fix((1/fout(n)-(tckv(n)-tR_dly_f(k))+
normrnd(0,sigma_tdc))/t_res_tdc)*t_res_tdc*fout(n);
%total error calculation
ph_e(k)=FCW_acc_int(k)-(Rv(n-1)+tdc_out(k))+tdc_os*f_desired;
rho_path(k)=rho_path(k-1)+rho*ph_e(k); % rho path integrator
NTW(k)=alpha*ph_e(k)+rho_path(k);      % phase error is filtered
OTW(k)=fref/KDCO_es*NTW(k);           % DCO gain normalization
int_path(k)=FCW_acc_int(k)-Rv(n-1);    % integer phase error
if k>640*5
    OTW(k)=fref/KDCO_es*NTW(k)+en_fout_jump*10e6/(KDCO_es);
end                                     % frequency jump

end
if en_inj==1
save period_inj.dat period -ASCII;
save Acc_out.dat Acc_out -ASCII;
save tckv_inj.dat tckv -ASCII;

else
save period_nor.dat period -ASCII;
save Acc_out.dat Acc_out -ASCII;
save tckv_nor.dat tckv -ASCII;
end

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% Plot Phase Noise
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% load period_nor.dat;
load period_inj.dat;

% % Calculate Sphi
nfft=2^22;           % number of fft points
winLength=nfft; % window length
overlap=nfft/2;
winNBW=1.5;

```

```

period_no=period_inj(0.7e5:length(period_inj)-100);
peri_length = length(period_no);% number of period points
T=mean(period_no)
f=1/T
J_no=std(period_no);
% compute the cumulative phase of each transition
phases_no=2*pi*cumsum(period_no)/T;
% compute power spectral density of phase
% winLength = hann(winLength);
[Sphi_no, f]=psd(phases_no, nfft, 1/T, winLength, overlap, 'linear');

% [Sphi, f]=pwelch(phases, nfft, overlap, nfft, 1/T);
% correct for scaling in PSD due to FFT and window
Sphi_no=winNBW*Sphi_no/nfft;

% plot the results (except at DC)
K = length(f);
rbw = winNBW/(T*nfft);
%norm to resolution bandwidth
pn_dco_no=Sphi_no/rbw;

k1=fix(10e3/(2432e6/2)*K);
k2=fix(10e7/(2432e6/2)*K);

figure(1);
semilogx(f(k1:k2),10*log10(pn_dco_no(k1:k2)), 'Color',[0 0.5 0]
, 'LineWidth',4);
xlabel('Offset Frequency [Hz]', 'FontSize',26, 'FontWeight','bold'
, 'FontName','Arial');
ylabel('Phase Noise [dB/Hz]', 'FontSize',26, 'FontWeight','bold'
, 'FontName','Arial');
grid on;
set(gca, 'Box', 'on', 'LineWidth', 1);
set(gca, 'FontSize', 26, 'FontWeight', 'bold', 'FontName', 'Arial')
grid on;
set(gca, 'GridLineStyle', '-');
ax.MinorGridLineStyle = '-';
set(gca, 'GridLineStyle', '-');
set(gcf, 'color', 'w');
ax.MinorGridLineStyle = '-';

%%%%%%%%%%%%%%
% Plot Spectrum
%%%%%%%%%%%%%%

```

```

load tckv_inj.dat;
% load tckv_nor.dat;
fs=100e9+9211.42e6;      % sample tckv timestamp
length1=1e6;           % number of rf cycle to sample
t=tckv_inj(10e5:10e5+length1); % choose tckv

% t=tckv(10e5:40e5);
t=t-t(1);              % start with 0
t_total=t(length(t));
nr_points=fix(fs*t_total);
ts=zeros(1,nr_points);
y=zeros(1,nr_points);
m=2;
for index=2:length1,

    while (ts(m)<t(index))

        if    ts(m)<(t(index)+ t(index-1))/2 && ts(m)>=t(index-1)
            y(m)=1;

        elseif ts(m)>=(t(index)+ t(index-1))/2 && ts(m)<t(index)
            y(m)=0;
        end
        m=m+1;
        ts(m)=ts(m-1)+1/fs;
    end
end

y=y(1:nr_points);
ffty = (fft(y'.*(1)));
% ffty = (fft(y'.*(kaiser(nr_points,500))));
% ffty = (fft(y'.*(nuttallwin(nr_points)))); % do fft and add window
ffty_magn = abs(ffty); % get AM
% Scale the spectrum, add floor -80dB and scale fundamental to 0 dB
ffty_dB = 20*log10(ffty_magn/max(ffty_magn)+0.5e-4)+3.92;

figure(2);
fres = fs/nr_points; % frequency resolution
% semilogx((1:nr_points/2)*fres, ffty_dB(1:nr_points/2));

fmin=fix((2.432e9+15.625e3-70e6)/fres);
fmax=fix((2.432e9+15.625e3+70e6)/fres);
plot((fmin:fmax)*fres, ffty_dB(fmin:fmax), 'LineWidth', 3);
%title('PLL spurs', 'FontSize', 20, 'FontWeight', 'bold', 'Color', 'r');

```

```
xlabel('Frequency [GHz]', 'FontSize', 26, 'FontWeight', 'bold',  
, 'FontName', 'Arial Narrow');  
ylabel('Spectrum [dBm]', 'FontSize', 26, 'FontWeight', 'bold',  
, 'FontName', 'Arial Narrow');  
grid on;  
%plot((fmin/4:fmax*4)*fres, ffty_dB(fmin/4:fmax*4));  
dcmObj = datacursormode;  
set(dcmObj, 'UpdateFcn', @updateFcn);  
grid on;  
set(gca, 'Box', 'on', 'LineWidth', 1);  
set(gca, 'FontSize', 26, 'FontWeight', 'bold', 'FontName', 'Arial Narrow')  
grid on;  
set(gca, 'GridLineStyle', '-');  
ax.MinorGridLineStyle = '-';  
set(gca, 'GridLineStyle', '-');  
set(gcf, 'color', 'w');  
ax.MinorGridLineStyle = '-';
```